МУЛЬТИМЕДИЙНЫЙ САМОУЧИТЕЛЬ НА CD-ROM

TeachPro™

WEB-дизайн

Мультимедиа Технологии и Дистанционное Обучение
МОСКВА
2006

Мультимедийный самоучитель на CD-ROM: **TeachPro™ WEB-дизайн**/Под ред. Катхановой Ю.Ф., Гринберг Г.С., Антонова Г., Соловьева И., Суховерхова Р. — М.: ММТиДО, 2006. — 560 с.: ил.

ISBN 5-9230-0154-3

Самоучитель **TeachPro™ WEB-дизайн** предназначен для тех, кто хочет создать качественный сайт и опубликовать его в Интернет. В нем Вы найдете все (или почти все), что должен знать человек, уже узнавший, что такое Internet. Изучая Web-дизайн, Вы приобретете умение грамотно и эффективно работать в сети Internet, реализуя свои творческие замыслы.

Обращаем Ваше внимание на то, что для Web-дизайнера важно не только знание языка программирования HTML. Хотя, профессиональный Web-дизайнер должен знать все тонкости и нюансы внутренней структуры своей Web-страницы.

Именно художественно-выразительные методы дизайна помогут вам реализовать творческие замыслы, проявить оригинальность мышления в процессе поиска и решения практических задач по созданию Web-страницы или сайта.

Мы познакомим Вас с основами компьютерной графики. По нашему мнению, программы для работы с изображениями и иллюстрациями, такие как CorelDraw, PHOTO-PAINT или Paint Shop Pro, являются основным инструментом Web-дизайнера. Именно в этих программах рождается все то, что принято называть Web-дизайном. Ну, а знание продуктов Adobe и Macromedia сделает Web-дизайнера гораздо привлекательнее в качестве профессионала Web-графики.

ISBN 5-9230-0154-3

© ООО «Мультимедиа Технологии и Дистанционное Обучение», 2006

Содержание

Руководство пользователя 11

Установка системы TeachPro на компьютер 13
Первый запуск программы 14
Первый вход в систему 14
Содержание курса. Выбор нужной лекции 16
Основной рабочий экран системы TeachPro 18
Обучение в непрерывном режиме 18
Обучение в пошаговом режиме 19
Контрольный режим 19
Тестовый режим 19
Область демонстрации урока 19
Панель управления системы TeachPro 19
Клавиши управления демонстрацией урока 20
Клавиши изменения режимов обучения 20
Дополнительные функциональные клавиши TeachPro 21
Подсказка 21
Закладки 22
Поисковая система 25
Устранение возможных проблем с программой 26
Проблемы при инсталляции программы 26
Проблемы при запуске программы 27
Затруднительные ситуации при проигрывании урока 28

TeachPro WEB-дизайн 29

Что необходимо знать Web-дизайнеру 31

Раздел 1. Основы Дизайна 35

1.1. История русского дизайна 35
1.2. История русского дизайна. Русский дизайн с *XIX* в. до наших дней 38
1.3. Формообразование 42
1.4. Основные правила композиции 45
1.5. Цвет в дизайне 48

Раздел 2. Компьютерная графика — что это такое? 51

2.1. Немного из истории 51

2.2. Основные виды компьютерной графики 54
2.3. Растровая графика 57
2.4. Векторная графика 60
2.5. Трехмерная графика 63
2.6. Фрактальная графика 66
2.7. Цветовые модели и палитры 68
2.8. Вспомним A*dobe Photoshop* 71
2.9. Размеры изображения, кадрирование. Цветокоррекция 74
Раздел 3. Общие сведения о Web-дизайне 77
3.1. Основы Интернет 77
3.2. WorldWideWeb. WWW, браузер, доменные имена 79
3.3. WorldWideWeb-2. Сайты и страницы, гипертекст и гиперссылки 81
3.4. Свойства документа 84
3.5. Особенности изображений для Web 86
3.6. Приемы оптимизации графики 89
3.7. О программе Macromedia Dreamweaver 92
3.8. Форматирование текста 94
3.9. Синтаксис текста 96
3.10. Разметка текста 97
3.11. Вставка изображения и редактирование его свойств 98
3.12. Создание простого Web-сайта 100
Раздел 4. Технологии Web-дизайна 102
4.1. HTML 102
4.1.1. Что такое HTML 102
4.1.2. Что собой представляют HTML-файлы 106
4.1.3. Синтаксис HTML и структура HTML-документа 107
4.1.4. Тело документа. Элементы BODY, DIV, SPAN 111
4.1.5. Разметка текста. Phrase elements, font-style-elements 114
4.1.6. Разметка текста-2 116
4.2. Технический уровень 119
4.2.1. Списки 119
4.2.2. Ссылки 121
4.2.3. Таблицы 125
4.2.4. Изображения 131
4.2.5. Формы 135

4.2.6. Фреймы ... 140
4.2.7. Изменение цветов фона. Горизонтальная линия ... 144
4.2.8. Стили шрифтов в HTML ... 146
4.2.9. Меню выбора в HTML ... 148
4.2.10. Создание формы отправки файлов в HTML ... 149
4.2.11. Карта изображения в HTML ... 151
Раздел 5. Программы для работы с Web-сайтами ... 154
5.1. Векторная графика: Macromedia Flash ... 154
5.1.1. Первое знакомство. Работа с панелью инструментов ... 154
5.1.1.1. Обзор интерфейса программы ... 154
5.1.1.2. Установка основных параметров фильма, настройка интерфейса редактора ... 158
5.1.1.3. Инструмент "Линия" ... 162
5.1.1.4. Инструменты "Карандаш", "Овал", "Прямоугольник" ... 164
5.1.1.5. Инструмент "Заливка" ... 166
5.1.1.6. Инструмент "Преобразование заливки" ... 169
5.1.1.7. Инструменты "Кисть", "Чернильница", "Пипетка", "Ластик" 171
5.1.1.8. Инструмент "Стрелка" ... 174
5.1.1.9. Инструмент "Свободная трансформация" ... 176
5.1.1.10. Инструмент "Лассо" ... 178
5.1.1.11. Инструменты "Перо", "Частичное выделение" ... 180
5.1.1.12. Инструмент "Текст" ... 183
5.1.2. Основы анимации во Flash, символы и экземпляры ... 186
5.1.2.1. Создание покадровой анимации, отображение кадров ... 186
5.1.2.2. Раскадровка формы (морфинг) ... 191
5.1.2.3. Раскадровка движения ... 196
5.1.2.4. Символы и экземпляры, изменение свойств экземпляров символов ... 200
5.1.2.5. Графические символы и символы видеоклипов ... 203
5.1.2.6. Символы кнопок и их особенности, создание пульсирующей кнопки ... 205
5.1.3. Принципы работы со специальными слоями.Применение сцен ... 207
5.1.3.1. Общие принципы работы со слоями, опорные слои ... 207
5.1.3.2. Слои траекторий. Принципы их использования ... 210

5.1.3.3. Примеры создания трехмерных эффектов с помощью слоев траекторий....213
5.1.3.4. Принципы работы с маскирующими слоями. Примеры их использования....215
5.1.3.5. Примеры применения текста при маскировании (**II** уровень)....218
5.2. Трехмерная графика. Создание рекламного ролика в 3*D Max* 6....221
5.3. Создание изображений для Web в программе Adobe Photoshop CS....229
5.3.1. Форматы графических файлов для Web....229
5.3.2. Команда Save For Web. Инструменты и режимы просмотра команды....231
5.3.3. Сохранение для Web в формате jpg....234
5.3.4. Цветовые модели изображения в формате jpg и Web....237
5.3.5. Изображения в формате jpg и прозрачность....239
5.3.6. Формат GIF....241
5.3.7. Цветовая таблица (Color Table)....243
5.3.8. Дизеринг. Поэтапный вывод изображения на экран....245
5.3.9. Изображения в формате GIF и прозрачность....246
5.3.10. Создание надписей для Web-страниц....250
5.3.11. Надпись с тенью и объемная надпись....251
5.3.12. Анимации в формате GIF....253
5.4. Разработка электронных документов в программе Adobe Illustrator CS....259
5.4.1. Создание Web страницы....259
5.4.2. Создание Web страницы с вырезками и гиперссылками....266
5.4.3. Создание анимационного клипа. Движение объекта....267
5.4.4. Создание анимационного клипа. Преобразование объектов и их появление....270
5.5. Создание электронных документов в CorelDRAW 12....273
5.5.1. Обработка исходного документа. Ввод гиперссылок и закладок....273
5.5.2. Ввод гиперссылок и закладок. Вставка объектов Интернета. Формирование изображения для Web....276
5.5.3. Порядок формирования изображения для Web....278
5.5.4. Создание документов в формате HTML....280
5.5.5. Создание документов в формате PDF....282
5.6. Вставка мультимедиа при помощи программы FrontPage 2003....285

5.6.1. Вставка фильма в формате Flash ... 285
5.6.2. Изменение свойств фильма в формате Flash ... 286
5.6.3. Вставка видеозаписи ... 288
5.6.4. Настройки автоэскизов ... 290
5.6.5. Создание коллекции фотографий на основе шаблона ... 292
5.6.6. Создание коллекции фотографий как Web-компонента ... 294
5.6.7. Вставка фотографии в коллекцию фотографий с помощью сканера ... 295
5.6.8. Изменение рисунка в коллекции фотографий ... 298
5.6.9. Макеты коллекции фотографий ... 300
5.6.10. Вставка меняющейся (интерактивной) кнопки ... 302
5.6.11. Изменение свойств меняющейся кнопки ... 303
Раздел 6. Web-сайты. Типы и разработка ... 306
6.1. Типы сайтов, назначение, цель, целевая аудитория ... 306
6.2. Кодировка русского языка. Мировая паутина. Браузеры ... 307
6.3. Обозреватели Интернет ... 310
6.3.1. Знакомство с браузерами ... 310
6.3.1.1. Что такое обозреватели? ... 310
6.3.1.2. Навигация в обозревателе ... 314
6.3.1.3. Любимые страницы. Печать ... 322
6.3.1.4. Автономный режим работы ... 326
6.3.2. Интерфейс Internet Explorer 6.0 ... 329
6.3.2.1. Панели браузера ... 329
6.3.2.2. Панели инструментов ... 331
6.3.2.3. Некоторые приемы работы ... 335
6.3.2.4. Кнопки поиск, избранное ... 337
6.3.2.5. Кнопки медиа, журнал ... 342
6.3.2.6. Кнопки почта, печать, показать в виде **HTML** ... 344
6.3.2.7. Главное меню. Меню Файл ... 346
6.3.2.8. Главное меню. Меню Правка ... 352
6.3.2.9. Главное меню. Меню Вид ... 355
6.3.2.10. Главное меню. Меню Избранное ... 358
6.3.2.11. Главное меню. Меню Сервис ... 361
6.3.2.12. Главное меню. Меню Справка ... 366
6.4. Путешествие в WEB ... 370

6.4.1. Путешествие по миру АВТО 370
6.4.2. Энциклопедия и WEB 372
6.4.3. Литература в WEB 374
6.4.4. Музеи в WEB 378
6.4.5. Отправление открытки 380
6.4.6. Покупка книги 382
6.4.7. Бронирование билетов 385
6.5. Отличия от поли графического дизайна на примере публикаций QuarkXPress 6.0 387
6.5.1. Создание простых Web-проектов 387
6.5.2. Палитра Web Tools 390
6.5.3. Диалоговое окно Preferences. Вкладка Export Выбор браузера. Палитры, ориентированные на Web 392
6.5.4. Преобразование печатных макетов в Web формат 395
6.5.5. Стилизованный текст 399
6.5.6. Создание гиперссылок 403
6.5.7. Создание карты изображения 405
6.5.8. Установка фона страницы. Использование изображения в качестве фона страницы 407
6.5.9. Анкеры 409
6.5.10. Редактирование гиперссылок. Создание мета-дескрипторов 412
6.5.11. Экспортирование Web-макета в HTML 417
6.6. XML 420
6.7. Общие сведения о программировании на Java 423
6.7.1. Введение 423
6.7.2. Среда программирования 427
6.7.2.1. Установка языка **Java** 427
6.7.2.2. Компиляция и выполнение простой программы 430
6.7.2.3. Текстовый редактор **TextPad** 432
6.7.2.4. Апплеты 434
6.7.3. Основные структуры языка 438
6.7.3.1. Простая программа на языке **Java** 438
6.7.3.2. Простая программа на языке **Java**. Комментарии 440
6.7.3.3. Типы данных. Переменные. Целые числа 443
6.7.3.4. 16-ричные, 8-ричные и действительные числа 446

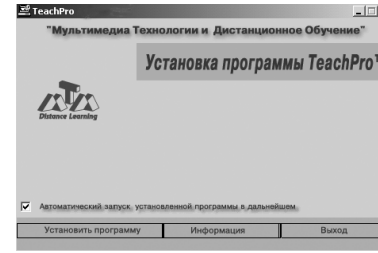
6.7.3.5. Символьный и булевый типы данных 449
6.7.3.6. Операторы. Арифметические действия 452
6.7.3.7. Битовые операции. Математические функции 455
6.7.3.8. Операторы. Преобразования типа 458
6.7.3.9. Строки: создание, слияние, выделение подстроки, длина 460
6.7.3.10. Строки. Изменение, сравнение, работа с ними 463
6.7.3.11. Ввод данных с диалоговым окном и операции с ними 466
6.7.3.12. Форматирование данных для отображения на консольном окне 469
6.7.4. Основные структуры языка (продолжение) 471
6.7.4.1. Оператор **if**. Синтаксис 471
6.7.4.2. Оператор **if**. Составной оператор. Вложение операторов **if** ... 474
6.7.4.3. Операторы цикла **do** 476
6.7.4.4. Операторы цикла **for** 479
6.7.4.5. Оператор **switch** 481
6.7.4.6. Прерывание потока управления. Оператор **break** 484
6.7.4.7. Большие числа. Арифметические операции 487
6.7.4.8. Большие числа. Сравнение. Вычисление с большой точностью 489
6.7.4.9. Одномерные массивы. Создание. Инициализация. Число элементов 492
6.7.4.10. Одномерные массивы. Копирование. Сортировка 495
6.7.4.11. Многомерные массивы. Создание. Инициализация 498
6.7.4.12. Многомерные неровные массивы 501
6.7.5. Апплеты 503
6.7.5.1. Создание простого апплета 503
6.7.5.2. Преобразование приложения в апплет 507
6.7.5.3. Архивы **jar** 510
6.7.5.4. Упаковка ресурсов в архивы **jar** 513
6.7.5.5. Всплывающие окна в апплетах Создание кнопки 516
6.7.5.6. Всплывающие окна в апплетах. Обработчик событий 518
6.7.5.7. Создание передаваемых параметров на **html** странице 520
6.7.5.8. Чтение параметров в апплете 522
6.7.5.9. Передача параметров апплетам. Построение гистограммы на их основе 524

6.7.5.10. Анимированные гистограммы на основе параметров из Web страниц 526
6.7.5.11. Создание цветных анимированных гистограмм на основе параметров из Web страниц. Создание нескольких вариантов страниц 529
6.7.5.12. Апплет с закладками. Создание 531
6.7.5.13. Апплет с закладками. Передача адресов для закладок в качестве параметров 534
Раздел 7. Публикация в WEB 539
7.1. Регистрация сайта и загрузка файла на сайт 539
7.2. Редактирование опубликованного сайта 541
7.2. Доступ к файлам по FTP 545
Предметный указатель 548

Руководство пользователя

Установка системы TeachPro на компьютер

Это производится один раз, а потом, чтобы начать работать, достаточно только вставить компакт-диск в дисковод и запустить соответствующее приложение из окошка ***TeachPro***. Если Вы хотите работать с курсом попеременно на нескольких компьютерах (скажем, в школе и дома), достаточно по одному разу установить ***TeachPro*** на каждом из них и в дальнейшем просто переставлять диск.

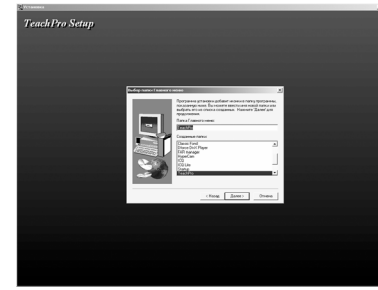


Следуйте нашей инструкции, и у вас все получится.

Вставьте компакт-диск с программой ***TeachPro*** в дисковод ***CD-ROM***. Данный диск снабжен автозапуском, и откроется окно, в котором нужно нажать кнопку «установить программу», если автозапуск отключен, проделайте пункты 2, 3, если нет, переходите к пункту 4.

Дважды щелкните мышкой по иконке ***Мой компьютер*** на Вашем рабочем столе, далее щелкните по иконке, изображающей ***CD-ROM***.

Сейчас перед Вами список файлов, содержащихся в корневом каталоге диска ***CD-ROM***. Найдите файл ***SETUP.exe*** и запустите его. Достаточно снова щелкнуть по нему два раза мышью.



Через несколько секунд появится окно Мастера установки программы ***TeachPro***. Для продолжения работы нажмите на Далее. После этого Мастер предложит Вам дать название программной группе, в которой будут размещены все приложения системы ***TeachPro***. Рекомендуем согласиться с предложенным по умолчанию названием и снова нажать на кнопку Далее. Весь процесс займет не более одной минуты.

Если Вы работаете в операционной системе ***Windows 95***, то необходимо установить библиотеку ***Microsoft DirectX 5 (Dx5eng.exe)*** из корневого каталога, позволяющую разворачивать окно программы ***TeachPro*** на весь экран независимо от разрешения экрана монитора. Если эта библиотека или ее более поздняя версия уже установлена или Вы работаете в операционных системах ***Windows 98/2000/XP***, то соответственно, приложение ***Dx5eng.exe*** инсталлировать не следует.

В конце установки Мастер предложит Вам перезагрузить компьютер, рекомендуем согласиться. В стартовом меню ***Пуск*** во вкладке ***Программы*** в меню ***TeachPro*** появились две вкладки: ***TeachPro*** и ***TeachPro Help***.

Первый запуск программы

Для запуска необходимо выбрать в стартовом меню ***Пуск*** в подменю ***Программы*** пункт ***TeachPro***, далее ***TeachPro Web-дизайн***. При запуске программы, если появится окно защиты от копирования ***StarForce***, надо будет ввести код с поверхности компакт диска. В случае неудачной идентификации диска надо дать диску остыть и попробовать ещё раз. Также проверьте, правильно ли вы набрали сам код, состоящий из английских букв и цифр.

В случае появления ошибки ***с*** синим экраном обращайтесь на наш сайт за обновлениями, возможна несовместимость с устаревшим процессором или с новым обновлением ***Windows Service Pack***.

Первый вход в систему

После запуска программы появится стартовое окно.

Рассмотрим поле для ввода имени пользователя и кнопки ***Регистрация*** и ***Удалить***.

Система ***TeachPro*** позволяет работать с одним пакетом (***CD-ROM***) не одному, а нескольким пользователям. Так, Вы можете учиться всем классом, всем коллективом, или обучаться на компьютерных курсах.

Разумеется, в определенный момент времени работать с курсом может только один человек, но зато ***TeachPro*** запоминает каждого зарегистрировавшегося пользователя и ведет по нему свою базу записей. Для каждого из обучаемых ***TeachPro*** запоминает:

- имя пользователя;

- последнюю лекцию;

- статистику по лекциям, т.е. что прослушано полностью, что частично, а к чему еще не приступали;

- дату последнего запуска программы;

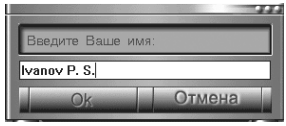
- закладки (пометки), которые пользователь создал для себя, чтобы лучше ориентироваться в лекциях;

- общее время работы с программой.

Поэтому, сколько бы людей ни работало в системе, Вы можете быть уверены, что компьютер знает все о Вашей личной работе с данным курсом.

Рассмотрим, как происходит регистрация.

Нажмите на клавишу ***Регистрация***.

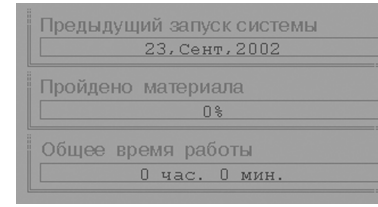


В появившемся окошке наберите свое имя или то имя, под которым Вы хотите видеть себя в списке пользователей.

Нажмите мышью на клавишу ***OK*** или на клавиатуре на ***Enter***. Если же Вы передумали регистрироваться — ***Отмена***.

Как видите, к списку имен добавилось Ваше имя, и теперь можно продолжать работу. Кстати, заметим, что при каждом следующем запуске программы окошко ***Имя*** будет содержать имя того пользователя, который работал в системе последним. Его, как бы, приглашают продолжить ранее прерванную работу.

Если имя надо поменять, достаточно нажать мышкой на клавишу в правой части окна, а в выпавшем списке выбрать нужное Вам имя.



Для удаления имени из списка необходимо:

– вывести в окошке ***ненужное*** имя;

– нажать на клавишу ***Удалить***.

Поскольку эта операция необратима, система обязательно переспросит ***«Хотите удалить имя из списка?»***. Нажмите на клавишу ***Ok***, и лишнее имя исчезнет из системы. Клавиша ***Отмена*** — отказ от удаления имени.

Рассмотрим остальные части экрана.

В графе ***Предыдущий запуск системы*** показана дата, когда пользователь, чье имя сейчас активно, в последний раз работал с системой

Графа ***Пройдено материала*** показывает, какой процент от общего объема уже пройден. Если Вы только в первый раз запустили программу и зарегистрировались, естественно, там будет показано 0 процентов.

Графа ***Общее время работы*** покажет то время, которое Вы затратили, обучаясь по курсу.

В нижней части экрана расположены еще три клавиши.

Кнопка ***О программе*** вызывает диалоговое окно с информацией о разработчиках.

Кнопка ***Помощь*** вызывает справочную систему. После нажатия на эту кнопку Вы попадёте в справочную систему (для её отображения вызывается ***Internet Explorer***), где в привычном гипертекстовом формате Вам расскажут о том, как работать с программой, и о ее характеристиках. Чтобы вернуться в программу ***TeachPro***, необходимо нажать кнопку ***TeachPro*** на панели задач.

Кнопка ***Выход*** в комментариях не нуждается. Нажмите ее — и Вы завершите работу с программой.

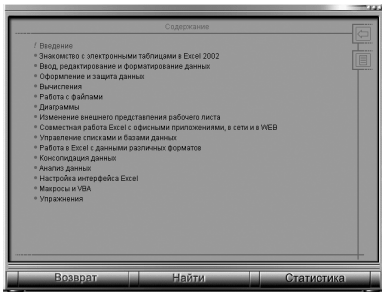
Содержание курса. Выбор нужной лекции

Все занятия с курсом расположены в виде некоторой иерархической структуры. Уроки в каждой подглаве представлены в следующем наборе:

- лекции;

- контрольные занятия;

Материал разбит на главы. Каждая глава в свою очередь состоит из тематических частей, причем количество колеблется от одной до двадцати. Часть содержит несколько лекций, с которыми пользователь и работает. Продолжительность одной лекции не превышает 12-15 минут, т.к. это время является оптимальным для освоения нового материала. Темп изложения материала на наших занятиях весьма высок, и Вам часто захочется прослушать некоторые отрывки лекции несколько раз. Реальное время работы с одним уроком, учитывая повторные прослушивания трудных мест, тестовый ставляет от 30 минут до 1 часа. Разумеется, учить материал лучше всего по порядку, не перескакивая через лекции.



Если Вы впервые работаете с курсом, то после заглавного экрана Вы попадете сразу в оглавление. Это сделано для того, чтобы можно было получить общее представление о том объеме материала, который Вам предстоит изучить. Но если Вы уже работали с системой, то появится страничка с той подглавой (и со всеми ее уроками), на которой Вы остановились в прошлый раз. Как видите, ***TeachPro*** помнит все. Допустим, мы в первый раз работаем в программе, и вот экран Содержание. Мы видим полный перечень разделов курса. Если навести курсор на любой раздел, и нажать на левую кнопку мыши, то мы перейдем на страничку с уроками (или подглавами) этого

раздела. Но не торопитесь это делать — мы еще не со всем разобрались. Так что продолжим.

Слева от строк с названием уроков расположены значки, которые несут полезную смысловую нагрузку. Смысл их в следующем: нам необходимо всегда знать, сколько материала мы уже изучили, какие уроки прослушали полностью, какие частично, а к каким даже не приступали. И такой значок, меняя свою форму, показывает нам, насколько изучен тот урок, слева от которого он стоит. Вот все варианты:

! *— **часть урока прослушана или весь урок***, но Вы не прошли контрольный и тестовый режимы;

√ — материал изучен полностью.

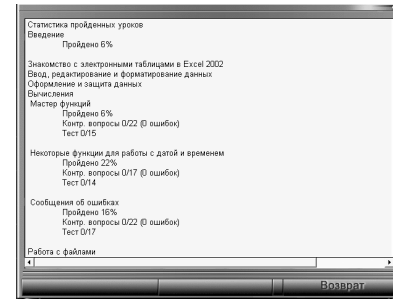
● — ***обозначает*** название главы или подглавы. Иерархическое содержание построено так, что имеет несколько уровней вложенности, и тем самым вы сможете перейти на вложенную подглаву, посвящённую какой-либо теме. Уроки, предназначенные для освоения тем глубже, более продвинуто, отмечены как ***II уровень***.

Для возврата назад к списку глав предусмотрено две кнопки в правом верхнем углу:

– возврат к листу содержания предыдущей главе

– возврат в корневой список глав

TeachPro меняет эти значки автоматически.



Кнопка ***Статистика*** вызывает детальную статистику по лекциям, которая запоминает для каждого урока:

– дату изучения этого урока;

– процент прослушанной части урока от длительности урока;

– сведения о прохождении урока в контрольном и тестовом режиме:

1. число правильных ответов и общее число вопросов;

2. число ошибок (число заданий, на которые был дан правильный ответ не с первой попытки);

3. счётчик использования подсказки.

Такая информация отображается только для тех уроков, которые загружались хотя бы один раз. При первом запуске программы окно «Статистика пройденных уроков» будет пустым. В дальнейшем появится

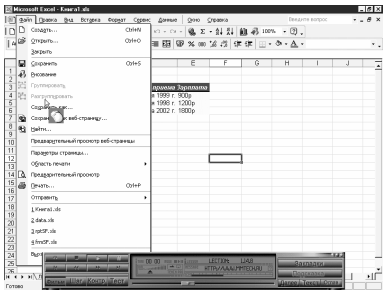
список названий глав, а между ними — информация о тех уроках, которые были пройдены.

Если при повторном запуске урока вместо неправильных ответов будут введены правильные, информация о прохождении урока обновится. При этом старые лучшие результаты не сбрасываются, и чтобы пройти курс сначала, надо зарегистрироваться под другим именем.

Теперь, когда мы описали страницу Оглавления, перейдем к системе управления уроком.

Основной рабочий экран системы TeachPro

После выбора нового урока в оглавлении и до начала реальной работы с ним пройдет времени минута, максимум две. За это время нужная часть курса, содержащая этот самый урок, загрузится в память Вашего компьютера, пройдет процесс распаковки данных.



Сначала разберемся, как происходит обучение пользователя в системе ***TeachPro***. Суть занятий предельно проста и максимально приближена к реальным занятиям в аудитории с преподавателем. Вы слышите голос лектора, и одновременно видите все те операции, которые лектор производит на своем компьютере. То есть, идет реальная работа с изучаемой программой, при этом все происходящее на экране подробно комментируется и объясняется. Дополнительно Вы имеете возможность отслеживать те действия лектора, которые на экране при работе с той или иной программой обычно не видны. Скажем, какая кнопка мыши и когда была нажата, какие клавиши, комбинации клавиш и в каком порядке нажимались на клавиатуре. Эффект достигается такой, как если бы рядом с Вами сидел опытный педагог, все бы Вам показывал и рассказывал, а Вы бы еще постоянно прерывали его и переспрашивали: «А сейчас что ты нажал, а как ты это сделал?».

Имеется 4 режима обучения, разберемся с ними подробно.

Обучение в непрерывном режиме

Как видно из названия, урок идет ***без перерыва***, поэтому около 15 минут придется воспринимать информацию «на одном дыхании». Вы, конечно, в любой момент можете ее остановить, нажав на ***Стоп*** или на ***Пауза***.

Обучение в пошаговом режиме

Урок разбит на некоторое количество ***частей,*** или ***шагов***. Их может быть 10 — 20 в уроке, и каждый содержит некоторый фрагмент материала, о котором говорит лектор. После прослушивания одного шага лекция прерывается, и Вы можете начать слушать следующую часть лекции, либо еще раз пройти предыдущую.

Контрольный режим

Здесь ***Вам предложат самостоятельно сделать то или иное действие***. Это может быть изменение соответствующей опции в меню, нажатие на клавишу на панели инструментов или применение комбинации клавиш клавиатуры. Чаще всего пользователя просят самостоятельно повторить те действия, которые лектор только что произвел.

Тестовый режим

Здесь от пользователя потребуется максимум внимания и знания предметной области. ***Вам придется самостоятельно решить несколько задач***, которые поставит лектор. То есть выполнить самому ряд действий, оперируя с мышью и клавиатурой. Это потребует твердого знания всего материала этого урока.

Область демонстрации урока

В рабочей области демонстрируется непосредственная работа лектора с этой программой: перемещается курсор мыши, набирается текст, формируются таблицы, открываются меню и т. д.

Панель управления системы TeachPro

Ее основная задача — дать Вам возможность самому управлять процессом обучения: вызывать систему помощи, ставить свои закладки в наиболее интересных местах и т.д. Забегая вперёд, скажем, что с этой панели можно вызвать дополнительные панели — панель закладок и панель текста лекции. Окно с автоматически прокручиваемым текстом лекции будет особенно полезно для глухих и для людей с ослабленным слухом, или же при отсутствии звуковой карты на компьютере.

Теперь рассмотрим, из чего состоит управляющая панель программы. Перед нами ряды клавиш и информационная панель. Начнем с клавиш. Каждая из них имеет надпись или значок, поясняющий действия. Они имеют две позиции, нажатую или отжатую.

Кроме того, клавиши могут быть доступными или недоступными в данный момент времени.

Доступные клавиши выглядят вот так.

А недоступные — так.

Это может означать, что в текущем режиме такая клавиша бессмысленна, как, скажем, ***Подсказка*** в режиме непрерывной демонстрации.

Далее представлен полный перечень всех клавиш и краткое описание их функций.

Клавиши управления демонстрацией урока

Повтор вопроса — в режимах «контрольный» и «тестовый» нажатие на эту клавишу приводит к повторению вопроса преподавателя.

Стоп — остановка прослушивания урока

Старт — начать (или продолжить) урок

Пауза — временная (короткая) остановка

Вернуться назад на один шаг

Перейти на шаг вперед

Перейти в конец урока

Перейти в начало урока

Клавиши изменения режимов обучения

Обучение в непрерывном режиме

Пошаговый режим

Контрольный режим

Тестовый режим

Каждый новый урок по умолчанию начинается в непрерывном режиме (клавиша ***Фильм*** утоплена в панель). Чтобы перейти в другой режим, достаточно нажать на нужную клавишу.

Дополнительные функциональные клавиши TeachPro

Подсказка

Используется только в контрольном и тестовом режимах. Суть ее проста — когда Вы совсем не знаете, как выполнить задание лектора, нажимайте на ***Подсказку***. Поскольку в тестовом режиме Вам придется самостоятельно выполнять не одно, а серию действий, то и подсказка может быть получена отдельно по каждому из них.

1. Если потребовалось щелкнуть мышью в определенную область экрана, курсор принимает форму красного вопросительного знака. При вызове подсказки ***TeachPro*** проявит искомую область:

– если необходимо кликнуть левой клавишей мыши — область загорится красным цветом:

– если необходим двойной щелчок левой клавишей мыши, область загорится синим;

– если необходимо кликнуть правой клавишей мыши (например, как для вызова контекстного меню) — область загорится зелёным цветом.

Будьте внимательны: если в уроке требуется двойной щелчок, а вы нажали на мышь один раз, ответ будет занесён в статистику с ошибкой.

Иногда (очень редко) может потребоваться нажатие мыши при зажатой клавише ***Shift***, ***Alt*** или ***Ctrl***. При этом подсказка загорится, как и прежде, то поверх появится надпись ***Shift*** и т.д.

В некоторых уроках реализован также механизм ***Drag and Drop***, так называемое перетаскивание объектов при помощи мыши. Вам потребуется нажать левую кнопку мыши в одной области экрана, протащить мышь по экрану и отпустить в другой области экрана. При подсказке первая область будет мигать красным, вторая — зелёным цветом.

В случае если потребуется выделить диапазон ячеек, в качестве подсказки будут указаны верхний левый и правый нижний угол. При проведении линий потребуется провести линию из начальной точки, в конечную.

При прохождении тестовых серий, при раскрытии вложенных подменю, бывает достаточно только подведения мыши, без нажатия. Такие области также будут отмечены красным цветом, но система среагирует уже при подведении мыши в некоторую область экрана, и урок пойдёт дальше.

В качестве областей для нажатия мыши могут быть не только прямоугольники, но, хотя и довольно редко, области с многоугольными границами. Или же потребуется перечислить все области на экране,

объединённые каким-либо признаком, или указать хотя бы одну из них. Если при этом вызвать подсказку, она будет оформлена в следующем виде:

– если необходимо указать несколько точек-областей на экране (в произвольном порядке), то такие области будут отмечены красным цветом;

– если важен порядок перечисления точек, при подсказке каждая область будет снабжена номером 1, 2, 3..;

– если достаточно указать одну область из всего набора (например, действие можно сделать любым способом среди альтернативных), такие области будут отмечены каждая оранжевым цветом.

2. Если же понадобится нажать клавишу или их комбинацию на клавиатуре, курсор принимает чёрно-белую форму. При вызове подсказки система ***TeachPro*** нарисует окошко, где отобразятся те клавиши, на которые требуется нажать для продолжения занятия.

Некоторые клавиши потребуется нажать в сочетаниях с ***Shift***, ***Ctrl***, ***Ctrl+Shift*** или ***Alt***.

Иногда при ответе потребуется нажать не одну клавишу, а ввести целую строку текста, если это обусловлено ситуацией в уроке. Например, ввести адресную строку текста, значение ячейки таблицы, или просто ввести число, которое только что видели на экране.

На экране отметится белый прямоугольник, в котором необходимо ввести этот текст последовательным нажатием клавиш, и нажатие каждой клавиши отобразится в строке текста на экране. Также можно получить подсказку по очередной клавише.

Обычно такой ввод нечувствителен к регистру, но в случае смены языка появится сообщение, если надо переключить клавиатуру с английского на русский.

3. Вопросы с вариантами ответов. По ходу лекции могут возникать текстовые вопросы с четырьмя или менее вариантами ответа. В таком случае надо будет указать мышью текст правильного ответа. Если же в качестве ответа на вопрос предлагается «да» или «нет», необходимо указать правильный вариант мышью. При необходимости можно воспользоваться подсказкой, и область правильного ответа отметится красным.

Закладки

Система закладок дает Вам возможность отмечать те места в лекциях, которые Вас особенно заинтересовали. Поскольку процесс обучения весьма индивидуален, то и наборы закладок у каждого свои. Когда на первом экране Вы выбираете свое имя, подгружаются

именно Ваши закладки. Вот несколько случаев, когда нужно поставить закладку:

1. Часть материала урока так и осталась непонятой. Отметьте это место, и в дальнейших лекциях, где такой материал будет использоваться на других примерах и в ином контексте, Вы сможете сопоставить приемы работы и разобраться в сложной теме.

2. Вам показали какой-нибудь интересный прием или метод, и Вы поняли, где в своей работе сможете его использовать. Перед началом проектирования можно быстро перелистать нужные, отмеченные в лекциях элементы.

3. Когда готовятся лекции для школьников и студентов

Как их ставить? Если нажать на клавишу ***Закладка***, то урок будет приостановлен, панель управления ***TeachPro*** заменится панелью управления закладками.

Вот она.

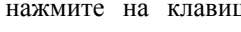
В левой части расположен список закладок, уже существующих в системе. Если названия всех закладок не поместились на экране, можно воспользоваться полосой прокрутки и найти нужную. Чтобы перейти по закладке, можно дважды щелкнуть мышью по ее названию, либо, отметив закладку (однократный щелчок мышью), нажать на клавишу ***Перейти к…*** А над списком находится комментарий к текущей (отмеченной) закладке, указывает полное название закладки, дату ее создания, а также номер лекции, на которой она поставлена.

Далее расположен набор клавиш. Они недоступны в панели управления уроком, ими можно воспользоваться только отсюда. Итак, клавиши управления системой закладок.

Главная. Возвращает Вас обратно к панели управления уроком.

Создать. Создает новую закладку. Если на эту клавишу нажать, появится небольшое меню, где в строке названия можно набрать текст, обозначающий закладку, а потом нажать на клавишу ***Да***, (если передумали, нажмите на клавишу ***Отмена***). Когда Вы добавляете в системе новую закладку, она, конечно же, появляется в общем списке.

Удалить. Если какая-то закладка больше не нужна, щелкните один раз по ее названию в списке, а потом нажмите на клавишу Удалить — и ее не станет.



Перейти к…. Область демонстрации урока изменится, отражая тот момент в лекции, на который поставлена закладка.

Продолжаем тему «Дополнительные функциональные клавиши на главной, управляющей панели курса».

Далее — переход к следующему уроку

Текст — вызывает окно с текстом лекции

Оглавление — возвращает Вас обратно к страничке с оглавлением данной лекции.

Панель с текстом лекции выглядит так

Нам осталось рассмотреть информационную панель главного экрана системы ***TeachPro***. Она имеет несколько областей, в каждой из которых содержится свой тип информации.

В самом центре управляющей панели курса находится вот такое окно.

Time — Общее время работы с уроком.

Vol. — Уровень громкости. Для того, чтобы изменить его, надо навести курсор мыши на ползунок панели, нажать левую кнопку мыши и, не отпуская ее, переместить ползунок до желаемого уровня.

Lesson — Номер текущего урока.

Message — В этой строке Вас информируют о фирме, предоставившей оболочку для данного курса, попеременно высвечивая название фирмы и адрес ее сайта.

Этот значок показывает, что после каждой лекции система остановится, и Вы сможете работать дальше по своему усмотрению.

Выбрав такой значок, Вы можете прослушать весь лекционный материал курса без перерыва. Этот режим предназначен для организации массовых лекционных занятий в больших аудиториях, где есть необходимость в непрерывном звучании нескольких лекций подряд. Выберите снова соседний значок, и после каждой лекции опять будет остановка.

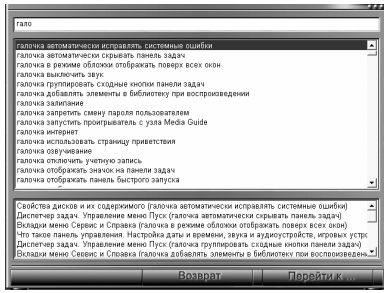
Линейка состояния — Вся она условно отображает общее время урока, а ее темная часть — уже прослушанный материал.

На линейке находятся деления. Это контрольные точки. Причем в разных режимах линейка отградуирована по-разному. Когда текущая позиция урока подойдет к такому делению (темная полоска «доползет» до очередной отметки), в пошаговом режиме изложение материала остановится, а в контрольном — после звукового сигнала Вас попросят самостоятельно произвести некоторое действие.

Клавиши ***Перейти на шаг вперед*** и ***Вернуться назад на один шаг*** переводят текущую позицию урока соответственно на одно деление вперед или назад. Для этого же предназначены деления линейки в режиме ***Фильм***.

В самом низу информационной панели Вы видите бегунок, уцепившись за него мышкой, можно перемещаться по уроку быстрее.

Поисковая система



Несколько слов о ***поисковой системе*** курса. Она чрезвычайно полезна и в процессе обучения, и в реальной работе с изучаемыми системами. Вы всегда сможете быстро и легко отыскать любой материал данного курса. От Вас требуется только четкость в формулировках.

Войдите в содержание и нажмите на клавишу ***Найти***. Перед Вами поисковая панель. Введите название искомой темы в поле редактирования или выберите ее из списка (он представлен в алфавитном порядке).

Затем выберите из нижнего окна название урока, в котором данная тема рассматривается (в этом окне даётся название раздела и урока, в котором рассматривается ключевое слово).

Далее нажмите на кнопку ***Перейти к….*** Вы окажетесь не только в нужном уроке, но и в нужном его фрагменте. То есть именно там, где описывается выбранное Вами понятие.

Устранение возможных проблем с программой

При возникновении любых неполадок в работе с программой обращайтесь в службу технической поддержки нашей фирмы:

Телефон: (495) 673-76-92

Проблемы при инсталляции программы

Проблема	Решение проблемы
Для пользователя Windows2000/XP (без прав администратора) появляется окно с красным знаком без текстового сообщения	Для установки необходимо иметь права администратора. Установите себе права администратора.
Программа каждый раз перезагружает компьютер.	После установки предлагается перезагрузка, и после этого не нужно повторять setup (установку) ещё раз (что закончится второй перезагрузкой), запускайте программу из главного меню.
Кнопки отображаются чёрными прямоугольниками	Установлен режим 16 или 256 цветов. Переключите монитор в режим с большим количеством цветов.
Диск не работает через локальную сеть.	Для обучения в локальной сети покупайте специальные сетевые версии, этот диск может работать только с CD-ROM.
Диск не загружается, например, через программу Word	Если вы плохо владеете компьютером, выполняйте только действия, предлагаемые автозапуском. Например, если вы умеете работать только в Word, и возникают трудности с установкой, не надо пытаться открывать setup.exe как кодированный текст через Word.

ВНИМАНИЕ! Программа не будет проигрываться в музыкальных центрах, так как это диск для компьютера и имеет специальный формат.

Проблемы при запуске программы

Проблема	Решение проблемы
Если программа не запускается (моргает чёрный экран)– при наличии двух CD-ROM, один из которых пишущий.	Повторите инсталляцию с устройства, буква которого меньше по алфавиту.
Не отображаются шрифты (в случае Windows98/ME).	Если у вас есть CD-ROM с коллекциями 6000, 12000, 15000 шрифтов, не устанавливайте все эти шрифты сразу(!), Windows нормально поддерживает до 500 шрифтов.
Не отображается видео (звук идёт).	Произведена усечённая выборочная установка Windows95/98, необходимо зайти в Панель управления — Установка и удаление программ — Установка Windows — Мультимедиа — установите кодеки видеосжатия.
Ошибка распаковки урока.	Возможно, не читается файл урока с CD-ROM, или же у пользователя без прав администратора нет доступа к временным файлам или к каталогу Windows.

Затруднительные ситуации при проигрывании урока

Проблема	Решение проблемы
Через 10-15 минут урок «стопорится».	Урок кончился, и необходимо загрузить следующий. При движении мыши вниз экрана появится всплывающая панель, нажмите «Оглав.».
Не играет звук.	Возможно, наушники вставлены в гнездо на передней панели CD-ROM. Вставьте наушники в гнездо звуковой платы сзади компьютера.
При запуске урока только чёрный экран..	Такое бывает, когда звук идёт, но не подключены колонки, и видео тоже идёт, но не установлены кодеки видеосжатия (в случае усечённой установки Windows). Произведите полную установку Windows.

TeachPro
WEB-дизайн

Что необходимо знать Web-дизайнеру

Мы очень рады, что Вы захотели войти в мир ***Web-дизайна***! Именно этот электронный учебник предназначен для вас! В нем Вы найдете все (или почти все), что должен знать человек, уже узнавший, что такое ***Internet***. Да, «всемирная паутина — «***WWW***», ***Web-сайт***, ***Web-страница*** — совсем недавно были окутаны туманом загадочности для большинства людей, работающих с компьютером.

А те немногие, кто имел доступ к сети, с большим восторгом рассказывают об удивительном «мире ***Internet***», в котором нет ничего невозможного, который живет своей жизнью, не зависимо от страны, правителей или отдельных людей. Можно сказать, что это двухмерная, возможно, трехмерная и красочная копия реального мира! И сейчас этот мир доступен, почти, для каждого человека, который сел за компьютер, соединенный различными каналами связи с компьютерами всего мира.

Именно на страницах ***Internet*** публикуются журналы и книги, собираются библиотеки и проводятся выставки и конференции. Установленные в разных городах мира видеокамеры позволяют транслировать события и передавать фотоизображения различных объектов. А Вы, находясь в любой точке мира, можете обращаться к информационным ресурсам на родном языке, просматривать интересующие вас журналы и слушать новости любимой радиостанции. Итак, Вам уже известно, что информационные возможности ***Internet*** практически не ограничены.А что же такое ***Web***-дизайн и с чем его едят? Есть ли необходимость Вам его изучать? Думаем, что есть, поскольку с помощью этого средства Вам можно размещать информацию о себе и своей деятельности. Вы можете публиковать свои работы (если они уже есть) и размещать любую информацию, которая у Вас имеется. Иначе говоря, изучая ***Web-дизайн***, Вы приобретете умение грамотно и эффективно работать в сети ***Internet***, реализуя свои творческие замыслы. Однако в связи с быстрым темпом развития ***Internet***-технологий некоторые важные вопросы не успевают решаться. Особенно остро это касается проблемы дизайна в ***Internet***.

Дело в том, что в основе создания ***Web***-страниц лежат простейшие языки программирования, базирующиеся на ***HTML-кодах***. ***HTML*** — это язык разметки документа, позволяющий форматировать и верстать страницы, используя самую различную информацию (текстовую, графическую, звуковую). Заметим, что совсем до недавнего времени созданием ***Web-страниц*** (что совершенно естественно) занимались только программисты. Однако вопросы эстетического характера при оформлении ***Web-страниц*** и сайтов отступали на второй план. Сейчас же требования к этим документам

настолько возросли, что к процессу их создания подключились дизайнеры-художники.

Вот и появилась новая профессия ***Web-дизайнер***. Однако ***Web-дизайнеров*** так мало, (и вы, вероятно, уже заметили) что ***Internet*** по-прежнему кишит не удобно сделанными, не привлекательными, мало интересными страницами. Появление специальных книг, освещающих вопросы художественной композиции и цветоведения, дизайна интерфейсов, шрифтов и текстов в применении к ***Web-страницам***, немного облегчило работу программистов. Вместе с тем, не имея навыков к дизайнерской деятельности, применить полученные знания на практике им бывает весьма затруднительно. Что же касается художников-дизайнеров, то при определенном желании освоить элементарные азы программирования совсем не сложно.

Тем более, что в настоящее время появляется все больше вспомогательных программных продуктов, облегчающих процесс написания кода. Кроме того, подобную проблему иногда решают очень просто. Над созданием ***Web***-документов одновременно работают программисты и художники, как же, как и многие специалисты. Исключает ли это знание методов работы напарника? Конечно же, нет. Особенно для разработки макета сайта, где решаются композиционные задачи, цветовое решение ***Web-страницы***, программирование и другие современные требования ***Web-дизайна*** для оптимального решения творческой задачи. Обращаем Ваше внимание на то, что важно для ***Web-дизайнера***. А именно, далеко не только знание языка программирования ***HTML***. Хотя, разумеется, профессиональный ***Web-дизайнер*** должен знать все тонкости и нюансы внутренней структуры своей ***Web-страницы***. По этому поводу можно заметить, что внимание к дизайну страницы и программированию должно быть разделено пополам. Поэтому, заранее говорим вам, что стоит обратить особое внимание на создание дизайна страницы.

Web-дизайнеры часто спорят о выборе программных средств компьютерной графики. Некоторые утверждают, что важны только профессиональные инструменты графического дизайна. Другие же выражают распространенную точку зрения, что важны не инструменты, а проблемы дизайна в ***Internet***. Инструменты, в конечном счете, ничего не значат. И все же, если вы хотите стать профессионалом в области ***Web-дизайна***, то вам необходимо в совершенстве владеть не только компьютерными инструментами высшего уровня, но и знаниями классического дизайна. Обойтись без него нам не представляется возможным, поскольку дизайн несет в себе различные виды информации. И мы видим в нем аналогию между языком ***Web*** и грамотным и эффективным подходом к любой визуальной коммуникации, чтобы говорить языком изобразительных форм. Итак, что вы узнаете в главе «Основы

дизайна»? Мы почти увенны, что в этом мало понятном еще ***Web*** не обойтись без классики дизайна? Без методов его осуществления, например, без основ формообразования, законов пропорционального соотношения частей объектов, композиции, цветового решения и многих других понятий и терминов.

Именно художественно-выразительные методы дизайна помогут вам реализовать творческие замыслы, проявить оригинальность мышления в процессе поиска и решения практических задач по созданию ***Web-страницы*** или сайта. Затем мы познакомим Вас с основами компьютерной графики. Учитывая, что Вы уже не новичок в ней, мы дадим только самые основные моменты этой графики, которые следует учитывать при создании той или иной страницы. Прежде всего, назначение и сферы использования компьютерной графики. Здесь, возможно, Вы узнаете для себя что-то новое. Не помешают Вам и знания видов компьютерной графики, ее основные понятия, цветовые модели и другие необходимые сведения для создания ***Web-страницы***. Ну, а знание продуктов ***Adobe*** и ***Macromedia*** сделает ***Web-дизайнера*** гораздо привлекательнее в качестве профессионала ***Web-графики***, чем знание, например, ***CorelDraw***, ***PHOTO-PAINT*** или ***Paint Shop Pro***. С другой стороны, если у Вас, как у Веб-мастера, более скромные потребности в дизайне, то любые из этих продуктов будут полезны. По нашему мнению, программы для работы с изображениями и иллюстрациями являются основным инструментом Веб-дизайнера. Именно в этих программах рождается все то, что принято называть Веб-дизайном.

Эти программы дают возможность работать с фотографиями, создавать изображения с помощью цвета и текста, сканировать их, а также анимировать, совершенствовать, оптимизировать графику. А вот дальше в учебнике мы определили три уровня обучения ***Web-дизайну***. На первом уровне Вы познакомитесь с общими сведениями о ***Web***-программировании. Сюда вошли уроки по созданию простых ***Web-страниц*** с помощью ***HTML-кода***, разметкой текста, ссылки. Вы узнаете, что такое «списки» и приемы форматирования текста, как сохранить страницу и посмотреть ее в браузере. Не менее важны для Вас сведения подготовки изображений для Вашей ***Web-страницы*** с помощью графического редактора ***Adobe Photoshop***. В Вашей работе будут также полезны и основы структурирования информации, цветовые схемы ***Web-сайта***, особенно редактор ***Makromedia Dreamweaver***.

Если Вы захотите перейти на второй уровень изучения ***Web-дизайна***, то Вы будете знать основные технические особенности ***Web***. На этом уровне Вы узнаете (или вспомните) много новых для Вас терминов — теги, атрибуты, ссылки, фреймы и т.д. В том числе мы расскажем и покажем Вам структуру ***HTML-кода***, назначение его тегов , научитесь создавать ***Web-документы*** в среде ***HTML-редактора*** и другие сведения. И, наконец, на третьем уровне Вы

станете почти ***Web-профессионалом***, если внимательно изучите вопросы разработки контента (содержания) сайта и его структуры.

Здесь Вы узнаете, что такое навигация интерфейса, сможете отличить ***Web-дизайн*** от полиграфического дизайна, познакомитесь с «клиентскими» и «серверными» программами ***Jawa Script*** и динамическим ***HTML*** и многим другим. Изучив все это, Вы получите возможность перейти к созданию своего сайта, узнав все, что необходимо для этой работы. Разумеется, Вам еще надо узнать — как разместить Ваш сайт и как дальше его поддерживать. Для чего следует познакомиться с ресурсовой поддержкой контента сайта, с требованиями стандарта к нему, чтобы он был функционален, маневренным, доступным, а также соответствовал международному законодательству.

В заключении мы проанализируем достигнутые вами результаты, дадим советы по корректировке сайта, а также мероприятия по его дальнейшему продвижению в этом направлении. Интересно и то, что в учебнике после каждого урока Вы сможете проверить свои знания тестовыми вопросами. Не проходи мимо них. Они помогут Вам объективно (в некоторой степени) оценить полученные Вами знания. Заметим, что начать обучение можно с любой главы или урока, если Вы уже кое-что знаете в работе с компьютером. Однако мы рекомендуем Вам пошагово познакомиться с материалом каждого урока, чтобы получить знания о ***Web-дизайне*** в определенной системе.

Не исключено, что вам понравится и подобранный к урокам видеоматериал, а также его звуковое сопровождение. Все это разработано нами на основе практического опыта преподавания этого курса для дизайнеров художественно-графического факультета Московского педагогического университета. Вот для чего Вам необходим данный учебник, в котором Вы по мере необходимости можете подготовить себя к более качественной работе над ***Web-страницей*** или сайтом. Итак, вступительный урок закончен. Желаем Вам удачи в изучении основ ***Web-дизайна***!

Раздел 1. Основы Дизайна

1.1. История русского дизайна

История русского дизайна. Дизайн — термин, обозначающий различные виды проектной деятельности, главной целью которых является формирование эстетических и функциональных качеств окружающей предметной среды. Английское «***design***» — производное от итальянского «***disegno***» означает не только чертеж, рисунок, замысел, проект, но и гораздо более сложные вещи, едва ли не всю область работы художника. В узком смысле дизайн — это художественное конструирование. В широком смысле — дизайн — это деятельность, направленная на создание нашего окружения, гармоничного, понятного и удобного в использовании предметного мира, в котором мы живем.

Создание предметного мира — одна из древнейших сфер художественного творчества. Уже на заре цивилизации, одновременно с совершенствованием первых орудий труда, предметов обихода формируются ранние формы художественного осмысления окружающего мира, преобразуемого человеком. На протяжении всей истории человечества развиваются различные техники художественной обработки изделий, осваиваются новые материалы, расширяется сфера деятельности художника, усложняются его задачи.

Особенности формы, орнамента, фактуры, цвета в изделиях разных народов неразрывно связаны со своеобразием их хозяйства, образа жизни, обычаев, культуры, с особенностями климатических условий, видами местных материалов и так далее. Поэтому специфика развития русского дизайна неразрывно связана с русской национальной культурой, историей, особенностями природно-климатических условий нашей страны.

Русский дизайн с 8-м до начала 17 веков. Археологические находки свидетельствуют о том, что уже в эпоху палеолита художественная деятельность первобытного человека проявляется не только в изобразительном творчестве (наскальные рисунки, пластика), но и в изготовлении и оформлении утилитарных предметов — орудий труда, бытовой утвари. В процессе совершенствования продуктов труда, постижения свойств разнообразных материалов, освоения способов и приемов обработки, отбора из них наиболее экономных, эффективных возникали первые представления о красоте изделий, формировалось универсальное эстетическое чувство человека.

Первобытные охотники, применявшие только грубые каменные инструменты, уже пытались украсить свои изделия из кости и рога простым орнаментом или изображениями животных, вырезать примитивные костяные

статуэтки. Эти древние формы искусства тесно связаны с поклонением силам природы и применялись при совершении магических ритуалов. Таково, например, «Изображение оленя. Пещерная роспись. Палеолит». Именно в эту эпоху появился первый орнамент, который стал средством художественно-образного украшения предметов. Появление орнамента связано с развитием технологии обработки различных природных материалов — камня, кости, глины.

Изобразительная символика русского орнамента зачастую была связана с религиозными представлениями славян, с языческими культами, на которые впоследствии наложилось христианское мировоззрение. Ни один элемент орнамента не был случаен, каждая черточка, каждый образ что-то обозначали. Уникальным явлением в русском орнаментальном искусстве был растительно-геометрический плетеный орнамент. В ***VIII—X*** веках искусно декорировались самые разные изделия. Художественная техника отличалась многообразием.

Поскольку Русь была по преимуществу лесной страной, дома строились из дерева, очень многие предметы быта так же изготовлялись из древесины. Например, изделия из дерева делались резными и расписными, точеными и инкрустированными. Металл ковали, чеканили, гравировали, чернили, эмалировали. Эмалевые, черневые и сканные изделия русских мастеров были известны и ценились далеко за пределами Руси.

Ткани красили, расшивали бисером, жемчугом, золотом и серебром. Объединение славянских племен, образование ими крупнейшего восточнославянского государства, принятие единой государственной религии (христианства), обусловили бурный рост художественных городских ремесел в конце ***X*** века. Обучение столяров, резчиков, золотников и других мастеров происходило, как правило, при монастырях. Так, в 1456 году в Троице-Сергиевской лавре под руководством мастера Амвросия была создана школа резчиков по дереву. Для покрытия нужд великокняжеского двора Василий ***III*** организовал в 1511 году Московскую оружейную палату — мастерские, вырабатывавшие различные предметы вооружения и быта. Основными потребителями изделий художественных ремесел явились на Руси в ***XI*** — ***XVII*** веках княжеско-боярские дворы и помещичьи усадьбы, а также монастыри и церкви, служившие в то время центрами культурной жизни.

Именно к этому периоду относится рождение и развитие нового типа поселения — усадьбы. Усадьба — комплекс жилых (хоромных) и хозяйственных помещений. Типичные хоромы — это несколько бревенчатых срубов, соединенных сенями, со временем обраставшими дополнительными постройками: прирубами, придельцами, присенцами, задцами, гульбищами. Вскоре в такой комплекс стала включаться и церковь. Вся композиция была подчиненная бытовым требованиям целесообразности и практичности. Общая

меблировка таких сооружений была пристенной. Предметы мебели были конструктивно просты, строги и относительно легки, декорировались традиционными русскими орнаментами.

Русский дизайн 17 века (петровская эпоха) ***XVII*** век ознаменовался общим подъемом производства на Руси бытовых изделий, расцветом всех видов прикладного искусства. Рынок становивится общенациональным, развивается торговля и культурные связи с Западной Европой (Польшей, Германией, Англией, Швецией и другими странами). Происходит обогащение прикладного искусства за счет новых стилевых мотивов и форм, прикладное искусство становится светским. История России конца ***XVII*** — первой четверти ***XVIII*** века неотделима от имени одного из крупнейших политических деятелей России — Петра I.

В ***XVII*** веке формируются новые принципы архитектуры и градостроительства. Постройки Петербурга являются примером комплексного решения интерьеров, представляющих в эстетическом плане единое целое. Убранство дворцовых помещений и предметы мебели начала ***XVII*** века характеризуется сдержанностью и утилитарностью решений. В ***XVII*** веке утверждается понимание необходимости художественного решения каждой отдельной вещи с учетом особенностей всей архитектурно-предметной среды. Складываются приемы оформления, характерные для разных типов помещений. Залы, приемные и другие многолюдные помещения решаются обычно очень декоративно, служебные и интимные — значительно более строго.

К середине ***XVIII*** века господствующие позиции в русском искусстве занимает так называемый стиль рококко, завершавший тенденции позднего барокко. Парадные интерьеры этого времени, например некоторые помещения Петергофского, Царскосельского и других дворцов, почти сплошь декорируются вычурной резьбой. Общие особенности орнамента рококко таковы: изогнутость линий, обильная и асимметричная компоновка стилизованных или близких к натуре цветов, листьев, раковин, глазков и т. п. Все это полностью воспроизводятся в русской архитектуре и мебели того времени, керамике, одежде, каретах, парадном оружии и т. д.

С 60-х годов ***XVIII*** века начинается переход к классицизму с его лаконичными и строгими формами, обращенными к античности и отмеченными большой сдержанностью и изяществом. В планировке, оборудовании и декоре городских особняков и дворцов появляется четкая симметричность, пропорциональная ясность. Стены помещений скрываются зеркалами и панелями из шелкового штофа, декоративных хлопчатобумажных тканей, сукна или грунтованных расписных холстов.

Полы набираются из дерева различных пород, а иногда обтягиваются холстом или сукном; потолки расписываются (например, в технике гризайль, имитирующей рельефную лепку). Переходы от стен к потолку в парадных помещениях решаются в виде карнизов, дверные проемы превращаются в порталы со скульптурными завершениями. Оконные проемы увеличиваются, получают красивые пропорции. Среди архитекторов этой эпохи можно назвать Кокоринова, Баженова, Кваренги, Старова. В конце ***XVIII*** века в оформлении крупных дворцов выявляется отделение проектирование изделий (мебели, светильников, часов, шпалер и др. утвари и предметов убранства) как особой области творческой деятельности. В роли проектировщиков выступают в основном архитекторы и профессиональные художники.

В ***XVIII*** веке при отдельных усадьбах для обслуживания возросших поместных нужд организуются крепостные мастерские, выпускающие довольно крупные партии мебели, фарфора, паласов и других изделий. В дальнейшем на базе помещичьих мастерских развиваются кустарные промыслы, а на базе купеческих — капиталистические предприятия. Именно поэтому, в производстве изделий на массовый рынок постепенно все больший вес приобретают сравнительно крупные частные мануфактурные (основанные на ручном труде) и механизированные предприятия. Таким образом, формируются объективные предпосылки для нового этапа развития прикладного искусства — индустриального периода.

1.2. История русского дизайна. Русский дизайн с *XIX* в. до наших дней

История русского дизайна. Русский дизайн с 19 века до наших дней. В ***XIX*** веке хронологически условно выделяются три основных периода развития искусства: так называемый ампир (первая четверть века, продолжения тенденций классицизма); поздний классицизм (примерно 1830—1860 годы) и эклектизм (после 1860-х годов).

Строгость и монументальность являются характерными особенностями архитектуры и форм различных предметов быта стиля ампир. Обогащаются декоративные мотивы за счет использования декоративной символики Древнего Египта и Рима — грифонов, сфинксов, фасций, военных атрибутов («трофеев»), перевитых гирляндой венков, и др. Формы становятся более монументальными из-за большей обобщенности и геометризованности классических орнаментальных мотивов — оканта, венков, лиры, доспехов и др., которые постепенно удаляются от своих реальных прообразов. Почти полностью исчезает живописная роспись предметов (сценки, пейзажи, букеты). Орнамент стремится к пятну, контурности, напоминает аппликацию.

Изделия в своем большинстве, особенно мебель, становятся крупными, массивными, но разнообразными по общей конфигурации и силуэту. С середины ***XIX*** века гарнитуры мебели по составу, типажу и размерам все более приспособляются к габаритам небольших квартир. Тяжеловесность ампира в предметах мебели постепенно пропадает.

Во второй половине ***XIX*** века происходит дальнейшее вытеснение ручного труда из производства утилитарно-бытовых изделий. Возникает противоречие с новыми экономическими тенденциями массовости и рентабельности производства вещей на рынок. Основная масса изделий представляет собой необычайно пеструю картину, в которой мелькают то явственные, то едва уловимые влияния античности, романской эпохи, готики, итальянского или французского Возрождения, искусства Византии и Древней Руси, барокко и т. д., зачастую эклектически смешиваясь в оформлении одного изделия, интерьера, здания.

В России впервые привлекают к себе внимание профессиональных художников кустарные артели и мастера, в творчестве которых еще сохранялись глубокие народные традиции. Ведутся поиски нового стиля с учетом реальной необходимости в условиях машинного производства. Так появляется новый стиль — модерн — «новое искусство» в Бельгии, Великобритании и США, «югендштиль» в Германии, «стиль Сецессионе» в Австрии, «свободный стиль» в Италии. Название его — «модерн» (от фр. ***moderne***) означало «новый, современный». Для модерна характерно практически полное искоренение всех ранее применявшихся декоративно-орнаментальных мотивов и приемов, коренное их обновление.

Традиционные карнизы, розетки, капители, каннелюры, пояски «набегающей волны» и т. д. заменяются стилизованными растениями местного произрастания (лилии, ирис, гвоздика и т. п.), женскими головками с длинными вьющимися волосами и др. Часто художественный эффект достигается за счет выразительности силуэта, членений формы, линий, как правило тонко прорисованных, как бы свободно текущих, пульсирующих. В формах изделий модерна почти всегда ощутима некоторая причудливая воля художника. Модерн принес новый подход к проектированию здания, интерьера, его обстановки. Планировка жилища — особняков, дач буржуазии — освобождается от каких-либо искусственных канонов, что было свойственно дворянским особнякам предшествующей поры. Она становится гибкой, приспосабливается к индивидуальному стилю и ритму жизни владельца.

Во многих случаях это повлекло к усиленному вниманию к функциональной стороне задачи, к комплексному решению всей архитектурно-предметной среды. Зарождаются идеи и ведутся первые опыты

создания изделий, красота которых обусловливается не их чисто декоративной проработкой, а логичностью функционального, конструктивного и технологического решения. Модерн обусловил возникновение новых эстетических и творческих принципов, ускорил появление новой художнической профессии — художественного проектирования (дизайна).

В 1918 году при отделе ИЗО Наркомпроса молодой Советской Республики образуются специальные подотделы по вопросам архитектуры и художественной промышленности. Видную роль в их создании и работе сыграл А. В. Луначарский. В стране начинается строительство электростанций, радиоцентров, рабочих клубов, детских дошкольных учреждений, домов-коммун и других общественных зданий и сооружений новых типов. Быстро растет спрос на изделия массового потребления, произведения прикладного искусства, особенно его монументальных видов, расцвет которых был связан с осуществлением ленинского плана монументальной пропаганды. Резко изменяется социальная и эстетическая направленность работы архитекторов и художников. Перед художниками всех видов искусства революция выдвинула задачи создания реалистических, проникнутых глубокой партийностью и народностью произведений.

Бурное развитие получили многообразные виды агитационно-массового искусства. В СССР в Москве в 1918 году был создан Художественно Технический Совет. В 1919 году создается Художественно Технический Совет в составе Совнаркома Труда и Обороны. В 1920 году В. И. Ленин подписал декрет о создании Государственных Высших Художественно-технических Мастерских с факультетами: архитектурный, живописи, скульптуры, текстильный, керамический, дерева и железо обработки. Создан Совет по промышленному искусству в составе ВСНХ. Тогда же в 1920 году были организованы Высшие Художественно-Технические Мастерские — ВХУТЕМАС, которые просуществовали до 1932 года.

ВХУТЕМАС — свободные художественные мастерские, основанные в 1918 в Москве на базе Московского училища живописи, ваяния и зодчества и Строгановского художественно-промышленного училища. В 1926-1927 ВХУТЕМАС был реорганизован во ВХУТЕИН. ВХУТЕИН — Высший государственный художественно-технический институт. В 1930 ВХУТЕИН был закрыт. Вместо него были созданы Московский архитектурный институт и Московский художественный институт (которому позднее было присвоено имя В. И. Сурикова).

Хронология российского промышленного дизайна: в 1923 году создан первый легковой автомобиль.

1925 год — участие в Парижской выставке, где с Российской стороны была представлена: Изба-читальня — рабочий клуб с полной предметной начинкой.

1926 год — выпушена серия малолитражных тракторов.

1935 год — метро в Москве.

1939 год — малолитражные автомобили.

1948 год — автомобиль «Победа М-20», серия трамваев. В послевоенные годы в истории дизайна наметились две тенденции. Приверженцы первого направления считали, что моральный долг дизайнеров — способствовать эстетическому развитию публики, тогда как другие, придерживаясь демократических взглядов, стремились дать публике то, что она, скорее всего, желала получить и что определялось коммерческим успехом.

1949 год — цельнометаллические вагоны поездов с откидывающимися полками.

1950 — 1960-е годы. Дизайнерские фирмы, стремительно развиваясь, использовали свой талант не только для оформления товара, но и для его упаковки, для оформления интерьеров торговых помещений, витрин, выставок, средств общественного транспорта, для изготовления фирменных знаков, логотипов, фирменных канцелярских бланков, для разработки общей концепции имиджа корпораций.

1962 год — разработка комплексных художественно-конструкторских программ (дизайн-программ или государственных стандартов, ГОСТов).

1962 год — вышло постановление Совета Министров СССР «Об улучшении качества продукции машиностроения и товаров культурно- бытового назначения путем внедрения методов художественного конструирования». Организация Всесоюзного Научно-Исследовательского Института Технической Эстетики ВНИИТЭ (с филиалами на Урале, Дальнем Востоке, Украине, Белоруссии, Литве, Армении, Грузии), внедрение методов художественного конструирования.

1964 год — появился журнал «Техническая эстетика». Художественно-конструкторским организациям принадлежит важная и ответственная роль. Социалистический дизайн устремлен к преобразованию окружающей человека предметной среды на основе социальных и эстетических идеалов коммунистического общества. Гармоничная среда труда и быта, эффективно служащая людям, воспитывающая их в идейно-нравственных принципах социалистического образа жизни, — такова коренная задача советского дизайна. Он призван сделать достоянием трудящихся высшие достижения научно-технического прогресса, формировать и развивать у них культурные потребности, разумное отношение к вещам и всей предметной среде как средству получения благ, комфорта и источнику эстетических наслаждений.

1970-е годы. В дизайнерской деятельности появилось несколько новых направлений:

1. Гуманитарный дизайн для широких слоев населения.

2. Компьютерный дизайн.

3. Создание энергоэкономичной бытовой аппаратуры.

В настоящее время художественные промыслы в нашей стране используют почти все традиционные виды и техники декоративно-прикладного искусства прошлого. География промыслов стала значительно шире, чем в прошлом веке, прежде всего за счет их возрождения в национальных республиках.

Постепенно дизайнеры стали больше уделять внимания вопросам надежности и эргономичности изделий.

Конец 1980-х — 2002 годы — быстро развивается технология компьютерного дизайна. Теперь один дизайнер может решать такие задачи, какие решить раньше было по силам лишь конструкторскому бюро. В связи с появлением персональных компьютеров быстро развивается промышленный дизайн, реклама, и многие другие направления дизайна:

а) дизайн интерьера,

б) веб-дизайн,

в) ландшафтный дизайн,

г) полиграфический дизайн.

Особенно интенсивно развивается сегодня веб-дизайн, это непосредственно связано с динамикой развития самой сети интернет. Дизайн интернет-сайтов, которые имеют самую разнообразную информационную направленность. Кроме того, информация, предоставляемая в Интернете, ориентирована как на самую широкую аудиторию пользователей, так и на специалистов. В сети Интернет существуют спортивные, новостные, научные, культурные и искусствоведческие сайты, а так же религиозно-направленные, рекламные, торговые, игровые и многие другие сайты.

1.3. Формообразование

Форму человек всегда находил у природы. С давних времен природа являлась неисчерпаемым источником творческого вдохновения, кладезем художественных образов. Ковш в виде утки или гуся, подъемный кран в виде журавля, цапли или аиста. Форма крыла самолета ассоциируется с формой птицы, с орлом, соколом, альбатросом. Геометрическая форма снежинок или морских звезд вызывает у человека положительные эмоции, ассоциируется с произведением искусства. Красота и геометрия природных форм не всегда прямолинейна и геометрически регулярна.

В природе достаточно много прекрасных криволинейных форм: спиральные формы раковин, плодов, фруктов, ягод, цветов, насекомых и так далее. Привлекательность формы может быть как в симметричности, так и в

асимметрии, в напряжении и движении, в простоте и структурной сложности. Вид красивой рукотворной формы органично связан с ее назначением (функцией), материалом, конструкцией, способом изготовления (технологией), эпохой, модой и многими другими условиями.

Считается, что форма предмета — это его геометрическая структура, организация и соотношение всех ее элементов и частей: точек, линий, поверхностей. С геометрической точки зрения форму подразделяют на плоскую и пространственную. Плоская форма или плоская фигура — это совокупность точек и линий, расположенных определенным образом в одной плоскости. К плоским формам относят, например, многоугольники, окружности, эллипсы, фигуры, составленные из лекальных кривых.

Пространственные формы или объемные фигуры — это совокупность точек, линий, поверхностей и их частей, расположенных определенным образом в пространстве. К ним можно отнести, например, шар, конус, цилиндр, пирамиду и т.п.

Известно, что красота и успех созданного дизайнером изделия состоит, прежде всего, в том, что его основная форма и ее элементы соответствуют задаче проектирования. Создавая какую-либо форму предмета, дизайнер оперирует различными средствами: пропорциями и пропорциональными соотношениями, симметрией и асимметрией; законами композиции; цветом и светом; различными видами декора и т.д. Эмоциональное воздействие природы, опыт повседневных наблюдений показывают дизайнеру, какие отношения пропорций, цвета, света и тени, пластичных форм в окружающей среде вызывают у человека то или иное впечатление.

Именно в природной среде художник находит удивительные геометрические признаки формы и ее частей. Целесообразная красота природы неисчерпаемый источник создания гармоничной и красивой формы. Именно природа проектирует любой живой или неживой организм согласно определенной геометрической схеме. Так паутина, с ее сложнейшими геометрическими переплетениями легла в основу плетения кружев многих народностей. Насекомые, грибы, цветы, деревья, птицы, рыбы, фрукты, камни, горные массивы и многие другие объекты живой и неживой природы дают дизайнеру идеи для разработки дизайн-проекта изделия.

Во всех природных объектах главенствует геометрия. Она, так или иначе, присутствует в форме и конструкции различных объектов, в том числе архитектурных, технических, а также в строении человека и животных. Среди самых распространенных типов симметрии в живой природе стоит назвать лучевую и двухстороннюю симметрию. Даже орнаменты, декорирующие изделия народного и декоративно-прикладного искусства черпают свои мотивы у природы. Известно, что форма современных самолетов и

космических кораблей ассоциируется с формой птиц, морских животных и т.д. Характерные элементы летательных аппаратов, форма фюзеляжа сравнимы с ласточкой, соколом альбатросом, дельфином и др.

Природная форма привлекает наше внимание, порождает эмоции и чувства. Такое очеловечивание всегда было традиционно для духовного отношения людей к природе. Активность природной формы является источником ее привлекательности. Иногда достаточно одного намека на природную форму, сразу создается «визуальный» язык — особый язык формы. Так, при восприятии формы подъемного крана, возникает сравнение его с жирафом. Расположение атомов внутри кристаллов, звезд в созвездиях и галактиках представляет собой картину, в основе которой лежит математическая закономерность прекрасного. Красивы современные мосты, где так хорошо поработали дизайнеры, их конструкции решают геометрические проблемы натяжения, давления и нагрузки, а так же следуют рациональности природных форм, окружающего ландшафта, геологического рельефа.

Другой пример, расстояния между этажными перекрытиями или членениями уменьшаются по мере возрастания структуры архитектурного строения в той же пропорции, как и интервалы между утолщениями ростков тростника. Если внимательно присмотреться с точки зрения геометрии форм к любому объекту, то можно заметить, что он состоит из известных геометрических фигур или простых форм к ним приближающиеся, например, квадрат, круг, прямоугольник, ромб, многоугольник, овал, трапеция и т.д. Известно, что геометрический вид формы можно определитьсоотношением величин измерения: длины, ширины, высоты, толщины, радиусами кривизны, характером поверхности, например, криволинейная, прямолинейная, топографическая и т. п.

Сложную форму можно также представить как сочетание объемных геометрических тел или сочетание плоских геометрических фигур. Простейшие фигуры играют роль «кирпичиков». Набор конструкционных геометрических «кирпичиков» закономерно называть графическим меню или палитрой форм. В этом ключе все объекты живой природы и неорганического мира можно описать одними и теми же условными понятиями, числами, геометрическими образами, одинаково понятными для каждого человека. В том числе таких понятий можно назвать геометрические: «точка», «линия», «поверхность», «плоскость», «угол», «многоугольник» и т.п. Эти абстрактные понятия происходят из древнейшей геометрии, что позволяет не только представить себе воображаемые объекты для дизайн-проектирования, но и с помощью математического абстрагирования лучше узнать о некоторых сторонах предмета, больше и глубже изучить его форму с целью ее совершенствования.

Форма и внешний облик многих вещей и объектов всегда несут в себе эмоциональный заряд, вызывая чувство радости и восхищения от характера плавных ее линий, расположения в пространстве. Мы говорим «строгая» форма, если в ней присутствуют простые геометрические тела с лаконичностью очертания. Линии строгой формы, как правило, немногочисленны и лаконичны. Строгая форма не терпит обильного декора, орнаментального и изобразительного излишества. Язык «строгой» формы скуп, но выразителен. Тонкость очертаний, плавность линий характерны для «изящной» формы. Ей свойственна художественная соразмерность, изысканная красота, грациозность. «Изящная» форма не допускает массивных составляющих частей, рубленности линий. Она стремиться к максимальной плавности, легкости, утонченности, гармоничности.

«Мягкая» форма характеризуется неопределенностью очертаний. «Мягкая» форма легко поддается давлению, меняет форму под воздействием внешних сил. Для «мягкой» формы не характерны резкие очертания, ярко выраженные плоскости. «Живая» форма выражает внешнюю энергию, подвижность. Ей свойственная большая выразительность, экспрессия. При отображении «живой» формы важно выявить движение (внешнее или внутреннее). Живая форма находится в постоянном стремлении к изменению своего качественного состояния. «Живая» форма не принимает неестественности. Она всегда связана с внешним миром.

Форму, с точки зрения пластической характеристики, дизайнеры-художники делят на «тяжелые» и «легкие». К «тяжелой» они относят форму, имеющую в своей основе куб, параллелепипед, шар. К «легкой» — сильно вытянутые вверх параллелепипед, пирамиду эллипсоид, вытянутые равнобедренные треугольники и т.п. Чтобы форма была еще легче, ее обычно разбивают на части, иногда декоративными накладками, вставками. В противопоставлении формы большую роль играет контраст. Например, форма большая и маленькая, вертикальная и горизонтальная, кривая и плоская, ровная и бугристая, простая и сложная и т.п.

1.4. Основные правила композиции

Закономерности и средства композиции. Наука о композиции изучает общие и внутренние закономерности строения форм в искусстве и дизайне, а также конкретные средства достижения их целостности и единства с содержанием вещей.

Цель композиции в дизайне — создать утилитарно оправданную форму вещи, имеющую функциональную, конструктивную и эстетическую ценность. Структура вещи, формируемая по законам композиции, получает такие функциональные и конструктивные особенности, которые наилучшим образом

отвечают назначению вещи. Композиционный поиск в художественном проектировании направлен на придание форме свойств, обеспечивающих получение потребителем полезных эффектов. Композиционный поиск в художественном проектировании направлен на придание форме свойств, обеспечивающих получение потребителем полезных эффектов. Композицию — «сочинение» — понимают в области искусства как систему построения художественного произведения. Это понятие применимо к процессу проектирования, к проекту и самому изделию.

Рассматривая различные закономерности композиции, надо познакомиться со свойствами пространственной формы материальных предметов. Под свойствами пространственной формы понимается совокупность всех ее зрительно воспринимаемых признаков: геометрический вид (конфигурация), величина, положение в пространстве, масса, фактура, текстура, цвет, светотень.

Геометрический вид — свойство формы, определяемое соотношением ее размеров по трем координатам пространства, а также характером (конфигурацией) поверхности формы. В зависимости от преобладания одного из трех основных измерений выделяются три вида формы:

1) объемный, характеризуемый относительным равенством всех трех пространственных измерений (трехмерный);

2) плоскостной, определяющийся резким (или полным) уменьшением размеров по одной из трех координат измерения;

3) линейный, для которого характерно преобладание какого-либо одного измерения над двумя другими при их относительно малой величине;

Другим признаком геометрического вида формы является прямолинейность либо криволинейность поверхности. Величина формы — свойство протяженности формы и ее элементов по трем координатам. Величина формы оценивается по отношению к размерам человека или другим формам или как соотношение величин элементов одной и тойже формы. При сопоставлении форм по величине наблюдается их равенство или неравенство.

Положение в пространстве — свойство формы, определяемое ее местонахождением среди других форм, а также относительно наблюдателя в системе трех координатных плоскостей: фронтальной, профильной и горизонтальной. Взаимное расположение форм в пространстве по отношению друг к другу и зрителю рассматривается и по другому признаку. Они могут быть расположены в отношении друг друга или зрителя ближе, дальше, выше, ниже, слева, справа и т.п. Форма может располагаться и на различных уровнях по отношению к линии горизонта, т. е. на уровне горизонта, выше или ниже ее.

Одна или несколько форм по отношению к другим могут быть расположены на одном или нескольких уровнях. Сочетание указанных типовых положений

дает сложные ситуации композиционного расположения форм в пространстве. При решении многих композиционных задач большую роль играет учет зрительного восприятия массы. Оно зависит от многих факторов. Большое значение имеют размеры и форма того или иного предмета.

«Зрительная масса» — свойство формы, определяемое визуальной оценкой количества вещества (материала), заполняющего пространство в пределах видимой геометрической формы. Как и при анализе других свойств формы, здесь можно установить степени массивности, зависящие от различных условий. Большей по величине форме зрительно соответствует и большая масса, если примерно одинаковы все другие их свойства и условия восприятия. Восприятие массы изменяется и в зависимости от геометрического вида формы. Наибольшей «зрительной массой» обладают формы, приближающиеся к кубу и шару, и все те, измерения которых по трем координатам равны между собой или близки к равным. Минимальной массой обладают формы, приближающиеся к линейным.

Изменение массы формы зависит, кроме того, от цвета, фактуры и текстуры материала, из которого она сделана, и от величины предмета или элементов, соседствующих с ней. Увеличение массы наблюдается при сопоставлении с данной формой предметов или деталей меньших размеров. При увеличении сопоставляемых деталей масса того же предмета уменьшается. Все эти изменения массивности форм иллюзорные, а не фактические и часто используются и проектировании изделий.

Большое значение в восприятии форм имеет фактура — свойство, характеризующее внешнее строение поверхности формы (шероховатая, гладкая др.). Фактурность материала зависит от плотности и величины микроискажений поверхности. Фактура создает зрительный образ изделия и выступает одним из основных источников осязательной информации. Именно различие фактур дает нам возможность различать на черно-белом изображении воду и металл, снег и бумагу.

При разработке художественно-конструкторского проекта любого изделия выбор фактуры поверхности столь же важная задача, как и выбор материала. Один и тот же материал может выглядеть весьма по-разному при различной ***о***бработке его поверхности. Фактура выступает активным свойством поверхности, способным влиять даже на восприятие пропорциональных отношений формы.

Восприятие фактуры зависит и от характера освещения поверхности. Используя в изделии ту или иную фактуру поверхности, проектировщик должен учитывать те конкретные условия, при которых она будет восприниматься: удаленность от наблюдателя, характер, угол, яркость, цветность освещения и др. В восприятии формы важное значение играет и

текстура — наблюдаем на поверхности внешние признаки структуры материала, из которого предмет изготовлен. Наиболее часто текстурой (рисунком) характеризуются изделия из дерева и ткани.

Рисунок текстуры древесины изменяется в зависимости от направления обработки. В выявлении текстуры значительную роль играет цвет, особенно разница (контраст) в естественной окраске волок древесины. Фактура и текстура представляют собой активные средства художественоой выразительности. Эффект фактуры и текстуры используется прежде все для того, чтобы передать естественные качества материала, раскрыть его эстетическое своеобразие. Если фактура или текстура материала очень выразительны, то их воздействие на наблюдателя может быть сильнее, чем воздействие самой формы изделия. Фактура и текстура поверхностей должны подбираться с учетом размеров изделия и величины пространства, в котором оно будет функционировать

1.5. Цвет в дизайне

Цвет — свойство тел вызывать то или иное зрительное ощущение в соответствии со спектральным составом отражаемого или излучаемого ими света. По другому говоря, цвет — это результат отражения света от поверхности предмета и восприятия части светового излучения зрительным аппаратом человека.

Цветоведение — наука, определившаяся как самостоятельная отрасль знаний лишь в прошлом столетии, привлекала исследователей давно. Среди них можно назвать титанов эпохи Возрождения — художников: Леонардо да Винчи, Леона Батиста Альберти, Вазари, Дюрера; физиков и ученых эпохи просвещения и нового времени: Ломоносова, Ньютона, Гельмгольца и пр. Цвет обладает такими основными характеристиками, как цветовой тон (различные оттенки цвета), насыщенность (степень яркости цвета), светлота (отражающая способность цветовой поверхности).

Крайние цвета спектра — вишнево-красный и сине-фиолетовый — несколько схожи: в одном чувствуется фиолетовость, в другом — красноватость. Если смешать их, получаются промежуточные по цветовому тону так называемые пурпурные, или пурпуровые, цвета. Добавив пурпурные цвета к спектральным (поместив их между вишнево-красными и сине-фиолетовыми), спектр можно замкнуть в кольцо, то есть представить его в виде круга, который так и называют цветовой круг. Цветовой круг можно разделить на две части так, чтобы в одну часть вошли красные, оранжевые, желтые и желто-зеленые цвета, а в другую — голубо-зеленые, голубые, синие, фиолетовые. Первые из них называют теплыми цветами, вторые — холодными. Отнесение цветов к теплым и холодным основано на том, что

красные, оранжевые, желтые цвета напоминают цвет огня, солнечного света, накаленных предметов; голубые, синие, фиолетовые цвета напоминают цвет воды, воздушной дали, льда.

Понятие о теплоте цвета в известной мере относительно. Любой теплый цвет по сравнению с еще более теплым может оказаться холодным и, наоборот, любой холодный цвет рядом с еще более холодным может оказаться теплым. Теплые и холодные цвета играют большую роль в дизайне.

Все разнообразие цвета можно свести к трем основным рядам:

1) ряд серых ахроматических тонов в пределах от белого до черного цвета;

2) хроматический ряд (цвета спектра), который можно разделить по следующим признакам:

а) теплая гамма: желтый, оранжевый, красный и их промежуточные состояния;

б) холодная гамма: зеленый, синий, фиолетовый и их промежуточные состояния;

в) дополнительные цвета: синий — оранжевый, зеленый — красный, фиолетовый — желтый. Дополнительные цвета располагаются в круге спектральных цветов (или цветовом круге) диаметрально, друг против друга;

3) ряды, идущие от хроматических (спектральных) цветов к ахроматическим, например: от зеленого к белому, от зеленого к серому, от зеленого черному.

Изменение цвета по указанным признакам создает бесконечное его разнообразие. Цвет может быть использован в качестве активного композиционного средства. Светотень — свойство, характеризуемое распределением светлых и темных участков на поверхности формы. Распределение светотени обусловлено формой предмета, рельефом его поверхности и освещением. Светотень облегчает зрительное восприятие объема и рельефа, способна обобщить или расчленить объем и поверхность предмета.

Вообще рельеф предмета и его трехмерная форма воспринимаются прежде го благодаря градациям и переходам от более освещенных участков к менее освещенным. Наиболее богаты нюансами переходы света и тени на мягко освещенных предметах. Форма изделия воспринимается отчетливо, если освещенные места и тени ее поверхности соответствуют реальной композиционной взаимосвязи элементов и частей предмета.

При неблагоприятном направлении света форма зрительно разрушается: наблюдатель видит только набор светлых и темных пятен. Отсутствие теней (бестеневое освещение) лишает округлую форму объемности, этому, если условия освещения предмета будут только такие, следует изменять форму или

исправлять ее, привлекая для этого такие средства, как цвет, фактура поверхности и т. п. На хорошо обработанной поверхности часто возникают светлые блики, которые в совокупности образуют так называемый световой каркас поверхности.

Форма светового каркаса должна быть согласована с формой предмета. При проектировании изделий с полированной поверхностью сложной формы следует обязательно испытывать получающийся световой каркас в условиях различие освещения. Фактурная поверхность исключает блики — характерную особенность гладких поверхностей. Приближая источник света к предмету вплотную, можно добиться более контрастных световых отношений с сильными рефлексами и с густой (плотной) тенью. На криволинейной поверхности, в тех местах, где падает скользящий свет, фактура выделяется яснее.

С удалением источника света светотеневые градации исчезают, «материальность» формы уменьшается, и с определенного момента предмет воспринимается как силуэт, лишенный объема и деталей. Свойства формы не изолированы друг от друга. Форма характеризуется их совокупностью и единством. Анализируя взаимосвязи между элементарными свойствами, мы изучаем более сложные закономерности объемно-пространственных форм, а именно композиционные, или художественные. Важнейшая предпосылка высокого художественного качества изделий — единство всех элементов их формы, т. е. их соразмерность и соподчиненность. Средствами приведения первичных свойств формы к композиционному единству являются пропорции, масштабность, ритм, контраст и нюанс. Применение этих средств композиции должно подчиняться функциональным и конструктивным требованиям, предъявляемым к изделиям, а также требованию оптимальной взаимосвязи изделия со средой и человеком.

Нужно помнить, что любые композиционные приемы не самоцель, а только средство для выражения в форме существенных, содержательных свойств изделия — его назначения, особенностей устройства, конструкции и др.

Раздел 2. Компьютерная графика — что это такое?

2.1. Немного из истории

Немного из истории компьютерной графики. Практически с самого момента появления компьютеров появилась и компьютерная графика. В настоящее время можно сказать, что нет ни одной области в деятельности человека, где бы она ни применялась. Редко какой фильм обходится без компьютерной графики, не говоря уж о рекламе, издательском деле, анимации и компьютерных играх. Виртуальная реальность находит свою нишу в индустрии развлечений и видеоиграх.

Число виртуальных галерей и развлекательных парков быстро растет. Практически ни одно производство не обходится без компьютерной графики. Такая отрасль как космическая, пользуется компьютерной графикой с самого её появления, особенно с приходом автоматических пилотируемых аппаратов.

"Классическая" векторная графика до сих пор используется в различных приложениях бизнеса, включая разработку концепции, тестирование и создание новых продуктов. Можно считать, что первые системы компьютерной графики появились вместе с первыми цифровыми компьютерами. Сейчас ее рассматривают как средство, которое обеспечивает мощную взаимосвязь между человеком и компьютером, заставляя компьютер говорить с человеком на языке изображений.

Прошло несколько лет, пока компьютерная графика стала основным средством связи между человеком и компьютером, постоянно расширяющим сферы своего применения. Проект "вихрь" Массачусетского технологического института был отмечен как начало эры компьютерной графики. "Вихрь" стал основой создания опытного образца командноуправляемой системы воздушной защиты, разработанной как средство преобразования данных, полученных от радара, в наглядную форму.

В конце шестидесятых — начале семидесятых в области компьютерной графики начали работать новые фирмы. Если ранее для выполнения каких-либо работ покупателям приходилось устанавливать уникальное оборудование и разрабатывать новое программное обеспечение, то с появлением разнообразных пакетов программ, облегчающих процесс создания изображений, чертежей и интерфейсов, ситуация существенно изменилась. За десятилетие системы стали настолько совершенны, что почти полностью изолировали пользователя от проблем, связанных с программным обеспечением.

В конце семидесятых в компьютерной графике произошли значительные изменения. Появилась возможность создания растровых дисплеев, имеющих множество преимуществ: вывод больших массивов данных, устойчивое, немерцающее изображение, работа с цветом. Впервые стало возможным получение цветовой гаммы. Растровая технология в конце семидесятых стала явно доминирующей. Наиболее знаменательным событием в области компьютерной графики было создание конце семидесятых персонального компьютера.

В 1977 году компания ***Apple*** создала ***Apple-II***. Появление этого устройства вызывало смешанные чувства: графика была ужасной, а процессоры медленными. Однако персональный компьютер стимулировали процесс разработки периферийных устройств. Конечно, персональный компьютер развивались как важная часть машинной графики, особенно с появлением в 1984 году модели ***Apple Macintosh*** с их графическим интерфейсом пользователя. Первоначально областью применения персонального компьютера были не графические приложения, а работа с текстовыми процессорами и электронными таблицами, но его возможности как графического устройства побуждали к разработке относительно недорогих программ как в области ***CAD***/***CAM***, так и в более общих областях бизнеса и искусства.

К концу 80-х программное обеспечение имелось для всех сфер применения: от комплексов управления до настольных издательств. В конце восьмидесятых возникло новое направление рынка на развитие аппаратных и программных систем сканирования, автоматической оцифровки. Оригинальный толчок в таких системах должна была создать магическая машина ***Ozalid***, которая бы сканировала и автоматически векторизовала чертеж на бумаге, преобразуя его в стандартные форматы.

Однако акцент сдвинулся в сторону обработки, хранения и передачи сканируемых пиксельных изображений. В 90-х стираются отличия между компьютерной графикой и обработкой изображения. Машинная графика часто имеет дело с векторными данными, а основой для обработки изображений является пиксельная информация. Еще несколько лет назад каждый пользователь требовал рабочую станцию с уникальной архитектурой, а сейчас процессоры рабочих станций имеют быстродействие, достаточное для того, чтобы управлять как векторной, так и растровой информацией.

Кроме того, появляется возможность работы с видео. Прибавьте аудиовозможности — и вы имеете компьютерную среду мультимедиа. Все области применения — будь то инженерная или научная, бизнес или искусство — являются сферой применения компьютерной графики. Возрастающий потенциал персональных компьютеров и их громадное число

— порядка 100 миллионов — обеспечивает устойчивый рост индустрии в данной отрасли. Графика все шире проникает в бизнес — сегодня фактически нет документов, созданных без использования какого-либо графического элемента.

Художники, архитекторы, дизайнеры, уже не мыслят своей работы без использования компьютерной графики. Трехмерная графика позволяет смоделировать архитектурный объект и позволяет оценить его достоинства более объективно, чем это, возможно, сделать на основе чертежей или макетов. Дизайнер по интерьерам сейчас может предложить заказчику почти фотографическое изображение его будущего жилья, тогда как раньше, возможно было довольствоваться только эскизами.

Особенно часто в повседневной жизни мы сталкиваемся с векторными изображениями. Почти на любом изделии есть логотип компании-изготовителя. Разрабатывается логотип в векторах. Но нельзя и переоценивать возможности компьютера. Ведь это всего лишь инструмент, каким бы совершенным он не был. Компьютер лишь облегчает работу человека с графическими изображениями, но не создает их. Сначала можно было создавать лишь простые векторные объекты — изображения, состоящие из, так называемых, «векторов» — функций, которые позволяют вычислить положение точки на экране или бумаге. Например, функция, графиком которой является круг, прямая линия или другие более сложные кривые.

С развитием компьютерной техники и технологий появилось множество способов выполнения графических изображений. Примерно в 1995 году в России появились свои разработчики мультимедиа программ, зародились электронные издательства. Качественный уровень программных продуктов, выполненных российскими художниками и программистами не уступал, а иногда и превосходил качество программ зарубежных авторов. Ещё одним направлением современной компьютерной графики стал «Веб-Дизайн». С 1995 года во всём мире наблюдается развёртывание глобальной мировой компьютерной сети — Интернет.

Интернет является самым большим в мире хранилищем информации и связывает сегодня почти 80% всех компьютерных систем мира. По нашему мнению, Интернет стал новым направлением для компьютерных художников-дизайнеров. По своему жанру он очень близок к книжной и журнальной графике. Заметим, что художественная графика, предназначенная для Интернет, должна быть лаконична. Это связано, прежде всего, с ограничениями в скорости передачи данных по телефонным и кабельным сетям, через которые осуществляется связь между компьютерами.

Однако этого вполне достаточно для того, чтобы осуществлять передачу видеоданных. На этом урок закончен. На следующих уроках мы поговорим с вами о видах компьютерной графике.

2.2. Основные виды компьютерной графики

Под видами компьютерной графики подразумевается способ хранения и отображения изображения на плоскости монитора. Как и в любом другом искусстве в компьютерной графике есть свои специфические виды графических изображений.

К ним относятся: растровое изображение, векторное изображение, трехмерное изображение и фрактальное изображение. Есть еще и символьное изображение. Но о нем мы говорить не будем, потому что оно устарело. И на сегодняшний день практически не используется. Дизайнеры работают с каждым изображением по-разному, используя различные графические пакеты программ.

Начнем знакомиться с растровой графикой. Надо сказать, что этот вид графики наиболее распространен и связанно это в первую очередь с особенностями восприятия человеком изображения. Свет, отражённый от поверхности предмета проецируется на сетчатку глаза, где он воспринимается миллионами светочувствительными клетками глаза. Происходит кодирование светового сигнала, он разбивается на множество частей, которые в свою очередь попадают в мозг, где и воспринимается как объёмный предмет. Тот же процесс напоминает и растровая графика при демонстрации на мониторе компьютера, только в обратном порядке. Растровая графика напоминает нам лист клетчатой бумаги или шахматную доску, на которой любая клетка закрашивается определенным цветом, образуя (в совокупности) рисунок. Основной элемент растровых изображений точка — она еще называется пиксель. Точка или пиксель — это основной минимальный элемент растрового изображения.

Его мы можем сравнить с одной клеточкой бумаги. Из множества пикселей (клеточек) и состоит растровое компьютерное изображение. А вот растр — это сетка или матрица, которая состоит из точек (пикселей). Растр имеет очень много различных характеристик, которые фиксируются компьютером. Запомните две важные характеристики: размер и расположение пикселей — характеристики, которые фиксируются компьютером. Файл растровых изображений должен их сохранить, чтобы создать картинку.

Еще одна характеристика — это цвет. цвет — тоже важная характеристика для растровых изображений. Так, например, изображение описывается конкретным расположением и цветом каждой точки сетки. Вы видели

мозаичное панно? Так вот, в растровой графике эти действия похожи на создание изображения в технике мозаики. Более подробно о растровой графике мы поговорим на третьем уроке, который так и называется растровая графика.

Следующий вид изображения является векторным. Чем же интересно векторное изображение, используемое в компьютерной графике? Во-первых, с помощью векторной графики можно решить много художественно-графических задач. Во-вторых, возможность масштабирования векторного изображения без потери качества может быть ценна, например, при создании большой по размеру рекламы. Увеличение или уменьшение объекта производится увеличением или уменьшением соответствующих коэффициентов в математических формулах. Любое векторное изображение можно представить в виде набора векторных объектов, расположенных определенным образом друг относительно друга.

Векторное изображение можно сравнить с аппликацией, состоящей из кусочков цветной бумаги, наклеенных (наложенных) один на другой. Однако, в отличие от аппликации, в векторном изображении легко менять форму и цвет составных частей. Векторный графический объект включает два элемента: контур и его внутреннюю область, которая может быть пустой или иметь заливку в виде цвета, цветового перехода (градиента), или мозаичного рисунка. Контур может быть как замкнутым, так и разомкнутым.

Контур в векторном объекте выполняет двойную функцию. С помощью контура можно менять форму объекта. Контур векторного объекта можно оформлять (тогда он будет играть роль обводки), предварительно задав его цвет, толщину и стиль линии. Именно этот вид изображений в компьютерной графике называют объектно-ориентированным. Почему? А потому, что каждый элемент изображения представляет собой отдельный объект, у которого можно изменить контур, заливку цветом, пропорции. Возможность редактирования (изменения) контура может применяться при работе над дизайном изделия из стекла, керамики, и, вообще, пластичных материалов. Очень хорошо применять векторное изображение при разработке орнамента (в круге, квадрате, полосе, овале) для украшения декоративного изделия (слайд-шоу из орнаментов).

Разработав всего один элемент орнамента, его можно много раз повторить (размножить) без дополнительной прорисовки, сэкономив много времени для другой работы. Особенно важно, что векторное изображение изначально позволяет выполнять точные геометрические построения, следовательно, чертежи и другую конструкторскую документацию

К большому сожалению, векторный формат становится невыгодным при передаче изображений с большим количеством оттенков или множеством

мелких элементов, например, фотографий. Ведь каждый мельчайший блик в этом случае будет представляться не совокупностью одноцветных точек, а сложнейшей математической формулой или множеством графических элементов (примитивов), каждый из которых, является формулой. Все это приводит к большому файлу. Файлы растровых изображений имеют гораздо больший размер, чем векторные, так как в памяти компьютера каждый из объектов этой графики сохраняется в виде математических уравнений. В то же время как параметры каждой точки в файле растровой графики задаются индивидуально. Вот откуда такие огромные размеры файлов в этой графике. Остается добавить, что наиболее популярными графическими программами, предназначенными для обработки векторных изображений, являются ***Adobe Illustrator*** и ***CorelDRAW***.

Одной из наиболее интересных, в то же время сложных видов изображений в компьютерной графике является трехмерное изображение (или его еще называют — трехмерной графикой). Заметим, что трехмерная графика уходит своими корнями и имеет много общего с векторной компьютерной графикой.

Она также может называться объектно-ориентированной. Это позволяет изменять как все элементы трехмерной сцены, так и каждый объект в отдельности. Применяется она при разработке дизайн-проектов интерьера, архитектурных объектов, в рекламе, при создании обучающих компьютерных программ, видео-роликов, наглядных изображений деталей и изделий в машиностроении и т. д. В трехмерной графике изображения (или персонажи) моделируются и перемещаются в виртуальном пространстве, в природной среде или в интерьере, а их анимация позволяет увидеть объект с любой точки зрения, переместить в искусственно созданной среде и пространстве, разумеется, при сопровождении специальных эффектов.

Эти свойства трехмерной графики позволяют создавать и кинопродукцию профессионального качества. Интересно, что в процессе разработки трехмерной графики и ее анимации человек выступает в качестве режиссера и оператора, поскольку ему приходится придумывать сюжет, содержание и композицию каждого кадра и распределять движение объекта или объектов сцены не только в пространстве, но и во времени. Что же требует трехмерная графика от человека? Конечно же, умение моделировать различные формы и конструкции при помощи различных программных средств, а также знания ортогонального (прямоугольного) и центрального проецирования. Последняя — называется перспективой.

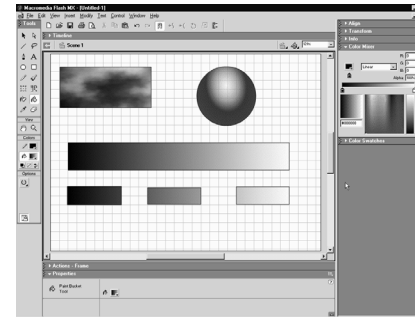
И последний вид компьютерной графики это фрактальное изображение. Фрактальная графика является на сегодняшний день одним из самых быстро развивающихся перспективных видов компьютерной графики. Математической основой фрактальной графики является фрактальная

геометрия. В основу метода построения изображений во фрактальной графике положен принцип наследования от, так называемых, «родителей» геометрических свойств объектов-наследников. О каждом из представленных видов компьютерной графики будет более подробно рассказано на следующих уроках, а этот урок закончен.

2.3. Растровая графика

Начнём знакомиться с растровой компьютерной графикой. Программный инструментарий ее наиболее развит и прост для усвоения. Способ выполнения изображения позволяет имитировать привычную работу с помощью графических инструментов, таких как, карандаш, уголь, сангина, ластик, кисть и многих других, а также позволяет передать фактуру бумаги или холста, ткани или металла. С помощью средств растровой графики можно выполнять учебные и творческие задания по композиции и рисунку. Кроме того, широкие графические, цветовые и колористические возможности программного инструментария растровой графики позволяют легко изменять цветовые и тоновые отношения, что ценно для решения живописных задач.

Вам уже известно, что растровые изображения напоминают лист клетчатой бумаги или шахматную доску, на которой любая клетка закрашена определенным цветом, образуя в совокупности рисунок. Пиксель — основной элемент растровых изображений, это одна клеточка. Именно из совокупности пикселов и состоит растровое изображение. Растровые изображения обладают множеством характеристик, которые должны быть фиксированы компьютером. Размеры изображения и расположение пикселов в нем, это две основные характеристики, которые файл растровых изображений должен сохранить, чтобы создать картинку. Еще — цвет. Например, изображение описывается конкретным расположением и цветом каждой точки сетки, что создает изображение примерно также как в мозаике.



Растровая графика зависит от разрешения, поскольку информация, описывающая изображение, прикреплена к сетке определенного размера. Разрешение — это количество пикселов на единицу длины, чаще всего на дюйм — ***dpi***, причем, чем выше разрешение, тем больше пикселов помещается в дюйме и тем качественней изображение. Основными понятиями,

непосредственно связанными с понятием пиксель считают понятия, определяющие то количество оттенков, в диапазоне которых точка может изменять свой цвет.

Они кодируются 24 ***bit*** на точку — это примерно 16 500 000 цветов. Этот режим называют «Настоящий цвет». Кодирование в 16 ***bit*** на точку позволяет различать 65 536 оттенков цвета. Этот режим получил название «Качественный цвет». Кодирование в 8 ***bit*** на точку позволяет различить всего 256 оттенков цвета. Этот режим известен как «Фиксированные цвета». Эти понятия непосредственно связаны со второй группой понятий «Цветовые форматы», о которых мы будем говорить на следующих уроках.

При редактировании растровой графики, качество ее представления может измениться, ведь меняются сами пиксели. В частности, изменение размеров растровой графики может привести к «разлохмачиванию» краев изображения, поскольку пиксели будут перераспределяться на сетке. К сожалению, масштабирование таких картинок в любую сторону также обычно ухудшает качество. При уменьшении количества точек теряются мелкие детали и деформируются надписи (правда, это может быть не так заметно при уменьшении визуальных размеров самой картинки — т.е. сохранении разрешения).

Добавление пикселей приводит к ухудшению резкости и яркости изображения, т.к. новым точкам приходится давать оттенки, средние между двумя и более граничащими цветами. Вывод растровой графики на устройства с более низким разрешением, чем разрешение самого изображения, то же понизит его качество. Не смотря на эти недостатки, только растровая графика эффектно представляет реальные образы. Реальный мир состоит из миллиардов мельчайших объектов и человеческий глаз как раз приспособлен для восприятия огромного набора дискретных элементов, образующих предметы, поэтому растровые изображения выглядят реально, конечно, если они были получены с высоким разрешением.

Помимо естественного вида растровые изображения имеют другие преимущества. Устройства вывода, такие как принтеры, для создания изображений используют наборы точек, поэтому растровые изображения могут быть очень легко распечатаны. Таким образом, растровое представление обычно используют при сканировании и обработке графических изображений с большим количеством деталей и оттенков, например, фотографий, при создании изображений для использования в других программах, в частности для передачи другим пользователям по сети ***Internet***, при создании различных художественных эффектов, которые возможны благодаря специальным программным фильтрам. Самые известные программы растровой графики —

Adobe Photoshop и ***Corel PHOTO-PAINT***. А теперь мы зададим вопрос себе. В каких случаях лучше использовать растровую графику?

Во-первых, как уже говорилось, способ выполнения изображений в этом виде графики позволяет имитировать привычную работу с помощью графических инструментов: карандаша, угля, сангины, ластика, кисти. В растровом изображении можно передать фактуру бумаги или холста, ткани или металла.

Во-вторых, широкие графические, цветовые и колористические возможности растровой графики позволяют легко изменить цветовые или тоновые отношения изображения — обычно при сканировании и обработке графических изображений с большим количеством деталей и оттенков. Ну, например, фотографий. Заметим, что этот вид графики часто используют при создании изображений для других программ. Например, для передачи другим пользователям по сети интернет.

В третьих, растровая графика незаменима при создании самых различных художественных эффектов, которые возможны только благодаря специальным программным фильтрам. Каждый объект растрового изображения находится в одном из слоев, имеющих прямоугольную форму. Слой можно представить в виде набора небольших квадратных ячеек, одинаковых по размеру, в которых можно сформировать некоторое изображение (растровый объект), состоящее из мозаичных элементов (пикселей).

Пиксель характеризуется не только цветом, но и прозрачностью при наложении элементов друг на друга. В случае, когда растровое изображение состоит из одного слоя, его можно сравнить с витражом, состоящим из небольших квадратных цветных стекол, или же с узором, вышитым крестиком. Растровые форматы файлов предназначены исключительно для сохранения растровых изображений. К числу наиболее популярных относятся следующие: ***BMP***, ***PCX***, ***TIFF***, ***CPT***, ***PSD***, ***GIF*** и ***JPEG***.

Форматы CPT и ***PSD*** используют для сохранения многослойных изображений, а форматы ***GIF*** и ***JPEG*** применяют главным образом при работе в ***Internet***, (они обеспечивают приемлемое качество изображений при небольших размерах файлов). В зависимости от того, какую обработку изображения планируется выполнить, может возникнуть потребность представления его в том или ином виде (растровом или векторном).

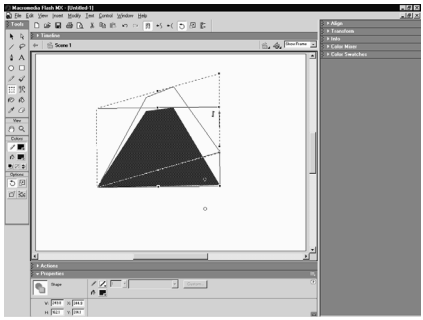
Для преобразования растровых изображений в векторные и наоборот используются соответствующие функции программ векторной графики, а также специализированные программы трассировки ***Adobe Streamline*** 4.0, ***CorelTRACE*** 9. Операция трассировки заключается в формировании в автоматическом или ручном режиме векторного изображения, являющегося копией исходного растрового. Создаваемое изображение состоит из отдельных

векторных объектов, раскрашенных определенными цветами и расположенных определенным образом друг относительно друга. Операция преобразования векторного изображения в растровое называется растрированием.

2.4. Векторная графика

Векторная компьютерная графика имеет на сегодняшний день очень широкую область применения в самых различных областях человеческой деятельности, начиная от рекламы на страницах газеты и заканчивая разработкой проектов в таком виде промышленности как космическая.

Особенно важно, что векторное изображение изначально позволяет выполнять точные геометрические построения, следовательно, чертежи и другую конструкторскую документацию. Заметим, что почти все системы автоматизированного проектирования в свою основу берут векторную компьютерную графику. Надо ли говорить, что векторная компьютерная графика наряду с растровой широко применяется художниками-дизайнерами и связанно это в первую очередь с особенностями этого вида компьютерной графики.



Векторное изображение (векторная графика) представляется в виде совокупности отрезков прямых (векторов), а не точек, которые применяются в растровых изображениях. Векторная графика описывает изображения с использованием прямых и изогнутых линий, называемых векторами, а также параметров, описывающих цвета и расположение. Например, изображение описывается точками, через которые проходит линия, создавая тем самым контур. Цвет задается цветом контура и области внутри этого контура. Остановимся на описании отличительных особенностей векторных и растровых изображений — объектов.

Каждое такое изображение может состоять из одного или нескольких графических объектов соответствующего типа. Векторный графический объект включает два элемента: контур и его внутреннюю область, которая может быть пустой или иметь заливку в виде цвета, цветового перехода (градиента), или мозаичного рисунка. Контур может быть как замкнутым, так и разомкнутым. В векторном объекте он выполняет двойную функцию. Во-первых, с помощью контура можно менять форму объекта. Во-вторых, контур

векторного объекта можно оформлять (тогда он будет играть роль обводки), предварительно задав его цвет, толщину и стиль линии.

Под стилем оформления линии подразумевается набор штрихов и полосок, из которых она состоит, а также параметры этой линии в точках перегиба и на концах. Любое векторное изображение можно представить в виде набора векторных объектов, расположенных определенным образом друг относительно друга. Векторное изображение можно сравнить с аппликацией, состоящей из кусочков цветной бумаги, наклеенных (наложенных) один на другой. Однако, в отличие от аппликации, в векторном изображении легко менять форму и цвет составных частей.

Таким образом, векторная графика в основном «живет» созданием новых объектов, широко используется в дизайнерских проектах, в то время как растровая графика изначально создавалась и существует для обработки фотографий, это видно и из самих названий программ — ***Adobe Photoshop***, ***Corel PHOTO-PAINT***. Следующая группа понятий раскрывает содержание «векторной компьютерной графики».

В основе векторной компьютерной графики лежит расчёт координат экранных точек, входящих в состав линии контура изображения, поэтому этот вид компьютерной графики называют вычисляемым. В основе векторной графики лежат математические представления о свойствах геометрических фигур (в основном этот процесс затрагивает расчёты, связанные с представлением линии). Векторная графика использует для построения изображений координатный способ. Основным базовым понятием в векторной компьютерной графике является — линия.

Для её математического представления используются: точка, прямая, отрезок прямой, парабола, отрезок параболы, функция $y = x3$, кривая второго порядка, кривая третьего порядка, кривая Безье. Соподчинёнными для понятия линия являются: точка, конец кривой линии, управляющая линия касательная к кривой, изгиб кривой, маркер управляющей линии. Также, как и линия, основным базовым понятием для векторной графики является объект.

Объектом называется любой графический элемент внутри векторного изображения, состоящий из отрезка прямой или кривой линии или замкнутого контура. При редактировании элементов векторной графики изменяются параметры прямых и изогнутых линий, описывающих форму этих элементов. Можно переносить элементы, менять их размер, форму и цвет, но это не отразится на качестве их визуального представления.

Векторное представление именно и заключается в описании элементов изображения математическими кривыми с указанием их цветов. Еще один пример: красный эллипс на белом фоне будет описан всего двумя

математическими формулами — прямоугольника и эллипса соответствующих цветов, размеров и местоположения.

Как следствие, еще одно преимущество — качественное масштабирование в любую сторону. Увеличение или уменьшение объектов производится увеличением или уменьшением соответствующих коэффициентов в математических формулах. Еще один плюс — векторная графика не зависит от разрешения, т.е. может быть показана в разнообразных выходных устройствах с различным разрешением без потери качества. Но, к сожалению, векторный формат становится невыгодным при передаче изображений с большим количеством оттенков или мелких деталей (например, фотографий). Ведь каждый мельчайший блик в этом случае будет представляться не совокупностью одноцветных точек, а сложнейшей математической формулой или совокупностью графических примитивов, каждый из которых, является формулой.

Это приводит к утяжелению файла. Теперь поговорим о проблеме размеры файлов. Файлы векторных изображений имеют гораздо меньший размер, чем растровых, так как в памяти компьютера каждый из объектов этой графики сохраняется в виде математических уравнений, в то время как параметры каждой точки (координаты, интенсивность, цвет) описываются в файле растровой графики индивидуально, отсюда — такие огромные размеры файлов. Наиболее популярными графическими программами, предназначенными для обработки векторных изображении, являются ***Adobe Illustrator*** и ***Corel DRAW***. Чем же интересно векторное изображение, используемое в компьютерной графике?

Во-первых, с помощью векторной графики можно решить много художественно-графических задач. Во-вторых, возможность масштабирования векторного изображения без потери качества может быть ценна, например, при создании большой по размеру рекламы. Увеличение или уменьшение объекта производится увеличением или уменьшением соответствующих коэффициентов в математических формулах, так как любое векторное изображение можно представить в виде набора векторных объектов, расположенных определенным образом друг относительно друга. Векторная компьютерная графика также может решить многие художественно-графические задачи.

Возможность масштабирования векторного изображения без потери его качества может быть очень ценна, например, для создания учебных плакатов. Векторная компьютерная графика является объектно-ориентированной, т.е. каждый элемент изображения является отдельным объектом, которому можно изменить контур, заливку, пропорции. Это ее свойство может быть использовано, например, при изучении законов композиции: ритма, поиска

сюжетно-композиционного центра, симметрии и асимметрии, параллельности в композиции, и пр.

Возможность простого редактирования контура может быть применима для работы над линейным рисунком, дизайном изделий из стекла, керамики и других пластичных материалов. Для декоративно-прикладного искусства векторная компьютерная графика позволяет работать над орнаментальной композицией в круге, квадрате или полосе. Разработав рапорт или элемент орнамента, векторная графика позволяет размножить его без дополнительной прорисовки, что особенно ценно в процессе творческого поиска.

Изначально этот вид компьютерной графики позволяет проводить любые геометрические построения. Это свойство необходимо для таких специальных дисциплин, как техническая графика. С помощью инструментария векторной компьютерной графики можно изучать все разделы черчения и начертательной геометрии. Используя системы автоматизированного проектирования, в основу которых положена векторная компьютерная графика, можно выполнять конструкторскую документацию любого уровня сложности. Типичным примером такой системы является пакет программ ***AutoCAD***. К средствам работы с векторной графикой, которые могут быть использованы для поддержки графических дисциплин, относят такие графические редакторы как, ***Corel DRAW***, ***Macromedia Freehand***, ***Adobe Illustrator***, ***AutoCAD***, ***ArhiCAD***. На следующем уроке мы поговорим о трёхмерной графике так как она вобрала в себя многое от векторной графики и от растровой. А этот урок закончен.

2.5. Трехмерная графика

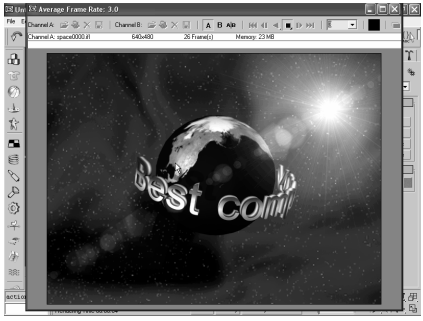
На этом уроке мы поговорим об одной из наиболее интересных, в то же время сложных видов изображений в компьютерной графике — это трехмерное изображение (или его еще называют — трехмерной графикой). Этот вид компьютерной графики вобрал в себя очень много из векторной, а так же и из растровой компьютерной графики.

Но начнём сначала. Применяется она при разработке дизайн-проектов интерьера, архитектурных объектов, в рекламе, при создании обучающих компьютерных программ, видео-роликов, наглядных изображений деталей и изделий в машиностроении и др. Трёхмерная компьютерная графика позволяет создавать объёмные трёхмерные сцены с моделированием условий освещения и установкой точек зрения.

Для изучения приёмов и средств композиции, таких как, передача пространства, среды, светотени, законов линейной, воздушной и цветовой перспективы здесь очевидны преимущества этого вида компьютерной графики над векторной и растровой графикой. В трехмерной графике изображения (или

персонажи) моделируются и перемещаются в виртуальном пространстве, в природной среде или в интерьере, а их анимация позволяет увидеть объект с любой точки зрения, переместить в искусственно созданной среде и пространстве, разумеется, при сопровождении специальных эффектов.

Трёхмерная компьютерная графика, как и векторная, является объектно-ориентированной, что позволяет изменять как все элементы трёхмерной сцены, так и каждый объект в отдельности. Этот вид компьютерной графики обладает большими возможностями для поддержки технического черчения. С помощью графических редакторов трёхмерной компьютерной графики, например ***Autodesk*** 3***D Studio***, можно выполнять наглядные изображения деталей и изделий машиностроения, а также выполнять макетирование зданий и архитектурных объектов, изучаемых в соответствующем разделе архитектурно-строительного черчения. Наряду с этим может быть осуществлена графическая поддержка таких разделов начертательной геометрии как, перспектива, аксонометрические и ортогональные проекции, т.к. принципы построения изображений в трёхмерной компьютерной графике частично заимствованы из них. Для декоративно-прикладного искусства трёхмерная компьютерная графика предоставляет возможность макетирования будущих изделий с передачей фактуры и текстуры материалов, из которых эти изделия будут выполнены.



Возможность увидеть с любых точек зрения макет изделия до его воплощения в материале позволяет внести изменения и исправления в его форму или пропорции, которые могут быть уже невозможны после начала работы (например, ювелирные изделия, декоративное литьё из металла и др.). В том же направлении трёхмерная компьютерная графика может быть использована для поддержки скульптуры, дизайна, художественной графики и др. Объёмная трёхмерная анимация и спецэффекты также создаются средствами трёхмерной графики. Создание учебных роликов для обучающих программ может стать основным применением этих возможностей трёхмерной компьютерной графики.

К средствам работы с трёхмерной графикой, относят такой графический редактор как, 3***D Studio MAX***. Это один из самых известных трёхмерных редакторов, он часто используется при создании фильмов. Разработка программы 3***D Studio*** MAX была начата в 1993 году. Версия 3***D Studio*** MAX 1.0 вышла в 1995 году на платформе ***Windows NT***.

Уже тогда некоторые эксперты осторожно высказывали мнение, что MAX может конкурировать с другими пакетами трехмерной графики. Осенью 2003 года ***discreet*** выпускает 3***d max*** 6. Новые инструменты анимации частиц в связке с модулями позволяют создавать фотореалистичные атмосферные эффекты. Появились встроенная поддержка капельно-сетчатых объектов, полноценная сетевая визуализация, импорт данных из CA***D***-приложений, новые возможности для моделирования. Но кроме 3***D Studio MAX*** есть и другие не менее популярные программы трёхмерного моделирования, например ***Maya***. ***Maya*** — эта программа аналог 3***D Studio MAX***, но она предназначена в первую очередь для анимации и, например, для передачи мимики на лице трёхмерного актёра, так же в ***Maya*** удобнее рисовать. 3***D Studio MAX*** направлен в первую очередь на качественную визуализацию предметов, ещё в нём можно выполнять примитивные чертежи.

Вообще для черчения существуют свои программы трёхмерного моделирования, самые известные из них ***AutoCAD***, ***ArhiCAD***. ***AutoCAD*** предназначен в первую очередь для машиностроительного черчения, а ***ArhiCAD*** для архитектурного моделирования. Что же требует трехмерная графика от человека?

Конечно же, умение моделировать различные формы и конструкции при помощи различных программных средств, а также знания ортогонального (прямоугольного) и центрального проецирования. Последняя — называется перспективой. Очень хорошее качество моделирования достигается при помощи тщательного подбора текстур и материалов в сочетании с правильным размещением в сцене источников освещения и камер. Основой для построения любой пространственной формы является плоскость и грань объекта. Плоскость в трехмерной графике задается с помощью трех точек, соединенных отрезками прямых линий.

Именно это условие дает возможность описать с помощью получаемых плоскостей «пространственную сетку», которая представляет собой модель объекта. Затем объекту дополнительно присваиваются характеристики поверхности объекта — материал. В свою очередь, материал характеризует качество поверхности, например, полированная, шероховатая, блестящая и др. Описывается и его текстура (камень, ткань, стекло и др.). Задаются и оптические свойства, например, прозрачность, отражение или преломление световых лучей и т.д. Наряду с этим, трехмерному объекту можно задать условия освещения и выбрать точку зрения (камеру) для получения наиболее интересного наглядного изображения. Постановка, состоящая из трехмерного объекта, условий освещения и выбранной точки зрения называется «трехмерной сценой». А вот для описания трехмерного пространства и

объекта, находящегося внутри его, используется хорошо уже знакомый Вам координатный метод.

Существуют различные методы моделирования трехмерных объектов. Например, метод текстового описания модели с помощью специальных языков программирования «Скрипт». Но об этом мы поговорим на одном из следующих уроках.

2.6. Фрактальная графика

Последней из рассматриваемых видов компьютерной графики — это фрактальная графика. Фрактальная графика является на сегодняшний день одним из самых быстро развивающихся перспективных видов компьютерной графики. Математической основой фрактальной графики является фрактальная геометрия. Здесь в основу метода построения изображений положен принцип наследования от, так называемых, «родителей» геометрических свойств объектов-наследников.

Понятия фрактал, фрактальная геометрия и фрактальная графика, появившиеся в конце 70-х, сегодня прочно вошли в обиход математиков и компьютерных художников. Слово фрактал образовано от латинского ***fractus*** и в переводе означает «состоящий из фрагментов». Оно было предложено математиком Бенуа Мандель-бротом в 1975 году для обозначения нерегулярных, но самоподобных структур, которыми он занимался.

Фракталом называется структура, состоящая из частей, которые в каком-то смысле подобны целому. Одним из основных свойств фракталов является самоподобие. Объект называют самоподобным, когда увеличенные части объекта походят на сам объект и друг на друга. Перефразируя это определение, можно сказать, что в простейшем случае небольшая часть фрактала содержит информацию обо всем фрактале. В центре фрактальной фигуры находится её простейший элемент — равносторонний треугольник, который получил название «фрактальный». Затем, на среднем отрезке сторон строятся равносторонние треугольники со стороной, равной (1/3***a***) от стороны исходного фрактального треугольника. В свою очередь, на средних отрезках сторон полученных треугольников, являющихся объектами-наследниками первого поколения, выстраиваются треугольники-наследники второго поколения со стороной (1/9a) от стороны исходного треугольника.

Таким образом, мелкие элементы фрактального объекта повторяют свойства всего объекта. Полученный объект носит название «фрактальной фигуры». Процесс наследования можно продолжать до бесконечности. Таким образом, можно описать и такой графический элемент, как прямую. Изменяя и комбинирую окраску фрактальных фигур можно моделировать образы живой

и неживой природы (например, ветви дерева или снежинки), а также, составлять из полученных фигур «фрактальную композицию».

Фрактальная графика, также как векторная и трёхмерная, является вычисляемой. Её главное отличие в том, что изображение строится по уравнению или системе уравнений. Поэтому в памяти компьютера для выполнения всех вычислений, ничего кроме формулы хранить не требуется. Только изменив коэффициенты уравнения, можно получить совершенно другое изображение. Эта идея нашла использование в компьютерной графике благодаря компактности математического аппарата, необходимого для ее реализации. Так, с помощью нескольких математических коэффициентов можно задать линии и поверхности очень сложной формы.

Итак, базовым понятием для фрактальной компьютерной графики являются «Фрактальный треугольник». Затем идет «Фрактальная фигура», «Фрактальный объект»; «Фрактальная прямая»; «Фрактальная композиция»; «Объект-родитель» и «Объект наследник». Следует обратить Ваше внимание на то, что фрактальная компьютерная графика, как вид компьютерной графики двадцать первого века получила широкое распространение не так давно.

Её возможности трудно переоценить. Фрактальная компьютерная графика позволяет создавать абстрактные композиции, где можно реализовать такие композиционные приёмы как, горизонтали и вертикали, диагональные направления, симметрию и асимметрию и др. Сегодня немногие компьютерщики в нашей стране и за рубежом знают фрактальную графику. С чем можно сравнить фрактальное изображение?

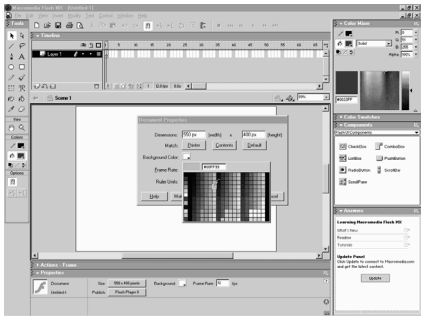
Ну, например, со сложной структурой кристалла, со снежинкой, элементы которой выстраивается в одну сложную структуру. Это свойство фрактального объекта может быть удачно использовано при составлении декоративной композиции или для создания орнамент. Сегодня разработаны алгоритмы синтеза коэффициентов фрактала, позволяющего воспроизвести копию любой картинки сколь угодно близкой к исходному оригиналу.

С точки зрения машинной графики фрактальная геометрия незаменима при генерации искусственных облаков, гор, поверхности моря. Фактически благодаря фрактальной графике найден способ эффективной реализации сложных неевклидовых объектов, образы которых весьма похожи на природные. Геометрические фракталы на экране компьютера — это узоры, построенные самим компьютером по заданной программе. Помимо фрактальной живописи существуют фрактальная анимация и фрактальная музыка.

Создатель фракталов — это художник, скульптор, фотограф, изобретатель и ученый в одном лице. Вы сами задаете форму рисунка математической формулой, исследуете сходимость процесса, варьируя его параметры,

выбираете вид изображения и палитру цветов, то есть творите рисунок «с нуля». В этом одно из отличий фрактальных графических редакторов (и в частности — ***Painter***) от прочих графических программ. Например, в ***Adobe Photoshop*** изображение, как правило, «с нуля» не создается, а только обрабатывается. Другой самобытной особенностью фрактального графического редактора ***Painter*** (как и прочих фрактальных программ, например ***Art Dabbler***) является то, что реальный художник, работающий без компьютера, никогда не достигнет с помощью кисти, карандаша и пера тех возможностей, которые заложены в ***Painter*** программистами.

2.7. Цветовые модели и палитры



Цветовая модель — это представление цвета путем его разложения на базовые компоненты. В цветовой модели для каждого оттенка, входящего в цветовую гамму модели, определяется доля цветовых компонентов, которые характеризуют выбранную модель. Наиболее распространенными цветовыми моделями являются следующие модели: ***RGB*** (***Red*** — красный, ***Green*** — зеленый, ***Blue*** — синий), ***CMYK*** (***Cyan*** — голубой, ***Magenta*** — пурпурный, ***Yellow*** —желтый, ***Black*** — черный) и ***HSB*** (***Hue*** — оттенок, ***Saturation*** — насыщенность, ***Brightness*** — яркость).

Большинство цветовых моделей описывают цвета, входящие в цветовое пространство соответствующего устройства. В частности, модель ***RGB*** используется при воспроизведении цветов на экране монитора, сканировании, а также печати на цветном струйном принтере. Модель ***CMYK*** применяют в полиграфии. Давайте поговорим про цветовую модель ***RGB***.

Глаз человека воспринимает длины волн в диапазоне от 400 до 700 нм. Длину волны от 430 нм до 470 нм человеческий глаз воспринимает как синий цвет, в диапазоне от 500 до 540 нм, как зеленый цвет и в диапазоне от 620 до700 нм, как красный цвет. В компьютерной промышленности эти цвета называются тремя первичными цветами. Для их обозначения и используется цветовая модель ***RGB***.

Все цвета, встречающиеся в природе, можно создать, смешивая свет трех этих длин волн, варьируя их интенсивности. Смесь, состоящая из 100% каждого цвета, дает белый свет. Смесь 0% от каждого цвета дает отсутствие света или черный свет. Искусство воспроизведения цвета путем сложения в

различных пропорциях трех первичных ***RGB*** цветов называется аддитивным смешением.

Путем проекции трех цветов: красного, зеленого и синего на светлую поверхность можно плучить большенство цветоввидемой области спектра, например голубой, пурпурный и желтый. Этот принцип используется для воспроизведения цвета в видео и в компьютерных мониторах. Будучи определена природой компьютерных мониторов, система цветов ***RGB*** является самой популярной и распространенной.

И это естественно, так как аппаратные средства, которые вы используете, явно или неявно влияют на характер и процесс вашей работы с цветом. Однако именно в этом проявляется и недостаток системы ***RGB***, она исключительно зависима от устройства. При замене устройства изменяются и цвета. Она не очень подходит для воспроизведения цвета, когда в одном комплексе должны работать сканер, принтер и монитор.

Поскольку она использует три аддитивных первичных цвета, она не подходит для раскраски или для красителей и пигментов, используемых при печати, поскольку те используют другой набор первичных цветов (голубой, пурпурный, желтый). Цветовые модели ***CMYK***. Три вторичных цвета ***CMY*** при смешивании создают черный цвет. Выражение цвета путем вычитания из белого света одной из компонент называется субтрактивным смешением.

Краски или красители создают цвет субтрактивным методом: когда краситель или пигмент поглощает красный и отражает зеленый и синий свет, мы видим голубой. Когда он поглощает зеленый и отражает синий и красный, мы видим пурпурный. Голубой, пурпурный и желтый являются тремя базовыми цветами, используемыми в субтрактивном смешении. Цвета в таких светящихся устройствах, как телевизоры и компьютерные мониторы формируются путем смешивания трех первичных цветов ***RGB***, но такие средства воспроизведения цвета, как печатные издания и картины работают на поглощении одних длин волн и отражение других три вторичных цвета ***CMY***. Поскольку реальные чернила ***CMY*** не создают чисто черного цвета он получается темно коричневым, а также при создании черного цвета с помощью модели ***CMY*** тратится в три раза больше красок, то к этим трем цветам добавляется отдельно черный цвет (К) и модель называется ***CMYK***. Диапазон представления цветов ***CMYK*** хуже, чем в ***RGB***, поэтому при преобразовании данных из ***RGB*** в ***CMYK*** цвета получаются более темными и приглушенными.

Для расширения цветового охвата применяют цветоделение на шесть красок. Модель ***HexaChom*** в ней кроме цветов ***CMYK*** используются дополнительные -***Green*** и ***Orange*** (зеленый и оранжевый). В современной

цифровой, широкоформатной и офсетной печати используется система 12 цветов.

Цветовые модели ***CIE***. Эти цветовые модели были разработаны Международной комиссией по освещению. Они основаны на реакции человеческого глаза на ***RGB*** и спроектированы так, чтобы точно представлять восприятие цвета человеком. Эти модели используются для того, чтобы определять так называемые аппаратно независимые цвета, которые могут правильно воспроизводиться устройствами любого типа: сканерами, мониторами и принтерами. Эти модели широко применяются, потому что их легко использовать на компьютерах и они описывают широкий диапазон цветов.

Системы цветов ***HSB*** и ***HSL*** Системы цветов ***RGB*** и ***CMYK*** базируются на ограничениях, накладываемых аппаратным обеспечением (мониторами компьютеров в случае ***RGB*** и типографскими красками в случае ***CMYK***). Более интуитивным способом описания цвета является представление его в виде тона, насыщенности и яркости — система ***HSB***.

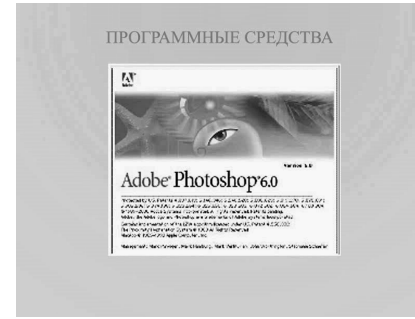
Она известна также как система ***HSL*** (тон, насыщенность, освещенность). Тон представляет собой конкретный оттенок цвета, отличный от других: красный, зеленый, голубой и т. п. Вы можете видеть как изменяется цвет фотографии с натюрмортом при воздействии на нее цветового тона. Насыщенность цвета характеризует его относительную интенсивность (или чистоту). Уменьшая насыщенность фотографии с натюрмортом мы делаем цвета более пастельными или блеклым и размытым, а увеличивая насыщенность получаем более яркие или ядовитые и резкие цвета.

Яркость (или освещенность) цвета показывает величину черного оттенка, добавленного к цвету, что делает фотографию с натюрмортом более темной или уменьшая что делает ее более светлой. Система ***HSB*** хорошо согласуется с моделью восприятия цвета человеком. Тон является эквивалентом длины волны света, насыщенность — интенсивности волны, а яркость — общего количества света. Система ***HSB*** имеет перед другими системами важное преимущество: она больше соответствует природе цвета. Недостатком этой системы является то, что для работы на мониторах компьютеров ее необходимо преобразовывать в систему ***RGB***, а для четырехцветной печати — в систему ***CMYK***.

Цветовые палитры. При использовании цветовых моделей цвет может быть задан путем ввода его числовых значений. В данном случае цвет выбирается из библиотеки (в графической программе таких библиотек несколько), и можно руководствоваться только названием цвета и его представлением на экране.

Во многих графических программах, особенно в тех, которые специализируются на обработке векторных изображений, предусмотрена возможность использования каталогизированных цветов не только при обработке изображений, но и при выводе на печать. Это означает, что при помощи программы можно создать цветоделенные пленки, каждая из которых предназначена для печати краской того цвета, который был выбран из библиотеки при создании изображения. Качество печати при этом возрастает, поскольку печать выполняется красками с чистым цветом и растрирование изображений не производится. На этом урок про цветовые модели и палитры закончен.

2.8. Вспомним A*dobe Photoshop*



Вспомним ***Adobe Photoshop***. На данном уроке мы поговорим об особенностях такой программы как ***Adobe Photoshop***. Сразу надо сказать, что программа ***Adobe Photoshop*** является электронным редактором и программой обработки растрового изображения, в частности фотографий переведённых в электронный (растровый) вид, а не программой для рисования, хотя и обладает большими возможностями в данной области.

Именно по этому разговор пойдёт о ней, а не о других не менее популярных редакторах растрового изображения. Что такое растровое изображение вы должны помнить из третьего урока данной главы, поэтому останавливаться на нём не будем. Ещё надо сказать, что у фирмы ***Adobe*** кроме программы ***Photoshop*** есть ещё очень много продуктов, вообще программы подобного вида не распространяются поодиночке они идут в пакете программ.

В программный пакет такой фирмы как ***Corel*** входят следующие редакторы: графический редактор векторной графики ***Corel Draw***, редактор растровой графики ***Corel PHOTO-PAINT***, программа конвертирования растровых и ображений в векторные ***Corel TRACE*** и другие. В программный пакет ***Adobe*** входят такие программы как: ***Adobe Photoshop***, программа растровой графики, программа векторной графики ***Adobe Illustrator***, ***Adobe Imaeg Ready*** программа по подготовки изображения к печати и другие.

Изначально эти программы были предназначены для обработки фотографических изображений, оцифрованных в растровую графику, коррекции цветовых и тоновых отношений; добавления оптических эффектов,

которые трудно выполнить с помощью традиционных фототехнологий. Основное назначение этих программ состояло в допечатной подготовке изображений для последующей их публикации в книгах, газетах и журналах. ***Adobe Photoshop*** стал индустриальным стандартом. В последнее время без него не обходится ни одно издательство. Были также «переписаны» для ***PC*** известные программы вёрстки для настольных издательских систем «Адоб Пайджмэйкер» (***AdobePagemaker***) и «Кварк Экспресс» (***QuarkXPress***). Так почемуже программы ***Adobe*** и в частности ***Adobe Photoshop***? Сами создатели называют ***Photoshop*** — «фотокамерой фантазии», и это действительно так. Какая бы фантастическая картина ни возникла в голове, с помощью мощных инструментов можно воплотить ее в жизнь.

В повседневной практике этой программой пользуются люди самых различных профессий: дизайнеры, художники, создатели ***Web***-страниц, фотографы. С ее помощью можно создавать собственные произведения искусства, ретушировать и корректировать уже готовые изображения, изменять отсканированные рисунки и создавать макеты печатных изданий. Данная программа также позволяет производить цветокоррекцию, коллажирование, всевозможные трансформации, цветоделение и другое.

Она располагает различными методами работы с точечными изображениями, при этом умело справляется со слоями и использует контуры. Ну и, конечно же, надо упомянуть о ***Image Ready***, который теперь встроен в ***Photoshop***. Это сделано из расчетов на то, что в данный момент очень много графики создается под ***Web***. Это программа для Веб-дизайна, ключевой "игрок" среди профессиональных инструментов для создания ***Web***-графики.

Изучение курса Веб-дизайн невозможно без знания этой программы, это доказало проводимое исследование среди студентов ХГФ. Большая часть того, что принято называть дизайном для Веб-страниц делается именно в этой программе, она также является связующим звеном между остальными программами компьютерной графики и анимации. Ниже перечислен ряд возможностей программы ***Adobe Photoshop***, которые применяются в Веб-дизайне.

1. Создание растровой графики, которая подходит для оптимизации ***Web***-изображений.
2. ***Web***-безопасная палитра, полезная при оптимизации графики для ***Web*** (появилась в версия 4.0).
3. Использование слоев — мощного средства работы с изображениями, позволяющее эффективно компоновать элементы Веб-страницы.
4. Функция экспорта ***GIF***89, позволяющая создавать прозрачные и чередующиеся ***GIF***-изображения.

5. Полный набор фотографических операций и фильтров, улучшающих качество

снимков и позволяющих делать в них изменения и исправления по вашему желанию.

6. Мощные функции типографского набора и возможности фильтрации,

такие как косая, падающая тень и источники света.

Комбинируя и редактируя слои теперь с помощью ***Photoshop*** появилась возможность создания динамичных баннеров. Кроме самой программы ***Photoshop*** существуют и приложения. Приложения имеют широкие возможности для обработки текста и изображений и удобный интерфейс. Многие возможности программ являются уникальными. При работе с приложениями одной фирмы можно не заботиться о совместимости форматов, поскольку об этом уже позаботились производители.

Приложения практически идеально сочетаются, и при их совместном использовании становятся доступными многие дополнительные возможности. ***Photoshop*** — самая распространенная программа обработки растровых изображений, и поэтому ее близкие отношения с ***Illustrator*** — весомый довод для дизайнера. Популярность этой программы среди профессиональных художников и дизайнеров не имеет аналогов. Практически 100 процентов мастеров, работающих с компьютерной графикой, используют ее в качестве средства обработки растровых изображений.

В последней версии этой популярной программы по обработке графики - ***Photoshop CS*** добавлено много новых возможностей. Как известно ***Photoshop*** всегда работал только с растровой графикой, видимо в компании ***Adobe*** немного подумали и решили составить конкуренцию продуктам компании ***Corel***. Так что, теперь для работы с векторной графикой Вам не будут нужны дополнительные программные продукты. Например, появилась возможность работы с векторной графикой, имеется встроенный ***Image Ready***, увеличены возможности при работе со слоями.

В программу встроены фильтры и пакеты по обработке графики и добавления всяческих эффектов от различных производителей. Увеличенные возможности по работе со слоями позволяют художникам творить настоящие чудеса с изображением. Удобный менеджер слоев позволяет с легкостью создавать, комбинировать, устанавливать коэффициент прозрачности для слоев. А во всем остальном данный программный продукт остался таким, каким его все привыкли видеть.

И в конце остаётся сказать что о всех качествах этой программы невозможно сказать в одном уроке, и для профессиональной работы её необходимо изучить отдельным курсом.

2.9. Размеры изображения, кадрирование. Цветокоррекция

Размеры изображения, кадрирование и цветокоррекция. На данном уроке мы поговорим с вами о предпечатной подготовки изображения и о таком термине как кадрирование. Кадрирование — это процесс определения границ фотокадра, при съемке и последующего выбора формата изображения на снимке при печати. Хотя размер кадрового окна у фотоаппарата строго определенный, но соотношение сторон отпечатанного снимка может быть разлитым, что зависит от содержания сюжета. Кадрирование является одним из эффективных средств придания снимкам композиционной завершенности и, в конечном счете, художественной выразительности.

Выбирая соответствующий формат снимка, автор решает вопрос рационального размещения в границах кадра сюжетных элементов изображения. Разумеется, можно, подобно театральному режиссеру, отобрать действующих лиц задуманной сцены, расположить их среди предметов окружающей обстановки, поставить перед исполнителями задачу по их общению и, выбрав соответствующую точку съемки, полностью вписать в кадр все, что потом воплотится на снимке.

В отличие от человеческого глаза, воспринимающего картину окружающего мира целостно, фотообъектив способен воспроизводить ее лишь небольшими участками, ограниченными рамками кадра. Эта специфичность фотографического видения заставляет фотолюбителя позаботиться о том, чтобы в кадр попали не какие-либо случайные и малозначащие объекты, а основополагающие, находящиеся в тесной взаимосвязи между собой.

Это даст возможность зрителям воссоздать целостную картину происходящего в данный момент события. Всё это касается в основном людей, которые связаны с фотографиями, а как быть ***Web***-дизайнерам? Дизайнеры по своей специфике для работы используют уже готовые фотографии. Для этого существуют различные источники — это электронный клипарт, компакт диски с множеством картинок, интернет — где просто море всяческих иллюстраций, ну и просто глянцевые журналы.

Ну и конечно дизайнер должен сам уметь обращаться с фото и видео камерой — ведь бывает, что не всегда можно найти нужное изображение. Получается что ***Web***-дизайнер зачастую использует уже готовую продукцию в своей работе. И это нередко можно увидеть во всемирной паутине, но под разным углом зрения художника предоставляются одни и те же фотографии. Это делается, для того чтобы сэкономить время и средства на изготовление художественных продуктов — например ***Web***-страниц в Интернете или рекламы.

Но вернёмся к ***Web***-дизайнеру и его способам работы с фотографией. Как уже было сказано, художник-фотограф применяет кадрирование с самого начала от поиска композиции в фотообъективе до печати фотографии. Фотограф является человеком, который получаеbт изображения. А что можно сказать про дизайнера, может ли он сам делать новые фотографии? Конечно, да и это относится не только к работе с фотоаппаратом, но и к работе с компьютером. Каждый художник по своему обрабатывает изображения: задаёт индивидуальные размеры, придаёт свой тон, применяет фильтры и просто из разных частей фотографий создаёт в результате новое изображение. Теперь давайте поговорим собственно о самом кадрировании, о том как это применяется компьютерной графике. В большинстве случаев соотношение сторон исходного файла не пропорционально соотношению сторон требуемого отпечатка. Это связано с тем, что почти у всех цифровых фотоаппаратов соотношение сторон кадра 4:3, а многие стандартные форматы рассчитаны на печать с пленки, где соотношение сторон чаще всего 3:2.

Так же этот вопрос возникает при печати фотографий с фотопленок, сделанных фотоаппаратами с нестандартным соотношением сторон кадра (например, 6***x***7 для широкой пленки). В результате при печати таких фотографий встает вопрос — либо уместить изображение на отпечатке целиком за счет возникновения белых полей, либо напечатать без белых полей, но за счет потери информации. У ***Web***-дизайнеров эта проблема решается гораздо проще. С помощью программного обеспечения, в частности редактора растровой графики ***Adobe Photoshop*** можно просто вырезать из первоначального изображения необходимые части и в бедующем с ними работать.

Как это делается. Да очень просто, берётся инструмент прямоугольная область выделения и выделяется необходимая область фотографии, затем копируется в новый файл. Так же можно просто обрезать края имеющейся фотографии и пересохранить файл под другим именем и т. д. Подробную техническую сторону мы опустим, лишь скажем, что способов работы в программах очень много и каждый дизайнер работает за компьютером по-своему. Всё это касается изготовления самого изображения, но во многих случаях после кадрирование следует предпечатная подготовка, теперь давайте поговорим о ней.

Подготовка издания к печати включает в себя ряд операций, объединяемых одним термином: допечатная подготовка. Иногда говорят «предпечатная подготовка», «форпресс», но, как ни назови, это комплекс действий по подготовке рукописи (текста), включающий в себя набор, корректуру, правку, дизайн, верстку и подготовку иллюстраций. А также цветоделение, фотовывод. Заканчивается допечатная подготовка передачей подготовленных

к печати полуфабрикатов в типографию или заказчику, который потом ищет типографию. Но нас больше интересует цветокоррекция или цветокорректирование.

Цветокорректирование — это изменение цветного изображения оригинала в соответствии с требованиями заказчика, технологического процесса и других причин или исправление фотоформ, полученных в результате цветоделения. В свою очередь цветоделение — это разделение цветного изображения оригинала с помощью светофильтров или селективных источников освещения на отдельные одноцветные равномасштабные изображения. При синтезе в процессе печатания с цветоделенных однокрасочных печатных форм с определенной точностью воспроизводится цветное изображение оригинала. Цветоделенное изображение — это одноцветное изображение, полученное на экране монитора издательской системы или на твердом носителе, после разделения на отдельные цвета многоцветного изображения оригинала.

Разделение проводят при помощи фильтров, освещением узкоспектральным светом или использованием фотоматериалов с узкозональной чувствительностью (в фотооптических системах и сканерах) или вручную художником при создании цветного оригинала (создание раскладок по цветам для штриховых изображений). Разделение изображения на отдельные цвета может быть проведено и по специальным программам, с использованием вычислительной техники на базе других параметров, например, — разность температур, влажность атмосферного давления, высоты над уровнем море, глубины водоемов (применяется при создании географических, медицинских или геологических карт).

И всё это нужно для качественной печати имеющихся изображений, что бы рассчитать характеристики красок. Цветовые характеристики печатной краски — это три основных показателя: цветовой тон, насыщенность и светлота краски. В производственных условиях их контролируют денситометром. Цветовые характеристики печатных красок на оттиске зависят от вида пигмента (красочного лака), его количества в краске, степени перетира краски, связующего вещества, толщины красочного слоя на оттиске, вида запечатываемой поверхности, а также от характеристик других печатных красок (например, их совместимости при наложении) и др. Вот какой долгий путь надо пройти изображению прежде чем попасть на станицы журнала или на ***Web***-сайт.

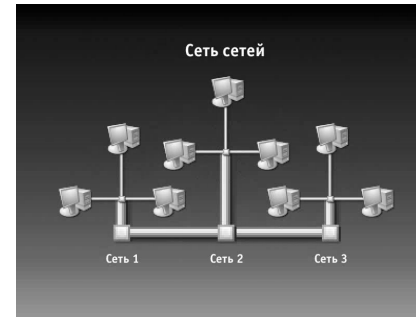
Раздел 3. Общие сведения о Web-дизайне

3.1. Основы Интернет

В этом уроке мы в самых общих чертах рассмотрим техническую сторону вопроса: познакомимся с основными концепциями Интернета, без понимания которых невозможно приступать к изучению веб-дизайна. Интернет — это компьютерная сеть. Назначение любой компьютерной сети — дать компьютерам возможность "общаться", то есть обмениваться друг с дружкой информацией. Когда такой обмен происходит нечасто (допустим, Вы выполнили какую-то работу и отвезли её показать заказчику), Вы можете воспользоваться внешним носителем: дискетой, компакт-диском и т.п.

Но представьте, например, что ваша работа требует постоянного обмена. Например, вы сидите в редакции, обрабатываете изображения и передаёте их верстальщику. Если этот верстальщик сидит в соседней комнате, то можно к нему "сбегать с дискеткой". Но если он находится в другом здании, а то и в другом городе, то нужно уже что-то другое. В прежние времена такой обмен информацией с помощью внешних носителей называли в шутку "***Floppy-net***", от английского названия дискеты: ***Floppy Disk***. Должен сказать, что "***Floppy-net***" был сомнительным удовольствием.

Итак, если компьютеры обмениваются информацией достаточно часто, то возникает необходимость объединить их в сеть. Каким образом они соединяются, сейчас не важно, но цель такого соединения — создать единое информационное пространство, в котором обмен информацией между компьютерами происходит почти так же легко, как внутри одного компьютера, как когда Вы копируете файлы с одного диска на другой. Компьютеры, объединенные в сеть, называют узлами сети.



Небольшие группы компьютеров, расположенных недалеко друг от друга, объединяют в так называемые локальные сети. Например, может существовать локальная сеть предприятия или института. Внутри такой сети узлы могут "общаться" между собой, но получить к ней доступ "извне", с компьютера, не подключенного к этой сети, невозможно. Локальные сети могут включать в себя довольно много узлов, но всё же имеется ряд ограничений, не

позволяющих построить "одну большую" локальную сеть на весь мир (поэтому они и называются локальными). Тем не менее, существует способ соединения нескольких локальных сетей между собой. Получается "сеть сетей". Тогда компьютеры из одной сети смогут через это соединение "общаться" с компьютерами другой сети.

Например, из офиса Вы сможет получить доступ к сети магазина или склада. "Сети сетей" также можно объединить между собой, а их объединения — между собой, и так далее, пока мы не получим глобальную сеть, объединяющую все компьютеры, в которой любой узел может найти "дорожку" к любому другому узлу. Тогда студент из Новосибирска сможет соединиться с библиотекой Конгресса США, а хакер из Сургута — с сервером Пентагона, как если бы все они находились по соседству. Такой глобальной "сетью всех сетей" и является Интернет. Благодаря Интернету единое информационное пространство расширилось до масштабов планеты.

Теперь перейдём к вопросам более практическим. Прежде всего, разберёмся, каким же образом компьютерам удается не заблудиться в этом "лесу" информационного пространства и отыскивать те самые "дорожки" к нужным узлам? Очевидно, что каждый узел сети должен иметь некий "адрес", по которому его можно безошибочно найти. Кроме того, нужна карта, позволяющая "добраться" по этому адресу. Например, в телефонной сети каждый абонент имеет принадлежащий только ему номер, а телефонная станция соединяет один номер с другим.

В Интернете происходит примерно то же самое. Каждый компьютер, подключенный к глобальной сети, получает уникальный для всей сети адрес, называемый ***IP***-адресом. ***IP***-адрес состоит из четырех чисел в диапазоне от 0 до 255, что теоретически позволяет закодировать чуть больше 4 миллиардов адресов. Что касается карты, то в Интернете скорее действует принцип "язык до Киева доведёт": там, где более мелкие сети соединяются в более крупные, установлены специальные устройства. Эти устройства, хотя и не содержат полной "карты" Интернета, то есть не могут знать точного маршрута до нужного вам адреса, но, проанализировав ***IP***-адрес, они подскажут, в каком направлении лучше двигаться. Там вы встретите другое такое устройство, которое также укажет вам дальнейший путь в пределах свое "зоны видимости". В конце концов, вы доберетесь до ближайшего к вашему адресату устройства, которое уже "подведёт вас за ручку" к нужному узлу.

Таким образом компьютеры в сети соединяются друг с дружкой. И с этой точки зрения они равноправны, то есть каждый может соединиться с каждым. Однако с точки зрения выполняемых функций существует два больших класса: клиенты и серверы. Назначение Серверов состоит в обслуживании клиентов.

Например существуют файловые серверы. Они служат хранилищем информации, которую клиенты могут черпать из них по мере необходимости, а не хранить у себя. Продолжая аналогию с телефонной сетью, можно сказать, что все абоненты в ней равноправны с точки зрения соединения, но когда вы собираетесь встречать родственника из другого города и звоните на вокзал, чтобы узнать расписание поездов, вы выступаете в роли клиента, а справочная служба вокзала, соответственно, в роли сервера. При этом ничто не мешает, поговорив с Вами, справочной самой превратиться в "клиента": допустим, позвонить в бухгалтерию и узнать, когда же, наконец, будут выдавать зарплату.

Клиенты и серверы не универсальны, а решают вполне определенные задачи. То есть нельзя позвонить в бухгалтерию, чтобы узнать, когда прибудет поезд; для этого нужен или другой "сервер" (не бухгалтерия, а справочная служба), или другой "клиент" (не Вы, ожидающий родственника, а сотрудник, ожидающий зарплаты). Поэтому существует масса "разнопрофильных" программ-серверов и соответствующих им программ-клиентов, которые в совокупности образуют различные системы Интернета.

3.2. WorldWideWeb. WWW, браузер, доменные имена

Так фабрикуют мысли. С этим можно
Сравнить хоть ткацкий, например, станок.
В нём управленье нитью сложно:
То вниз, то вверх снуёт челнок,
Незримо нити в ткань сольются;
Один толчок — сто петель вьются.
Иоганн Гёте. "Фауст", часть 1, сцена 4

Одной из наиболее распространенных и знакомых, наверное, Вам систем Интернета является ***Worldwide Web*** (сокращенно ***WWW***). "Всемирная паутина" или просто Веб, как её называют — это те самые, сайты, которые вы посещаете, выходя в Интернет, и создание которых называется "Веб-дизайном", чем мы с Вами и будем заниматься.

Иногда некоторые люди ошибочно отождествляют Интернет и ***WWW***, хотя это абсолютно неправильно. Веб — лишь одна из систем Интернета. Кроме неё существует электронная почта, пересылка файлов и множество других систем. ***WWW*** изобрел человек по имени Тим Бёрнерс-Ли, ныне член Английского Королевского научного общества, и даже обладатель звания рыцаря Ордена Британской Империи.

В 1989 году Бернерс-Ли работал в Европейской лаборатории элементарных частиц, что в Женеве. Бернерс-Ли задался целью разработать систему для объединения разрозненных работ по физическим исследованиям, систему для связывания частей информации. Он её успешно разработал, и эта система включала все базовые элементы современного веба.

Во-первых, язык для установления связей между разрозненными кусками информации (***HTML***, который мы будем изучать в дальнейшем).

Во-вторых, правила циркуляции информации в компьютерных сетях (так называемый протокол ***HTTP***).

И в-третьих, систему адресов (тех, что у нас у всех на слуху: ***www***, точка и т.д.).

Кроме того, Бернерс-Ли с коллегами разработал соответствующие программы: сервер и клиент для своей системы. Первый Веб-сервер для хранения фрагментов информации был запущен на компьютере ***NeXT*** все в той же Женевской лаборатории. Первый веб-клиент для доступа к этой информации, её просмотра назывался, как и сама сеть, ***WWW*** и в нем были реализованы практически все элементы современного браузера.

"Браузерами" называют веб-клиенты, т.е. программы для просмотра сайтов (от английского слова ***browse*** — пролистать, проглядеть). Браузер является "посредником" между пользователем и сервером: когда вы пишите в адресной строке браузера адрес некоего сайта, браузер отправляет запрос серверу, на котором хранится этот сайт, получает от него информацию, переводит её с языка ***HTML*** на "человеческий" и показывает Вам в виде запрошенного Вами сайта.

История развития браузеров — это история ожесточенной борьбы, взлётов и падений. Она, поистине, заслуживает отдельной главы. Исторически сложилось так, что на сегодня мы имеем одним из наиболее распространённых браузеров встроенную в операционную систему ***Windows*** программу ***Internet Explorer***.

В 1991 году проект "***WWW***" влился в тогда ещё только американскую сеть Интернет, и с тех пор, как говорится, началось… Система очень быстро набирала популярность благодаря таким своим качествам, как простота и универсальность: по задумке Бернерса-Ли размещенные в ***www*** документы должны были открываться на любом компьютере, вне зависимости от

операционной системы и возможностей вывода. Именно универсальность является одним из главных свойств ***WWW***.

В 1994 году объем циркулирующей в ***WWW*** информации превысил объем информации в других Интернет-системах, ну а какое место всемирная паутина занимает в жизни современного человека, я думаю, можно не уточнять. Находится всё меньше сфер человеческой деятельности, где бы не были протянуты её нити.

Как вы помните, адресом в сети Интернет является ***IP***-адрес. Следовательно, теоретически, чтобы дать браузеру команду показать тот или иной сайт, Вы должны указать в адресной строке ***IP***-адрес компьютера, веб-сервера, на котором хранится этот сайт. Но, что компьютеру хорошо, то человеку смерть. Если бы нам пришлось пользоваться ***ip***-адресами, возможно, всемирная паутина и не получила бы столь широкого распространения.

Вы, наверное, имеете некоторый опыт работы с Сетью и адрес сайта для Вас — это не циферки, а нечто вроде ***www***.что-то такое.***ru***. Как же подобные привычные нам веб-адреса соотносятся с реальными ***IP***-адресами серверов? Для этого была введена система так называемых доменных имён, поставившая в соответствие ***IP***-адресам "человеческие" слова, как правило, отражающие содержание сайта или его название. Система называется ***DNS*** (***Domain Name System***). Благодаря этой системе вы можете вместо 213.180.216.200 набрать адрес ***www.yandex.ru*** и попасть на один и тот же сайт. Точно так же в мобильном телефоне вам достаточно выбрать имя того, кому вы хотите позвонить, а телефонный номер будет взят из адресной книжки автоматически.

Разница в том, что в Интернете такая "адресная книжка" расположена вне вашего компьютера, а распределена по специальным ***DNS***-серверам; она едина для всех, и любой человек, набравший ***www.yandex.ru***, попадет на сайт поисковой системы Яндекс, а не, допустим, Рамблер.

3.3. WorldWideWeb-2. Сайты и страницы, гипертекст и гиперссылки

Итак, с помощью доменного имени (ну, или с помощью ***IP***-адреса, если вы питаете слабость к цифрам) мы можем обратиться к некоторому сайту "всемирной паутины". Что же происходит дальше? Как этот сайт показывается нам? Сайт состоит из так называемых страниц, и браузер показывает нам его постранично, т.е. в один момент времени мы видим на экране одну страницу сайта. Таким образом, кроме доменного имени браузер должен "знать" ещё и адрес интересующей нас страницы. С точки зрения хранения на сервере, страницы — это просто файлы, и адресом страницы является имя файла. От доменного имени оно отделяется слешом — символом косой черты "/".

Кроме того, файлы, как известно, могут быть "разложены" по папкам. В таком случае между доменным именем и именем страницы указывается "путь" к файлу. Имена папок также отделяются слешами. Наконец, существует ещё один вариант: мы вообще не указываем имя страницы, а это случается в большинстве случаев. На такой случай на сервере имеется страница по умолчанию, называемая "главной" или "домашней" страницей. Это, так сказать, парадный вход на сайт.

Описанная выше система адресов называется ***URL*** (***Uniform Resource Locator***, Единый указатель ресурсов). Эта система была разработана Тимом Бернерсом-Ли, и она позволяет однозначно определить любой документ во всемирной сети. ***URL*** включает в себя имя домена, как обязательную составляющую, "путь", т.е. имена папок, каталогов, в которых находится документ, и собственно название документа. Кроме этого ***URL*** может содержать ещё некоторые элементы, которыми мы не будем сейчас забивать себе голову.

Страница — это составная часть сайта, законченная порция информации. В рамках сайта странцы связаны между собой, как в книжном переплёте, и мы можем "перелистывать" их, переходить от одной страницы к другой, оставаясь в пределах сайта, как бы читая одну книгу.

Веб-Сайт — это совокупность связанных друг с другом "страниц", объединенных по какому-то признаку: по тематике содержащейся на них информации, по автору или по какому-либо другому. В конце концов, просто по объединению под одним доменным именем. Однако аналогия с книгой уместна лишь отчасти.

Во-первых, в отличие от книги, сайт, как правило, делится на страницы не по объёму размещаемой на них информации, а по содержанию. Название "страница" достаточно условно. Это, прежде всего, некая законченная порция информации. Страница может состоять из двух строчек, а может вмещать текст, объёмом с хороший том. Поэтому и листают страницы сайта обычно не последовательно, как в книге, а сразу переходят к странице, содержащей нужную информацию.

Во-вторых, и это главное, границы "книги", которую вы читаете, достаточно расплывчаты. Перелистнув очередную страницу, Вы запросто можете оказаться в совершенно другой книге, т.е. на другом сайте, иногда даже и не заметив этого.

Родоначальником этой идеи является Теодор Нельсон, предлагавший создать "документарную вселенную", связав все тексты, произведенные человечеством, перекрёстными ссылками. Такую совокупность связанных текстов Нельсон назвал "гипертекстом".

Другим пионером гипертекста был Даглас Энгельбарт, который, кроме всего прочего, изобрёл компьютерную мышь. Под его руководством в конце 60-х годов была разработана система, включавшая ссылочные связи между фрагментами информации.

Прародителем же гипертекста можно считать Ванневара Буша, в конце 40-х годов работавшего над машиной ***Memex*** для механизации поиска данных в научной литературе. ***Memex*** был не компьютерной системой, а электронно-механической — он использовал микрофильмы и фотоэлементы. Главной особенностью системы была возможность вводить взаимную связь элементов. И когда пользователь просматривал один из документов, он мог тут же вызвать связанный. Классическое определение гипертекста, которое дал Нельсон в 1987 г., — это "форма письма, которое ветвится или осуществляется по запросу". Иначе говоря, это "нелинейное письмо", которое "больше чем текст", т.е. сверх-текст, ***hypertext***. Более формальное определение гипертекста звучит так: гипертекст — это представление текстовой информации как сети, в которой читатели получают свободу перемещаться нелинейным образом.

Переход от одного фрагмента гипертекста к другому осуществляется с помощью гиперссылок. Идею гипертекста и гиперссылок и заложил Тим Бернерс-Ли в основу ***WWW***. Любой фрагмент страницы может быть помечен как ссылка на другую страницу данного или любого другого сайта. Увидев интересующую нас ссылку, мы можем моментально "перейти по ней", т.е. попросить браузер показать страницу, на которую эта ссылка ведёт. Благодаря гиперссылкам и "плетётся всемирная паутина": миллиарды страниц ссылаются друг на друга, и, переходя с одной на другую, мы находим нужную нам информацию.

Язык для разметки гипертекста был назван языком гипертекстовой разметки или ***hypertext markup language***, сокращенно ***HTML***. На этом языке и пишутся веб-страницы. Вы можете возразить: мы, мол, учимся веб-дизайну, а не программированию. Дизайнер должен картинки рисовать, а не программы писать.

Ну, во-первых, ***HTML*** — это не язык программирования, а язык разметки, т.е. основу его составляет обычный человеческий текст, дополненный специальными знаками. И для его освоения вовсе не нужно быть программистом.

Во-вторых, веб-дизайн далеко не ограничивается рисованием картинок. Веб дизайн — это не столько оформление информации, сколько "дизайн", так сказать, самой этой информации: её структурирование, выделение этих самых законченных фрагментов, установка между ними связей и т.д..

В-третьих, освоение веб-дизайна не обязательно начинать с изучения ***HTML***'я, также как освоение работы с компьютером не стоит начинать с двоичной системы или с внутреннего устройства машины. Мы и не будем с этого начинать.

Существуют специальные программы — визуальные ***HTML***-редакторы, которые по создаваемому нами макету сами формируют необходимый ***HTML***-код, а работа в таких программах — не намного сложнее работы в том же Ворде. Тем не менее, должен оговориться: поскольку в любом случае результатом деятельности веб-дизайнера является документ на языке ***HTML***, в профессиональной работе обойтись без его знания затруднительно. Изучению ***HTML***'я посвящена соответствующая глава, а здесь мы сделаем первые шаги в веб-дизайне с помощью одного из ***HTML***-редакторов.

3.4. Свойства документа

Свойства документа. На данном уроке мы поговорим о том из чего состоит Веб-сайт. И первое понятие на котором стоит остановиться — это Веб-страница.

С чего начинается Веб-сайт? Веб-сайт начинается с первой (главной) страницы (на англ. ***front page***). Часто ошибочно считают, что веб-сайт и веб-страница это одно и тоже. Для того чтобы разобраться в этом надо привести точные определения каждого термина. Итак, веб-страница это простой гипертекстовый документ, который содержит некую текстовую или графическую информацию для просматривания веб браузерами. Проще говоря, это файл с текстом и рисунками, который можно разместить на сети.

Обычно веб-страница имеет один уникальный адрес (***URL***) по которому пользователи Интернет могут открыть только данную страницу. Веб-сайт это

множество веб-страниц имеющих общую тему и общий интерфейс, физически размещенных в одном месте сети Интернет. Обычно несколько веб-страниц соединяются для создания одной структуры, чтобы описать организацию, индивидуального человека, мероприятие и что-либо другое. Иначе говоря веб-сайт состоит из веб-страниц, которые логически и физически соединены между собой в одну систему.

Следующее понятие — это комплектующие сайта. Так, мы более или менее разобрались с значением и сущностью веб-страниц и веб-сайтов. Учитывая то, что грубо говоря веб-сайт это несколько веб-страниц, можно перейти к вопросу составляющих веб сайтов. Каждый веб-сайт состоит из трех относительно независимых составляющих: содержания (контента), оформления (дизайна) и кода.

Создание сайта заключается как раз в разработке и состыковке всех этих частей, а его поддержание в их обновлении. Рассмотрим эти составляющие по отдельности. Контент — информационное наполнение сайта; адекватное информационное наполнение для большинства сайтов является необходимым фундаментом. Контент, пожалуй, является самой сложной и важной частью сайта. Действительно, как бы замечательно сайт не был выполнен, вряд ли неинтересное содержание привлечет множество посетителей.

Поэтому перед каждым, решившим делать сайт, встает вопрос: а чем этот сайт наполнить? Дизайн сайта это проектирование и художественное оформление сайта. Интересное содержание надо еще красиво и удобно оформить. Это включает не только рисование, чертеж и красивую графику, но и разработку максимально удобного пользовательского интерфейса, чтобы ваш посетитель получал нужную ему информацию на сайте без каких либо трудностей и максимально быстро. Дизайн сайта, как понятно из слов означает, как будет выглядеть внешне Ваш сайт.

Также на этом этапе Вы определяете полностью графическую часть. Код является скелетом всего сайта, который командует где, что и когда поставить. Обычно роль исходного кода для веб страниц играют языки гипертекстовой разметки. Самый популярный и основной язык это ***HTML***, который является простым средством разметки, предназначенным для создания гипертекстовых документов. Кроме ***HTML*** существуют много других языков как модификаций, так и независимых. Немного о браузерах. Браузер — это программа просмотра Веб-страничек во всемирной компьютерной сети Интернет. Иначе говоря, браузер интерпретирует информацию, составляющею веб-сайт, и отражает ее на экране пользователя.

Данная информация может быть представлена в виде текста, фотографий, звука, видеоизображения или анимации; веб-браузер интегрирует данные названные элементы и представляет их пользователю в виде набора страниц.

По сути дела, браузеры является основной программой для доступа к службам Интернет. Они предназначены для того, чтобы соединяться по Интернет с удаленными машинами, запрашивать определенные документы, а затем форматировать полученные документы для просмотра на локальной машине.

В качестве языка, или протокола, в Веб-транзакциях используется ***Hypertext Transfer Protocol***, или ***HTTP***. На удаленных машинах, содержащих необходимые документы, работают ***HTTP***-серверы, которые ожидают запросов браузеров и по получении запроса выдают браузеру затребованные документы. В реальности браузеры являются HTTP-клиентами.

Существует более десятка различных браузеров выпущенных разными разработчиками и имеющие различные возможности. Основными веб браузерами используемыми около 90% пользователей Интернет являются ***MS Internet Explorer*** и ***Netscape Navigator***.

Основной задачей веб-мастера при проектировании нового веб сайта является тестирование своего сайта на нескольких платформАХ и веб браузерах. Дело в том, что различные браузеры имеют различную конфигурацию и возможности, и порой очень трудно создавать странички, которые можно было бы видеть одинаково на основных браузерах. Для этого существует целая система правил кодирования на ***HTML***.

3.5. Особенности изображений для Web

Особенности изображений для веб. При всем многообразии графических форматов для представления изображений в Веб используется малое их количество. Основным ограничителем здесь выступает размер файла. А потому выбор формата при подготовке изображения для Веб определяется оптимальным соотношением двух взаимоисключающих параметров: размеров изображения и объема файла.

Однако из теории вероятности известно, что большинство схем представления информации обладают той или иной степенью избыточности. К примеру, составляя конспект на лекциях, мы пользуемся некой системой сокращений слов и фраз, не теряя при этом смысла содержания. Этот принцип положен в основу большинства систем сжатия информации, в том числе и форматов графических файлов, используемых в Веб.

Однако при одинаковом принципе алгоритмы его реализации разрабатываются разными людьми, а потому имеют весьма существенные различия между собой. Более того, не стоит забывать, что каждый формат имеет и другие особенности, поэтому при его выборе следует учитывать прежде всего исполнение рисунка. В мире Интернет безраздельно господствую два графических формата: ***GIF*** и ***JPEG***. Причем сферы

применения каждого четко определены: ***GIF***-картинки, как правило, применяются при оформлении страницы как элементы навигации и фона, а ***JPEG*** — для "высококачественного" репродуцирования.

Сейчас продвигается новый формат для Веб-графики под названием ***PNG***, но он пока еще мало распространен, и не все браузеры его понимают, поэтому с этим форматом пока разбираться не будем. ***JPEG***. ***JPEG*** (***Joint Photographic Experts Group***). Разработан группой экспертов по фотографии (что видно из названия) под эгидой ***ISO*** (Международная организация по стандартам). Вообще этот формат довольно уникален тем, что использует алгоритм сжатия, отличающийся от применяемых во всех остальных графических форматах, — сжатие с потерями.

Этот алгоритм ранее использовался на телевидении в схеме телевизионной трансляции США (***NTSC***). Основан он на все той же ограниченности человеческого зрения, неспособности глаза не замечать некоторые искажения в восстановленном изображении. На сегодня этот алгоритм является одним из самых эффективных (коэффициент сжатия достигает 1:100), однако он не очень хорошо обрабатывает изображения с малым количеством цветов и резкими границами.

Вообще ***JPEG*** можно назвать противоположностью ***GIF***. Он позволяет отображать 24-битную палитру, т. е. все 16,8 млн. цветов, что дает возможность отображать градиенты с фотографической точностью, но при этом не может иметь прозрачных областей. Однако этот формат таит в себе одну особенность, которую нельзя не учитывать. При повторном сохранении изображения в ***JPEG*** он повторно запускает алгоритм сжатия, естественно, с ухудшением качества. Поэтому сохранять изображение в нем следует только после окончательной обработки. Он поддерживает миллионы цветов и оттенков, палитра не настраиваемая, предназначен для представления сложных фотоизображений.

Разновидность ***progressive JPEG*** позволяет сохранять изображения с выводом за указанное количество шагов (от 3 до 5 в ***Photoshop'e***) — сначала с маленьким разрешением (плохим качеством), на следующих этапах первичное изображение перерисовывается все более качественной картинкой. Анимация или прозрачный цвет форматом не поддерживаются. Уменьшение размера файла достигается сложным математическим алгоритмом удаления информации — заказываемое качество ниже — коэффициент сжатия больше, файл меньше.

Главное, подобрать максимальное сжатие при минимальной потере качества. Кроме коэффициента сжатия еще приходится делать выбор между типами формата — стандартный, оптимизированный или прогрессивный. Наиболее подходящий формат для размещения в Интернете полноцветных

изображений. Вероятно, до появления мощных алгоритмов сжатия изображения без потери качества останется ведущим форматом для представления фотографий в Веб. Формат ***JPEG***: — Позволяет сохранять полноцветные изображения с количеством цветов 16,7 млн. цветов (или 24***bpp***), причем, если в рисунке меньше цветов, то перед сохранением файла он все равно преобразуется в полноцветное изображение; — Использует сжатие с потерями информации, за счет чего достигает диких степеней сжатия файлов; — Поддерживает прогрессивную развертку, т.е. изображение появляется вначале с плохим качеством, и в процессе загрузки постепенно улучшается. Пожалуй, это все преимущества формата ***JPG***, и основное его предназначение — хранение изображений фотографического качества.

GIF (***Graphics Interchange Format***, формат взаимообмена графикой) разработан ***CompuServe Incorporated***, последняя версия ***GIF***-89***a***. Первоначально, как можно понять из названия, этот формат разрабатывался для передачи графической информации в потоке данных, а потому, в отличие от остальных, представляет собой последовательную организацию, а не произвольную, что позволяет использовать минимум ресурсов процессора при его распаковке. Для компрессии файлов ***GIF*** использует ***LZW***-алгоритм сжатия, или, как его еще называют, сжатие без потерь, при этом он наиболее эффективен при больших однотонных областях с четкими границами.

А так как сканирование изображения происходит по горизонтали, то и сжатие будет более эффективно при больших горизонталях таких областей. Однако ***GIF*** не способен хранить неиндексированные изображения, то есть может отображать не более 256 цветов. Эта ограниченность формата не позволяет добиться плавного перехода от одного цвета к другому, что особенно заметно при использовании градиентов и размывок.

Как вы узнали, ***GIF*** поддерживает не больше 256 цветов, а это значит, что все изображения, которые мы сохраняем в ***GIF***-формате, явно или неявно уменьшают количество цветов, чтобы уложиться в этот лимит (разные программы с разным успехом).

А отсюда вывод — если у Вас красивая фотография с плавными переходами и едва уловимыми оттенками цвета, то после преобразования все будет гораздо хуже — оттенки перестанут быть неуловимыми, и вся фотография приобретет неестественный, нереалистичный вид. Еще одной особенностью, введенной в последнюю версию формата, является создание прозрачных областей в изображениях, открывающее интересные возможности в Веб-дизайне.

"Потоковая" природа ***GIF***, относительно малые размеры его файлов, возможность компрессии за счет использования прозрачных областей в кадрах сделали его прекрасным инструментом для создания анимации в Веб.

Использование ***GIF*** целесообразно, прежде всего, для так называемых плоскоцветных изображений с четко обозначенными границами переходов между цветами, а также малоразмерных изображений типа кнопок, превьюшных картинок и т. п.

3.6. Приемы оптимизации графики

Приёмы оптимизации графики. Проблемой номер один при работе над графикой для ***Web***-сайта является размер файла, который напрямую связан со скоростью появления изображения в окне браузера. Дело в том, что самая лучшая графика — та, которая имеет приемлемое качество изображения и небольшой размер файла.

Подготовку графики для веб-страницы условно можно разбить на 3 этапа: создание графических элементов в векторном редакторе (в нашем случае это — ***CorelDRAW***! версии 8), экспорт в полноцветный растровый формат и, наконец, оптимизацию полученной графики с одновременным переводом ее в формат с ограниченной цветностью (***GIF***) или в формат, реализующий сжатие с потерями (***JPEG***).

Деление на этапы условно: между вторым и третьим этапами на изображение можно наложить какие-либо специфически растровые эффекты (например, тени или размывки); или второй этап может отсутствовать вообще, если векторная программа способна экспортировать графику сразу в ***GIF*** или ***JPEG*** и предоставляет при этом достаточные для профессионала возможности настройки этого процесса.

Таким образом, оптимизацию графики можно характеризовать как поиск компромисса между ее качеством и объемом файла, как выбор одного из двух форматов и параметров сжатия в выбранном формате. Рассмотрим подробнее возможности сжатия графики в этих форматах. Сжатие графики в формате ***JPEG*** определяется одним параметром, называемым уровнем качества (***quality***) и измеряемым в относительных единицах — чаще всего от 0 (максимальное сжатие) до 100 (максимальное качество). Большинство ***JPEG***-файлов сохраняются с уровнем качества в диапазоне от 50 до 100. Чем плавнее и размытее цветовые переходы в изображении, тем меньшим может быть этот параметр и тем большего сжатия удается достичь.

Наоборот, четкие и контрастные цветовые границы требуют повышения уровня качества, иначе возле них появляется муар. В большинстве случаев можно сразу выбрать либо ***JPEG***-формат (для фотографий или коллажей, где они доминируют), либо ***GIF*** (для неполноцветных изображений с ограниченным количеством используемых цветов, надписей, заголовков и т.п.). Если нельзя сразу определить, в каком формате выгоднее сохранять изображение, надо попробовать сначала сохранить его в формате ***JPEG***, повышая степень сжатия до тех пор, пока качество не перестанет нас удовлетворять. Даже если полученный файл будет слишком велик и мы решим воспользоваться форматом ***GIF***, у нас будет цифра, с которой можно будет сравнить результаты.

Приспособленность формата ***JPEG*** для фотографических текстур столь велика, что размер файла в этом формате при заданном уровне качества можно вполне использовать как объективную меру фотореалистичности изображения. Стоит отметить, что для изображений, размер которых меньше ста пикселов по одному из измерений, более правильным выбором будет формат ***GIF***: если изображение содержит даже небольшой фрагмент фотографии, ***GIF*** справится с его воспроизведением не хуже, чем ***JPEG***, который плохо приспособлен для графики малых размеров.

Если ***JPEG*** оперирует единственным и довольно абстрактным параметром ***quality*** для сжатия, то в ***GIF*** присутствует целый ряд параметров, главным из которых является количество цветов, или размер палитры. В отличие от других форматов, которые имеют только стандартные градации цветовой глубины (2 цвета, 16, 256, 215- ***high color*** , 224-***true color***), ***GIF*** может иметь любое количество цветов от 2 до 256.

Если при сохранении изображения в ***GIF*** не применяется безопасная палитра, графическая программа сама решает, какие именно цвета останутся при редукции полноцветного изображения в ограниченную палитру. Общее правило таково: чем больше какого-либо цвета в оригинале, тем выше вероятность того, что он в неизменном виде войдет в редуцированную палитру; те же цвета, которые встречаются в оригинале реже, будут заменены на ближайшие цвета палитры либо переданы смесью пикселов близких цветов (если включена диффузия).

Хотя количество цветов, необходимое для изображения, подбирается опытным путем, есть некоторые закономерности. Например, одноцветному тексту на одноцветном фоне должно хватить палитры из 8, а в небольших размерах даже и 4 цветов — два из них станут основными, а остальные отойдут промежуточным тонам для анти-алиасинга (сглаживания) на общих границах основных цветов.

С другой стороны, 256 (а иногда и 128) цветов с диффузией обычно вполне достаточно для полноценной передачи фотографического изображения средних размеров (хотя с этой задачей лучше справится ***JPEG***). Основная часть веб-графики располагается где-то в промежутке между этими крайностями. На величину палитры сильно влияет наличие или отсутствие диффузии — метода, подменяющего смешение цветов внутри пиксела смешением пикселов разных цветов. Основанная на псевдослучайном распределении пикселов, диффузия обнаруживает свое несомненное сходство с фотографическими текстурами: хотя на первый взгляд диффузная зернистость совсем не похожа на фотографическую плавность и размытость, для глаза сочетание двух аморфных текстур гораздо естественнее, чем диффузия изображений с плоским цветом и четкими границами объектов.

Если "сыпь" пикселов на плоскоцветных участках можно ликвидировать увеличением размера палитры (занимающие определенную площадь цвета в итоге получат свои собственные клетки в цветовой таблице и тем самым избавятся от диффузии), то на резких цветовых границах с анти-алиасингом диффузия приводит к появлению принципиально неустранимых "зубчиков". Человеческий взгляд не может заметить фальшь в отдельных граничных пикселах — но программа добросовестно старается обработать края объектов диффузией, для которой там нет места.

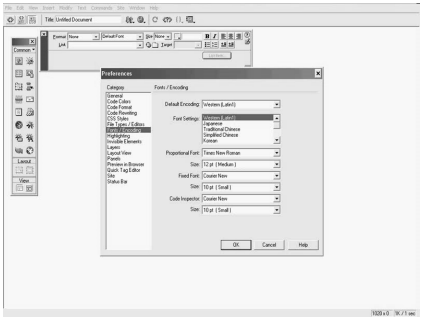
Главный смысл использования диффузии состоит в том, чтобы изображение становилось гораздо терпимее к размеру палитры. Например, если без применения диффузии картинка не выдерживает редукции даже до 128 цветов, то с диффузией ее можно ограничить до 64 или даже 32 цветов без особой потери качества, С другой стороны, случайная диффузия резко ухудшает сжимаемость графики. Вот почему иногда, снизив цветность изображения с 256 до 128 цветов, из-за этого противоположно направленного эффекта мы вместо уменьшения размера файла получим увеличение. В ряде случаев даже диффузия неспособна "вытянуть" такие сугубо фотографические элементы изображения, как размывки и градиенты. При воспроизведении в ограниченной палитре градиент распадается на сильно портящие впечатление поперечные полосы (хотя без диффузии эти полосы были бы несоизмеримо заметнее).

Степень сжатия графической информации в ***GIF*** сильно зависит от уровня ее повторяемости и предсказуемости, а иногда еще от ориентации изображения. Поскольку ***GIF*** сканирует изображение по строкам, то градиент, направленный сверху вниз, сожмется лучше, чем тех же размеров градиент, ориентированный слева направо, а последний — лучше, чем градиент по диагонали. Диффузия, хотя и сильно ослабляет эффект зависимости степени сжатия от ориентации, все же не отменяет его.

Процесс оптимизации может быть применен не только к статичному, но и анимированному ***GIF***-файлу. Жесткие требования к объему файла будут продиктованы в данном случае не столько из-за количества кадров в фрагменте, сколько благодаря физиологическим особенностям восприятия движущихся изображений: здесь уместно пользоваться гораздо более скромной палитрой и во многих случаях отказываться от таких вспомогательных эффектов, как диффузия и анти-алиасинг. Быстро мелькающие изображения не только не позволяют зрителю заметить возможные изъяны оптимизации, но и компенсируют недостатки друг друга, накапливая визуальное впечатление предсказуемо движущегося или изменяющегося объекта и нейтрализуя случайно расположенные дефекты. То же самое можно наблюдать в кинематографе, где субъективное качество движущегося изображения

3.7. О программе Macromedia Dreamweaver

О программе ***Macromedia Dreamweaver Macromedia Dreamweaver*** — это программа, которая содержит программный инструментарий для визуальной компоновки Веб-страниц и для работы с ***HTML***-кодом. Интуитивно понятный интерфейс ***Dreamweaver*** прост, не требующий длительного изучения.



Непосредственно в ***Dreamweaver*** можно создавать анимацию, использовать данные из ***Microsoft Office***, легко импортировать графику, меню и кнопки.

Использование программы ***Macromedia Dreamweaver*** позволяет в первую очередь экономить время и внимание при создании Веб-страниц. При этом множество функций программы уникальны, для их достижения, при кодировании ***HTML*** вручную, потребовалось бы множество времени и специальных знаний. Первым этапом в изучении программы ***Dreamweaver*** — это изучение интерфейса и настройки программы. Благодаря применению диалоговых окон можно без труда освоить основные настройки программы. Стоит отметить, что несмотря на то, что интерфейс программы прост, все расширенные возможности программы становятся доступны только после соответствующей настройки. Множество функций скрыто в меню программы и количество и размер кнопок интерфейса минимально. Подобное расположение в

значительной мере позволило сэкономить рабочее пространство, не позволяя отвлекаться от основной задачи создания Веб-сайтов.

Использование встроенной системы шаблонов и библиотечных элементов для ускорения процесса создания сайта позволяет пользователям, совершенно не знакомых с навыками программирования ***HTML***, создавать свои Веб-страницы без заучивания синтаксиса языка ***HTML***, тегов, не тратя время на исправление ошибок в коде Веб-страницы. Задание основных свойств страницы. В меню "***Modify***" выбираем пункт "***Page Properties*** ". Появляется следующее диалоговое окошко: свойства страницы. ***Encoding*** (кодировку символов) придется задавать для каждой страницы, если сразу не заменить его на кириллицу в свойствах ***Dreamweaver*** (***Edit*** / ***Preferences*** / ***Encoding***).

В окошке "***Title***" прописываем название страницы (которое будет отображаться в верхней части панели браузера). При условии задания верной кодировки русский язык поддерживается абсолютно корректно. В окошке "***Background Image***" задается фоновый рисунок, если надо. При наведении курсора на квадратики рядом с пунктами "***Background***", "***Text***", "***Links***", "***Visited Links***", "***Active Links***" вид курсора меняется на пипетку. Если нажать этой пипеткой на квадратик, получим следующее диалоговое окно. Внизу отображается образец цвета, над которым расположен в данный момент курсор, и его код.

Три кнопочки справа: пипетка — выбор цвета из данной палитры, ластик — выбор цвета, отображаемого браузером по умолчанию — "***Default***" (если сделан данный выбор, то в код страницы не будет добавлено никаких тегов относительно цвета данного элемента.

Далее можно задать значения краев страницы. Если вы решили воспользоваться этой возможностью, то обязательно укажите одинаковые значения в ячейках "***Top Margin***" и "***MarginHeight***", а также в ячейках "***Left Margin***" и "***MarginWidth***" (например, 0 и 0; 7 и 7), чтобы страница отображалась корректно в различных браузерах — все они понимают "края" по своему (***Top Margin*** и ***Left Margin*** прописываются для Эксплорера, а ***MarginHeight*** и ***MarginWidth*** — для Навигатора). После заполнения всех этих полей можно нажать кнопку "***Apply***", посмотреть, что получилось, и если не понравится — сразу переделать. Можно также нажать кнопку "ОК" и вернуться к основному окну, а переделать потом.

Одно из основных достоинств ***Dreamweaver*** состоит в том, что в текущем сеансе работы функция "***Undo***" ("отменить") применима даже к изменениям, сделанным до сохранения файла. То есть, если вы сделали что-то, сохранили файл, а потом решили отменить это действие — это возможно (во ***Front Page*** возможность отмены действия после сохранения исчезает). Задание текущей директории и редактирование свойств сайта в целом

Чтобы иметь возможность создавать короткие гиперссылки и работать с вложенными папками сайта, нужно задать директорию, в которой располагается сайт в целом, и определить его остальные свойства. В ***Dreamweaver*** также имеется собственный встроенный ***FTP***-клиент, который в 3-ей версии ни в чем не уступает ***Cute FTP***, а в чем-то даже его превосходит.

Например, можно попросить загрузить на сервер страницу вместе со всеми связанными изображениями, ***a Dreamweaver*** сам определит, какие картинки использованы на странице, и подгрузит их в нужные директории автоматически. Причем это именно ***FTP***-клиент, который не позволяет себе ни в чем превышать полномочия.

Чтобы пользоваться встроенным ***FTP***-клиентом, также необходимо задать параметры сайта. Параметры сайта и определение рабочей директории ("***root folder***") так же нужны для того, чтобы ***Dreamweaver*** корректно прописывал пути к вставляемым изображениям. Итак, в меню "***Site***"выбираем пункт "***New site***" и в результате получаем следующее диалоговое окно

Заполняем все требуемые пункты в разделе "***Local Info***" (в "***Link Management Options***"можно ничего не прописывать, если вы предпочитаете использовать короткие ссылки вида: <a ***href***="../***folder***/***file***.***htm***">). Главное — указать "***Local Root Folder***", остальные параметры особого значения не имеют. Заполнение раздела "***Web server info***" требуется в том случае, если вы собираетесь пользоваться встроенным ***FTP***-клиентом. Настройки похожи на настройки ***Cute FTP***. Раздел "***Site Map Layout***" является настройкой карты сайта. В него надо вникать, если вы собираетесь использовать автоматическую навигацию. Если нет — оставьте все, как естъ.Все остальные настройки тоже лучше оставить по умолчанию.

Когда вы все заполните и нажмете кнопку "ОК", вас спросят, создать ли кэш для ускорения работы. Для основных сайтов кэш лучше создать. Если же вы делаете что-либо одноразовое, типа обоев в виде страницы, то от создания кэша можно отказаться.

Примечание: вы можете вообще не определять сайт, если все используемые изображения у вас заранее собраны в одну директорию и при этом вы не собираетесь пользоваться встроенным ***FTP***-клиентом, а созданные ***HTML***-файлы сохраняете в этой же директории. В данном случае важно сохранить файл до вставки первого изображения (иначе путь к изображению будет прописан некорректно). Следующий урок мы начнем с форматирования текста.

3.8. Форматирование текста

Форматирование текста. На данном уроке мы познакомимся с приемами работы с текстом. Как вы, наверное, уже заметили, что основа практически

любой страницы — это текст. Создайте новую страницу и наберите в ней любой текст. Если текст уже готов и находится, в каком либо текстовом редакторе то его можно импортировать. Сохраните эту страницу дав ей какое-либо имя. Теперь переёдём к самому форматированию текста. Функции, которые необходимы для этого находятся в следующем меню ***Text***. Здесь мы видим множество инструментов, которые предназначены для форматирования текста: это стили начертания, размеры шрифта, цвет и т. д. Но гараздо проще работать с отдельным меню которое всегда перед глазами. Такая возможность в ***Macromedia Dreamweaver*** предусмотрена она находится в меню ***window*** — ***Properties***. Это меню форматирования текста. Сейчас оно находится в верхнем левом углу рабочего файла и закрывает текст. Его можно с помощью нажатия левой клавиши мыши перетащить в любое свободное место.

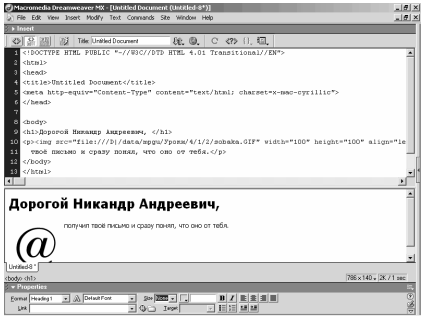
Теперь давайте подробнее остановимся на самом меню форматирования текста и его функциональных особенностях. Надо сказать, что во всех программах которые поддерживают функцию форматирования текста есть похожее меню. И если человек умеет хорошо работать в текстовом редакторе или в графическом, но с текстом то работа с текстом в ***Macromedia Dreamweaver*** у него не должна вызывать затруднений. Теперь непосредственно приступим к работе с текстом, будем его форматировать. Для начала его необходимо весь выделить или ту часть, с которой вы хотите работать.

Первое на чём остановимся это функция формат, изменение текста без измене5ния стиля начертания или просто говоря заголовок. Особенность работы с этим инструментом в ***Dreamweaver*** состоит в том, что он работает с текстом вообще, выделен тот или нет.

Следующее на чём мы остановимся это функция изменения стиля начертания текста без изменения его размеров. Следующая функция отвечает за размер текста. Подставляя цифровые значения в окошечко мы видим как изменяется текст. Следующая функция это цвет текста. Здесь мы видим два окошечка, в первом отображается палитра цветов, а во втором цифровое значение цвета. С помощью следующих двух кнопок тексту можно придать либо жирное начертание, либо наклонное. Следующие три кнопочки отвечают за расположение текста: селевого края, по центру, справного края. Снизу под инструментом выравнивания расположена функция маркирования текста. Маркирование текста может быть цифровым или маркированным. О функциях расположенных в нижней левой части меню мы поговорим на одном из следующих уроках. Предназначены они для гиперссылок.

На этом уроке мы в крациях коснулись проблем по работе с текстом в программе ***Dreamweaver*** более полную и точную информацию можно узнать из следующих уроков.

3.9. Синтаксис текста



На этом уроке мы познакомимся с ***HTML***-кодом и разберемся с множеством непонятных знаков (синтаксисом гипертекстовой разметки). Для этого нам необходимо создать какой-нибудь не сложный ***HTML***-документ на примере которого мы и будем разбирать этот синтаксис.

Большинство ***HTML***-редакторов в том числе и ***Dreamweaver*** позволяют работать как в режиме визуального редактирования так и в режиме непосредственной правки ***HTML***-кода. Мы переключимся в промежуточный режим, когда виден документ и соответствующий ему документ, так нам будет удобнее разобраться, что где находится. Итак перед нами ***HTML***-код в чистом виде. Вглядитесь в него, вы найдёте там фрагменты своего документа, а также массу чужеродных фрагментов заключённых в угловые скобки. Эти фрагменты называются тегами с их помощью производится гипертекстовая разметка. Тег состоит из названия заключённого в угловые скобки.

Название тегов могут писаться как строчными, так и прописными буквами. У тегов есть вторая половина дополненная косой чертой и такие поры окружают текст документа, например заголовок окружён тегом H1. это называют элементом, элемент и есть часть разметки. Например элемент H1 помечает текст как заголовок. В браузере такой текст будет отличаться от остального. Давайте попробуем изменить разметку текста, вместо H1 напишем P. В панели свойств появляется сообщение что код был изменён и для отображения изменений нужно нажать кнопку ***refresh F***5.

Нажимаем и видим что вместо заголовка получили обычный абзац текста. Итак элемент состоит из открывающих тегов, данных и закрывающих тегов. Назначение тегов обозначение границы элемента. Очень важно, что бы открывающему тегу соответствовал закрывающий. Бывают случаи когда элемент состоит только из открывающего тега, это случается когда элемент, предназначенный для разметки текста несёт самостоятельную нагрузку. Таким элементом является ***img src***, обозначающий вставку изображения.

Элементы могут быть вложенными друг в друга пример элемент ***img*** вложен элемент p обозначающий абзац, можно вложить в этот абзац и другие элементы. Например, пометим несколько слов тегом ***n***, эти слова выделились курсивом. Теги не могут пересекаться, вложенный элемент должен закрыться

до того, как закроется элемент, в который он вложен. В случае возникновения ошибки программа сообщит об этом.

Некоторые теги помимо имени могут включать в себя атрибуты, которые содержат дополнительную информацию о элементе. Атрибут состоит из имен и атрибута и значения разделённого знаком равенства. Например, атрибут ***img*** содержит атрибут ***width*** отвечающий за ширину изображения. При изменении атрибута ширина картинки изменится. Итак, мы познакомились с основными элементами, из которых состоит ***HTML***-код. В ***HTML*** есть свои законы которые находят своё отражение в структуре документа.

3.10. Разметка текста

Разметка текста. На этом уроке мы вкратце коснемся понятия разметки гипертекстового документа. В настоящее время существует масса редакторов ***Web***-страниц, которые не требуют от вас знаний основ ***HTML***, например ***Macromedia Dreamweaver***. Но для того чтобы уметь профессионально подготавливать гипертекстовые документы, вы должны знать их внутреннее строение, то есть код документа ***HTML***.

HTML является стандартным языком, предназначенным для создания гипертекстовых документов в среде Веб. ***HTML***-документы могут просматриваться различными типами Веб-браузеров, когда документ создан с использованием ***HTML***. Веб-браузер может интерпретировать ***HTML*** для выделения различных элементов документа и первичной их обработки.

Использование ***HTML*** позволяет форматировать документы для их представления с использованием шрифтов, линий и других графических элементов на любой системе, их просматривающей. ***HTML*** позволяет вам формировать различную гипертекстовую информацию на основе структурированных документов. Обозреватель определяет сформированные ссылки и, через протокол передачи гипертекста ***HTTP***, открывает доступ к вашему документу другим пользователям Интернет. Разумеется, для успешной реализации всего этого необходим софт, полностью совместимый с ***WWW*** и поддерживающий ***HTML***.

HTML-документ — это обычный текстовой файл. Используя, например, обозреватель ***Netscape Navigator*** вы можете просмотреть результат вашей работы, просто загрузив в него созданный на основе синтаксиса ***HTML*** текстовой файл. Гипертекстовый язык предоставляет только информацию для чтения. Это означает, что редактировать ***Web***-страницы может лишь тот, кто их создал, а не простой пользователь сети.

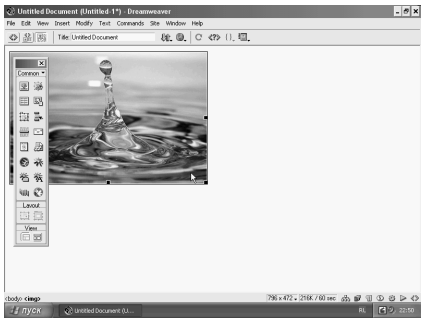
Термины гипертекста. Самый смак гипертекстового языка — это ссылки. В мировой паутине вы просто нажимаете на ссылку и мгновенно оказываетесь в другой точке земного шара на выбранной вами страничке. В языке описания

гипертекстовых документов все тэги парные. В конечном тэге присутствует слеш, который сообщает обозревателю о завершении. Не все тэги совместимы с обозревателями. Если обозреватель не понимает тэг, то он его просто пропускает. Любой документ ***HTML*** имеет название. По названию вашего документа ***HTML*** другие обозреватели могут найти информацию. Место для названия всегда определено — оно находится вверху экрана, и отдельно от содержимого документа. Максимальная длина названия — 40 символов.

Форматирование. Форматирование может быть непосредственным или авторским. Если вы используете тэг <pre>, то форматирование считается авторским. Следующие тэги присущи непосредственному форматированию: <p> — параграф. <***hr***> — горизонтальная линия. <***br***> — обрыв строки. Заголовки и подзаголовки

Язык ***HTML*** позволяет вам работать с шестью уровнями заголовков. Первый заголовок — самый главный. На него обращается особое внимание. Остальные заголовки могут быть оформлены, например, жирным шрифтом или прописными буквами. В ***HTML*** разбиение на строки не принципиально. Это означает, что вы можете разбить строки вашего документа в любом его месте. Связано эго с тем, что в гипертекстовом документе идущие подряд отбивки превращаются в одну. Более подробную информацию можно будет узнать из следующих уроков, а на этом урок закончен

3.11. Вставка изображения и редактирование его свойств



Вставка изображения и редактирование его свойств. Для вставки картинки следует воспользоваться кнопочкой с изображением картинки на панели "***Objects***". После нажатия этой кнопки вы автоматически обозреваете свою рабочую директорию на предмет пригодных для употребления картинок.

Если картинка находится за пределами рабочей директории, вы ее спокойно находите и вставляете. Большой плюс: "***Preview***" (предварительный просмотр) работает во всех директориях, а не только в директории сайта. После того, как вы выбрали картинку и нажали "***Select***", возможны два варианта развития событий. Выбранная картинка находится в рабочей директории сайта (или в одной из ее поддиректорий). В этом случае она спокойно вставится, и все. После вставки изображение автоматически

оказывается выделенным, при этом меняется панель свойств объекта (не вылезает дополнительная, а просто меняется существующая!). Чтобы в дальнейшем изменить какие-то свойства, достаточно выделить нужное изображение и обратиться к этой панели. В верхнем углу панели отображается миниатюрная копия изображения, чтобы вы были уверены, что работаете именно с тем изображением, с которым собирались.

Справа от этой копии окошечко, в которое можно вписать значение параметра "пате" (название рисунка, используется поисковыми системами и браузером при кэшировании страниц). Еще правее — ширина и высота рисунка (изменять не рекомендуется, т.к. браузеры отвратительно масштабируют графику. Лучше отмасштабировать картинку в каком-нибудь графическом редакторе, и на сайт ее выкладывать с текущими размерами).

Удалять эти параметры тоже нежелательно — это уменьшит, конечно, размер кода, но в то же время увеличит время загрузки страницы, т.к. браузер не сможет корректно отформатировать текст, пока не загрузит графику. Далее идут параметры ***src*** (путь к рисунку) и ***link*** (следует заполнять, если рисунок является гиперссылкой).

Dreamweaver не вставляет автоматически рамку вокруг рисунка, являющегося ссылкой. Если вы хотите, чтобы рисунок был обведен рамкой определенной толщины, независимо от того, является он ссылкой или нет, следует установить соответствующее значение параметра ***border*** (поле ***border*** в правой нижней части панели). В правом верхнем углу панели доступны параметры ***align*** (выравнивание рисунка относительно текста) и ***alt*** (альтернативный текст). В левой нижней части панели можно сделать из своего рисунка ***image map*** -мультиссылку (просто выбираете соответствующую фигуру и рисуете части ***image map*** на картинке).

Название тэг целесообразно задавать не только для рисунков, фактически являющихся картами изображений, но и для любых картинок, часто повторяющихся на разных страницах вашего сайта (например, логотип) — если на всех страницах задать для одной и той же картинки одно и то же название тар, то браузер вообще не будет убирать ее с экрана при переходе между этими страницами, что ускоряет загрузку страниц.

Далее идут параметры ***hspace*** и ***vspace*** — вертикальный и горизонтальный отступ от текста. Далее ***target*** — задание окна, в котором будет открываться гиперссылка (для рисунка, не являющегося гиперссылкой, недоступно), далее параметр ***Low Src*** (копия рисунка с низким разрешением), поле ***border*** и кнопки настройки выравнивания абзаца

(если рисунок находится не в теге-контейнере , а непосредственно в теле страницы, нажатие этих кнопок создает для рисунка тег-контейнер <***div***> </***div***> с соответствующим выравниванием). Правый нижний угол

занимают кнопки "***Refresh***" (применить измененные параметры) и ***Edit*** (вызывает ***Fireworks*** — графический редактор, иногда поставляющийся вместе с ***Dreamweaver***, но если рисунок не в формате ***png***, то в ***Fireworks*** окажется довольно сложно его редактировать).

Примечание: ***CSS*** и ***DHTML***, весьма корректно реализованные в 3-ей версии ***Dreamweaver***, позволяют применять к рисунку дополнительные эффекты (анимированное появление на странице, полное скрытие и появление по нажатию кнопки "Показать" и т.д.) и фильтры, реализуемые браузером (размытие, прожектора и др.), которые также могут накладываться в виде анимации, но это тоже отдельная песня.

3.12. Создание простого Web-сайта

На этом уроке мы сделаем простой веб-сайт в среде ***Macromedia Dreamweaver***. На предыдущих уроках вы ознакомились с тем как можно форматировать текст и обрабатывать изображения, а также вы познакомились с понятием и ***HTML***-кодом и основами разметки текста и из чего состоит веб-сайт. Теперь все те знания которые были получены вами на прошлых урока нужно применить для создания простого сайта. С чего стоит начать? Для начала необходимо настроить свойства документа. Свойства документа находятся в меню ***modify*** > ***page properties***. В начале в строке ***title*** необходимо написать имя будующего сайта. Следующая строка отвечает за изображение на заднем плане.

Следующая строка отвечает за цвет на заднем плане. Следующая строка отвечает за цвет текста. Нажмём кнопочку ОК и увидим результат изменений. Теперь необходимо вставить текст, если текст скопирован и находится в буфере обмена то его можно вставить с помощью кнопки ***paste***, которая находится в меню ***edit***.

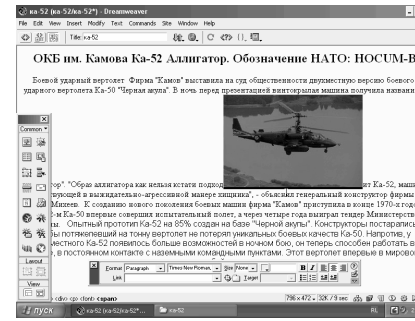
Текст появился в нашем документе. Теперь его необходимо отформатировать. Вызовем меню форматирования текста меню ***window*** > ***properties***. Теперь необходимо определиться что мы хотим получить от текста, то есть где он должен располагаться. Расположение текста зададим по центру, а цвет и начертание текста оставим без изменений. Изменим всего лишь заголовок, для начала отделим его от основного текста. В случае если этого не сделать то даже если мы выделим нужный нам текст, то изменится всё равно весь текст. Предадим нашему заголовку необходимые начертания.

Теперь когда текст у нас отформатирован, неплохо бы добавить в наш сайт картинки. Для того чтобы вставить картинки в наш документ необходимо войти в меню ***window*** > ***objects*** и в появившемся меню с помощью кнопки ***insert imadge*** вставить в документ выбранную картинку. Лучше всего если эта

картинка будет подготовлена в каком-нибудь другом редакторе например ***Adobe Photoshop***, а не ***Macromedia Dreamweaver***, потому как ***Dreamweaver*** очень плохо редактирует изображения. Здесь давайте остановимся ещё на одном вопросе касающимся веб-сайтов — это веб-безопасная палитра. Все изображения для веб-сайтов должны быть сделаны с применением веб-безопасной палитры. Обозреватели ***microsoft Internet explorer*** и ***netscape novigator*** изменяют цветовую палитру любых изображений и выводят её на экран. Эти программы подстраивают палитру изображений под системную палитру. По этому дизайнеры используют набор 216 цветов.

Это означает, что если ваша картинка составлена из набора безопасных цветов, то в любом обозревателе она отобразится, так как видите её сами. Во всех современных редакторах есть пункт экспортирования изображения в веб. Но с развитием Интернета система веб безопасной палитры потихоньку отходит на задний план и всё меньше применяется.

Так вот мои картинки уже обработаны с использованием веб-безопасной палитры и их остаётся только вставить после чего необходимо произвести их позиционирование. Само изображение можно перетаскивать мышкой в любое удобное место. Но более профессионально перемещать изображение с помощью специального инструмента. Как обычно на веб-страничке присутствует несколько изображений, остальные изображения можно так же подгрузить и расположить их как нужно.



Как уже говорилось масштабирование изображения в ***Macromedia Dreamweaver*** проходит так же плохо как и редактирование и поэтому к этим операциям нужно прибегать крайне редко. Запомните, изображение нужно готовить в графическом редакторе. Ну вот наш сайт готов, есть текст, есть изображение, осталось только сохранить документ.

Сохраняется документ в ***Macromedia Dreamweaver*** стандартным образом меню ***file*** > ***Save as*** в появляющемся окне вводится имя и расширение файла. Нажимаем кнопку сохранить и документ сохраняется как документ ***Internet explorer***. Можно посмотреть что у нас в результате получилось. На этом пожалуй стоит остановиться.

Раздел 4. Технологии Web-дизайна

4.1. HTML

4.1.1. Что такое HTML

Говорят, что любая новая технология становится по-настоящему массовой только тогда, когда появляется возможность пользоваться ею, не понимая, как она "устроена". Например, огромное количество людей смотрит телевизор, не имея ни малейшего представления о том, что такое телевизионный сигнал и как он преобразуется в картинку на экране. Компьютерные технологии в этом смысле, пожалуй, ещё не стали истинно массовыми.

Хотя, конечно, благодаря развитию программного обеспечения, сегодня пользователю компьютера уже не требуется тех знаний, что требовалось ещё несколько лет назад. Тем не менее, согласитесь, что управление компьютером по своей простоте всё-таки ещё уступает нажатию кнопок на телевизионном пульте.

Кроме того, стабильность работы компьютеров всё ещё далека от идеала, особенно в руках неумелого пользователя. А это значит, что углубленные знания всё ещё актуальны. К чему я это всё говорю? К тому, что ***Web***-технология ещё достаточно молода. Совсем недавно заниматься веб-дизайном не-технарю было весьма затруднительно. Нужно было обладать массой технических знаний, проделывать какие-то загадочные пассы, писать таинственные заклинания, чтобы всё это, наконец, заработало.

Информационное пространство было заполнено дискуссиями на тему "кто виноват и что делать", чтобы исчезли поля вокруг страницы, фон таблицы совпал с содержимым, а картинки встали вплотную друг к другу. И чтоб потом это работало и в Ескплорере и в Нетскейпе. О работе в ***HTML***—редакторах, подобных дримвиверу, можно было говорить лишь с большой натяжкой, поскольку код, выдаваемый тогдашними редакторами был не очень высокого качества. Вся эта ловля блох, безусловно, была интересна только настоящим энтузиастам и не располагала, прямо скажем, к производительной профессиональной работе.

Сегодня, безусловно, всё несколько проще. Появились островки стабильности в этом бушующем море новых технологий. Стандарты несколько устаканились, накал браузерных войн ослаб, ***HTML***—редакторы совершенствуются. Поэтому мы начали изучение веб-дизайна не с основ, а с практических вещей. Вы убедились, что можно получить некий продукт веб-дизайнерства, не обладая, так сказать, глубоким знанием технологии.

Тем не менее, от технологии никуда не деться. И, уж извините меня за резкость, то, что достаточно для создания домашней странички Васи Пупкина, то для целеустремленного будущего веб-дизайнера — лишь затравка. Итак, пускай же ***HTML***и, ***URL***и, ***CSS***-ки и ***XML***и, наряду с прочими тегами и джава-скриптами войдут в вашу плоть и кровь. И пусть ни один мускул не дрогнет при этом на вашем лице.

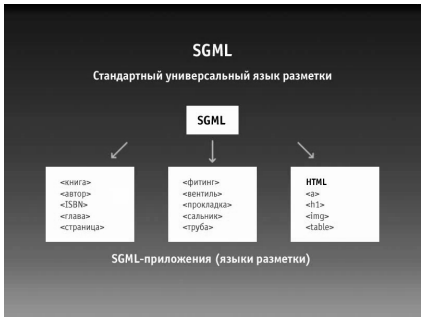
Итак, если вы помните, в первых уроках предыдущей главы мы говорили о том, что основа веба — гипертекст — создается при помощи специального языка разметки ***HTML***. Но что значит "язык разметки" — спросите вы. Тут нам опять нужен небольшой исторический экскурс.

В 1969 году человек по имени Чарльз Голдфарб, работавший в компании ***IBM***, занимался созданием системы хранения юридических документов в электронном виде. Он задался вопросом: как, например, получить список заголовков документов или найти документы, написанные конкретным юристом. То есть, с одной стороны, система должна была выдавать цельные документы, с сохранением исходного форматирования, а с другой стороны, документ должен состоять из смысловых кусков: заголовок, автор, тема и т.д. И тогда родился так называемый ***GML*** (***Generalized Markup Language*** — универсальный язык разметки). Этот язык содержал специальные знаки, которые встраивались прямо в текст документа и отмечали, грубо говоря, какая часть текста что обозначает. То есть вводилась мета-информация, по-русски говоря, информация об информации. Расстановка в тексте таких указателей и называется разметкой. Речь идёт исключительно о структурной разметке, позволяющей компьютеру в какой-то степени понять смысл текста.

Что делать с этим пониманием — это уже дело программы. Например, найдя в тексте имя автора, программа может вывести его на экран курсивным начертанием, а может показать биографию автора или список всех его произведений, в зависимости от поставленной перед программой задачи.

В середине 80-х, продолжая идеи ***GML***, был создан ***SGML*** — стандартный универсальный язык разметки, который успешно используется до сих пор. ***SGML*** не является сам по себе инструментом разметки текстов. Он лишь закрепляет общие правила создания таких инструментов. А разработка конкретных языков (так называемых ***SGML***-приложений) состоит в определении синтаксиса создаваемого языка, т.е. фактически в определении набора элементов, на которые может быть разделен текст.

Понятно, что для книгоиздателей нужен один язык, в котором фигурируют такие элементы, как книга, автор, название и т.д., а, допустим, для сантехников — совсем другой, описывающий фитинги, вентили, прокладки и фановые трубы. ***HTML*** как раз и является одним из ***SGML***-приложений, предназначенным для разметки гипертекста во всемирной паутине.



Как вы помните, гипертекст — это нелинейный текст, по которому читатель может перемещаться произвольным образом, и главный элемент гипертекста — это гиперссылка. Идея такого текста не нова: всегда было очевидно, что, если художественные произведения, имеющие определенную композицию, предполагают последовательное восприятие, то документы, научные труды и т.п. удобнее читать, перемещаясь от одного интересующего нас фрагмента к другому. Ещё в папирусах и глиняных табличках, датированных 3000 годом до н. э. были обнаружены перекрестные указания на другие работы.

А в начале второго тысячелетия н. э. в печатных изданиях Библии использовались различные наборы пометок, отсылающих человека к схожим по тематике разделам. Для этого была разработана система адресации внутри священных текстов: каждая из книг Библии делится на главы, главы — на стихи, а стихи состоят из отдельных фраз. Такая структура значительно облегчала подготовку цитат и комментариев.

Итак, В 1990 году Тим Бернерс-Ли разработал в составе совей системы язык гипертекстовой разметки ***HTML*** и, естественно, он находился в полном согласии с идеологией ***SGML***, т.е. размечал только структуру документа. Визуальное же отображение было исключительно на совести браузеров. Если браузер отображал заголовок так-то, то автор документа никак не мог на это повлиять.

Развитием языка занимался специально Созданный в 1994 году Консорциум ***WorldWide Web***. Имея в то время, по преимуществу, академический состав, консорциум подходил к развитию языка и утверждению новых стандартов очень основательно. В 1995 году началась работа над 3 версией ***HTML***, где, наконец, должны были появиться мощные инструменты работы с отображением информации.

В соответствии с идеологией ***SGML*** оформление гипертекстовых документов должно было храниться в отдельных "таблицах стилей", где описывались правила отображения всех элементов документа. Дизайнер мог бы редактировать эти правила, не трогая сам документ, который по-прежнему содержал бы исключительно структурную разметку. То есть оформление содержания отделялось от самого содержания. Такое разделение имеет очень большое значение и глубокий философский смысл, что-то из области связи

предмета с его наименованием. Но, не углубляясь в лингвистические дебри, отметим, что:

Во-первых, не "засоренное" оформлением содержание гораздо лучше поддается компьютерной обработке, поскольку оформление — это нечто из человеческой сферы. А во-вторых, мы можем прилаживать разное оформление к одному и тому же содержанию, в зависимости от ситуации. Один и тот же документ может быть показан на экране, распечатан на бумаге, прочитан вслух и так далее. При этом правила, так сказать, его материализации будут совершенно разные.

Но пока консорциум работал над этой замечательной идеей (которая, в конце концов, была реализована, но несколько позже), ***Web*** распространялся со скоростью пожара. Началась и его коммерциализация. Появился браузер ***Netscape Navigator***, в котором, для привлечения пользователей, был поддержан ряд элементов, не имевших никакого отношения к структуре, зато позволявших управлять внешним видом документа.

Эти новшества были восприняты далекими от науки пользователями "на ура", популярность браузера стала расти, а с выходом на браузерный рынок в конце 95 года корпорации ***Microsoft*** со своим ***Internet Explorer***'ом разгорелась настоящая браузерная война, когда каждый производитель, стремясь обойти конкурента, добавлял в свой браузер разные фирменные фишки, типа бегущей строки или мигающего текста.

Всё это грозило закончиться появлением двух абсолютно несовместимых версий ***HTML*** и вообще было весьма далеко от первоначальной идеи гипертекста. В конце концов, проект ***HTML***3 был заморожен и выпущен стандарт ***HTML*** 3.2, который, фактически, зафиксировал современное состояние браузеров, более-менее приведя в порядок появившиеся к тому времени новшества.

В 1997 году появился стандарт ***HTML*** 4, который унаследовал большинство недостатков своих предшественников, но во многом шагнул вперед. В частности, в нем, наконец-то были поддержаны таблицы стилей, позволившие аккуратному веб-дизайнеру более-менее приблизиться к светлой идее разделения оформления и содержания.

По всей видимости, ***HTML*** исчерпал возможности своего развития. На смену ему потихонечку движется язык ***XML***, свободный от родовых болезней своего предшественника. Но об этом мы поговорим позже. ***HTML*** всё ещё остается главным инструментом веб-дизайнера, а актуальность свою он потеряет и вовсе не скоро, поскольку миллиарды созданных за эти годы ***HTML***-страниц в ближайшее время никуда не денутся.

4.1.2. Что собой представляют HTML-файлы

Не пора ли нам от теории перейти к практике? Мы много поговорили о том, как и для чего возник язык ***HTML***, какими извилистыми путями шло его развитие. Теперь пришло время посмотреть ему в глаза и в ужасе отшатнуться. Или же наоборот, с замиранием сердца припасть к этому живительному источнику гипертекстовых страничек. Выглядит ***HTML***-код примерно вот так. Как говорил Карлсон: "Тебе страшно? Мне нет". Вы можете сами посмотреть код любой заинтересовавшей Вас страницы.

Для этого (будем рассматривать на примере ***Internet Explorer***'а) нужно зайти в меню ***View*** и выбрать пункт ***Source***. ***HTML***-файлы — это текстовые файлы. Это значит, что если просмотреть содержимое файла каким либо его текстовым редактором, только не ***Word***'ом, а, например, Блокнотом, ***Notepad***'ом, потому что ***Word*** — это своего рода браузер, который зачастую показывает нам не исходное содержимое файла, а неким образом его форматирует; так вот, если открыть текстовый файл ***Notepad***'ом, то мы увидим там нечто более-менее членораздельное. Я не уверен, что можно назвать членораздельным для Вас то, что Вы видите, но всё-таки, если Вы присмотритесь, то увидите не какие-то непонятные закорючки, а некоторые английские слова. В противоположность текстовым существуют так называемые бинарные файлы.

Попробуйте открыть ***Notepad***'ом какую-нибудь картинку. Эти закорючки понятны только компьютеру. Человеку тут делать нечего: ни понять, ни отредактировать здесь мы ничего не можем. Кстати, ***Word***'овские файлы ***doc***, казалось бы, должны быть текстовыми. Но давайте попробуем открыть такой файл ***Notepad***'ом.

Как видите, этот файл оказался бинарным и найти в нём какое-либо понятное человеку содержимое невозможно. Так вот, благодаря тому, что ***HTML*** — это обычные текстовые файлы, и в таком виде они и приходят к нам с сервера, мы всегда можем посмотреть их "в оригинале", до перевода браузером на человеческий язык.

И с точки зрения освоения веб-дизайна это прекрасно. Побольше изучайте исходники хороших сайтов — в этом залог успешного освоения Вами веб-дизайна. Программистам в этом смысле живется труднее: Программы обычно компилируются в бинарный код,

те самые закорючки, по которым невозможно понять, как программа была написана. Поэтому для компилируемых программ существует понятие "исходника", т.е. текста программы, который написал программист.

К пользователю, как правило, попадает уже откомпилированная программа, а исходники, представляющие, на самом деле, большую ценность, разработчики держат у себя и никому не показывают. Хотя есть и исключение — программы с открытым исходным кодом, но это отдельная песня. Это целая, можно сказать, философия.

Итак, да простят меня программисты за не вполне корректное сравнение, ***HTML*** — в отличие от программ — сам себе исходник. Это, в частности, обусловлено требованием универсальности: если программы пишутся под определенную операционную систему, то текст "понятен" абсолютно любому компьютеру.

Теперь о порядке работы. Было бы жестоко из нежных объятий ***Dreamweaver***'а сразу бросать Вас в омут чисто текстовых редакторов. Поэтому поначалу мы будем продолжать пользоваться нашим привычным инструментом, но ***HTML*** будем писать вручную, благо ***Dreamweaver***, как и большинство других редакторов, позволяет помимо работы в визуальном режиме, работать и в режиме ручной правки ***HTML***-кода. Переключение осуществляется с помощью кнопок на панели инструментов. Нажимаем кнопку ***Code View*** и можем теперь редактировать наш ***HTML***-код.

Затем мы плавно перейдем на полностью ручную работу в текстовом редакторе. Результат будем смотреть сразу в браузере. Это, конечно, занятие не для слабонервных, но что может быть прекрасней, чем почувствовать себя крутым ***HTML***-верстальщиком, пишущим код вручную?!

4.1.3. Синтаксис HTML и структура HTML-документа

Любой язык имеет синтаксис, т.е. набор правил построения фраз, позволяющий определить осмысленные предложения в этом языке. С изучения этих правил мы и начнем изучение языка ***HTML***. Для начала откроем наш ***Dremweaver*** и создадим какой-нибудь несложный ***HTML***-документ в привычном нам визуальном режиме. Посмотрим, какой ***HTML***-код соответствует нашему документу.

Как Вы помните, переключение в режим просмотра ***HTML***-кода производится с помощью кнопки ***Code View***. В ***Dremweaver***'е есть ещё один режим, очень удобный, особенно когда у Вас большой монитор. Это "промежуточный" режим который показывает одновременно и код, и соответствующий ему документ. Так гораздо проще разобраться, где у нас что: мы отмечаем некий фрагмент в визуальном режиме и тут же видим этот

фрагмент в режиме ***HTML***-кода. Итак, вы можете заметить, что ***HTML*** содержит наш исходный текст в чистом виде и, кроме того, массу каких-то непонятных слов, заключенных в угловые скобки. Эти непонятные конструкции и есть основные строительные блоки нашего языка. Называются они тэгами. Тег состоит из названия тега, заключенного в угловые скобки. Названия тегов могут быть написаны как строчными, так и прописными буквами — это не важно.

Если вы присмотритесь повнимательнее, то заметите, что многие теги имеют свою "вторую половину", дополненную косой чертой. И такие "пары" окружают фрагменты нашего "содержательного" текста. Например, заголовок окружен тегами <***h***1> и </***h***1>, абзац окружен тегами <***p***> </***p***>. Такая конструкция называется Элементом. Элемент — это и есть единица разметки. Элемент состоит из Открывающего тэга, данных (т.е. содержательной информации) и закрывающего тега. Таким образом, назначение тегов состоит в обозначении границ элементов.

Давайте попробуем изменить разметку текста: например, заголовок вместо элемента <***h***1></***h***1> заключим в элемент <***p***></***p***>. Когда мы что-то меняем в коде, то изменения не сразу отображаются в визуальном режиме, а в панели свойств появляется сообщение, что мы изменили код и, дабы увидеть изменения, необходимо нажать кнопку "***refresh***" или клавишу "***F***5".

Вместо заголовка мы получили обычный абзац текста. Поздравляю! Вы сделали первый шаг к ручному редактированию ***HTML***-кода. При разметке текста на элементы очень важно, чтобы открывающему тегу соответствовал такой же закрывающий. Если бы мы изменили открывающий тег на <***p***>, а закрывающий оставили </***h***1> — это было бы грубейшей ошибкой. ***Dreamweaver*** способен "отлавливать" такие ошибки и сообщать нам об этом. Бывают элементы, для которых указывать закрывающий тег не обязательно. Таким элементом, например, является элемент ***P***. Конец первого абзаца может быть обозначен началом второго.

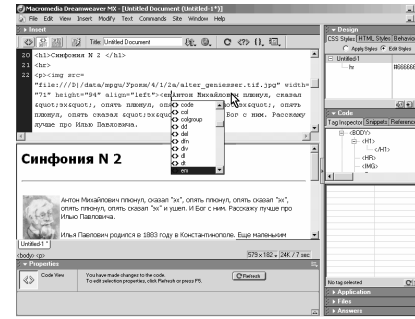
Бывают элементы, которые вообще не могут иметь закрывающего тега, состоит только из открывающего. Это случается, когда элемент предназначен не для разметки текста, а несет самостоятельную смысловую нагрузку. Например изображение, элемент <***img***>. Элементы могут быть вложены друг в друга. Например, элемент <***img***> вложен в элемент <***p***>. Это значит, что картинка находится внутри абзаца. Мы можем вложить в этот абзац и другие элементы.

Например, давайте какой-нибудь фрагмент текста заключим в элемент <***em***></***em***>. Этот фрагмент выделился курсивом. Элементы не должны не должны при этом "пересекаться": если элемент <***em***> открылся внутри элемента <***p***>, то закрыться он должен также внутри <***p***>. Если же мы закроем

элемент "снаружи" его родительского элемента, это будет грубейшей ошибкой, о чём ***Dreamweaver*** нам так же сообщит. Вот, он сообщает, что у нас имеются пересекающиеся теги. Некоторые теги помимо имени могут включать в себя атрибуты, содержащие дополнительную информацию о данном конкретном элементе. Атрибут состоит из имени атрибута и значения, разделенных знаком равенства. Например, тег <***img***> содержит атрибут ***width***, отвечающий за ширину изображения.

Мы можем изменить значение этого атрибута, и ширина картинки изменится. Особыми атрибутами являются ***id*** и ***class***. Эти атрибуты применимы почти к любому элементу. ***Id*** позволяет дать элементу некоторое уникальное имя и таким образом выделить его из ряда аналогичных элементов. Это имя затем может использоваться:

Во-первых, для нестандартного оформления конкретного элемента(например, у нас есть 2 абзаца — одинаковые элементы, выглядят,соответственно, одинаково; присвоив им имена с помощью атрибута ***id***, мы можем контролировать оформление каждого из них по отдельности). Во-вторых, ***id*** может использоваться в качестве цели в гиперссылке, т.е. мы можем ссылаться не просто на страницу, а на конкретное её место. В-третьих, для программистских нужд (для определенных манипуляций с конкретным элементом). ***Class***, в отличие от ***id***, не должен быть уникальным. Он задает принадлежность одного или нескольких элементов к определенному классу. Классы используются в основном для оформительских нужд.



Например, мы можем пометить несколько абзацев как принадлежащих к одному классу. Сейчас у нас этот и этот абзацы принадлежат к одному классу и мы можем теперь одним махом изменить их оформление. Помимо тегов в ***HTML*** используются так называемые "подстановки" (***entities***). Они нужны в двух случаях: для вставки в документ символов, отсутствующих на клавиатуре (например, "правильных" кавычек) и для вставки символов, имеющих в ***HTML*** служебное значение. Например, если нам нужна угловая скобка и мы напишем её в коде как есть, то, естественно, это будет воспринято браузером как тег. В данном случае вместо угловых скобок нужно указать специальные подстановки, которые будут восприняты браузером, как угловые скобки.

Подстановки имеют следующий вид: &код; Код может быть либо мнемоническим, специальные слова, либо числовым, когда мы просто

указываем номер символа. Числовой код предваряется "решеткой". Существует огромное количество мнемонических кодов, для самых разных символов, но с помощью числового кода мы можем указать абсолютно любой символ, имеющийся в данном шрифте, даже такой, для которого не существует мнемонического кода.

Итак, элементы, теги, атрибуты и подстановки — это основные "кирпичики", из которых строится ***HTML***-документ. Но в любом строительстве мы должны придерживаться некоторых базовых правил: сначала заложить фундамент, возвести стены, накрыть строение крышей и т.д. В строительстве ***HTML***-документа также есть свои законы, которые находят отражение в структуре документа. С разметкой текста мы более-менее разобрались. Вы убедились, что в основе документа лежит ваш текст, разбитый на элементы с помощью тегов. Однако вы можете видеть ещё ряд элементов, непонятно к чему относящихся. Располагаются они строго определенным образом.

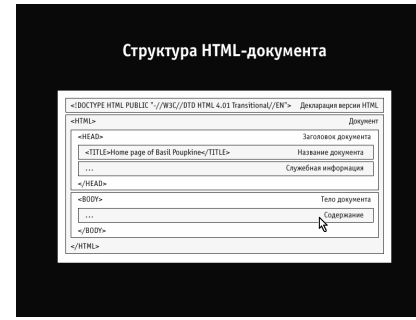
Начинается документ с такой строки: <!***DOCTYPE HTML PUBLIC*** "-//***W***3***C***//***DTD HTML*** 4.01 ***Transitional***//***EN***">. Это так называемая декларация версии ***HTML***. Дело в том, что, как Вы помните, за недолгую, но насыщенную историю своего развития, ***HTML*** вышел в нескольких версиях. Чтобы браузер мог правильно показать данный документ, он должен знать, на каком из ***HTML***-ей он написан. В данном случае в этой строке указано, что мы имеем дело с ***HTML*** версии 4.01. После декларации начинается сам документ. Он заключен в элемент <***HTML***>. Мы видим открывающий тег, и в конце документа находится закрывающий.

Внутри элемента <***HTML***> располагаются друг за другом два обязательных элемента: <***head***></***head***> и, следом за ним, <***body***></***body***>. <***head***>, т.е. заголовок, содержит служебную информацию, не отображаемую в окне браузера, например, элемент <***title***> содержит название документа, которое выводится в заголовке окна. <***body***> — это тело документа — оно в себя включает содержимое нашего документа, т.е. то, что мы видим в окошке браузера.

Если всё это изобразить графически, то получится следующая схема. Интересно заметить, что с точки зрения спецификации минимальный ***HTML***-документ должен содержать только два элемента: декларацию версии ***HTML*** и элемент ***TITLE*** — название документа. Все остальные элементы, строго говоря, являются опциональными. Правда, понятно, что информационная ценность такого документа будет весьма сомнительна.

Дальше мы будем двигаться последовательно, рассматривая группы элементов тег за тегом, изучая их назначение, особенности использования и атрибуты.

4.1.4. Тело документа. Элементы BODY, DIV, SPAN



Начнем мы не совсем последовательно, не с заголовка, а с тела документа, ибо, не освоившись с "содержательными" элементами, переходить к мета-тегам, фреймам и кодировкам может быть небезопасно для Вашего психического здоровья. Однако помните, что тело документа всегда идёт только после заголовка.

Итак, элемент ***BODY***. Не будем нажимать на правильное английское произношение, пусть будет просто Боди. Этот элемент включает в себя содержание нашего документа. Например, то, что мы видим в окне привычного нам визуального браузера. Или, например, читается вслух аудиобраузером (говорят, существуют на свете и такие). Строго говоря, по спецификации, открывающий и закрывающий теги для элемента ***BODY*** указывать необязательно, т.е. он может присутствовать только своим содержимым: подразумеваться, но не указываться явно. То же самое, кстати, относится и к элементам ***HEAD*** и ***HTML***. Т.е. мы можем их теги поудалять, и с точки зрения спецификации такой документ будет вполне корректным. На практике, впрочем, эти теги обычно присутствуют в документе, хотя бы для того, чтобы удобнее было читать ***HTML***-код. Элемент ***BODY*** может присутствовать в документе только один раз.

Элементы, входящие в состав ***BODY***, составляющие содержимое документа, бывают двух типов: блоковые или блочные элементы (по-английски это называется ***block-level***) и текстовые элементы (***inline*** или ***text-level***). Это требует пояснения. Блочные элементы являются более крупными, если так можно сказать, образованиями. С их помощью содержимое делится на большие логические куски. Откроем, например, какую-нибудь страничку и посмотрим: заголовок является блочным элементом, абзац является блочным элементом.

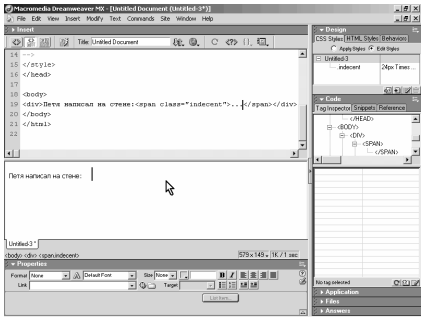
Текстовые же элементы служат для "тонкой настройки", т.е. работа ведется внутри текста: например, внутри абзаца мы можем какую-то часть его отметить в качестве гиперссылки. Или вставить внутрь текста картинку. Конкретные элементы мы будем рассматривать в последующих уроках, а здесь нам важно усвоить разницу между блочными и текстовыми элементами. Помимо описанной смысловой разницы эти типы отличаются ещё тем, что они могут в себе содержать: блочные элементы (в данном случае — элемент абзаца

"***P***") могут включать в себя просто текст, текстовые элементы. Например, элемент ***IMG*** (картинку), элемент ***EM***, элемент ***STRONG***.

Кроме того, блочные элементы могут содержать в себе другие блочные элементы. Это случается не так уж и часто. Например, существует элемент ***FORM***, для ввода пользователем каких либо данных. Этот элемент является блочным (он окружен пунктирной рамкой) и может содержать в себе другие блочные элементы.

Например, заголовок, абзац. Дальше пошли текстовые элементы, относящиеся непосредственно к форме. Текстовые же элементы, в отличие от блочных, например, элемент ***EM***, могут содержать в себе только текст или другие текстовые элементы. Блочные элементы внутри текстовых появляться не могут. Еще одно различие между блочными и текстовыми элементами, наиболее заметное невооруженным глазом, заключается в том, что блочные элементы всегда начинают новую строку, а текстовые появляются прямо посреди текста. Чтобы у Вас голова окончательно не пошла кругом, изобразим всё это схематически.

Итак, чтобы разбить содержимое на крупные логические куски, мы пользуемся блочными элементами. Блоки могут содержать (а могут и не содержать) в себе другие блочные элементы (как, например, форма содержала абзац и заголовок). Так же блочные элементы могут содержать в себе текстовые элементы и просто текст. При выводе на экран блоки начинаются с новой строки. Текстовые элементы не начинают новой строки, т.е. как бы плавно встраиваются в общее течение содержания. Текстовые элементы могут содержать (а могут и нет) в себе текст или другие текстовые элементы и не могут содержать блочные элементы.



Хотя о конкретных элементах мы будем говорить в соответствующих уроках, здесь мы сделаем исключение для двух из них, довольно любопытных. Это "просто блочный элемент" ***DIV*** и "просто текстовый элемент" ***SPAN***. Эти элементы, в отличие от других, не имеют никакого конкретного значения и служат исключительно для "голой" разбивки содержимого либо на блоки, либо на фрагменты текста.

На первый взгляд это кажется бессмысленным, однако таким образом мы можем расширять границы языка ***HTML*** и фактически создавать новые элементы. Из прошлого урока Вы помните, что любой элемент можно отнести к определенному классу. Если

нам нужен какой-нибудь нестандартный элемент, например, для указания в тексте неприличных слов, мы заключаем эти слова в элемент ***SPAN***, задаем ему какой-нибудь соответствующий класс и затем, с помощью таблиц стилей, можем эти элементы каким-либо образом "оживлять". Таблицы стилей мы будем изучать позже, суть в том, что с их помощью мы можем, например, выделить все эти слова красивыми большими красными буквами или, наоборот, скрыть их в документе. А несложными программными средствами можно делать и более интересные вещи: например, выяснить процентное содержание этих слов в тексте документа.

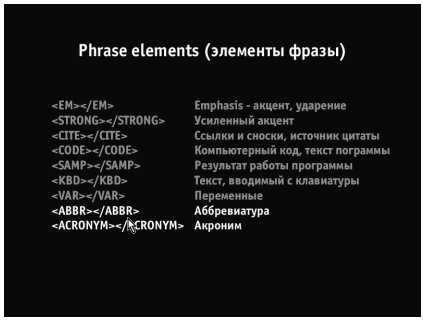
И в заключение поговорим о возможных атрибутах элемента ***BODY***. Вообще говоря, в 4 версии ***HTML*** все атрибуты этого элемента кроме стандартных ***id*** и ***class*** (и некоторых других) являются нерекомендованными к использованию, т.к. все они относятся к оформлению, а оформление должно осуществляться через таблицы стилей. Тем не менее, браузеры до сих пор понимают эти атрибуты. Мы рассмотрим их вкратце. Итак, атрибут ***bgcolor*** отвечает за цвет фона страницы.

Цвет указывается в виде шестнадцатеричного числа (об этом мы поговорим, когда будем изучать таблицы стилей). Вот появился у элемента этот атрибут, и цвет фона изменился. Атрибут ***background*** позволяет задать для фона не просто цвет, а некоторое изображение. Выбираем картинку, и она заполнила весь фон. Давайте пока этот атрибут лучше удалим. Атрибут ***text*** задает цвет основного текста. Указывается в таком же виде, как и цвет фона. Атрибут ***link*** отвечает за цвет гиперссылок. Если у нас будет в тексте какая-нибудь гиперссылка, то она предстанет в этом цвете. Атрибут ***vlink*** задает цвет ссылок, по которой мы уже "ходили", которые посещали, а атрибут ***alink*** — цвет гиперссылки, на которую мы собираемся в данный момент перейти, т.е. фактически это цвет ссылки в момент, когда мы по ней щёлкаем мышкой. ***Marginheight*** и ***topmargin*** задают размер верхнего и нижнего полей документа. Они абсолютно идентичны, но один был сделан для браузера ***Netscape Navigator***, а другой — для ***Internet Explorer***.

Это типичное "наследие тёмных времен" — можете представить себе, как жилось дизайнерам, когда одинаковые, по сути, элементы в разных браузерах назывались по-разному. Между прочим, сделать так, чтобы поля были одинаковыми в Нетскейпе и в Эксплорере было весьма затруднительно. Это было одной из актуальных тем вебдизайнерского общения, и пути решения этой проблемы отличались невероятной изощренностью. ***Marginwidth*** и ***leftmargin*** — это то же самое, только для боковых полей.

Вот, собственно, и всё об элементе ***BODY***, о теле документа. Дальше мы перейдем к рассмотрению элементов, которые составляют это тело.

4.1.5. Разметка текста. Phrase elements, font-style-elements



Основу гипертекста, как Вы помните, составляет обычный текст. Поэтому и начнем мы с элементов для разметки текста. Всё это очень похоже на работу в текстовом редакторе: заголовки, абзацы и т.д.

Но для начала скажем несколько слов о пробелах. В ***HTML*** имеющиеся в тексте символы пробела, табуляции, переноса строк, а также несколько пробелов, идущих подряд, в браузере остаются в виде единичных пробелов. Если мы хотим получить на экране подобную картину, мы должны воспользоваться специальными ***HTML***-элементами для принудительного перевода строк, для вставки нескольких пробелов подряд, либо использовать специальный элемент ***PRE***, который отображает в браузере текст в таком виде, в каком он имеется в исходнике. Ещё один тонкий момент: с какой стороны от тегов ставить пробелы. Например, если мы хотим выделить некоторые слова каким-нибудь текстовым элементом, то правильнее ставить пробелы с внешней стороны элемента, т.е. перед открывающим и после закрывающего тега.

Теперь перейдем к рассмотрению собственно элементов. Большая группа элементов, названных ***phrase elements***, элементы фразы, служит для придания некоторого специального значения фрагментам текста. Элементы эти, конечно же, являются текстовыми (не блочными) и могут включать как собственно текст, так и другие текстовые элементы. Открывающий и закрывающий теги для этих элементов являются обязательными.

Элемент ***EM*** (***emphasis***) делает акцент на данном фрагменте текста. Обычно выделяется курсивным начертанием. Элемент ***STRONG*** делает более сильный акцент. Обычно выделяется полужирным начертанием. Как Вы помните, элементы могут вкладываться друг в друга. Мы можем, например, скомбинировать эти два элемента. Элемент ***CITE*** отмечает ссылки или сноски к другим источникам, источники цитат. Об элементе ***Q*** мы поговорим попозже. В принципе, он выделяет саму цитату, в отличие от элемента ***CITE***, который описывает именно источник цитаты. Важно понять, что ***CITE*** не является элементом для расстановки гиперссылок, он просто придает тексту определенное значение.

Если мы хотим сделать имя автора гиперссылкой, мы должны добавить соответствующий элемент. Если бы Диоген жил чуть попозже или Интернет

появился бы чуть пораньше, то мы могли бы зайти по этой ссылке, почерпнуть новые интересные мысли философа, пообщаться с ним в чате. Следующий элемент — ***CODE*** — служит для обозначения компьютерного кода, как правило, текста программ. Если мы хотим показать какой-то фрагмент программы, мы должны его заключить в элемент ***CODE***. Напишем вот такую "высоко интеллектуальную" программу… Программисты, я думаю, оценят. Как видите, переводы строк браузер не воспринимает, поэтому не дурно было бы элемент ***CODE*** заключить ещё и в элемент ***PRE***.

Далее, элемент ***SAMP*** (от слова ***sample***). В отличие от ***CODE***, он обозначает текст, выводимый в результате работы программы, выглядит так же, как и элемент ***CODE***. Элемент ***KBD*** обозначает текст, который пользователь должен ввести с клавиатуры. Обычно используется в разного рода инструкциях, документации. Оформляется так же, как ***CODE*** и ***SAMP***, при этом выделяется полужирным начертанием.

Последний вот такой "компьютерный" элемент (наберитесь терпения) — это элемент ***VAR***. Предназначен для обозначения переменных. В данном случае имеется в виду, что вместо "***filename***" вы должны в реальной работе написать название файла. Последние два элемента — это ***ABBR*** и ***ACRONYM*** (отмечает аббревиатуры и акронимы). Аббревиатура — это стяжение нескольких слов, означающих единое понятие, в одно слово.

Акроним — это аббревиатура, составленная только из первых букв входящих в неё слов. Внешне похожей на предыдущую группу является группа так называемых ***font-style*** элементов, т.е. элементов для форматирования фрагментов текста. Поскольку эти элементы не предназначены для структурной разметки, а только для оформления, их использования следует по возможности избегать, а для форматирования пользоваться таблицами стилей. Элемент ***I*** — ***italic*** — задает курсивное начертание текста. Элемент ***B*** — ***bold*** — даёт полужирное начертание. Элемент ***BIG*** увеличивает размер шрифта. К сожалению, в ***Dreamweaver***'е мы этого не увидим, нужно открыть нашу страничку в браузере. Вы видите: текст стал слегка побольше.

Противоположный ему элемент — ***SMALL*** — уменьшает текст. Элемент ***TT*** (***teletype***) — выводит текст моноширинным шрифтом, в котором все символы имеют одинаковую ширину. Таким шрифтом оформляются уже знакомые Вам элементы ***CODE***, ***SAMP*** и т.д. Элемент ***S*** или его синоним ***STRIKE*** позволяют зачеркнуть какую-то часть текста. Элемент ***U*** подчеркивает текст. Этим элементом лучше не злоупотреблять, потому что у пользователей Интернета подчеркивание достаточно прочно ассоциируется с гиперссылками. По подчёркнутому тексту сразу хочется кликнуть мышкой.

И ещё один элемент, очень популярный в своё время, но сейчас, к счастью, перешедший в разряд нерекомендованных — элемент ***FONT*** (шрифт). Позволяет задавать размер, цвет и гарнитуру шрифта. До появления таблиц стилей это был единственный механизм управления параметрами шрифта. Понятно, что к структурной разметке этот элемент не имеет никакого отношения. У элемента ***FONT*** есть три атрибута: ***size***, который отвечает за размер шрифта. Размер задается числом от 1 — самый маленький до 7 — самый большой. Другой вариант задания размера шрифта — увеличение или уменьшение кегля относительно текущего. Т.е. мы можем шрифт слегка увеличить или слегка уменьшить с помощью всё тех же семи чисел, но добавляя знаки + или -.

Другой атрибут — ***color*** — позволяет задавать цвет. И замечательный атрибут ***face***, который позволяет указывать гарнитуру шрифта. На практике с этим атрибутом связано большое количество проблем, т.к. указанный шрифт может отсутствовать на компьютере пользователя, либо, что ещё хуже, присутствовать, но не содержать нужных символов. Тогда пользователь имеет все шансы увидеть нечто подобное. Если нам так уж нужно как-то особо оформить фрагмент текста, не выделяя его никаким логическим элементом (что само по себе не очень правильно), то правильнее пользоваться "просто текстовым элементом" ***SPAN*** и задавать его оформление через таблицу стилей. А в данном случае было бы правильнее использовать элемент заголовка.

Последний элемент в этой группе — ***BASEFONT***. Он состоит только из открывающего тега, не имеет содержимого и задает базовый размер шрифта для всего документа с помощью атрибута ***size***. Размер задается всё в тех же 7 числах, значением по умолчанию является 3. На заголовки и текст, заключенный в таблицы, действие этого элемента не распространяется. Рассмотрение других элементов мы продолжим в следующем уроке.

4.1.6. Разметка текста-2

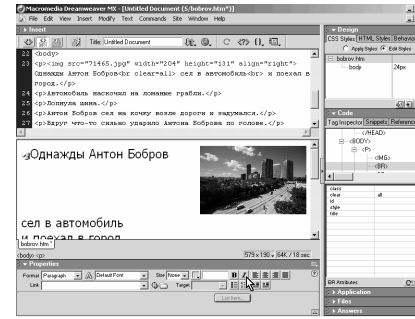
Продолжаем изучение элементов разметки текста. Элемент ***P*** (***paragraph***) служит для разбиения текста на абзацы. Открывающий тег для этого элемента является обязательным, а закрывающий — необязательным. Элемент этот, естественно, является блочным, но при этом может включать только текст и текстовые элементы. Другие блоки он включать в себя не может. По умолчанию абзацы отделяются от других блоков увеличенным интервалом, в отличие от традиционного набора, где абзацы выделяются красной строкой.

Элемент ***P*** имеет атрибут ***align***, отвечающий за выравнивание текста в абзаце. Атрибут является нерекомендованным, правильнее контролировать выравнивание через таблицы стилей. Атрибут ***align*** может принимать

значения: ***left*** (по левому краю), ***right*** (по правому краю), ***center***(по центру), ***justify*** (по правому и левому краям).

Вертикальными чёрточками мы будем обозначать в дальнейшем альтернативные варианты. Значением по умолчанию является ***left***. Т.е. если мы не указываем этот атрибут или браузер не может понять его значение (старые браузеры, например, не понимают равномерное выравнивание ***justify***), то текст выравнивается по левому краю.

Для начала новой строки служит элемент ***BR*** (от слова ***brake***, ***line brake*** — перевод строки). Не следует путать этот элемент с элементом ***P***. Элемент ***BR*** является текстовым элементом. Он не начинает нового абзаца, а просто заставляет браузер перевести строчку до того, как он сделает это сам, дойдя до правого края страницы. Визуально элемент не приводит к увеличению интервала между строками. Состоит только из открывающего тега. Если нужно сделать несколько переводов строк, то нужно вставить соответствующее количество элементов ***BR***. Имеет атрибут ***clear***, который определяет поведение текста, обтекающего изображение. Допустим, текст обтекает некоторое изображение, и мы хотим где-то вставить переводы строк. По умолчанию текст продолжает обтекать картинку.



Но если мы поставим атрибут ***clear***, то текст окажется под картинкой. Принимает значения: ***none*** (когда текст продолжает обтекание). ***left*** (когда обтекание прекращается в случае, если картинка выровнена по левому краю, как раз как в нашем случае). Если бы картинка была выровнена по правому краю, то обтекание текста продолжалось бы. Для этого случая атрибут должен иметь значение ***right***. И, наконец, значение ***all***, когда обтекание прекращается в любом случае, вне зависимости от выравнивания картинки.

Бывают случаи, когда нам нужно наоборот, запретить браузеру перевод строки в определенном месте. Браузер осуществляет перевод строк по словам, т.е. нам нужно отметить некоторый пробел как пробел, в котором перенос осуществляться не может. Для этого пробел заменяется на специальную подстановку *** ***;, которая расшифровывается как ***non-breaking space*** и интерпретируется браузером как неразрывный пробел. Теперь, если перенос и осуществится, то уж никак не в этом месте.

Для обозначения заголовков разделов и подразделов документа предназначены 6 элементов с ***H***1 по ***H***6 (от слова ***heading*** — заголовок,

рубрика). Эти элементы соответствуют 6 уровням "важности".разделов. ***H***1 — самый важный, верхнего уровня, ***H***6 — наименее важный. Эти элементы обязаны иметь как открывающий, так и закрывающий теги. Даже если документ не имеет разделов, у него, как правило, есть хотя бы один заголовок — его название. Обычно заголовки отображаются увеличенным кеглем: ***H***1 — самый большой, а дальше идёт постепенное уменьшение.

Элементы заголовков являются блочными, однако, как и абзац, могут содержать только текст и текстовые элементы. Имеют атрибут ***align***, аналогичный такому же атрибуту элемента ***P***. Заодно покажем, как ***align*** работает на параграфе. Уже упоминавшийся элемент ***PRE*** (от слова ***preformatted*** — предварительно отформатированный) заставляет браузер выводить текст точно в том виде, в каком он фигурирует в ***HTML***-коде, со всеми пробелами и переводами сток. Выводится обычно "моноширинным" шрифтом.

PRE является блочным элементом, должен иметь как открывающий, так и закрывающий теги. Может включать текст и текстовые элементы за исключением элементов изображений ***IMG***, встроенных объектов ***OBJECT***, а также элементов, изменяющих размер шрифта: ***BIG***, ***SMALL*** и некоторых других.

Для обозначения верхних и нижних индексов служат элементы ***SUP*** (***superscript***) и ***SUB*** (***subscript***). Элементы являются текстовыми, должны содержать как открывающий, так и закрывающий теги. Вот пример нижних и верхних индексов. Элементы могут включать в себя текст или другие текстовые элементы. Для выделения цитат предназначены два элемента: блочный ***BLOCKQUOTE*** и текстовый ***Q***. Оба элемента обязаны иметь как открывающий, так и закрывающий теги. ***BLOCKQUOTE*** предназначен для больших цитат, которые можно выделить в отдельный блок. Может включать текст, текстовые элементы и другие блочные элементы. Отображается обычно в виде абзаца с увеличенными отступами по бокам.

Текстовый элемент ***Q*** предназначен для цитат внутри текста. Согласно спецификации браузеры должны автоматически окружать такие цитаты кавычками. На практике же пока не все браузеры делают это или делают неправильно. Проблема с кавычками заключается в том, что в разных языках правила их использования разнятся. В русском, например, используются "елочки", а в английском — "лапки". Для учета таких вот национальных особенностей в спецификации предусмотрен атрибут ***lang***, который задает язык для подобных элементов. Т.е. теоретически если мы напишем вот так, то должны отобразиться "елочки", а если мы зададим английский язык, то "лапки".

Но пока, если браузеры и ставят кавычки, то ограничиваются примитивным знаком дюйма. Следовательно, пока этот элемент приходится считать малоприменимым и просто ставить кавычки в тексте вручную, с помощью специальных подстановок. ***BLOCKQUOTE*** и ***Q*** имеют атрибут ***cite***, который должен содержать ссылку на источник цитаты. В браузере, как видите, этот источник нигде не отображается, он нужен только для структурной разметки.

И в заключение несколько слов о переносах. Браузеры пока не умеют автоматически расставлять переносы, как текстовые редакторы или программы верстки, и переносят текст только по словам. Мы можем поставить в нужном месте слова дефис, и тогда браузер сможет сделать в этом месте перенос. Но, поскольку никто не гарантирует идентичного отображения нашей страницы во всех браузерах, может оказаться, что слово помещается в строку, перенос в этом месте не нужен, а посреди слова будет красоваться наш дефис.

Более правильный способ — использовать подстановку ***&shy***;, которая указывает браузеру место, где можно сделать перенос. Когда он не нужен, слово отобразится слитно, а когда нужен — браузер автоматически должен поставить дефис и перенести оставшуюся часть слова на другую строку. Получается, что, испещрив все слова в нужных местах этими символами, мы получим в документе правильные переносы. Впрочем, на практике, к сожалению (а может, к счастью), мало кто этим занимается.

Итак, мы рассмотрели элементы ***P*** (абзац), ***BR*** (перевод строки), подстановку *** ***; (неразрывный пробел), элементы заголовков, элемент ***PRE*** для предварительно отформатированного текста, элементы для верхних и нижних индексов ***SUP*** и ***SUB***, элементы для выделения цитат ***BLOCKQUOTE*** и ***Q*** и подстановку ***&shy***; для расстановки переносов.

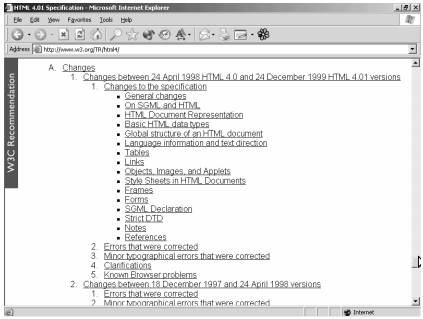
4.2. Технический уровень

4.2.1. Списки

В ***HTML*** имеется несколько механизмов упорядочивания данных в списки или перечни, по-английски ***lists***. Существуют нумерованные списки, ненумерованные списки и списки определений, терминов. Списки могут вкладываться друг в друга, образуя как бы подразделы. Причем вложенные списки не обязательно должны быть одного типа, т.е., например, нумерованный список может содержать в себе ненумерованный. Для создания нумерованных списков служит элемент ***OL*** (***ordeRed list***), для ненумерованных — ***UL*** (***unordeRed list***). Элементы должны иметь открывающий и закрывающий теги. Эти элементы являются блочными.

Вообще, эти элементы представляют собой лишь контейнеры для пунктов списка и поэтому не могут содержать ничего, кроме элементов для

обозначения этих пунктов — ***LI*** (***list item*** — элемент списка). Элемент ***LI***, обозначающий пункт списка, может содержать текст и текстовые элементы. Закрывающий тег у него может отсутствовать. Этот элемент должен находиться только внутри ***OL*** или ***UL***.



Давайте же попробуем разметить некий список. Пусть он будет нумерованным. Ставим открывающий тег <***OL***>. Закрываем этот элемент. Теперь обозначаем пункты списка с помощью элементов <***LI***>. Переключимся в режим визуального просмотра и посмотрим, что у нас получилось. Вот такой планчик.

Если мы хотим сделать вложенный список, то мы должны в один из пунктов вложить элемент списка (например, вложим сюда ненумерованный список) и, естественно, проставить его пункты. Вот как это будет выглядеть. Количество таких вложений практически не ограничено. Все атрибуты этих элементов являются нерекомендованными. Элемент ***OL*** имеет атрибут ***start***, указывающий, с какого числа начинать нумерацию, если это число должно отличаться от 1. Например, по какой-то причине мы хотим начать нумерацию с 10. Пожалуйста. Мало того, элемент ***LI*** имеет атрибут ***value***, позволяющий задать значение конкретному пункту списка. С помощью этого атрибута мы можем пронумеровать список совершенно невообразимым образом. В каких-то случаях это действительно бывает нужно. Все три элемента имеют атрибут ***type***, позволяющий изменить внешний вид маркеров списка.

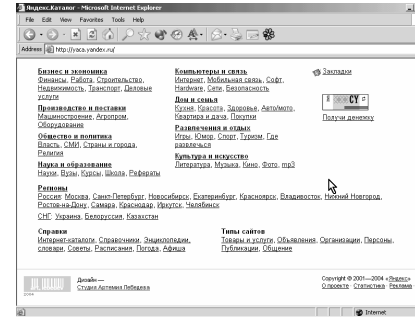
Возможные значения этого атрибута разные для нумерованных и ненумерованных списков. Для ненумерованных это может быть: ***disc*** (залитый кружочек). Давайте посмотрим, как он выглядит. С***ircle*** (контурный кружок) И ***square*** (квадратик). Контурного квадратика почему-то не предусмотрено. Значение по умолчанию зависит от уровня вложенности списка. Например, ненумерованный список первого уровня по умолчанию имеет тип ***disc***, второго уровня — ***circle***, третьего и ниже — ***square***. Для нумерованных списков предусмотрены следующие типы: 1 (арабские цифры). То, что мы видим по умолчанию.

Строчная ***a*** (латинские буквы в нижнем регистре). Прописная ***A*** (латинские буквы в верхнем регистре). Строчная ***i*** (римские цифры в нижнем регистре). И, наконец, прописная ***I*** (римские цифры в верхнем регистре). Мы можем менять тип как для всего списка, так и для отдельных пунктов. Например, создать такое вот приятное разнообразие.

Еще раз оговорюсь, что все эти атрибуты являются нерекомендованными, то же самое можно и нужно делать через таблицы стилей. Помимо нумерованных и ненумерованных списков существует специальная группа элементов для списков определений. Создается список с помощью элемента ***DL*** (***definition list*** — список определений) и включенных в него пар ***DT*** () — ***DD***. ***DL*** является блочным элементом, должен иметь оба тега — открывающий и закрывающий и может содержать только элементы ***DT*** и ***DD*** — очень похоже на списки, на элементы ***OL*** и ***UL***. ***DT*** (***definition term***) отмечает определяемый термин. Может содержать текст и текстовые элементы. Закрывающий тег может отсутствовать. ***DD*** (***definition description***) — содержит собственно определение термина. Может включать текст и текстовые элементы, а также блочные элементы. Закрывающий тег необязателен.

В спецификации приведено ещё одно интересное назначение этих элементов: для описания диалогов (***DT*** в этом случае содержит имя персоны, а ***DD*** — его или её речь). Давайте же создадим небольшой словарик. Заключаем всё в элемент <***DL***>, определяем термины. И их определения. Вот как это будет выглядеть. Из соображений совместимости поддерживаются ещё два устаревших элемента: ***MENU*** и ***DIR***. Отображаются они точно так же, как ненумерованный список ***UL*** и именно ***UL*** и нужно использовать вместо них.

4.2.2. Ссылки



Как вы помните, возможность связывать отдельные фрагменты информации перекрестными ссылками является главной особенностью гипертекста. Собственно, гиперссылки — это основное, что отличает гипертекст от других форм представления информации. Гиперссылка связывает две точки и имеет определенное направление: "отправная точка", т.е. фрагмент информации, помеченный в качестве ссылки, ведёт, указывает на "пункт назначения", т.е. другой фрагмент, расположенный где-либо в Интернете. Когда пользователь активирует ссылку (например, кликает по ней мышкой или переходит с помощью клавиатуры или с помощью голосовых команд), то браузер переходит по этой ссылке, то есть показывает пользователю информацию, расположенную в "пункте назначения". "Пунктом назначения" не обязательно должна быть ***HTML***-

страница. Это может быть самая разная информация: изображения (например, часто дают маленькое изображение, которое является ссылкой на его более крупную версию), звуковые файлы, программы, ну и конечно гипертекстовые страницы или даже отдельные элементы на них.

Соответственно, и поведение браузера при переходе по гиперссылке будет разным. Если это ***HTML***-страница, она будет показана пользователю; если адрес электронной почты, то при переходе по такой ссылке будет запущена почтовая программа и создано новое письмо; если программа, то будет предложено сохранить её на диск… Ну и так далее.

Для разметки гиперссылок используется элемент <***A***></***A***>. Это текстовый элемент. Должен содержать как открывающий, так и закрывающий теги. Содержимым этого элемента могут быть текст и текстовые элементы, в том числе и изображения. Ссылки не могут вкладываться друг в друга, т.е. внутри одного элемента <***A***></***A***> не должно быть других элементов <***A***>.

Такая вот разметка будет неправильной. Помеченный как гиперссылка текст обычно оформляется синим цветом и подчеркиванием. Это, конечно, не догма, но, чтобы пользователь мог легко находить в тексте ссылки и не путать их с другим каким-либо оформлением, следует более-менее придерживаться принятого способа обозначения ссылок. А также не использовать его в других целях. Нет ничего неприятнее в Интернете, чем подчеркнутый текст, не являющийся ссылкой. Если ссылкой является изображение, оно по умолчанию обычно обводится синей рамкой. Но об этом мы поговорим при изучении изображений.

Немаловажным является вопрос, какую часть текста делать гиперссылкой. Однозначно можно сказать, что не следует делать гиперссылок вроде "нажмите здесь". Это абсолютно бессмысленно. Подчеркнутый текст и так говорит пользователю, что на него нужно кликнуть. Это всё равно, что писать "нажмите здесь" на всех в мире кнопках. Тем более, что "нажимают" только пользователи визуальных браузеров. Достаточно совсем небольшого опыта работы в сети, чтобы выработался условный рефлекс на подчеркнутый текст. Именно поэтому так важно то, о чём мы говорили выше: не оригинальничать в оформлении ссылок.

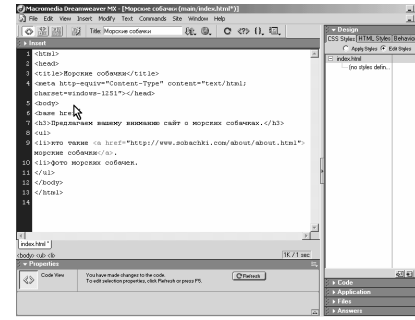
Их подчеркивание — не чья-то прихоть, а необходимое условие формирования в сознании пользователя стереотипа, помогающего ему ориентироваться в гипертекстовой среде. Что же тогда следует заключать в элемент <***A***>? Прежде всего, это должны быть информационно значимые слова. Ссылка не должна быть слишком длинной. Не нужно заключать в элемент <***A***> целые предложения. Гораздо лучше заключить в него наиболее важную часть предложения. Такие ссылки будут более заметны в тексте и пользователю будет легче "зацепиться" за них взглядом. Хотя, конечно, нет

правил без исключений. Существуют, например, сайты, представляющие из себя каталоги ссылок, где 90% страницы заняты именно ссылками.

Но это — отдельный случай. Теперь подробнее о том, как же, собственно, нам сделать некий фрагмент нашего документа гиперссылкой, а точнее говоря, сделать его "отправной точкой" гиперссылки. Как вы помните, его нужно заключить в элемент <***A***></***A***>, но этого ещё не достаточно. Элементу нужно задать атрибут ***href***. Значением этого атрибута будет как раз адрес "пункта назначения", т.е. документа, на который эта ссылка ведет.

Адрес этот может быть абсолютным или относительным. Что это значит? Абсолютный адрес — это ***URL***. Надеюсь, вы помните, что это такое: ***Uniform Resource Locator***, адрес, однозначно определяющий расположение документа в Интернете. При этом "адрес отправителя", т.е. страницы, с которой ведёт ссылка, не имеет никакого значения. Так же как не важно, из какого почтового отделения вы отправляете письмо. Для ссылок на страницы вне вашего сайта это единственный способ. Однако если документ находится на этом же сайте, можно использовать так называемые относительные ссылки. В таких ссылках путь к "пункту назначения" прокладывается от "исходного" документа, т.е. от того, в котором находится ссылка.

Допустим, у нас в документе ***index.HTML*** находится ссылка на документ ***about.HTML***. Документы эти, как видите, лежат в одной папке. В этом случае в ссылке нам достаточно указать только название файла. Если "пункт назначения" находится в подпапке, то нам уже нужно указать имя этой папки, слэш, имя файла. Может быть и такая структура, что нам нужно подняться из текущей папки на уровень выше, зайти в другую папку, и в ней уже находится наш "целевой" документ. Тогда мы ставим в ссылке две точки — это обозначает подъем на один уровень. И затем, опять же, пишем название папки и файла.



Тем, кто работал с командной строкой ***DOS***, буде такие найдутся, всё это должно быть хорошо знакомо. Сайт с относительными ссылками будет работать вне зависимости от того, где он находится: на домене таком или сяком, или вообще у вас на винчестере. Главное, чтобы сохранялась его структура, т.е. взаимное расположение папок и файлов. К полюсам относительных ссылок относится и некоторое сокращение объёма файлов. Если у вас на странице достаточно много ссылок и от каждой отрезать "лишний" хвост абсолютной ссылки, то экономия может

получиться довольно существенная. "Исходной точкой" для относительных ссылок можно принудительно сделать не текущий документ, а любой другой. Для этого предназначен элемент <***BASE***>, у которого в атрибуте ***href*** можно задать эту самую "исходную точку".

Например, перенести туда доменное имя. Тогда все ссылки можно делать относительными, они будут относительно этого доменного имени. Получится нечто среднее между абсолютными и относительными ссылками. Ссылка в конечном счёте будет абсолютной, но мы получим экономию на длине адресов. Как упоминалось выше, ссылка может вести на конкретное место ***HTML***-страницы. В этом случае в атрибуте ***href***, в адресе, после имени файла ставится решетка и некое название метки, имя целевого элемента.

Каким же образом мы можем поставить в определенном месте документа такую метку? Для этого используется всё тот же элемент <***A***>. Вот он.Только в данном случае мы вместо атрибута ***href*** применяем атрибут ***name***, значением которого как раз является имя метки, на которую мы можем затем ссылаться. Для этих имён меток существует ряд правил. Во-первых, они, естественно, должны быть уникальны в пределах документа, то есть не должно быть двух меток с одинаковыми именами, иначе будет не понятно, на какую из них ведёт ссылка. Во-вторых, в них не стоит использовать какие-либо символы кроме латинских букв и цифр. И последнее замечание — по поводу регистра.

Согласно спецификации атрибут ***name*** чувствителен к регистру, т.е. одно и то же имя, написанное строчными или прописными буквами, должно считаться разными именами. На практике же большинство браузеров эту разницу регистров не учитывают, поэтому желательно называть метки действительно по-разному. Некоторые браузеры не понимают пустого элемента ***A***, то есть, если бы мы написали вот таким образом, что содержимого у нас не было бы, некоторые браузеры могли бы это не воспринять. Поэтому следует по возможности заключать в этот элемент какие-то данные. Итак, если элемент ***A*** имеет атрибут ***href***, то он становится отправной точкой ссылки, а если атрибут ***name*** — то пунктом назначения. Один и тот же элемент может иметь и атрибут ***href***, и атрибут ***name***, т.е. быть и "пунктом назначения" для других ссылок и в тоже время сам ссылаться на другой ресурс.

Есть ещё один способ задания имён меток помимо элемента ***A*** с атрибутом ***name***. Вы помните, что практически любому элементу можно задать атрибут ***id***. Так вот на этот самый ***id*** и можно ссылаться, так же, как на элемент ***A*** с атрибутом ***name***. В данном случае мы достигли того же результата, только без использования фактически не нужного здесь элемента ***A***. К сожалению, некоторые старые браузеры этот атрибут ***id*** не понимают. Вы, наверное, заметили, что здесь мы ссылаемся на определенное место этого же самого документа, а не какого-то внешнего. Ссылка наверху страницы ссылается на

некоторое место внизу страницы. В таком случае писать в ссылке имя страницы не нужно, достаточно решетки и имени целевого элемента, как, собственно, и было в оригинале. Ссылки внутри одной страницы — это как раз типичный пример использования ссылок на именованные элементы.

И последнее о таких ссылках: при всей их полезности они таят в себе некоторое противоречие. Порцией гипертекста является всё-таки страница. И переход по ссылке подразумевает переход на другую страницу, другой файл. Возможность перехода на отдельное место страницы нарушает эту логику. Все ссылки выглядят одинаково, и пользователь не знает, куда он сейчас попадёт: на другую страницу или останется на этой, но перейдёт в другое её место. Кнопка ***back***, которая служит для возврата на предыдущую страницу, в данном случае должна вернуть пользователя то ли к предыдущему месту на текущей странице, то ли всё-таки на предыдущую страницу. Для опытного пользователя, конечно, всё это не составляет проблемы: он видит в адресе решетку и понимает, что происходит. Но многие пользователи не знают ничего ни об адресах, ни об именах элементов. Они просто кликают по ссылке и попадают… не совсем понятно куда.

Часто рядом с таким вот "целевыми элементами" ставят ссылку наверх страницы, чтобы пользователь мог по крайней мере вернуться к шапке документа и понять, где же он находится. Одним словом, механизм таких вот "локальных" ссылок, полезный сам по себе, скорее всего, просто не очень продуман пока с точки зрения интерфейса. Ну и о других атрибутах . Ссылка может иметь атрибут ***title***, содержащий некую дополнительную информацию. Эта информация может отображается, например, в виде так называемых всплывающих подсказок.

Ещё один атрибут — ***target*** — указывает "окно", в котором должна открыться ссылка. Подробнее об этом мы поговорим при изучении так называемых "фреймов", а сейчас нас может заинтересовать значение этого атрибута "***_blank***", при котором ссылка открывается в новом окне.

4.2.3. Таблицы

Без преувеличения можно сказать, что таблицы в ***HTML*** — это больше, чем таблицы. Вообще, надо сказать, ***HTML*** полон таких вот странных примеров использования вещей "не по назначению", приспосабливания их для решения каких-то задач ввиду отсутствия нормальных механизмов решения этих задач. И таблицы — один из наиболее характерных примеров. Помимо вывода собственно табличной информации они долгое время использовались (да и сейчас продолжают использоваться) для создания средствами ***HTML*** какого-либо более-менее вразумительного макета. Дело в том, что ***HTML*** предполагает последовательную загрузку: пришла порция текста —

отобразилась в браузере, пришла следующая — отобразилась дальше. И так, пока не загрузится вся страница.

При медленном соединении это особенно актуально: Вы можете начинать чтение, пока остаток страницы догружается. Естественно, никакая верстка при этом невозможна: пока мы не получим весь текст, непонятно, как его верстать: может быть там осталось две строчки, а может — 10 страниц. А вдруг там внизу окажется какая-нибудь огромная картинка? Поэтому изначально в ***HTML*** вообще не было заложено никаких механизмов верстки: гипертекст аккуратненько заполнял всё окно браузера. Но когда появились механизмы отображения таблиц, кто-то подумал: а почему бы не заключить в таблицу всё содержимое страницы?

Ну и пошло-поехало. Так вот и вбивались гвоздики в крышку гроба ***HTML***-я как языка исключительно структурной разметки: таблицами стали размечать вовсе не табличную информацию. Сейчас многие подобные несуразности уходят в историю, появились и более правильные, чем таблицы, механизмы позиционирования информации на странице (через таблицы стилей). Помимо теоретической неправильности такого использования таблиц, оно имеет и практические недостатки. Главный из них: браузер не может отобразить таблицу, не дождавшись всего помещённого в неё текста. Т.е. мы лишаем пользователя чудесной возможности начинать просмотр страницы, не дожидаясь её полной загрузки.

Вообще же подход к верстке гипертекстовых страниц должен быть иной, нежели в полиграфии. Дизайнерам очень трудно с этим смириться, но здесь невозможно точно скомпоновать всё, как на листе. Прежде всего потому, что у нас нет этого "листа": мы не можем предугадать размер окна, в котором будут просматривать нашу страницу. Кроме того, мы не можем предугадать ещё массу факторов, влияющих её внешний вид. Информация должна быть пригодна к восприятию в самых разных условиях. И вместо того, чтобы искать способы "запереть" размер шрифта или задать ширину страницы, следует разрабатывать дизайн таким образом, чтобы он нормально смотрелся при разной ширине страницы и разном размере шрифта.

Итак, таблицы… Создаются они с помощью элемента <***TABLE***>. Элемент блочный, закрывающий тег — обязательный. Имеет нерекомендованный атрибут ***align***, определяющий положение таблицы в документе. Может

принимать значения: ***right*** — выравнивание по правому краю, ***center*** — выравнивание по центру и ***left*** — значение по умолчанию — выравнивание по левому краю. Также имеет атрибут ***width***, позволяющий задать ширину таблицы и, собственно, выстраивать некий макет.

Ширина задается либо жестко — в пикселях, либо в процентах от ширины окна браузера (в этом случае к значению атрибута приписывается знак процента). Теперь наша таблица растягивается в соответствии с шириной окна. Она может иметь ширину 100% или любой другой процент. Тут опять же есть тонкости: если заданная ширина ячейки не позволяет вместить содержимое (например, в ячейке находится большая картинка или длинное слово), то, не взирая на заданную ширину, ячейка растянется так, чтобы вместить всё содержимое. Если ширина таблицы вообще не задана, то браузер сам определяет её, исходя из содержимого. Внутри себя элемент ***TABLE*** может содержать собственно ячейки таблицы (об этом — буквально через секунду), а также элемент <***CAPTION***>. Причем этот элемент должен размещаться до ячеек, т.е. сразу после открывающего тега <***TABLE***>. Этот элемент задает подпись к таблице. Этот элемент может иметь нерекомендованный атрибут ***align***, определяющий положение подписи относительно таблицы. Может принимать значения: ***top*** (над таблицей), ***bottom*** (под таблицей), ***left*** (слева) или ***right*** (справа).

А теперь самое интересное — собственно, содержимое таблицы. Оно может распределяться по 3 группам. Первая группа — шапка таблицы — элемент <***THEAD***>. Обычно это первая строка (или строки) таблицы, содержащая заголовки столбцов. Вторая группа — "подвал" таблицы — элемент <***TFOOT***> — это те же самые заголовки, только расположенные в конце таблицы. И, наконец, третья группа — собственно "тело" таблицы — элемент <***TBODY***>.

К шапке и подвалу мы вернемся позже, тем более, что это вещи пока довольно экзотические, а сейчас рассмотрим тело таблицы. Элемент этот, вообще говоря, является элементом подразумеваемым, так же как <***HTML***> или <***BODY***>. Т.е. если у вас в таблице нет шапки и подвала, а одно только "тело", то явно указывать <***THEAD***> не обязательно. Содержимое таблицы состоит из строк, разбитых на ячейки. Вот таким образом. Устроить таблицы наоборот, "поколоночно", невозможно, хотя в каких-то случаях это было бы удобнее.

Итак, строка… Задается с помощью элемента <***TR***> (***table row*** — строка таблицы). Закрывать этот элемент необязательно. Строка содержит ячейки, задаваемые элементами <***TD***> и <***TH***>. Закрывающие теги у этих элементов также необязательны. Какая же разница между ***td*** и ***th***? <***TD***> — это ячейка с данными, а <***TH***> — ячейка с "названием" строки или столбца. Например, если

у нас, как в данном случае, есть ячейки, содержащие фамилии людей, и есть ячейки, содержащие некоторые данные по этим людям, то фамилии следует заключать не в <***TD***>, а в <***TH***>. В визуальных браузерах <***TH***> обычно выделяется полужирным начертанием и выравниванием по центру ячейки.

О выравниваниях мы поговорим позже. С помощью атрибута ***width*** можно указывать ширину ячеек, однако, как и в случае таблицы в целом, указание ширины носит для браузера скорее рекомендательный характер. Широкое содержимое всё равно при необходимости "растянет" ячейку, а некоторые браузеры могут наоборот, сужать их до размеров содержимого, не глядя на указанный атрибут. Ещё у ячеек есть атрибут ***height***, задающий высоту, но он работает ещё менее четко, чем ***width***, и надеяться с его помощью корректно задавать высоты ячеек не стоит. Для этого приходится пользоваться разными ухищрениями, о которых речь пойдёт дальше.

Если нужно внутри ячейки запретить перенос по словам, можно задать этой ячейке атрибут ***nowrap***. Ему не присваивается никакого значения, просто пишется такое слово. Это значит, что содержимое ячейки должно вытягиваться в одну строчку. Однако механизм этот не очень надежен. Например, если задана ширина ячейки, то перенос всё-таки осуществится. Надёжнее пользоваться элементом <***NOBR***>. Этот элемент часто применяется и в других случаях. Все, что заключено в него, выводится в одну строку, без переносов.

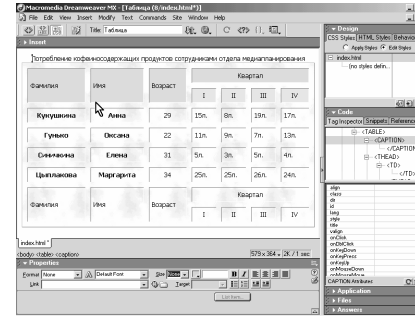
Несколько ячеек можно объединять в одну. Для этого служат атрибуты ***colspan*** и ***rowspan***. Разберём на простом примере. Допустим, мы хотим объединить несколько ячеек по горизонтали. Скажем, первую и вторую. Для горизонтального объединения служит элемент ***colspan***. Значением этого атрибута является количество объединяемых ячеек, включая и первую ячейку. В данном случае их будет две. Соответственно, вторую ячейку мы удаляем, так как она у нас слилась с первой. И вот, что у нас получилось. Мы можем объединить и три ячейки. В таком случае у нас всю верхнюю строку займет одна ячейка. ***Rowspan*** объединяет ячейки "по вертикали". Берет ячейки не из соседних столбцов, а из соседних строк. Можно, конечно, захватить для объединения ячейки и больше, чем из одной строки. Можно одновременно применять и ***colspan***, и ***rowspan***.

Естественно, мы не можем объединять ячейки какой-нибудь "буквой Г". При объединении ячеек нужно быть очень внимательным, чтобы не запутаться, где у нас сколько элементов ***TD***. Например, в данном случае мы имеем три строки по три ячейки. Но первая ячейка объединяет две по горизонтали и две по вертикали. Соответственно, в первой строке остается одна неприсоединенная ячейка. Во второй строке две первых ячейки относятся к первой строке, то есть остается также одна ячейка. Ну а третья строка не

тронута. Справедливости ради надо сказать, что составлять таблицы, конечно, удобнее в ***HTML***-редакторе, а писать их вручную — не самое эффективное занятие. Но надо уметь по крайней мере разобраться в коде, чтобы найти ошибку, если таблица "разъехалась", а редактор найти эту ошибку не может.

Ячейки таблиц могут содержать вложенные таблицы. Особенно это актуально как раз для "неправильного" использования таблиц — для создания макета страницы. Вот таким необычным образом несчастные дизайнеры вынуждены реализовывать свои смелые идеи.

Теперь вернемся к шапке и подвалу — элементам <***THEAD***> и <***TFOOT***>. И шапка, и, как это ни удивительно, подвал в ***HTML***-е должны идти до тела таблицы. Однако это перестает казаться удивительным, если разобраться, для чего они нужны. Правда, браузеры пока толком не поддерживают все эти замечательные функции, но замысел здесь следующий: во-первых, реализовать скролиинг содержимого таблицы независимо от шапки и подвала (такая возможность есть, например, в Экселе: вы проматываете содержимое большой таблицы, при этом названия столбцов остаются у вас перед глазами), во-вторых, при распечатке длинных таблиц, не помещающихся на одной печатной странице, шапка и подвал могут автоматически выводится на каждой странице.



Теперь поговорим об оформлении таблиц. Опять же оговорюсь, что правильнее контролировать оформление через таблицы стилей, но перечисленные ниже атрибуты всё ещё поддерживаются браузерами. Прежде всего, конечно, какие у нас есть механизмы раскрашивания, прошу прощения за грубое слово, таблиц? Таких механизмов два: атрибут ***bgcolor***, задающий фоновой цвет. Если мы применяем этот атрибут к таблице, то она вся заливается этим цветом. Либо мы можем задать цвет фона отдельных ячеек. Тогда нужно проставить атрибут соответствующим элементам ***TD***. Вот таким, например, образом. Итак, атрибут ***bgcolor*** может быть установлен как для всей таблицы, так и выборочно, для отдельных её ячеек.

Второй механизм — атрибут ***background***, позволяющий задать в качестве фона не цвет, а некое графическое изображение. Значением атрибута является адрес файла фоновой картинки. Об этом мы подробнее поговорим при изучении изображений. Вот такая красота. Теперь остановимся на

выравнивании. Содержимое любой ячейки может выравниваться тем или иным образом по горизонтали и по вертикали.

Горизонтальное выравнивание контролирует знакомый вам атрибут ***align***. Он может принимать значения ***left***, ***center***, ***right***, ***justify***. Все эти значения Вам уже должны быть знакомы. По умолчанию обычно ячейки с данными <***TD***> выравниваются по левому краю, ячейки-заголовки <***TH***> — по центру. Как раз с помощью атрибута ***align*** мы можем их выравнивание изменить (например, выровнять возраст по центру).

За вертикальное выравнивание отвечает атрибут ***valign***. По умолчанию содержимое вертикально выравнивается по центру. Мы можем это изменить. Вот этот атрибут, который может принимать значения ***top*** (выравнивание по верхнему краю), ***bottom*** (выравнивание по нижнему краю) или ***middle*** -это то, что мы видим по умолчанию — выравнивание по центру.

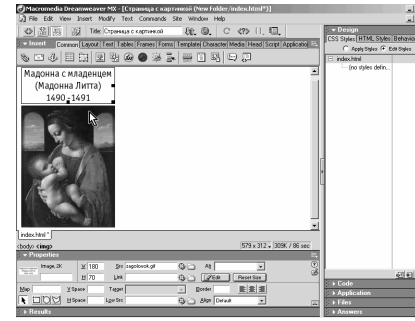
Следующий инструмент оформления таблиц — это расстояние между ячейками таблицы и между их содержимым. Для этого предназначены два атрибута: ***cellspacing*** и ***cellpadding***. ***Cellspacing*** задает расстояние между ячейками таблицы. ***Cellpadding*** задает отступ от границы ячейки до её содержимого. Значения эти задаются разом для всей таблицы. Таблицы стилей позволяют задавать значения и для отдельных ячеек. Обратите внимание, что в промежутки, образованные атрибутом ***cellspacing***, просвечивает фон таблицы. Это позволяет нам имитировать разные интересные обводки (например вот такие, или такие) варьируя ***cellspacing***, цвет таблицы и ячеек. Очень часто, особенно при использовании таблиц для создания макетов, значения этих атрибутов (так же как и толщину рамки) обнуляют, чтобы содержимое ячеек вплотную прилегало друг к другу. Таблица тогда фактически не видна, а видно только, каким образом сверстан текст.

Граница (рамка, обводка) таблицы и её ячеек контролируются атрибутами ***border***, ***frame*** и ***rules***. ***Border*** задает ширину рамки таблицы в пикселях. Как видите, это рамка самой таблицы, а не ячеек. Атрибут ***frame*** указывает, с каких сторон таблицы нужно рисовать границу, а с каких — нет. Значения могут быть следующие: ***void*** (рамка не отображается), ***box***, ***border*** (все 4 стороны — это значение по умолчанию), ***above*** (только верхняя граница), ***below*** (только нижняя граница), ***hsides*** (верхняя и нижняя границы), ***vsides*** (левая и правая), ***lhs*** (только левая), ***rhs*** (только правая). Вроде, ничего не забыл.

Rules отвечает не за внешнюю границу таблицы, а за границы между ячейками. Тут возможных значений чуть меньше: ***none*** (границ между ячейками нет), ***all*** (значение по умолчанию — изображаются все границы), ***rows*** (видны границы только между строками), ***cols*** (только между столбцами)

и, наконец, ***groups*** (границы проводятся только между группами строк, то есть шапкой, телом, подвалом).

4.2.4. Изображения



Вы, наверное, давно ждёте этого урока, изнывая от всех этих тегов и элементов. Ну когда же можно будет навести красоту и поместить среди всех этих блоков текста, заголовков и таблиц потрясающую воображение картину, которая одна скажет зрителю всё лучше, чем десять тысяч чудесно структурированных гипертекстовых слов?! Что ж, ликуйте: вот вам элемент <***IMG***>. Элемент этот состоит только из открывающего тега. Главный его атрибут — ***src*** — задаёт имя файла изображения. Поскольку картинка — это не текст, сам по себе текстовый ***HTML***-файл не может содержать в себе картинку. Он может только ссылаться на неё, т.е. иметь некоторую метку, говорящую браузеру, что "вот в этом месте должна быть вон та картинка".

Из этого следует, во-первых, что картинки лежат отдельно от ***HTML***-файлов, а во-вторых, что атрибут ***src*** принимает значения, аналогичные значениям ***href*** атрибута <***A***>. Т.е. мы фактически ставим ссылку на изображение, а элемент <***IMG***> заставляет браузер самостоятельно "сходить" по этой ссылке, получить картинку и отобразить её в указанном месте нашего документа. Соответственно, всё, что было сказано относительно гиперссылок, справедливо и для изображений: адрес картинки может быть относительный или абсолютный.

Как вы знаете, изображения могут быть в формате ***GIF*** или ***jpg***. Сейчас многие браузеры поддерживают также формат ***png***, обладающий рядом замечательных свойств. Многие, но не все. Поэтому универсальными форматами веб-графики по-прежнему остаются ***GIF*** и ***jpg***. Атрибут ***alt*** позволяет задать описание изображения. Значение этого атрибута больше, чем кажется. Изображение само по себе — это некий чуждый гипертексту элемент, менее универсальный: в отличие от текста, воспринять изображение можно только визуально. Соответственно, для невизуальных браузеров, а также различных автоматических систем, поисковых, например, содержание изображений остается тайной за семью печатями.

Мало того, в визуальных браузерах также есть возможность отключить показ графических элементов (с целью экономии времени или трафика —

объема скачиваемой информации). Тогда на месте изображений появляются прямоугольники с ***alt***-текстом. Это позволяет нам продолжать пользоваться сайтом даже с отключенной графикой. Итак, атрибут ***alt*** позволяет создать "текстовую" версию изображения. Если на картинке изображен текст, то сам бог велел этот текст продублировать в атрибуте ***alt***. Если что-то содержательное, то дать краткое описание этого содержания.

А вот если картинка выполняет чисто декоративную функцию, то ***alt***-текст следует оставить пустым, не забивать страницу лишней информацией. Вряд ли слушателю звукового браузера будет интересно узнать, что между заголовком и картинкой располагается какая-то линеечка. У изображений имеются атрибуты ***width*** и ***height***, задающие ширину и высоту изображения. Это может показаться странным, но значения этих атрибутов могут не совпадать с шириной и высотой самого изображения. И браузер худо-бедно растянет изображение под заданный формат (менее качественно, конечно, чем графический редактор, но растянет).

Однако атрибуты эти созданы в основном не для деформации изображений (хотя иногда такая деформация очень даже бывает нужна), а для того, чтобы браузер мог зарезервировать на странице место нужного размера и загружать следующий текст, не дожидаясь загрузки картинки. Дело в том, что пока картинка не начала загружаться, браузер не знает её размера.

Это приводит к тому, что если эти атрибуты не заданы, то уже загруженный текст страницы "переверстывается" на ходу по мере загрузки картинок. Выглядит это, прямо скажем, не ахти. А в некоторых старых браузерах даже останавливалась дальнейшая загрузка текста, пока не с сервера не "приходило" изображение. В свете сказанного рекомендуется всегда явно указывать размеры всех изображений на странице.

Что касается умышленной деформации, то она позволяет делать некоторые фокусы. Например, в своё время дизайнер по имени Дэвид Сигель придумал т.н. "графическую распорку". Это изображение в формате ***GIF*** размером 1***x***1 пиксель, прозрачное. На странице такое изображение не видно. Теперь, вставляя в нужном месте документа это изображение и задавая ему нужный размер с помощью атрибутов ***width*** и ***height***, мы можем точно управлять пустыми пространствами на нашей странице. Например, ячейка таблицы, в которую заключена такая "распорка" уже не сожмётся меньше размера этой распорки. С точки зрения содержания это, конечно, полный нонсенс, но до появления таблиц стилей распорки были вообще одним из основных механизмов управления размещением информации на странице: отступы, межстрочные интервалы и т.д., всё это имитировалось с помощью распорок.

Другой фокус: если нам нужно изобразить нечто вытянутое в одном из двух измерений, например, какую-нибудь тень, нам достаточно взять "сечение"

этого предмета шириной 1 пиксель и вытянуть его с помощью атрибута ***width*** настолько, насколько нам нужно. При этом, как вы понимаете, мы существенно экономим на объеме графических файлов. Другие атрибуты изображения: ***hspace*** и ***vspace*** задают поле вокруг изображения. Соседняя с картинкой информация (текст или другие картинки) может прилегать к ней вплотную, а может отстоять на некоторое расстояние. ***Hspace*** задаёт это расстояние по горизонтали, а ***vspace*** — по вертикали. Особенно это актуально как раз при обтекании картинки текстом, поскольку текст, обтекающий картинку вплотную в большинстве случаев смотрится неудовлетворительно.

Теперь о самом этом обтекании. Задается оно атрибутом ***align***. Вообще элемент <***IMG***> является текстовым и не начинает новой строки. Следующие за ним тексты или изображения по умолчанию примыкают к нему справа, но выравниваются по нижнему краю. Иногда такой вариант не подходит. С помощью атрибута ***align*** мы можем задать следующие варианты: ***bottom*** (это то, что мы имеем по умолчанию: текст выравнивается с изображением по нижнему краю), ***top*** (текст выравнивается по верху картинки), ***middle*** (текст выравнивается по центру картинки).

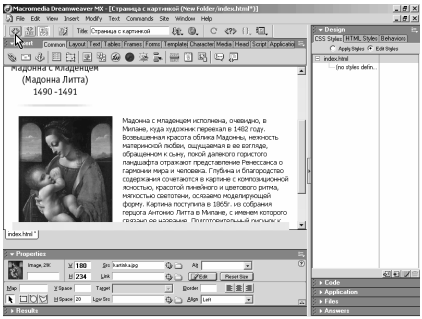
Пока об обтекании картинки текстом речи не идёт: вторая строка текста окажется под картинкой. Если же нам нужно встроить картинку в текст, то атрибут должен принять одно из значений ***left*** (картинка слева, текст обтекает её справа) или ***right*** (картинка выравнивается по правому краю, текст обтекает её слева). Обтекать картинку может не только текст, но и, например, другие изображения меньшего размера. Таким образом можно элегантно создавать конструкции, для которых, на первый взгляд, необходима таблица. Здесь же мы добиваемся того же эффекта без таблиц.

Здесь нам невредно будет вспомнить элемент <***BR***> с его атрибутом ***clear***. Как вы помните, если мы поставим перенос строки в текст, обтекающий изображение (или даже два переноса), то перенос осуществится, но обтекание продолжится и с новой строки. С помощью атрибута ***clear*** мы можем прекратить обтекание и перенести новую строку под изображение, причем сделать это на выбор: только в случае, если текст обтекает картинку справа (***clear=left***), слева (***clear=right***) или в любом случае (***clear=all***). Относится это не только к тексту, но и к "обтекающим" изображениям.

Вначале мы говорили, что элемент <***IMG***> похож на элемент <***A***>. Ещё одно их сходство — это атрибут ***name***. Правда, назначение этого атрибута в данном случае несколько иное: использовать его в качестве "пункта назначения" ссылки не получится, а нужен он для разных программистских нужд, для манипулирования изображением. Правда, атрибут этот несколько устарел, т.к. изображению, как и любому другому элементу, может быть присвоено имя с помощью универсального атрибута ***id***. Атрибут ***border*** задает ширину рамки

вокруг изображения. Атрибут, естественно, устаревший в пользу таблиц стилей, но тут есть одна тонкость: когда у нас в документе просто стоит изображение, его рамка по умолчанию равна нулю, и беспокоиться нам не о чем. Но вот если изображение является гиперссылкой, а это происходит довольно часто, то по умолчанию вокруг него появляется синяя рамка. Выглядит это обычно довольно несуразно. Соответственно, для изображений-ссылок, заключенных в элемент <***A***>, нам необходимо задать нулевую ширину рамки. Можно сделать это либо с помощью таблиц стилей, либо с помощью атрибута ***border***, задав его значение равным нулю.

Теперь рассмотрим ещё один хитрый механизм. Как сделать изображение гиперссылкой, вы знаете, но можно, оказывается сделать гиперссылкой часть изображения, причём произвольной формы. И таких "частей" на одном изображении может быть несколько. Таким образом мы можем обойти прямоугольность картинки и сделать ссылки какой-то необычной формы. Делается это с помощью так называемых ***Image maps*** (карт изображений). Карты эти задаются отдельно от изображений с помощью специального элемента <***MAP***>. Элемент этот имеет обязательный атрибут ***name***, по которому его сможет найти изображение. Закрывающий тег — обязателен.



Элемент <***MAP***> содержит элементы <***AREA***>, каждый из которых описывает одну из таких вот "чувствительных областей" изображения с помощью следующих атрибутов: ***shape*** — форма области. Может быть многоугольником (***poly***), прямоугольником (***rect***) или окружностью (***circle***). ***Coords*** — задает координаты области. Координаты эти, задаются в пикселях и, естественно, должны укладываться в размер изображения. Способ их задания зависит от формы области. Для прямоугольника это две пары координат: ***x*** и ***y*** верхнего левого угла и ***x*** и ***y*** правого нижнего угла.

В данном случае будет создана квадратная область размером 100x100 пикселей. Для окружности задаются координаты центра и радиус. В данном случае будет создана окружность радиусом 20 пикселей с центром в точке, отстоящей на 50 пикселей от верхнего края и на 50 — от левого. Ну а для многоугольника перечисляются через пробел координаты всех его точек. Следующий атрибут элемента <***AREA***> — ***href*** — полностью аналогичен атрибуту ***href*** гиперссылки, то есть задаёт собственно ссылку.

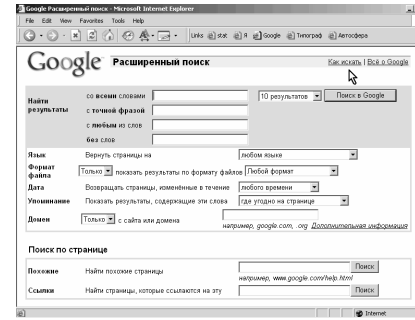
Наконец, знакомый вам по изображениям атрибут ***alt*** позволяет задать ***alt***-тексты не для всей картинки, а для каждой из таких областей. Вероятно, у вас уже зашевелились волосы на голове от всех этих игреков, но, конечно, никто не заставляет вас корпеть с измерителем над картинкой. Существуют специальные программы, где вы размечаете все эти области с помощью специальных инструментов, а вам уже выдается готовый ***HTML***-код.

И, наконец, как всю эту чудовищную конструкцию присобачить теперь к изображению? Изображению этому нужно задать атрибут ***usemap***, значением которого будет имя карты, заданное через атрибут ***name*** элемента <***MAP***>. И имя это должно предваряться решеткой. Вот и всё об элементе <***IMG***>, являющимся одним из основных инструментов веб-дизайнера. Осталось осветить ещё одну тему для продвинутых учеников. Вообще-то, изображения — это не единственный пример внешних данных, включаемых в ***HTML***-документ. Таким же образом в наши странички могут встраиваться и другие виды информации, не являющейся гипертекстом.

Например, звук, анимация, ***flash***-ролики, целые программные модули и даже другие ***HTML***-документы. До сих пор с подобным встраиванием царит некий сумбур, нет единого механизма, хотя по сути с точки зрения встраивания нет такой уж большой разницы между картинкой, звуком или ещё чем-то. Главное, чтобы браузер мог получить этот файл, понять, что это за файл и что с ним делать.

Именно поэтому в последней версии ***HTML*** рекомендуется все подобные внешние файлы, в том числе и изображения, включать при помощи универсального элемента <***OBJECT***>. Этот элемент уже давно используется, например, для вставки ***flash***-роликов, но использование его вместо элемента <***IMG***> пока что остается экзотикой.

4.2.5. Формы



Как вы могли заметить, веб-дизайн унаследовал очень много от дизайна графического и полиграфического. Но элементы и технологии, делающие его собственно веб-дизайном, не имеют (или почти не имеют) прямых аналогов в традиционном дизайне. В первую очередь таким элементом, конечно, является механизм гиперссылок. Но кроме того имеется ещё один замечательный механизм, который

называется Формы обратной связи или просто Формы. Всё, что мы обсуждали ранее, касалось информации, приходящей от сервера к клиенту.

Клиент только запрашивает определенный документ, а дальше как в телевизоре: сидим — смотрим. Или читаем. Формы же позволяют организовать с сервером обратную связь. Т.е. посылать некую информацию в обратном направлении — от клиента к серверу. Это выводит нас более высокий уровень интерактивности. Рассмотрим на примерах. Самый простой и знакомый всем вам — это поисковые системы. Когда вы отправляете поисковый запрос, то делаете это как раз с помощью формы. Сервер получает вашу фразу, осуществляет поиск и возвращает вам страницу с результатами. Это уже самое настоящее интерактивное "общение" с сервером, т.к. страницы с результатами поиска как таковой (в виде ***HTML***-файла) не существует, а сервер создает её для вас в соответствии с тем, какой запрос был послан с помощью формы.

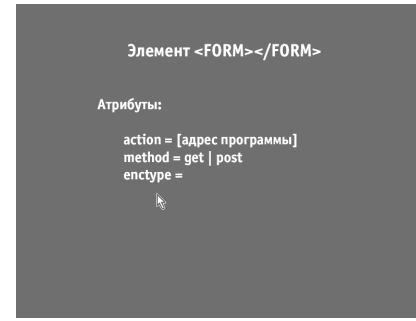
Более сложный пример — это форумы. Вы можете что-то написать, и это "что-то" появится на сайте. Т.е. с помощью форм мы можем даже удаленно изменять содержимое сайта. И форумы — это не единственный пример, хотя и достаточно характерный. Возможны и более сложные вещи, например, загрузка на сервер файлов прямо со страницы сайта. Например, сейчас очень популярны сервисы по размещению персональных фотографий. Таким вот образом мы можем разместить на сайте нашу фотографию.

Как же работает механизм форм? На одном конце мы имеем клиента — веб-браузер. В нём, собственно, и отображаются формы, которые мы заполняем. На другом конце — сервер, на котором работает некая специальная программа, получающая информацию, присланную через форму. На основе этой информации программа выполняет какие-то действия. Например, формирует ***HTML***-страницу и заставляет веб-сервер прислать её обратно клиенту. Или, как в случае с форумом, добавляет на сайт наше сообщение.

Или отправляет наше сообщение по ***e-mail***'у и т.д. Серверная часть с программой нас сейчас интересовать не будет, нам важно научиться встраивать формы в страницы нашего сайта. Собственно форма определяется элементом <***FORM***>. Элемент должен иметь как открывающий, так и закрывающий теги. Обязательным атрибутом элемента ***FORM*** является атрибут ***action***. Этот атрибут задаёт ***URL***-адрес, куда должны отправиться данные, введенные пользователем в форму. То есть, грубо говоря, расположение той самой программы на сервере, которая будет эти данные обрабатывать. Откуда же берутся эти программы?

Во-первых, провайдер, у которого будет размещён ваш сайт, как правило, предоставляет доступ к нескольким стандартным программам. Например, для отправки сообщения по ***e-mail***'у. Соответственно, достаточно вписать адрес

этой стандартной программы в атрибут ***action***. Во-вторых, если провайдер позволяет, можно установить на сервере программу самостоятельно. В Интернете можно найти огромное количество готовых программ для самых разных задач. Наконец, в-третьих, вы можете освоить веб-программирование и писать программы самостоятельно.



Следующий атрибут элемента ***FORM*** — атрибут ***method***. Этот атрибут определяет способ, каким данные будут отсылаться на сервер и может иметь одно из двух значений: ***get*** или ***post***. Объяснение разницы между этими двумя методами выходит за рамки данного урока. Какое значение устанавливать — зависит от того, как написана программа. Соответственно, ответ нужно искать в документации к программе. Очень грубо можно сказать, что ***get*** больше приспособлен для передачи сравнительно небольших объёмов текстовой информации. Если же форма предполагает отправку более сложных данных, например, файлов, то это, скорее всего, ***post***. Последний атрибут — ***enctype***. Он устанавливается как раз при передаче через форму разных хитрых данных. О нём мы поговорим ниже. Элемент ***FORM*** как таковой в браузере не виден. В ***Dreamweaver***'е форма обведёна пунктриной рамочкой, чтобы не потерять её. Элемент этот служит своего рода контейнером для других элементов, с которыми как раз и оперирует пользователь. Элементы эти соответствуют элементам интерфейса операционной системы. Это поля ввода текста, выпадающие списки, переключатели, кнопки и т.д. Кроме того внутри элемента ***FORM*** могут располагаться и другие элементы (блочные и текстовые). Мы можем заключить внутрь формы хоть всю нашу страницу.

Одним из основных органов управления, составляющих форму, является элемент <***INPUT***>. Этот элемент не имеет закрывающего тега. В зависимости от значения атрибута ***type*** этот элемент может отображаться в виде самых разных органов управления. Несколько особняком стоят элементы, которые дают команду браузеру начать отправку данных на сервер. Прежде всего это ***INPUT*** со значением атрибута ***type*** равным ***submit***. Отображается в виде кнопки. Может иметь атрибут ***value***, определяющий надпись на кнопке.

INPUT типа ***image*** создаёт графическую кнопку. То есть вместо стандартной кнопки мы можем подставить нарисованную нами красивую кнопочку. В этом случае элементу ***INPUT*** необходимо указать те же атрибуты, что присутствуют у элемента ***IMG***, то есть ***src*** (с указанием пути к файлу

картинки), ширину, высоту и даже ***alt***-текст. Клик по этой картинке будет приводить к отправке данных на сервер.

INPUT типа ***text*** создаёт поле для ввода текста. Как раз такое, как в поисковых системах. Чтобы программа могла понять, к какому из элементов формы относится та или иная порция данных, каждый элемент должен иметь атрибут ***name*** с уникальным в пределах формы значением. Также элемент может иметь атрибут ***value***. Если мы зададим его, то текстовое поле отобразиться не пустым, а с текстом, указанным в значении этого атрибута. Атрибут ***size*** задаёт ширину поля. Однако единицы, в которых она задается, разнятся от браузера к браузеру и для точного указания ширины следует пользоваться таблицами стилей.

Атрибут ***maxlength*** позволяет ограничить количество символов, которое пользователь может ввести в данное поле. Например, если имеется поле для ввода года, то, очевидно, имеет смысл ограничить его длину четырьмя символами. В этом случае пользователь не сможет ввести в это поле более четырёх символов.

INPUT типа ***password*** полностью аналогичен текстовому полю, за исключением того, что вводимые пользователем символы не отображаются на экране, а заменяются, например, звёздочками или точечками. Это нужно для того, чтобы при наборе паролей злоумышленник не мог подсмотреть через плечо, какой пароль вы набираете.

INPUT типа ***checkbox*** создает нечто наподобие выключателя: чекбокс работает по принципу вкл./выкл. Отображается обычно в виде квадратика: пустого или с "галочкой", когда включён. Как и любой ***INPUT***, чекбокс должен иметь атрибут ***name***. А вот атрибут ***value*** работает несколько необычно. Можно было бы предположить, что определенные значения этого атрибута соответствуют его включенному или выключенному состоянию, однако это не так. Если чекбокс не включен, то он вообще ни в каком виде не передаётся программе. Для программы его как бы и не существует. А вот если включен, то передается переменная с именем из атрибута ***name*** и значением из атрибута ***value***, т.е. как если бы у нас было текстовое поле, в которое мы ввели значение атрибута ***value***. Отличие состоит в том, что пользователь это значение изменить не может. По умолчанию чекбокс появляется на странице в выключенном состоянии. Чтобы сделать его включенным по умолчанию, следует указать атрибут ***checked***. Атрибуту этому значение не присваивается.

INPUT типа ***radio*** очень похож на ***checkbox***, но состоит из нескольких переключателей (так называемых радиобАттонов) и напоминает кнопки на магнитофоне: одну нажали, другая выскочила. Т.е. фактически это группа переключателей: несколько элементов ***INPUT*** типа ***radio***, имеющих одинаковое значение атрибута ***name*** (собственно, по этому признаку они и

группируются) и разные значения ***value***. В результате переменная с именем из атрибута ***name*** будет иметь значение из атрибута ***value*** того переключателя, который был включён пользователем. Как и в случае с чекбоксом мы можем указать атрибут ***checked***, чтобы "включить" по умолчанию один из переключателей. Имеется ещё один хитрый ***INPUT*** типа ***hidden***, который никак не отображается в браузере и не доступен пользователю. Он позволяет с помощью атрибутов ***name*** и ***value*** передать программе некие служебные данные, в которые не должен вмешиваться пользователь.

INPUT типа ***reset*** создаёт кнопку, при нажатии на которую форма "очищается", т.е. все данные, введённые пользователем сбрасываются, возвращаются в исходное состояние. Атрибут ***value***, как вы могли догадаться, задает надпись на кнопке. ***INPUT*** типа ***button*** создаёт, так сказать, "просто кнопку". Она ничего не делает, чтобы её оживить, нужно написать так называемый "скрипт" — маленькую программку, встроенную прямо в ***HTML***-документ, которая выполняется браузером. Но это мы сейчас не будем обсуждать.

INPUT типа ***file*** создаёт поле с кнопкой, позволяющее выбрать какой-либо файл для отправки его на сервер. Например, можно вообразить программу, которая анализирует каким-то образом картинку и осуществляет поиск на основе этих данных. Естественно, элемент должен иметь атрибут ***name***. Если в форме используется данный элемент, то форме (самому элементу ***FORM***) необходимо задать атрибут ***enctype*** (помните, мы упоминали его выше). При этом значение его должно быть "***multipart/form-data***".

Теперь переходим к органам управления, создаваемым с помощью других элементов, отличных от ***INPUT***. Во-первых, это элемент <***TEXTAREA***>. По сути это тот же самый ***INPUT*** типа ***text***, т.е. поле для ввода текста, но, состоящее не из одной строки, а из нескольких. Количество строк задается атрибутом ***rows***, а количество колонок, то есть ширина — атрибутом ***cols***. Как и в случае с атрибутом ***size*** текстового поля ширину лучше задавать таблицами стилей. Синтаксис ***TEXTAREA*** несколько иной. Он, естественно, имеет атрибут ***name***, но у него нет атрибута ***value***, зато есть обязательный закрывающий тэг. А роль ***value*** выполняет текст, расположенный между открывающим и закрывающим тегами.

И, наконец, выпадающие меню. Вообще говоря, они не обязательно должны быть выпадающими, в чём мы вскоре и убедимся. Смысл меню в том, что пользователь может выбрать из списка один или несколько пунктов. Создается меню с помощью элемента <***SELECT***>. Элемент должен обязательно иметь закрывающий тег. Элемент, как и другие, имеет обязательный атрибут ***name***. Может иметь атрибут ***size***. Как ни странно, атрибут ведёт себя не так, как в текстовых полях. Он определяет не ширину меню, как вы могли подумать, а

высоту. Если ***size*** не задан, меню будет выпадающим, а если задан, то отобразится список высотой как раз в кол-во строк, заданное атрибутом ***size***. Если количество пунктов больше высоты, будет осуществляться прокрутка.

Ещё один атрибут — ***multiple***. Значение ему не присваивается. Если он указан, пользователь может выбрать одновременно несколько пунктов из списка. При этом нужно удерживать клавишу ***Ctrl***. В противном случае, если ***multiple*** не указан, из списка можно выбрать только один пункт. Подобно элементу ***FORM***, ***SELECT*** является лишь контейнером для элементов, определяющих сами пункты меню. Поэтому закрывающий тэг у него обязателен.

Ну а для определения пунктов меню предназначены элементы <***OPTION***>. Имеет атрибут ***value***. Т.е. после отправки формы в программу попадёт переменная с именем, заданным атрибутом ***name*** элемента ***SELECT*** и значением из атрибута ***value*** элемента ***OPTION*** того пункта, который мы выбрали. Ну а пользователь в меню увидит не значение атрибута ***value***, а текст, заключенный между открывающим и закрывающим тегами элемента ***OPTION***. Это важно запомнить: пользователю мы показываем одно, а в программу передается совсем другое.

Чтобы "включить" определенный пункт меню по умолчанию, соответствующему элементу ***OPTION*** следует задать атрибут без значения ***selected***. Было бы, наверное, логичнее назвать этот атрибут ***checked***, как в чекбоксе и радиобАттоне, но — се ля ви — ***HTML*** полон таких вот несуразностей. Как видите, функционально меню чем-то напоминает радиобАттоны: несколько переключателей (***OPTION***) каждый со своим значением ***value*** объединены в группу (***SELECT***) с одним именем (***name***).

4.2.6. Фреймы

Как сказал когда-то Якоб Нильсен, большой специалист в области веб-юзабилити, т.е. удобства пользования сайтами, — "скажите фреймам НЕТ!". Такое вот не слишком оптимистичное начало. Но действительно, при всей своей привлекательности, особенно для начинающего веб-дизайнера, механизм так называемых фреймов таит в себе массу проблем и противоречий. Значит ли это, что мы никогда не должны использовать этот механизм? Нет, но мы должны чётко понимать эти проблемы и учитывать их при разработке сайта на фреймах.

Итак, что же это за такие страшные звери — фреймы? Это несколько окон браузера, определенным образом скомпонованных в одном "родительском" окне. Каждый из фреймов — это, можно сказать, отдельный браузер в том смысле, что в каждый из них загружается отдельный ***HTML***-документ. Но смысл фреймов состоит в том, что между ними имеется связь: из одного

фрейма мы можем ссылаться на другой. Наиболее распространённый пример: в одном фрейме у нас расположено меню со ссылками, при клике на которые целевой документ загружается в соседний фрейм. Зачем это нужно? Во-первых, это даёт некоторую экономию пересылаемых данных. При переходе по ссылке мы перегружаем только фрейм с содержимым, а меню загружено один раз и уже не перегружается. Во-вторых, прокрутка осуществляется в каждом фрейме индивидуально. Это, пожалуй одно из наиболее ценных свойств фреймов. Независимо от прокрутки содержимого меню всегда остается у нас перед глазами.

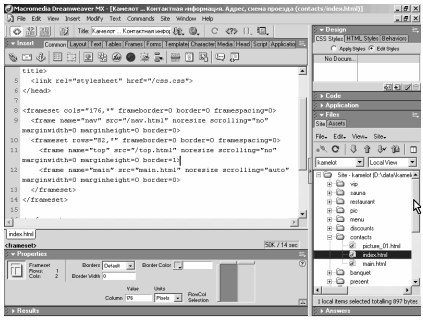
Чем же тогда не угодили фреймы Якобу Нильсену и прочим сознательным веб-дизайнерам? Если вы помните, по задумке Тима Бернерса-Ли неким "атомом" веба является страница. Понятие страницы включает в себя и единицу просмотра (пользователь видит её в окне браузера), и единицу навигации (кликнув по ссылке, мы переходим на другую страницу), и единицу, образно говоря, географическую (***URL***-адрес ведёт на страницу), и, наконец, единицу физическую (страница лежит на сервере в виде файла). Последние 2 пункта, прада, уже несколько размыты из-за динамических страниц, которые нигде не лежат, а создаются налету программой. Фреймы же вступают в противоречие с этой концепцией. К чему это приводит на практике?

Во-первых, единица навигации не соответствует единице просмотра. Т.е. то, куда ведёт ссылка не соответствует тому, что мы видим в браузере. Пока мы ходим по сайту, это не заметно. Но представьте, что мы захотим эту ссылку сохранить или отправить другу по почте. Давайте попробуем её скопировать и представим, что друг пошёл по этой ссылке. Что же он увидит? Он увидит страницу с содержанием. Что слева у неё должно быть меню, об этом браузеру узнать неоткуда. То же самое касается поисковых систем. Найдя нужную страницу, поисковая система никак не сможет догадаться, что вместе с ней в браузере нужно показать ещё пару других страниц.

Во-вторых, основной механизм адресации — ***URL*** — перестаёт работать вообще. На какую бы страницу мы не перешли, в адресной строке браузера мы продолжаем видеть одно и то же. В-третьих, навигация становится непредсказуемой. Мы кликаем в одном месте окна, а страница загружается в другом. В-четвёртых, не понятно, как печатать фреймы. Поэтому браузеры вынуждены предлагать различные опции: печатать всё в том виде, как мы видим на экране, печатать только выбранный фрейм, печатать содержимое каждого фрейма отдельно.

Можно ли как-то обойти эти проблемы при использовании фреймов? Можно. Как — обсудим ниже. Нужно ли оно нам? Это решать вам в каждом конкретном случае. Разрабатывать сайт с фреймами сложнее. Если эта сложность оправдана, то почему бы и нет? Как же создать фреймосодержащий

документ? Очевидно, нам нужен набор обычных ***HTML***-страниц, которые будут отображаться во фрейMAX, а также нам нужна некая страница-каркас, описывающая расположение фреймов в окне браузера.



Эта страница имеет несколько иной синтаксис. Вместо элемента <***BODY***> у неё идёт элемент <***FRAMESET***> (по-русски — набор фреймов). Этот элемент имеет атрибуты ***cols*** или ***rows***. ***Rows*** описывает горизонтальные фреймы, ***Cols*** — вертикальные. Если использовать ***rows*** и ***cols*** одновременно, получится сетка. Аналогов табличных ***colspan*** и ***rowspan*** здесь нет, для более сложных макетов нужно использовать вложенные фреймсеты, но об этом — ниже. Размер фреймов (ширина для ***cols*** или высота для ***rows***) может задаваться в пикселах, в процентах от размера окна или в долях так называемого "оставшегося" места. С первыми двумя пунктами, очевидно, всё ясно, тут всё аналогично таблицам.

А вот третий требует пояснения. Допустим, нам нужна колонка слева фиксированной ширины, скажем, 100 пикселей, а остальное пространство нужно разбить на 2 колонки одинаковой ширины. Как это сделать? Задать им по 50% будет некорректно, т.к. проценты — от ширины окна, а в нашем распоряжении уже не всё окно, а оставшаяся от колонки часть. Как покажет браузер эти якобы 50-процентные колонки — большой вопрос. Поэтому правильнее двум оставшимся фреймам поставить звёздочки, что будет означать, что оставшееся пространство нужно разделить пополам. К звёздочкам можно добавлять коэффициенты. Например, если написать вот так, то оставшееся пространство будет разделено в пропорции 1:2.

Как уже говорилось, фреймсеты могут вкладываться друг в друга. Например, мы можем один из фреймов заполнить вложенным фреймсетом. Таким образом мы можем создавать сколь угодно сложные макеты. Как же теперь указать, какие ***HTML***-файлы загружать в каждый из фреймов? Для этого в элемент ***FRAMESET*** вкладываются элементы <***FRAME***>, ответственные за каждый отдельный фрейм. Количество этих элементов, естественно, должно соответствовать количеству фреймов, созданных фреймсетом. Содержимое каждого фрейма задается с помощью атрибута ***src***. Его значение, как вы можете догадаться, — это ***URL*** загружаемой в данный фрейм страницы.

Атрибут без значения ***noresize*** не позволяет пользователю изменить размер фрейма (в противном случае, по умолчанию, пользователь может спокойно

перетаскивать мышью границу фрейма, изменяя таким образом весь макет). Атрибут ***frameborder*** определяет, будет ли эта самая граница фрейма видна или только "подразумеваться". Может принимать значения 1 — значение по умолчанию, когда граница видна, или 0, если границу нужно спрятать. Атрибут ***scrolling*** определяет, видна ли линейка прокрутки в данном фрейме. Может принимать значения ***auto***(по-умолчанию) — прокрутки нет, но она появляется, если содержимое страницы не помещается во фрейм. Может принимать значение ***yes*** — в этом случае линейка прокрутки будет показываться в любом случае. И, наконец, значение ***no***, когда прокрутка не появляется ни при каких обстоятельствах (даже если информация не помещается во фрейм). Атрибуты ***marginwidth*** и ***marginheight*** аналогичны таковым у элемента ***BODY***, то есть устанавливают горизонтальные и вертикальные отступы от края фрейма до его содержимого. Обычно значения этих атрибутов обнуляют, а с отступами разбираются уже на самой странице. И, наконец, один из самых главных атрибутов — ***name***. Он задаёт имя фрейма, что, собственно, и позволяет нам ссылаться из одного фрейма на другой.

Итак, переходим к самому интересному. Возможно вы помните, когда мы изучали гиперссылки, был у элемента ***A*** такой атрибут ***target***. Тогда мы рассмотрели только одно его значение — "***_blank***", которое позволяло открывать ссылку в новом окне. Так вот, в этом атрибуте как раз и следует указывать имя фрейма, в котором должен открыться документ, на который ведёт ссылка, то есть имя, заданное в атрибуте ***name*** целевого фрейма. Если в ссылке атрибут ***target*** не задан, то документ откроется в текущем фрейме (в котором находится ссылка). Если уж мы вспомнили про "***_blank***", то здесь же рассмотрим и другие "специальные" значения этого атрибута. Итак, кроме "***_blank***" это может быть "***_self***" — значение по умолчанию, когда ссылка открывается в текущем фрейме. Это может быть "***_parent***" — ссылка открывается в "родительском" окне. Что это значит? Фреймы — это как бы отдельные браузеры, вложенные в наш основной браузер. Если мы имеем дело с вложенными фреймсетами, то получается уже 2 и более уровня вложенности.

Так вот, мы можем обращаться к фрейму, в который вложен текущий, не по имени, а с помощью значения "***_parent***". Особенно это актуально для "главного" окна, которое не имеет имени. Таким образом мы можем выйти из фреймовой структуры и попасть на обычную страницу. Однако для обращения к окну самого верхнего уровня лучше использовать значение "***_top***". В отличие от ***_parent***, оно не относительно текущего фрейма, а абсолютно и из любого уровня вложенности ведёт на самый верхний уровень — на главное окно браузера. И тут мы подходим к ответу на вопрос, как корректно разрабатывать сайт на фрейMAX. Итак, если вы взвесили все "за" и "против" и решили, что без фреймов вам никак не обойтись, то нужно соблюдать, по крайней мере,

одно правило: механизм ***URL*** должен работать. То есть при переходе по ссылке должен меняться адрес в адресной строке браузера.

Что для этого нужно? Для этого каждый документ, на который может вести ссылка, должен иметь свой отдельный фреймсет, а все ссылки должны вести не на документ, а на этот его фреймсет и загружать этот фреймсет в главное окно браузера, т.е. все ссылки должны иметь атрибут ***target***="***_top***". Вот как это выглядит: каждая страница имеет "содержательный" ***HTML***-файл, а также имеет документ с фреймсетом. И именно этот документ показывается по умолчанию. Содержание остальных фреймов (например, фрейма с навигационным меню) одинаково для всех страниц, поэтому навигационный ***HTML*** лежит отдельно и загружается во все фреймсеты. "Свой фреймсет для каждого документа?!" — ужаснётесь вы. А кто сказал, что будет легко? Зато так каждая страница будет иметь полноценный адрес — ведущий не на отдельную страничку, а на полный комплект фреймов.

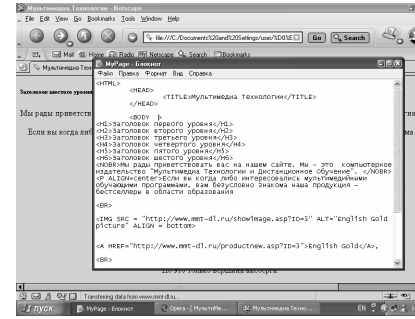
Но остаётся проблема поисковых систем — в поиск попадает только "содержательная страница". Но и эта проблема решаема, правда, её решение уже не так просто и требует некоторого программирования, на чём мы здесь не будем останавливаться. Теоретически же если мы наберем адрес "содержательной" страницы, то будем автоматически перенаправлены на страницу с соответствующим фреймсетом. И напоследок рассмотрим ещё один вид фреймов, достаточно радикально отличающихся от "обычных" фреймов. Это так называемые ***IFRAMES*** или ***inline-frames***.

Как следует из названия, такой фрейм встраивается прямо в содержание страницы, подобно текстовым элементам. Никаких фреймсетов для него не надо, он создаётся с помощью элемента <***IFRAME***>. Как и обычный фрейм, он имеет атрибут ***src***, определяющий его содержимое, атрибут ***name***, для использования в ***target***'ах ссылок. Также от обычных фреймов он унаследовал атрибуты ***frameborder***, ***scrolling***, ***marginwidth*** и ***marginheight***. Этот элемент чем-то похож на картинку: он так же имеет атрибуты ***width*** и ***height***, задающие его размеры, а также атрибут ***align***, управляющий выравниванием текста вокруг фрейма. Элемент должен иметь закрывающий тег, а то, что находится между открывающим и закрывающим тегами, будет показано в браузерах, не поддерживающих ***IFRAME***.

4.2.7. Изменение цветов фона. Горизонтальная линия

На прошлом уроке мы с Вами изучили работу с картинками. На этом уроке мы попробуем задать цвета. Надо сказать, что многие авторы ***HTML*** любят использовать заранее предопределенные цвета фона документа, обычного текста и ссылок. Для того, чтобы задать эти цвета, мы включаем в тэг ***Body***

соответствующие параметры. С тэгом ***Body*** мы с Вами работали на прошлом уроке при включении фоновой картинки. Для того чтобы ввести цвет фона, мы пишем ***Background color***, но только в сокращенном виде, то есть, пишем ***B COLOR***.



Равняется. Здесь мы задаем цвет в шестнадцатиричном формате по схеме ***RGB***. Что значит Цвет по схеме ***RGB***? Дело в том, что любой цвет состоит из трех цветов: красного, зеленого и голубого. Если цвет не содержит ни один из этих цветов, то он представляет собой черный цвет. Мы сейчас попытаемся изобразить черный цвет. Сначала ставим знак Диеза, чтобы указать, что мы начинаем изображение цвета. Первая составляющая ***R*** у нас равна двум нулям (то есть, этот цвет в общей сложности — нулевой). Далее. Зеленый цвет тоже равен нулю. Синий — также. Дело в том, что каждый из цветов ***RGB*** кодируется одним байтом. Это означает, что максимальное значение этого кода может составлять 255 (от 0 до 255). Попробуем понять, что это значит на следующем примере. Итак, цвет фона мы задали черный.

Зададим точно таким же образом цвет текста... Сначала вводим знак Диеза. Мы хотим задать для цвета текста белый цвет. Белый цвет является максимальным в палитре. То есть, для белого цвета и ***R*** — максимально, и ***G***, и ***B***. Все составляющие (красный, зеленый и синий) — максимальные. Максимальный для шестнадцатиричных цветов — это ***FF***, то есть, фактически, 255. Пишем 255, 255 и 255. Итак, мы с Вами получили белый цвет. Теперь — ссылки. Ссылки у нас должны быть в ***Link***-е... Сейчас посмотрим... Диез. Составляющая красного цвета — 96, зеленого цвета — 90 и синего — CC. Посмотрим, что теперь у нас получилось. Сохраним этот файл и обновим его вот здесь. Что мы теперь имеем? Сейчас у нас текст изображен белым цветом, то есть, именно так, как мы с Вами планировали. Фон у нас черный, как мы и задавали. А ссылки имеют вот такой непонятный цвет.

Посмотрим, что мы можем сейчас сделать. Мы можем, к примеру, обнулить составляющие красного и зеленого цветов. Нет, все-таки, лучше обнулим составляющие красного и синего цветов. А составляющие зеленого цвета сделаем чем-то средним. Было бы логично ожидать, что после того, как мы перезагрузим страницу в нашем браузере, эти ссылки станут зеленого цвета. К сожалению, сейчас они не станут зелеными, потому что в данный момент они

являются посещенными ссылками. Мы с Вами уже посетили их. Поэтому они у нас стали вот такими темно-синими.

Давайте сейчас попробуем поменять какую-либо из ссылок. Например, поставим что-то в этом роде вот сюда. А здесь, к примеру, поставим 4 или 5. Здесь закроем этот тэг. Фактически, мы с Вами сейчас поставили новую ссылку. Сохраним этот файл и обновим его. В этом случае, как видите, не посещенная ссылка отображается зеленым цветом, то есть, именно тем цветом, который мы задали вот здесь. Здесь установлено 88 — средне-зеленый цвет. Прекрасно. Мы получили то, что и хотели получить.

А теперь попытаемся изобразить горизонтальную линию. Текст можно разделить горизонтальной линией, используя тэг, называемый ***Horizontal*** (***HR***). Тэг ***Horizontal*** имеет следующие параметры. Один из параметров — ***Size***. Это величина этой самой горизонтальной линии. Здесь мы ошиблись... ***Size***. Равняется. Величина линии означает ее толщину в точках.

Давайте, попробуем задать здесь, к примеру, 5 (вот такую толщину линии). Затем зададим ее ширину. Ширина линии может быть задана в точках или в процентах (от ширины окна браузера). Зададим 50% от ширины окна браузера (можно задать и в точках). И, соответственно, у нас сейчас идет выравнивание, то есть, как конкретно расположить эту самую линию относительно экрана (***ALIGN***).

ALIGN у нас может быть равен, например, ***Left***-у. Это означает расположение слева. Посмотрим, что при этом получится. Закроем этот тэг. Сохраним и нажмем на кнопку Обновить. Теперь Вы видите на экране выровненную с левой стороны вот такую линию. Эту линию мы с Вами сумели отобразить здесь с помощью этого тэга, который называется ***HR***. Мы теперь можем увеличить ширину этой линии. Например, сделаем ее вот такой, равной 10 пикселям. Сохраним и обновим изображение в браузере. Как видите, ширина линии соответственно увеличилась. Если мы зададим здесь 100%, то линия, фактически, займет собой весь экран. Проверим это. Обновили — и линия, соответственно, увеличилась.

На этом мы завершим наш урок. Мы с Вами научились менять цвета в нашем документе и изображать горизонтальные линии. На следующем уроке мы начнем изучать стили нашего ***HTML***-документа.

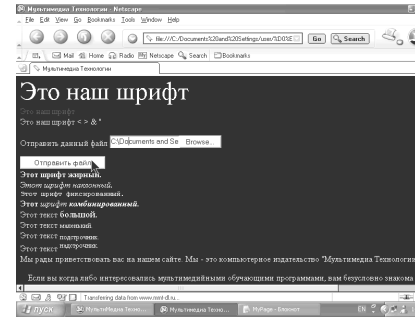
4.2.8. Стили шрифтов в HTML

На этом уроке мы с Вами научимся добавлять стили в ***HTML***-документ. Как Вы знаете, ***HTML***-документ позволяет использовать различные стили шрифтов для выделения текстовой информации в документах. Список стилей, которые поддерживает большинство браузеров, следующий. Здесь имеется: ***Bold*** (Жирный), ***Italic*** (Наклонный) и ***MonoSpaced*** (иногда его называют ***Type***

Writer). Это — с использованием фиксированных шрифтов. Стили, соответственно, можно комбинировать. Можно использовать жирный и наклонный шрифт одновременно.

Попытаемся сделать следующий эксперимент. Введем сюда, к примеру, следующий текст. Этот шрифт — жирный. Скопируем это предложение. Здесь Этот шрифт будет наклонный, здесь — фиксированный, а здесь — комбинированный. Посмотрим, что у нас получилось. Для того, чтобы изобразить этот шрифт с жирным начертанием, мы используем тэг, который называется ***Bold*** и обозначается строчной буквой ***b***. Соответственно, здесь используется закрывающий тэг ***b***. Сохраним этот текст и посмотрим, что при этом получается. Обновляем. Как видите, Этот шрифт у нас жирный. Здесь используется жирное начертание.

Вы можете видеть, что этот текст отличается вот от этого наклонного текста. Давайте, здесь поставим еще тэг ***BR***, чтобы каждое из этих предложений изображалось на новой строке. Вот так. Сюда мы поставим тэг ***Italic*** (наклонный). Соответственно, ***Italic***-ом мы и закроем (вот так). Сохраняем. Переходим сюда. Обновляем. Как видите, шрифт сейчас у нас жирный и наклонный. А теперь посмотрим, что означает фиксированный шрифт и какими тэгами он обозначается. Фиксированный шрифт обозначается тэгами TT. Сохраняем. Обновляем и смотрим. Вы видите, что теперь шрифт изменил свое начертание.



Это — фиксированный шрифт. Нам осталось рассмотреть комбинированный шрифт. Кстати, вот здесь нам был необходим закрывающий тэг. Вот так. Посмотрим. Теперь Вы видите, чем отличается фиксированный шрифт от комбинированного шрифта. Он отличается тем, что содержит буквы одинаковой ширины. Иначе говоря, и буква Т, и буква О, и буква Э имеют одинаковую ширину. Вернемся сюда и поработаем с комбинированным шрифтом. Дело в том, что тэги можно использовать во вложенном виде. Рассмотрим пример. Пусть это слово имеет жирное начертание. Вставим также и закрывающий тэг ***Bold***. Далее. Здесь у нас будет ***Italic***. А здесь у нас ***Italic*** заканчивается. Но слово Комбинированный, в свою очередь, также будет ***Bold***. Вставим и закрывающий тэг ***Bold***. Это означает, что слово Комбинированный в итоге окажется ***Bold*** и ***Italic***. Убедимся в этом. Обновим. Мы убеждаемся в том, что

так оно и есть. Слово Комбинированный оказалось изображенным жирным и наклонным.

Это произошло по той причине, что открывающий и закрывающий тэги ***Bold*** расположены внутри тэга ***Italic***. Кроме того, имеются еще и дополнительные стили. Что они представляют собой? Имеются следующие дополнительные стили: Большой, Маленький, Подстрочник и Надстрочник. Давайте, изобразим это здесь. Не забудем поставить ***BR*** и напишем: Этот текст большой... Затем напишем: Этот текст маленький...Здесь напишем: Подстрочник... А здесь — Надстрочник. Прекрасно. Соответственно, сюда мы ставим тэг ***BIG***... А здесь мы его закрываем. Сохраним и посмотрим, что у нас получилось. Сохранить. Обновить. Вы видите, что слово Большой заметно больше, чем стоящее рядом выражение Этот текст. А теперь поставим здесь ***Small***, а здесь, соответственно, — тоже ***Small***.

Вы теперь видите, что слово Маленький оказалось меньше, чем Этот текст. Подстрочник... Этот ***SUB***... Переходим сюда. Обновляем. Вы убеждаетесь в том, что слово Подстрочник располагается внизу. Именно таким образом очень часто изображают индексы. Аналогичным образом используется Надстрочник. С помощью этого тэга изображаются степени (вот так). Вы видите, что слово Надстрочник расположилось выше, чем Этот текст. На этом завершим текущий урок, который был посвящен использованию стилей шрифта в ***HTML***. На следующем уроке мы с Вами будем экспериментировать с размером шрифта и его цветом.

4.2.9. Меню выбора в HTML

На этом уроке мы с Вами продолжим изучение форм и начнем изучение меню выбора в форMAX. Под меню выбора в форMAX понимают, например, такой элемент интерфейса, как ***LIST BOX***. Существуют три типа тэгов меню выбора для форм. Один из таких тэгов называется ***SELECT***. В этом случае пользователь выбирает одно значение с фиксированного списка значений.

Список значений представляется тэгами ***OPTIONS***. Попробуем изобразить вот какой список. Сделаем это следующим образом. Откроем тэг и напишем здесь ***SELECT***. Затем данному элементу даем имя. Заметим, что элементу всегда нужно присваивать имя. Каждый элемент формы имеет свое имя. Итак, мы задаем, например, имя ***MyList***. Сейчас мы с Вами допустили здесь незначительную ошибку: мы начали писать это внутри тэгов ***INPUT***. Предварительно занесем это в буфер обмена, а затем удалим. Сделаем то, что мы делали ранее, вот здесь.

После этого закроем данный тэг и с помощью тэга ***OPTION*** заполним список этого ***LIST BOX***. Первый элемент, который мы поместим сюда, так и назовем Первый Элемент. Закроем ***OPTION***. Теперь вставим сюда. Второй

элемент. Вообще-то, мы можем сейчас занести ***OPTION*** в буфер обмена, чтобы каждый раз не вводить его заново. Итак — ***OPTION***. Затем — третий элемент. Три элемента нам достаточны. А теперь закроем тэг ***SELECT***. Сохраним все это и Обновим нашу страницу в браузере.

Как видите, у нас получился элемент, который содержит список из трех элементов. Мы сейчас можем выбрать любой из этих элементов. Вот так симпатично у нас все получилось. Прекрасно. А теперь вернемся на ***MyPage*** и попробуем поработать со вторым типом тэгов в меню. Второй тип тэга называется ***SELECT SINGLE***.

Он представляет собой такой же тэг, что и ***SELECT***, но в данном случае пользователь видит на экране одновременно уже несколько элементов выбора. Обычно по умолчанию количество этих элементов равно трем. Мы сейчас поступим следующим образом. Для начала вставим эти элементы после каждого элемента формы. Иначе говоря, мы хотим, чтобы данные элементы ***GUI*** располагались в столбик (друг под другом) для наглядности. Это мы расположим вот сюда. Этого будет достаточно. А теперь здесь мы создадим элемент типа ***SELECT SINGLE*** и посмотрим, в чем заключается разница. Для того чтобы не вводить все заново, мы скопируем эту группу вот сюда.

А здесь поменяем этот тэг на ***SELECT SINGLE***...Здесь установим ***MyList***2, а эти элементы оставим без изменения. Если сейчас мы сохраним данный файл и Обновим его здесь, то получим нечто вроде этого. Для того чтобы эти два элемента отличались друг от друга, нам необходимо в ***SELECT SINGLE*** вставить еще атрибут ***Size***. Зададим ***Size***, равным четырем. Сохраним и обновим данную страницу. Вы можете видеть, что теперь этот элемент определенно отличается от верхнего элемента.В тот момент, когда мы ***Size*** не устанавливали, они, практически, не отличались между собой.

И, последнее, это, так называемый, ***SELECT MULTIPLAY***. От ***SELECT SINGLE*** он отличается только тем, что в этом случае пользователь может одновременно выбирать несколько элементов списка. Если, к примеру, мы поставим здесь атрибут ***MULTIPLAY*** и установим его, например, равным двум, то мы получим следующее. Сохраним и Обновим этот элемент. Теперь мы видим, что при нажатой клавише ***Ctrl*** пользователь может выбирать несколько элементов. На этом завершим наш урок. На следующем уроке мы с Вами научимся отправлять файлы.

4.2.10. Создание формы отправки файлов в HTML

На этом уроке мы с Вами попытаемся создать форму для отправки файлов. Мы, например, очень часто встречали ситуации, при которых можно загрузить какой-нибудь файл и отправить его, то есть, выполнить, так называемый, ***Attache*** (Вложение). Мы попробуем сделать следующее. Попытаемся удалить

все, что было сделано нами ранее, кроме тэга ***FORM***. Удалим все это, чтобы оно нам не мешало, и начнем свою работу. Для случая отправки файлов мы, например, желаем, чтобы наша форма содержала соответствующий текст.

Отправить данный файл... Итак, текст Отправить данный файл мы уже имеем. Далее. Что нам теперь нужно сделать? ***Method Post*** для данной формы мы оставляем. ***Action URL*** мы также оставляем. Продолжаем. Для начала мы создаем тэг ***INPUT*** для этого элемента ***GUI***. Затем ставим его имя (это обязательно). ***Name*** — это обязательный атрибут. Зададим в качестве имени, например, ***MyFile***. Закроем кавычку. Теперь мы выполняем ***Type***. В данном случае для ***Type*** имеется элемент, который называется ***File***. После того, как такой элемент был создан для посылки файла, мы с Вами должны еще создать кнопку для подтверждения отправки файла, так называемую, кнопку ***Submit***.

Это, опять-таки, делается с помощью тэга ***INPUT***. ***Name***. Вообще-то, для элемента кнопки ***Name*** можно и не задавать в том случае, если мы используем тип, который называется ***Submit***. Поэтому, давайте, мы с Вами его используем. ***Type***. В данном случае имя его и будет ***Submit***. Только в этом случае можно не ставить имя. ***Submit***. Допустим, мы желаем, чтобы на кнопке имелась надпись Отправить файл. Для осуществления этого в тэге ***INPUT*** существует атрибут, который называется ***Value*** (Значение).

Если для кнопки в качестве такого значения мы напишем Отправить файл, то именно эта надпись и появится на кнопке ***Submit***. Отлично. На этом мы завершаем создание своей формы. Правда, в данном случае здесь стоит поставить тэг ***PARAGRAPH***, чтобы ***INPUT Type*** получился на новой строке. Сохраним, Обновим и посмотрим, что же у нас получилось. Мы видим здесь кнопку Отправить файл. Элемент Отправить данный файл у нас находится вот здесь. И тут у нас имеется вот этот самый элемент для отправки файлов.

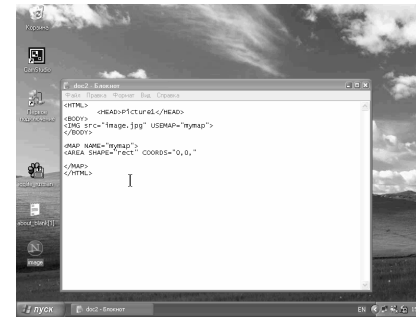
Если сейчас мы нажмем на кнопку ***Browse***, то перед нами появится окно открытия файла. Мы можем выбрать здесь любой файл. Выберем, например, тот файл, который мы с Вами скачали ранее, то есть, ***ICQ Russian***. Теперь, когда мы уже выбрали этот файл,мы видим здесь соответствующую ссылку на ***ICQ Russian***. Если бы мы правильно указали путь к файлу в атрибуте ***Action***, то при нажатии на кнопку Отправить файл, данный файл был бы скачан на наш сервер.

Но, к сожалению, ***URL*** мы не указали, и соответствующей программы у нас на сервере нет. Именно поэтому сейчас этого попросту не произойдет. Но, если бы мы это имели, то все это отлично сработало бы. Итак, при помощи незначительных действий, мы получили вот такие интересные элементы на нашей странице. На этом завершим текущий урок. На следующем уроке мы с Вами начнем изучение таблиц.

4.2.11. Карта изображения в HTML

Мы с Вами начинаем изучение создания карты изображений. Создание карты изображений является одной из самых привлекательных возможностей ***HTML***. Эта возможность позволяет пользователю привязывать ссылки на другие документы к отдельным частям изображений.

Вы щелкаете мышью на отдельных частях изображения — и можете переходить по той или иной ссылке на другие документы. Сначала мы с Вами создадим некоторое подобие картинки. Это, в общем-то, будет сама картинка, которую мы сейчас и разметим. Для этого в пункте Стандартные мы выберем ***Paint*** и создадим такую небольшую картинку. Вот так. Предположим, что у нас имеется такая небольшая картинка (вот она). В ней мы проведем вот такую линию. Верхнюю часть мы зальем красным цветом, а нижнюю часть — зеленым. На верхней части напишем: Переход на документ 3... Вот так. А в нижней части напишем: Переход на документ 4... Мы можем написать это, например, красным цветом... Вот так. А теперь сохраним этот рисунок. Давайте, сохраним его в формате ***JPEG*** и поместим на Рабочий стол. Назовем этот рисунок ***Image***.



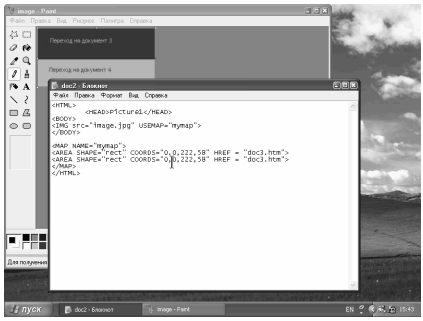
Что теперь мы можем сделать? У нас тут при сохранении несколько испортился внешний вид данного рисунка. Дело в том, что ***JPEG*** является сжатым форматом, а при сжатии изображения иногда теряется его качество. В данном случае нам это не помешает. Изображение у нас так и останется. А теперь, давайте, определим координаты. Если мы установим курсор вот сюда, то увидим здесь внизу, что весь прямоугольник имеет размер 223 пикселя на 110 пикселей. Эти данные нам пригодятся при создании карты этого рисунка. А сейчас мы с Вами создадим главный документ. Его мы будем создавать не с нуля, а создадим на основе документа 2, который мы создали на уроке при изучении фреймов. Главный документ мы создадим вот здесь. Для начала мы возьмем этот ***BASE TARGET*** (с фреймами мы сейчас не работаем) и вставим заголовок, назвав его, например, ***Picture***1.

А сейчас мы здесь поставим картинку. Как Вы помните, это можно сделать посредством тэга ***IMAGE***. Картинка находится прямо на нашем Рабочем столе, поэтому мы можем ее так и назвать, а именно, ***Image.jpg***. Назвали ее. Разумеется, здесь нужно указать ***SRC***, затем знак равенства. И это мы должны указать в кавычках (это понятно). После того, как мы указали картинку, мы

должны включить поддержку карты для данного изображения. Поддержка карты включается с помощью атрибута, который называется ***USEMAP***.

Допустим, что ***USEMAP*** использует ***MAPNAME***. В нашем случае ***MAPNAME*** — это ***MYMAP***. А сейчас мы уберем эти ссылки и создадим синтаксис определения карты изображения. Создадим его где-нибудь здесь. Синтаксис карты изображения создается с помощью тэга ***MAP***, которому дается имя ***NAME*** (это его атрибут). А ***NAME*** у нас равен ***MYMAP***-у (мы с Вами это уже определили — ***MYMAP***). Мы это определили вот здесь: ***MAPNAME — MYMAP***.

Закроем тэг ***MAP***. Сразу же поставим и завершающий тэг ***MAP***, чтобы потом не забыть об этом. После того, как мы определили эти два тэга, мы начинаем описание самой карты. Описание карты осуществляется с помощью тэга, который называется ***AREA***. Первый тэг ***AREA*** имеет атрибут, который называется ***SHAPE***. Атрибутом ***SHAPE*** мы указываем на форму определяемой области изображения. Она может быть по умолчанию (***Default***) прямоугольником. Но мы можем задать также и другую форму. Например, можно задать в качестве такой формы или ***RECT*** (прямоугольник), или ***CIRCLE*** (круг) или Поле, то есть, многоугольник произвольной формы. В данном случае мы задаем ***RECT***. Это — прямоугольник. Как Вы помните, наш рисунок имел две прямоугольные области: красную и зеленую.



Итак, ***SHAPE*** у нас равняется ***RECT***, причем ***RECT*** нужно заключить в кавычки (это мы уже определили). Следующий атрибут называется ***COORDS*** (это координаты). Это, фактически, размерность нашей области. Размерность области мы задаем в пикселях. Для прямоугольника нужно будет задать четыре координаты левого верхнего и правого нижнего углов. Давайте, мы с Вами зададим эти координаты. Итак, для левого верхнего угла задаем (0;0). А для правого нижнего угла... Кстати, мы с Вами, к сожалению, забыли параметры нашей картинки. Для того чтобы вспомнить их, откроем нашу картинку в приложении ***Paint***. Координаты мы можем определить здесь. Посмотрим вниз. Мы видим 222 58. Прекрасно. Здесь так и пишем: 222 и 58. Это — первая координата. После того, как мы с Вами задали координаты (***AREA COORDINATS***), здесь мы ставим соответствующую ссылку. Ссылка ставится с помощью тэга ***HREF*** (это мы с Вами знаем). Пишем: ***HREF***, равняется ***Doc***3.***htm***.

Первый элемент мы уже создали. Аналогичным образом создается второй элемент. Также пишется ***AREA*** с закрывающимся этим, но координаты мы уже должны задать по-другому. Координаты в нашем случае (откроем ***Paint***) уже будут определять вот этот угол, то есть, 0 58 до 223110. Так и задаем: 058 до 223110. А здесь, соответственно, задаем ***HREF***, равняется ***Doc***4. Посмотрим, что у нас получилось. Сохраним этот ***Doc***2, свернем наш Блокнот и попробуем его запустить. Вы видите, что у нас имеется вот такая картинка. Если Вы обратите внимание вот сюда на строку состояния, то увидите, что каждая область представляет собой ссылку на соответствующий документ.

Вы видите, что сейчас здесь курсор мыши превратился в ладонь, а внизу имеется ссылка на ***Doc***3. Здесь — ссылка на ***Doc***4. Убедимся в этом. Нажали сюда — попали на третий документ. Вернулись сюда — попали на четвертый документ. Вы убедились в том, что нам с Вами удалось создать, хотя и не сложную, но, все же, карту нашего изображения.

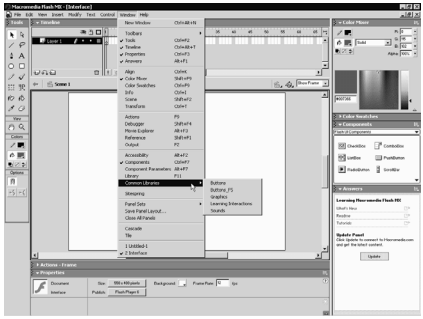
На этом мы завершаем изучение языка ***HTML***. Мы достаточно ознакомились со всеми основными тэгами этого языка и, в общем-то, научились с ним работать. На следующем уроке мы с Вами уже начнем постройку нашей веб-страницы, но будем пользоваться не ручным способом (не написанием ***HTML***-тэгов), а уже будем пользоваться специальной программой, которая позволяет делать это более удобным способом. Это программа называется ***Composer***-ом комплекса ***Netscape***.

Раздел 5. Программы для работы с Web-сайтами

5.1. Векторная графика: Macromedia Flash

5.1.1. Первое знакомство. Работа с панелью инструментов

5.1.1.1. Обзор интерфейса программы



Уважаемые дамы и господа. Вы проинсталлировали и открыли замечательную программу компании ***Macromedia — Flash MX***. На первом этапе обучения мы познакомимся с интерфейсом программы и научимся настраивать некоторые его элементы. Итак приступим. Если разрешение Вашего монитора составляет 1024 на 768 пикселей, то при первом запуске программы на своем экране Вы увидите аналогичную картинку. Обзор интерфейса начнем сверху вниз. В верхней части окна находится строка заголовка. Левая сторона, которой содержит имя программы и имя файла, открытого в текущем окне. С правой стороны расположены три кнопки. Первая слева позволяет свернуть окно программы.

Это бывает необходимо сделать, когда вы работаете одновременно с несколькими приложениями. Восстановим окно, щелкнув по пиктограмме внизу экрана. С помощью второй кнопки можно минимизировать окно программы. Например, следующим образом.

Теперь вы сможете подобрать такой размер окна, чтобы его было удобно просматривать одновременно с окнами других приложений. Повторный щелчок на этой же кнопке вернет окно ***Flash*** в полный экран. Щелчок на кнопке-крестике завершит работу программы. Чего мы делать не будем и продолжим обзор интерфейса. Ниже находится три аналогичные кнопки, которые таким же образом воздействуют на окно открытого файла. Под строкой заголовка расположена строка основного меню, знакомая Вам по другим программам для ***Windows*** и ***Macintosh***. Каждое из этих меню содержит список команд для управления редактором ***Flash***. Меню — это весьма большая тема. Подробный рассказ о назначении всех этих команд вызвало бы

у начинающих головную боль или в лучшем случае зевоту. Поэтому разбираться со всеми этими опциями мы будем постепенно, по ходу изучения программы. Сейчас я только обзорно познакомлю Вас с назначением основных команд меню поскольку они сгруппированы по тематическим признакам. Меню ***File***.

Команды этого меню позволяют создавать, открывать, сохранять, экспортировать и импортировать файлы ***Flash***-фильмов и статических изображений любого из кадров фильма в различных форматах. В нижней части списка команд находятся имена файлов, с которыми работали последними и которые можно быстро открыть щелкнув на соответствующей строчке. Последней в списке расположена команда ***Exit*** — для выхода из программы. Обратите внимание, что многим командам соответствует сочетание клавиш, которые Вы можете использовать для ускорения работы. Меню ***Edit*** — Правка.

Обеспечивает доступ к командам отмены и повторения операций, а также копированию, вставки, удалению, выделению графических объектов и кадров фильма. Команда ***Preferences*** — Предпочтения в нижней части списка устанавливает глобальные параметры для текущего и всех последующих фильмов. Начинающим флэшерам лучше оставить эту команду до поры до времени в покое. Разработчики позаботились об эффективной работе программы при настройках заданных по умолчанию. Предпоследняя команда списка ***Keyboard Shortcuts*** — оперативные клавиши, устанавливает соответствие используемых в ***MX*** клавиш, тем клавишам, которые применяют обычно в других приложениях.

Кто ранее работал с предыдущей версией программы, т.е. с ***Flash*** 5 и не хочет менять свои привычки. Или имеет большой опыт работы с такими приложениями как: ***Freehand*** 10, ***Adobe Illustrator*** 10, или ***Photoshop*** 6, — могут воспользоваться этой командой. Кроме того здесь имеется возможность устанавливать свои собственные уникальные комбинации клавиш для выполнения различных команд. И последняя команда меню ***Edit***, ***Font Mapping***

Соответствие шрифтов позволяет работать с файлами, содержащими шрифты, которые не установлены на Вашей системе и производить нужную Вам замену шрифтов. Эта команда появилась только в последней версии ***Flash***. Меню ***View*** — Вид. Это меню управляет видом рабочего пространства, качеством графического изображения, видимостью и настройкой некоторых элементов интерфейса, таких как: линейка, сетка, направляющие и некоторых других. Меню ***Insert*** — Вставка. Содержит команды, которые используются для ввода в фильм новых элементов, таких как: символы, слои, кадры и сцены. Не волнуйтесь если некоторые из перечисленных мною терминов, еще не

знакомы. По ходу изучения темы, мы будем неоднократно возвращаться к командам данного меню. Меню ***Modify*** — Видоизменять или модифицировать.

Команды этого меню позволяют: масштабировать, поворачивать и группировать графические объекты, сглаживать их форму, выравнивать и распределять их на сцене, а также делать многие другие полезные вещи. Хорошее знание команд этого меню позволяет добиться превосходных результатов рисования во ***Flash***-е. Меню ***Text***. Позволяет управлять параметрами выделенного текста, в частности: выбирать любой из установленных в системе шрифтов, размер шрифта, его начертание, выравнивание и межбуквенный интервал.

Дополнительные характеристики текста, в частности: цвет, абзацный отступ и другие устанавливаются с помощью специальной панели, о которой мы поговорим на соответствующем уроке. Меню ***Control*** — Управление. Не оказывает влияния на содержание ролика, а лишь служит для управления воспроизведением анимации. Меню ***Window*** — Окно.

С помощью команд этого меню можно отображать и скрывать вспомогательные панели, расположенные в правой части и внизу экрана, а также просматривать разные части фильма в отдельных окнах. Кроме того команда ***Common Libraries*** позволяет воспользоваться поставляемыми с программой библиотеками с кнопками, графикой и звуками. И наконец, меню ***Help*** — помощь. Команды этого меню предоставляют доступ к справочной системе ***Flash***. С помощью команд ***Lessons*** и ***Samples*** можно посмотреть уроки и примеры, встроенные в программу.

Новичкам настоятельно рекомендую посмотреть эти примеры, чтобы получить наглядное представление о возможности редактора ***Flash MX***. Возможно вы уже заметили, что при первом запуске ***Flash*** после инсталляции на экране отсутствует стандартная панель, знакомая нам по другим приложениям ***Windows***. Давайте отобразим ее на экране и придадим интерфейсу более привычный для нас вид. Для этого воспользуемся командой меню ***Window-Toolbars-Main***

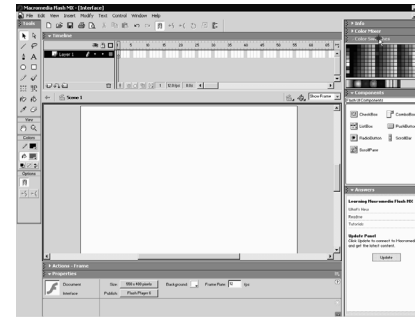
Основная. Теперь при последующих запусках программы эта панель будет всегда отображаться на экране. На стандартной панели находятся кнопки, известные Вам по другим программам, такие как: создать новый файл, открыть файл, сохранить, печать, предварительный просмотр, вырезать в буфер, копировать в буфер, вставить из буфера, отменить команду, повторить команду. Последние шесть кнопок характерны только для ***Flash***, о них я расскажу на следующих уроках

Ниже основной панели находится хронометрическая линейка или шкала времени ***Timeline***, предназначенная для компоновки и монтажа фильмов. Сейчас я только перечислю основные ее элементы. Сразу отмечу способность

редактора сворачивать шкалу при щелчке мышью по ее заголовку, что позволяет увеличивать область для рисования. Повторный щелчок снова развернет шкалу. В левой части линейки находится перечень слоев. По умолчанию программа при запуске создает проект с одним единственным слоем. Под ним расположены кнопки, позволяющие добавлять и удалять слои.

В последней версии также появилась новая кнопка с помощью которой можно создавать папки для слоев. Теперь слои можно укладывать в иерархическое дерево папок. Весьма полезное новшество, особенно для сложных проектов, имеющих большое количество слоев. Выше расположены кнопки с помощью которых каждый слой можно сделать невидимым, запретить его редактирование, а также отображать содержимое слоя в контурном виде.

Правую часть линейки занимает собственно шкала кадров — поле, где вы можете добавлять или удалять кадры фильма. Выше находится маркер, который указывает на текущий кадр отображаемый в окне. По умолчанию программа создает проект с одним единственным кадром под номером один. Ниже расположены кнопки, позволяющие отображать соседние кадры как бы через кальку, чтобы видеть разницу между предыдущим и последующими кадрами. Правее этих кнопок находятся поля со справочной информацией: о номере текущего кадра, частоте кадров в секунду, т.е. скорости проигрывания анимации и времени, прошедшем с начала фильма до выбранного кадра. Ну и как видите шкала времени имеет свои собственные полосы прокрутки.



Идем дальше. Большую часть экрана под шкалой времени занимает рабочая сцена или полотно, на котором вы будете создавать графику и анимацию. По умолчанию размеры полотна составляют 550 на 400 пикселей и оно имеет белый цвет. На следующем уроке я покажу Вам как можно изменять его параметры. В левой части экрана расположена панель инструментов, предназначенная для создания и редактирования графических объектов.

Работе с инструментами этой панели будет посвящено несколько подробных уроков. Поэтому сейчас я не буду задерживаться на этом вопросе. А теперь поговорим о вспомогательных панелях в правой части и внизу экрана. С помощью этих панелей вы сможете управлять характеристиками объектов, текстом, звуком, кадрами временной шкалы, сценами, выравнивать и распределять объекты, а также делать многие другие вещи. В новой версии

Flash вместо ***Adobe***-овских панелек с закладками появились складывающиеся панели "докерного типа". Одним кликом мыши по заголовку они разворачиваются или сворачиваются, экономя драгоценное место на экране. Во ***Flash MX*** появилась новая панель ***Properties*** — Инспектор свойств, которая выполняет множество функций. Ее содержимое контекстно меняется в зависимости от выбранного инструмента или объекта.

Так если ничего не выбрано, так как сейчас, в ней отображаются основные свойства фильма, т.е. размеры полотна его цвет и частота кадров в секунду. Если же, например, выбрать текст, то на панели ***Properties*** появится множество кнопок и полей для редактирования текста. По сути эта панель заменила множество других панелей ***Flash*** 5, отвечающих за свойства звука, текста, символов, кадра и т.д. Кроме того во ***Flash MX*** появилось еще несколько новых панелей. Например, панель ***Components*** для быстрого создания элемента управления, например, прокручивающихся текстовых полей, списков и кнопок. Как я уже отмечал, все панели могут быть отображены или скрыты с помощью команд меню ***Window***. Отобразим, например, панель ***Info*** — Информация, щелкнув на соответствующей строчке. Теперь появившуюся панель можно поместить в общий блок.

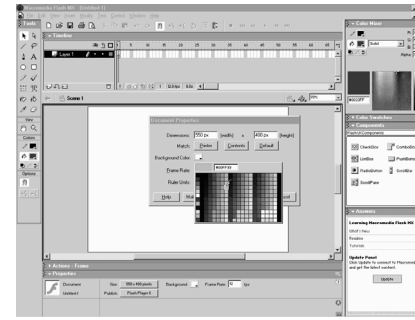
Для этого подведите курсор к сетке слева от заголовка и как только он примет вид четырехнаправленной стрелки, переместите панель в нужное место. Например, вот так. Таким образом, ***Flash MX*** позволяет быстро перестраивать рабочее пространство в соответствии со вкусом пользователя. При этом любое расположение панелей можно запоминать с помощью специальной команды, о которой я расскажу на следующем уроке. Скрыть и отобразить все панели удобно с помощью горячей клавиши Табуляция.

Ну и в заключении обзора интерфейса программы упомянем о стандартных полосах прокрутки и окне в правом верхнем углу под шкалой времени, в котором можно устанавливать произвольный или один из заданных разработчиками масштаб рабочей сцены.

5.1.1.2. Установка основных параметров фильма, настройка интерфейса редактора

Уважаемые слушатели на этом уроке мы научимся изменять основные параметры фильма с установки которых обычно начинают создание нового ролика а также познакомимся с принципами настройки интерфейса редактора ***Flash MX*** Как я уже отмечал на предыдущем уроке каждый новый фильм имеет стандартные основные параметры белый фон, ширину в 550, высоту в 400 пикселов и частоту равную 12 кадрам в секунду. Но эти параметры легко изменить. Обратимся к команде меню ***Modify — Document*** или воспользуемся клавиатурной комбинацией ***CTRL+J***. Перед нами появится окно свойства документа. В верхней части диалога находятся два поля, значения которых

определяют ширину и высоту полотна. ***Flash MX*** позволяет устанавливать эти параметры в пределах от 1 до 2880 пикселов. Ниже расположено окно с помощью которого можно изменять цвет фона. При щелчке по нему появится палитра в которой можно выбрать любой из 256 так называемых безопасных ***WEB*** цветов.



Если вы раздумали менять цвет, то просто нажмите клавишу ***Esc*** — палитра закроется. Еще ниже находится поле, в котором устанавливается скорость воспроизведения фильма, заданная в кадрах в секунду. Для ***WEB*** сайтов обычно устанавливается частота от 8 до 18 кадров в секунду. Помните чем меньше частота кадров, тем меньше окончательный размер файлов фильма и меньше нагрузка на процессор компьютера.

С другой стороны увеличение этого параметра обеспечивает более плавную анимацию. В раскрывающемся списке ***Ruler Units*** — можно задавать единицы измерения. При работе с ***Flash*** практически всегда используются пикселы, но при желании здесь можно установить единицы измерения в миллиметрах, сантиметрах, пунктах и дюй MAX.

А теперь давайте изменим некоторые из перечисленных параметров. Установим размеры фильма равными, например, 500 на 200 пикселов. А цвет фона, скажем, светло-голубым. Остальные опции оставим без изменения. Подтвердим выбор сделанных установок кликом на кнопке ОК. Вы видите, что у нас получилось. Вновь вернемся к диалогу Свойства Документа, но сделаем это по-другому. Если открыта панель инспектора свойств, то достаточно просто щелкнуть по кнопке ***Size***, где отображается размер фильма. Появится знакомое окно. Здесь имеется еще ряд кнопок, о которых я еще не рассказывал. Щелчок по кнопке ***Default*** быстро восстановит все параметры принятые по умолчанию.

Если вы часто работаете над фильмом с другими параметрами, то после установок нажмите кнопку ***Make Default*** — Сделать по умолчанию. В этом случае программа будет создавать проекты с этими параметрами и вам не нужно будет заниматься рутиной в начале каждой работы. Кнопка ***Match Printer*** подгоняет размер фильма под максимально доступную область печати, в соответствии с установленным в системе принтером и выбранным размером бумаги. Кнопка ***Match Contents*** Подогнать под содержимое, устанавливает размеры полотна так, что вокруг содержимого остается свободное рабочее

пространство с противоположных сторон. Точкой отсчета служит левый верний угол полотна Не трудно догадаться, что если фильм пустой как сейчас, то при щелчке по этой кнопке ничего не произойдет. Возможно не совсем понятно, ну что же давайте продемонстрируем работу этой установки. Закроем окно, щелкнув по кнопке ОК. И нарисуем что-нибудь на сцене. Например, овал. Снова вызовем окно свойств и нажав кнопку ***Contents***, подтвердим установку. Я думаю, что комментарии больше не нужны. Очевидно, что эту опцию удобно применять в конце проделанной работы. Инструментом Стрелка выделим и удалим овал, нажав клавишу ***Delete***.

Еще разок откроем окно свойств документа. Для этого можно воспользоваться другими двумя способами. Либо через контекстное меню, для чего нажмем правую кнопку мыши и выберем пункт меню ***Document Properties***. Либо сделаем двойной щелчок в поле, где отображается скорость анимации. Установим, принятые по умолчанию параметры и закроем окно. Обратите также внимание, что цвет полотна и скорость воспроизведения могут быть определены на панели ***Properties*** и без вызова диалога Свойства Документа.

А теперь займемся настройкой интерфейса программы и сделаем его более удобным для последующей работы. Как вы уже заметили, ***Flash*** по умолчанию открывает в правой части экрана четыре вспомогательные панели. Но их в программе значительно больше. Для начала уберем некоторые панели, которые нам не понадобятся на ближайших уроках и вместо них установим другие. Щелкнем правой кнопкой мыши на заголовке панели ***Answers*** Ответы, предназначенной для получения справки, в том числе и с помощью интернета. Кстати говоря, эта панель — новинка редактора ***MX***. Закроем панель, выбрав команду ***Close Panel*** .

Аналогичным образом поступим с панелью ***Components***. Далее выведем на свет Божий другие панели, которые будут нам нужны для освоения основ рисования во ***Flash***-е. Обратимся к меню ***Window*** и отобразим на экране панель ***Align*** — Выравнивание. Подведем курсор к сетке, слева от заголовка и поместим панель в общий блок. Аналогичным образом откроем и разместим панель ***Info*** — Информация. И панель ***Transform*** — Трансформация. Развернутые панели можно свернуть, щелкнув мышью по их заголовкам. Вы можете, конечно, разместить панели по своему усмотрению в зависимости от характера предстоящей работы и разрешения экрана Вашего монитора.

А теперь давайте запомним это расположение панелей, чтобы в последующем заново не настраивать интерфейс. Выполним команды ***Window — Save Panel Layout*** Сохранить расположение панелей. В появившемся одноименном диалоговом окне, зададим имя заданной компоновке панелей. Например ***My*** и кликнем кнопку ОК. Теперь если в

процессе работы мы будем перемещать панели, открывать и закрывать другие панели, то можно всегда вернуться к сохраненному расположению. Посмотрим как все это работает. Вернемся к компоновке панелей заданных по умолчанию. Для чего выполним команду ***Window — Panel Sets — Default Layout***. Затем попробуем установить ранее настроенный интерфейс. Для этого выполним те же операции ***Window — Panel Sets***. А в списке настроек панелей выберем появившийся пункт ***My***.

Как видите, все получилось. Таким образом мы можем создать и сохранить несколько настроек интерфейса. И быстро воспользоваться каждой из них в зависимости от характера выполняемой работы. Может случиться так, что с некоторого времени, сохраненный когда-то набор панелей оказывается не нужным. Такие наборы можно удалять из меню.

Но возможность удаления не встроена во ***Flash***. Файл соответствующий набору панелей нужно удалять вручную. Он находится в папке с названием ***Panel Sets*** в каталоге ***Flash MX***, куда вы установили программу. Правда возможны варианты. Если Ваш компьютер подключен к сети, то ищите этот файл в подкаталоге ***Application Date*** каталога ***Windows***. Для увеличения рабочего пространства можно проделать еще некоторые действия. Уменьшим мышью ширину шкалы времени.

А затем с помощью окна под шкалой отмасштабируем размер полотна, выполнив команду ***Show Frame***, которая точно впишет полотно в рабочую область. Эту же операцию можно произвести командой меню ***View — Magnification*** Масштабирование ***Show Frame***, или, что быстрее нажав сочетание клавиш ***Ctrl***+2. Теперь мы готовы к освоению основ рисования в редакторе ***Flash***. В первую очередь вам необходимо научиться работать с панелью инструментов, с помощью которой создаются графические объекты.

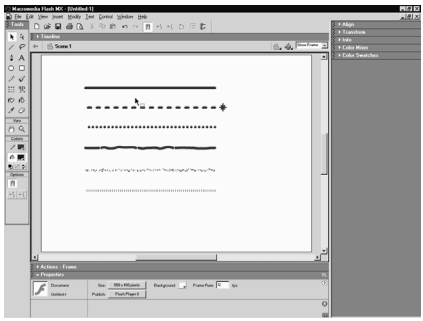
Познакомимся с ней чуть подробнее. Панель инструментов разделена на четыре раздела. В верхнем разделе ***Tools*** Инструменты находятся средства для рисования и редактирования графических объектов.Во втором разделе ***View*** просмотр, расположены инструменты для управления отображением. Эта так называемая Рука ***Hand*** и инструмент Масштаба — ***Zoom***. С подобными инструментами вы наверняка встречались в других графических редакторах. Ниже в разделе ***Colors*** Цвета находятся инструменты выбора цвета контуров и заливок, которые составляют основу векторной графики.

И, наконец. в самом нижнем разделе ***Options*** Параметры, находятся модификаторы инструментальных средств, которые определяют конкретное поведение выбранного инструмента. Вы можете посмотреть как меняются пиктограммы этой группы при смене инструментов. Для тех кто уже работал с предыдущей пятой версией ***Flash***, замечу, что панель инструментов

претерпела мало изменений. И лишь в раздел ***Tools*** добавлено два новых средства, а именно, ***Free Transform***

Свободная трансформация и ***Fill Transform*** Трансформация заливки. Впрочем, последний уже был в предыдущей версии редактора, только находился в разделе модификаторов инструментов заливки. Принципам работы с отдельными инструментами этой панели с приведением конкретных примеров будет посвящены последующие урок

5.1.1.3. Инструмент "Линия"



Уважаемые слушатели, на этом уроке мы начнем подробно изучать панели инструментов программы ***Flash MX***, а также научимся использовать панель Инспектора Свойств для редактирования графики. Рисовать мы будем в одном кадре, поэтому давайте свернем шкалу времени, щелкнув по ее заголовку, а затем отмасштабируем полотно командой ***SHOW FRAME***.

Теперь у нас имеется больше пространства для рисования. Рассмотрим верхний раздел панели инструментов ***TOOLS***, где находятся средства для рисования и редактирования графических объектов. Для быстрого выбора этих инструментов вы можете воспользоваться клавиатурой. Для определения "горячей клавиши" наведите курсор мыши на соответствующий инструмент. При этом появится динамическая подсказка с его названием и клавишей быстрого выбора. Постарайтесь постепенно запоминать эти клавиши, что в дальнейшем существенно ускорит вашу работу.

В первом ряду инструментов панели находится инструмент ***Arrow*** Стрелка, которая служит для выделения, перемещения и редактирования графических объектов. Однако прежде чем выделять и редактировать, наверное, нужно что-нибудь нарисовать, поэтому изучение панели мы начнем с инструментов рисования. Во ***Flash***, как и в других редакторах векторной графики, существует два основных типа графических объектов — это линии, или контуры, и заливки. Линии создаются тремя инструментами панели: ***Line*** (Линия), ***Pencil*** (Карандаш) и ***Pen*** (Перо). Инструмент ***Pen*** наиболее сложен в освоении, учиться работать с ним мы будем несколько позже.

Если необходимо нарисовать прямую линию, воспользуйтесь инструментом ***Line***. Курсор примет вид перекрестия. Выберите точку, где должна начинаться линия, щелкните левой кнопкой мыши и не отпуская ее проведите мышью по полотну. За курсором потянется прямая линия. Как только вы отпустите

кнопку мыши, прорисуется отрезок, соединяющий начальную и конечную точки рисования.

Если во время перемещения мыши удерживать клавишу ***SHIFT***, то у вас получится либо строго горизонтальная линия, либо строго вертикальная, Нажмите клавишу на клавиатуре позволяющую рисовать линии под прямыми углами либо под углом 45 градусов, в зависимости от того, в каком направлении вы начнете перетаскивание в начальный момент. Вот, пожалуй, и все премудрости использования этого простого инструмента. Однако нарисованные линии вовсе не обязаны быть именно такими, как получились у нас на экране. Посмотрим, как можно изменять параметры нарисованной линии. Это делается с помощью панели ***Properties*** (Инспектор Свойств).

Замечу, что в 5-ой версии редактора для этого применялась отдельная вспомогательная панель ***Straight***. Удалим нарисованные линии, чтобы они нас не отвлекали, для чего выделим их с помощью инструмента ***Arrow*** и нажмем клавишу ***DELETE***. Теперь нарисуем 6 одинаковых горизонтальных линий. Проще сначала с помощью инструмента Линия нарисовать одну, затем инструментом ***Arrow*** выделить её и удерживая клавишу ***CTRL*** создать пять её копий. Какой клавишей вместе с мышкой можно делать одинаковые копии Обратите внимание, что при перемещении выделенной линии рядом с курсором появился знак "+", что говорит о копировании объекта.

Почему я нарисовал 6 линий? Через пару минут вы увидите, что редактор предоставляет в ваше распоряжение именно 6 основных стилей контура. Выделим все линии, что необходимо для модификации, и обратимся к панели ***Properties***. Здесь находится поле, в котором вы можете задать толщину контура, начиная от самого тонкого и до 10-ти пикселлов. Щелкнув по кнопке со стрелкой вниз это можно сделать с помощью появившегося ползунка.

Установим толщину всех линий равную, например, 5-ти пикселлам. Левее находится окошко,щелкнув по которому вы можете с помощью появившейся палитры выбрать необходимый цвет линии, ну, например, красный. Разумеется, каждой линии вы можете задать свой собственный цвет и толщину. Для этого нужно предварительно выделить лишь одну из них и сделать соответствующие установки. Цвет линии также можно устанавливать с помощью окна в разделе ***Colors*** Панели инструментов. Правее расположено окно со списком выбора для стиля линии или контура. Как я уже отмечал, в редакторе имеется 6 основных стилей. Сейчас для всех линий установлен пункт ***Solid*** Сплошной, который предлагается по умолчанию. Выделим только одну — вторую линию.

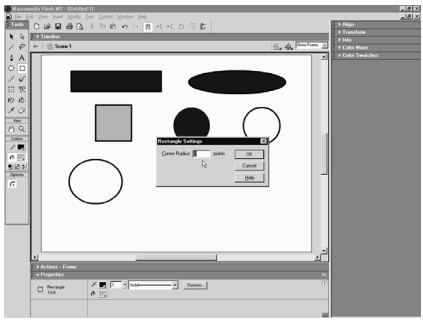
На панели ***Properties*** установим для нее пункт ***Dashed*** Пунктир. Для третьей линии после ее выделения установим пункт ***Dotted*** Точечный, для четвертой ***Ragged*** — Рваный, для пятой ***Stipple***- Мелкий пунктир,и наконец для

последней — пункт ***Hatched*** Поперечные штрихи. Но это еще не все. Редактор предлагает дополнительные настройки для каждого стиля контура, за исключением сплошного, для которого можно изменять только толщину и цвет. Выделим, например, пунктирную линию и щелкнем по кнопке ***Custom*** Другой на панели ***Properties***. Появится диалоговое окно, в котором можно определить длину пунктира и расстояние между черточками. Здесь также можно изменить толщину линии и ее стиль.

Введем, например, в правое поле, определяющее расстояние между пунктирами, цифру 12 и подтвердим установку. Вы видите, что расстояние между черточками увеличилось. Точно так же можно настраивать и другие стили. Так для точечной линии можно изменить расстояние между точками, для штриховой — длину и ориентацию штрихов и т.д. Поэкспериментируйте с этими настройками самостоятельно, иначе у нас останется мало времени для освоения остальных инструменто

5.1.1.4. Инструменты "Карандаш", "Овал", "Прямоугольник"

Другим средством для рисования контуров является инструмент ***Pencil*** Карандаш. В отличие от инструмента Линии он позволяет создавать контуры произвольной формы. Карандаш очень прост в использовании. Выбрал его, настроил параметры будущего контура, щелкнул на полотне и рисуй себе контур произвольной формы, удерживая нажатой левую клавишу мыши. Об этом инструменте не стоило бы больше и говорить, если бы некоторые его особенности. Обратите внимание, что при выборе инструмента карандаш в разделе ***Options*** — Модификаторы — появилась одна единственная кнопка. При щелчке на ней откроется меню из трех пунктов. Первый из них, выбранный по умолчанию, называется ***Straighten*** — Выпрямление.



Вы этом режиме программа ***Flash*** пытается определить, какие формы рисуются, и исправляет неточности человеческой руки. Например, попробуем нарисовать неровный прямоугольник. При отпускании кнопки мыши программа превратит его в ровный. А в непонятной округлой фигуре редактор распознает практически идеальный эллипс. Проделаем те же операции при включенных модификаторах ***Smooth*** — Сглаживание, и ***Ink*** — Подпись. Выберем режим Сглаживание и рисуем. Как видите, программа сглаживает шероховатости, сохраняя кривизну линии

оригинала. А в режиме ***Ink*** редактор старается отобразить любую проведенную кривую как можно точнее, хотя самые мелкие неровности тоже сглаживаются. Таким образом программа ***FLASH*** проявляет довольно-таки гибкий подход к пользователю как к рисовальщику. Ненужные неровности могут быть исправлены, а нужные сохранены. Разумеется, все контуры, нарисованные инструментом Карандаш, могут быть модифицированы с помощью панели Инспектора Свойств точно таким же образом, как я показывал на примере инструмента Линия. Не забывайте только выделять контур перед модификацией. Поехали дальше, и сейчас разберемся с такими инструментами рисования как Овал и Прямоугольник.

Пользоваться этими инструментами так же просто, как и инструментом Линия: следует перемещать мышь при нажатой левой кнопке. При этом вычерчивается соответствующая фигура. Если при рисовании удерживать нажатой клавишу ***SHIFT***, Какой клавишей на клавиатуре можно добиться отрисовки правильных геометрических форм то создаются, соответственно, круг или квадрат. Особенность использования этих инструментов в том, что они создают не только контуры, но и заливки. Цвет заливки определяется соответствующей установкой в разделе ***Colors***.

После создания фигуры можно с помощью инструмента Стрелка выделить только контур или только заливку, после этого их нужно перемещать или использовать самостоятельно. Цвет заливки или контура легко поменять. Для этого инструментом Стрелка выделите, например, заливку, а затем в разделе ***Colors*** щелкните по окну Цвет Заполнения и в появившейся палитре выберите подходящий цвет. Такую же операцию можно проделать и с помощью панели ***Properties***. Если вы хотите создать прямоугольник или овал без заливки, необходимо после выбора инструмента сбросить цвет заливки, щелкнув по значку ***No Colors*** — Без Цвета.

Аналогичным образом можно сбросить цвет контура и создавать только заливки овальной или прямоугольной формы. У инструмента Прямоугольник есть еще одна особенность, о которой следует знать, а именно возможность создавать прямоугольники с закругленными углами. Для этого после выбора инструмента следует щелкнуть по кнопке модификатора ***Round Rectangles Radius***.

Вы увидите окно, предлагающее выбрать радиус закругления углов. Введем в единственное поле, например, 20 пунктов и кликнем кнопку ОК. после рисования у нас получится вот такой прямоугольник. В качестве примера раздельного использования контура и заливки попробуем быстро нарисовать фигуру полумесяца. В начале с помощью инструмента Овал создадим круг с голубой заливкой. Инструментом Стрелка щелкнем на контуре, выделив его.

Переместим контур немного влево и щелкнем мышью где-нибудь на пустом месте,сняв выделение с контура. При этом контур разрежет заливку на две части. Кликнем на нужной нам части заливки и переместим ее куда-нибудь в сторону. Наш полумесяц готов, осталось только удалить ненужную часть графики. Какой кнопкой можно удалить выделенный объект Я не просто так привел этот пример, а хотел, чтобы вы сразу заметили особенность графики ***Flash*** редактировать саму себя при наложении. Поначалу эта особенность может причинить существенное неудобство, ведь если ошибочно поместить одно изображение на другое, можно потерять плоды длительного труда. Однако в нашем распоряжении есть такой инструмент, как ***Undo*** — Отменить, позволяющий легко исправить ошибку. С другой стороны подобный метод позволяет создавать сложные изображения простыми средствами и по мере совершенствования ваших навыков превращается во благо. Кроме того, существует способ предотвратить нежелательное редактирование готовых изображений.

Например, группирование объектов, размещение графики в разных слоях и некоторые другие, о которых мы поговорим позже. Ну и в заключение этого урока рассмотрим два вспомогательных инструмента раздела ***View*** — Просмотр, которые помогут нам более эффективно работать в программе. Это так называемая Рука — ***Hand*** и инструмент масштаба ***ZOOM***. Пользоваться этими инструментами очень просто. С помощью руки можно перемещать всю рабочую область. Если выбран какой-либо другой инструмент, например, Стрелка, то при нажатии клавиши ***SPACE*** (ПРОБЕЛ) временно активизируется инструмент Рука. Как только клавиша ПРОБЕЛ отпущена, Какую клавишу нужно удерживать чтобы двигать все полотно то вновь активизируется ранее выбранный инструмент. С помощью инструмента ***ZOOM*** можно увеличивать масштаб рабочей области, а при нажатой клавише ***ALT*** уменьшать ее видимый размер. Какую клавишу нужно удерживать чтобы масштаб уменьшался.

Кроме того, при использовании инструмента Масштаб можно перетащить выбранный прямоугольник для указания участка фильма, который требуется сделать крупнее. Как только кнопка мыши будет отпущена, изображение увеличится до указанной степени. Замечу, что оба указанных инструмента никак не влияют на содержание ролика, а используются только для удобства просмотра

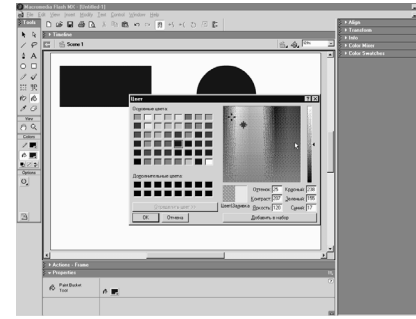
5.1.1.5. Инструмент "Заливка"

Уважаемые слушатели, на этом уроке мы продолжим знакомиться с панелью инструментов редактора и научимся работать инструментами Заливка и ***TRANSFORM FIL*** (Преобразование заливки). Рассмотрим инструмент Заливка, который позволяет закрашивать замкнутые контуры любым цветом, а также изменять уже созданные заливки. Нарисуем фигуры, заливку которых

мы будем изменять, пусть это будут прямоугольник и круг. Для того, чтобы поменять цвет заливки существующего объекта, выберем инструмент ***Paint Backet*** и щелкнем по окну выбора цвета заливки раздела ***Colors*** — Цвета.

Откроется знакомая цветовая палитра. Если вас не устраивает ни один из предложенных цветов, то кликните по кнопке в правом верхнем углу палитры. Появится диалоговое окно, в котором можно различными способами настроить дополнительные цвета. Подберем нужный оттенок, например, такой, подтвердим выбор и щелкнем инструментом Заливка по нашему прямоугольнику — его цвет мгновенно изменится. Точно так же можно заполнить любой замкнутый контур, не имеющий заливки. Нарисуем Карандашом любую фигуру и постараемся сделать ее замкнутой, затем выберем инструмент Заливка, цвет и щелкнем внутри нарисованного контура — он будет заполнен выбранным цветом. Интересно, что программа ***Flash*** позволяет закрашивать не только замкнутые контуры, но и контуры, имеющие небольшие разрывы. Для этого в программе предусмотрено несколько режимов работы инструмента Заливка. Для их выбора щелкнем на пиктограмме раздела модификаторов.

Откроется меню из четырех пунктов: это и есть режимы заливки. Пункт Не закрывать зазоры соответствует нормальному режиму, при котором закрашиваемый контур должен быть полностью замкнут, остальные три режима: закрывать малые зазоры, закрывать средние зазоры и закрывать большие зазоры допускают наличие разрывов в контуре и отличаются допустимой величиной этого разрыва.



Проведем маленький эксперимент: нарисуем Карандашом замкнутый контур и с помощью инструмента Ластикделаем в нем маленький разрыв. Выберем инструмент Заливка и попробуем закрасить этот контур в нормальном режиме работы. Ничегошеньки у нас не получилось. Попробуем поставить режим закрывать малые зазоры — опять ничего. Ну что же, установим следующий режим — закрывать средние зазоры — и повторим попытку. "Ура, заработало! — как сказал бы кот Матроскин". Что касается конкретной величины допустимого зазора контура в различных режи MAX заливки, то программа ***Flash***, к сожалению, не дает на этот вопрос однозначного ответа, поэтому мой вам совет — при наличии разрывов в контуре подбирайте режим заливки опытным путем.

Пока я рассказал вам только о самом простом виде заливки, а именно о заливке одним цветом или так называемой сплошной заливке. К счастью, программа ***Flash*** предоставляет нам возможность использования и других, более интересных режимов заливок: градиентных и растровых. Чтобы создать эти виды заливок необходимо воспользоваться вспомогательной панелью ***Mixer***. Здесь имеется раскрывающийся список, в котором можно выбрать вид заливки. Пункт ***Solid*** Сплошная означает сплошную заливку, с которой мы уже знакомы, при этом на панели отображается кнопка выбора цвета заливки, которая открывает знакомую палитру. Следующие пункты этого меню предназначены для создания градиентных и растровых заливок. Пункт ***Linear Gradient*** определяет линейную градиентную заливку, ***RADIAL GRADIENT*** — круговую градиентную, а пункт ***BITMAP*** — растровую заливку.

При выборе градиентной заливки, например, линейной, на панели появляется линейка, которая отображает вид будущего градиента, на ее концах всегда находится два маркера, показывающие, между какими цветами осуществляется цветовой переход. По умолчанию нам предлагается черно-белый градиент. Сначала посмотрим, как это работает. Выберем инструмент Заливка и щелкнем внутри прямоугольника. Он будет закрашен черно-белым линейным градиентом. Направление перехода цвета линейного градиента можно задать, нажав кнопку мыши и вычерчивая линию внутри фигуры вот таким образом.

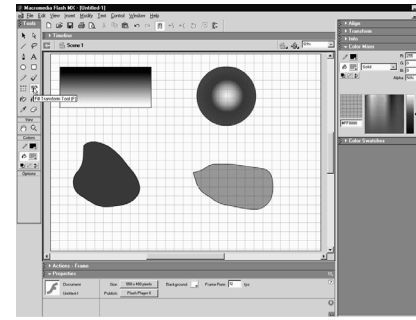
При выборе кругового градиента имеет значение место щелчка внутри контура фигуры, в этом месте располагается центр идеальной заливки. Однако черно-белый градиент — это несколько скучно. Чтобы выбрать, между какими цветами осуществляется переход, проделаем следующее: отметим один из маркеров линейки, щелкнем на окне выбора цвета и выберем в палитре подходящий оттенок. Затем настроим цвет второго маркера. Это можно сделать более гибким способом с помощью окна и ползунка в нижней части панели. Щелкнем по нужному оттенку, после чего можно подрегулировать его яркость с помощью ползунка. Теперь настроенный градиент можно применить к любому нашему объекту, щелкнув по нему инструментом ***Paint Backet***.

Таким образом можно получить любые двухцветные градиенты. А что делать, если надо получить постепенный цветовой переход между тремя и более цветами? Все очень просто: подведите указатель по нижнюю часть линейки, рядом с ним появится знак "+", щелчок мыши создаст новый маркер, цвет которого можно настроить показанными ранее способами. Так создаются интересные многоцветные градиенты. Если вы ошибочно добавили новый маркер, то нажав левую кнопку мыши потяните его вниз — маркер исчезнет.

И еще одна замечательная возможность редактора по созданию заливок, а именно: цвет любого маркера можно сделать прозрачным или

полупрозрачным. Для этого активизируем один из маркеров и в поле ***Alfa*** установим прозрачность около 40%. Применим полученный градиент к нарисованному кругу. Чтобы эффект прозрачности был больше заметен, выведем на свет божий вспомогательную сетку: для этого выполним команду ***VIEW***, ***Grid***, ***Show Grid***. Теперь хорошо видно, что центральная часть круга является полупрозрачной. Собственно говоря, с помощью панели ***Mixer*** можно сделать полупрозрачной и сплошную заливку. Так в списке заливок выберем пункт, ***Solid*** — Сплошная, установим прозрачность около 50% и применим эту заливку к какой-нибудь фигуре. На фоне сетки хорошо виден полученный эффект.

5.1.1.6. Инструмент "Преобразование заливки"



Как я уже отмечал, редактор позволяет создавать еще один не менее интересный вид заливки — растровую заливку. Но прежде чем познакомиться с ней нам желательно освоиться с инструментом ***Transform Fill*** — Преобразование заливки. Этот инструмент применим только для градиентных и растровых заливок, на сплошную заливку он не оказывает никакого влияния. Инструментом Стрелка выделим нижние объекты и удалим их клавишей ***DELETE***. Выберем инструмент ***Transform Fill***, вид указателя мыши изменится. Щелкнем теперь по прямоугольнику с линейным градиентом. На экране появились три маркера. Круглый маркер в центре фигуры позволяет переместить центр заливки в нужную точку вот таким образом, квадратный маркер предназначен для изменения размера заливки, а с помощью круглого углового маркера заливку можно повернуть, тем самым задав ей необходимый наклон.

Если же щелкнуть инструментом ***Transform Fill*** по круговому градиенту, то появятся четыре маркера. Центральный маркер так же служит для перемещения центра заливки, средний круглый маркер позволяет изменить размер градиента, квадратным маркером можно задать отклонения от круговой формы, а с помощью еще одного круглого маркера заливку можно повернуть на любой угол. Таким образом, используя сочетания показываемых настроек, линейные и круглые градиенты можно изменить как только душе угодно. А теперь снова вернемся к инструменту Заливка и посмотрим, как в программе создаются растровые заливки. Если сейчас на панели ***Mixer*** установить пункт ***Bitmap***, Растр, то этот выбор сам по себе ничего не даст, сначала нужно

импортировать в файл проекта какое-нибудь растровое изображение. Это делается с помощью команд ***FILE***, импорт или импорт в библиотеку.

Если все-таки щелкнуть по пункту ***Bitmap***, то редактор сам предложит произвести импорт растра и откроет диалоговое окно импорт в библиотеку. Выберем в диалоговом окне файл подготовленного заранее растра, например, формата ***.JPEG*** и подтвердим установку, при этом миниатюрная копия импортированного изображения отобразится на панели ***Mixer***, а сам файл будет помещен в библиотеку фильма. Импортировать можно сколько угодно растров, и все они будут отображаться в уменьшенном виде на панели ***Mixer*** при выборе пункта Растр.

Только не перестарайтесь при импорте, поскольку размер файла проекта будет при этом существенно увеличиваться. Что касается библиотеки и принципов ее использования, то это тема отдельного урока. Выбрав нужную пиктограмму, впрочем, в данном случае выбирать не приходится, щелкнем инструментом Заливка по какой-нибудь фигуре.

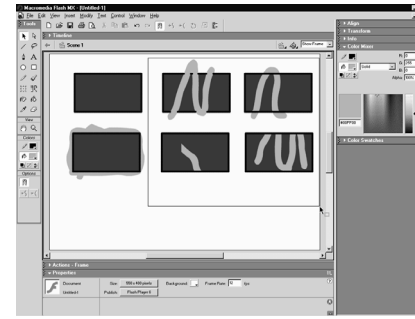
Она равномерно заполнится уменьшенными копиями растров. Затем используя инструмент ***Transform Fill*** изменим эту заливку по своему вкусу, например, следующим образом. Снова выберем инструмент Заливка и теперь обратим внимание на еще одну кнопку в разделе ***Options***. Этот модификатор называется ***Lock Fill*** — Блокировать заливку, и применяется только для градиентных и растровых заливок. После активации этой кнопки закрашиваемая форма рассматривается редактором как часть текущей, которая редактировалась или создавалась последней. Этот параметр блокирует угол, размеры и исходную точку текущего градиентного или растрового заполнения. Если модификатор не применяется, заливка будет создана заново. Возможно, не очень понятно? Ладно, согласен. Лучше один раз увидеть, чем сто раз услышать. Выберем инструмент Прямоугольник, сбросим цвет заливки и нарисуем один большой и ниже несколько маленьких прямоугольников. Установим линейную градиентную заливку и настроим черно-белый градиент.

Выберем инструмент ***Paint Backet*** и закрасим нарисованные фигуры, начиная с самых маленьких. Теперь нажмем кнопку ***Lock Fill*** и повторно кликнем по маленьким прямоугольникам. Они заполнятся градиентом таким образом, как будто представляют собой единый объект — большой прямоугольник, который редактировался последним перед выбором модификатора. Я думаю, дополнительные комментарии излишни. Этот модификатор удобно использовать, например, при создании рисунка закатного неба с чередой облаков, где все облака должны быть закрашены одним общим последовательным градиентным заполнением.

5.1.1.7. Инструменты "Кисть", "Чернильница", "Пипетка", "Ластик"

Уважаемые слушатели, на этом уроке мы продолжим изучение панели редактора ***Flash MX*** и подробно познакомимся с такими инструментами, как ***Brush Tool*** — Кисть, ***Incle Boatle*** — Чернильница, ***Eyedropper*** — Пипетка и инструментом ***Eraser*** Ластик. С помощью инструмента ***Brush*** можно создавать заливки произвольной формы без контуров.

Он применяется для рисования мазков заливки и заполнения замкнутых фигур. При выборе Кисти в разделе ***Options*** появляется несколько модификаторов. Параметр ***BRUSH SIZE*** представляет собой раскрывающийся список из 10-ти раскрывающихся параметров кисти, начиная от самого маленького и заканчивая вот таким.



Список ***BRUSH SHAPE*** позволяет задать форму кисти. Здесь содержатся 9 профилей, созданных на основе круга, эллипса, квадрата, прямоугольника и линии. Таким образом вы можете получить различные варианты мазков. При использовании профиля отличного от окружности размер, определяемый установкой ***BRUSH SIZE***, относится к самой широкой части профиля. Цвет мазков кисти устанавливается в разделе ***Colors*** панели инструментов. Можно также использовать градиентные и растровые заливки, для чего нужно будет сделать необходимые настройки на панели ***Mixer***.

Например, если здесь установить линейную градиентную заливку, то получатся вот такие мазки. Щелчок по кнопке ***BRUSH MAUNT*** открывает меню из пяти возможных режимов работы инструмента Кисть. Чтобы проиллюстрировать эти режимы, нам понадобится какой-нибудь графический объект. Пусть это будет прямоугольник с черным контуром и красной заливкой. Нарисуем его, но сначала очистим экран. Выберем инструмент Прямоугольник, на панели ***Properties*** назначим цвет контура черным, толщину равной 4 пикселлам, а цвет заливки красным.

Теперь рисуем. Создадим пять копий этого прямоугольника. Какой кнопкой нужно воспользоваться, чтобы создавались копии Выберем инструмент ***Arrow***, выделим прямоугольник двойным щелчком, нажмем клавишу ***CTRL*** и удерживая ее выполним перетаскивание копии. Этим же способом сделаем еще 4 копии нашей фигуры. Скопировать выделенный объект можно также командой ***EDIT Duplicate***. Сделаем отмену.

Оригинал оставим в покое, а на копиях продемонстрируем 5 различных режимов работы Кисти. Выберем инструмент ***BRUSH*** и назначим цвет заливки отличный от цветов прямоугольника, ну, например, зеленый. Щелкнем по кнопке ***BRUSH MAUNT*** и выберем режим ***NORMAl***, впрочем, его выбирать не надо, он стоит по умолчанию. Попробуем закрасить первую копию прямоугольника. Вы видите, что мазки кисти закрашивают все изображение: и заливку, и контур. Установим режим ***Paint Fills*** и повторим операцию. Как видите, закрашивается только заливка, а контур объекта сохраняется как ни в чем ни бывало. В режиме Раскраска сзади закрашивается только часть полотна, на которой ничего не нарисовано. Этот режим можно использовать, например, для ручного раскрашивания фона. Для демонстрации режима ***Paint Selection*** Раскраска по выбранному, необходимо выделить какой-либо объект или его часть. Для этого воспользуемся инструментом ***Arrow***.

Вы видите, что раскрашивается только выделенный участок заливки. И, наконец, режим ***Paint Inside*** Раскраска внутри позволяет ограничивать область раскраски каким-либо замкнутым контуром. Весьма полезный режим для раскраски, например, усеченных объектов. Установим нормальный режим работы кисти и освободим место, удалив часть фигур.

Теперь рассмотрим вспомогательные инструменты, которые часто могут существенно облегчить жизнь пользователю. Это инструменты ***Ink Boatle*** — Чернильница, инструмент ***Dropper*** Пипетка и инструмент ***Eraser*** — Ластик. Инструмент Чернильница позволяет изменять оформление уже существующих контуров, а также создавать контуры с различными параметрами вокруг заливок произвольной формы. Предположим, вы нарисовали какую-нибудь фигуру, пусть это будет круг, и захотели изменить часть контура. На панели ***Properties*** настроим нужные параметры будущего контура, например. такие: толщина 5 пикселлов, стиль точечный, цвет синий.

Выберем инструмент Чернильница и щелкнем в любом месте фигуры. Новый контур готов. Вообще-то говоря, этот результат мы могли бы получить и без Чернильницы. Для этого нужно было предварительно выделить контур, а затем сделать новые настройки на панели ***Properties***. Поэтому Чернильница чаще всего применяется не для изменения, а для создания нового контура вокруг заливки или для переноса параметров контура одной фигуры на другую.

Предположим, вы нарисовали кистью заливку произвольной формы без контура, а потом захотели обвести ее. Выберем Чернильницу и настроим параметры контура: толщину 6 пикселлов, цвет черный, стиль — поперечные штрихи. Щелкнем теперь инструментом по заливке и получим рисунок,

похожий на ветку кактуса. для переноса контуров одной фигуры на другую инструмент Чернильница используется совместно с инструментом Пипетка.

После его выбора указатель мыши принимает соответствующий вид. Если провести мышью над контуром, около пипетки появится небольшой карандаш, а если над заливкой — изображение кисти. При щелчке мышью автоматически выбирается соответствующий инструмент и копируются параметры контура или заливки. Например, при щелчке на контуре выберется Чернильница, и автоматически устанавливаются параметры этого контура на панели ***Properties***. Теперь если мы щелкнем на контуре другой фигуры, его параметры изменятся. Если же снова выбрать Пипетку и щелкнуть ей по заливке, то будет автоматически выбран инструмент Заливка и теперь можно перенести скопированный цвет заливки на другую фигуру.

Таким образом можно с помощью пипетки перенести на любой другой объект нужную заливку или стиль контура, даже не глядя на их параметры. Вы можете заранее создать коллекцию собственных линий и заливок вне рабочего полотна, а затем использовать нужные стили при необходимости. Это значительно ускорит работу. У инструмента Пипетка есть еще одна особенность: если нажать клавишу ***SHIFT*** и навести инструмент на контур или заливку, то возле курсора появится крошечное изображение, напоминающее перевернутую букву ***U***. Щелчок мышью приведет к одновременному изменению цвета всех инструментов рисования и окраски программы.

Какой клавишей нужно воспользоваться, чтобы рядом с пипеткой появилась буква ***U*** Ну и в заключение этого урока рассмотрим инструмент ***Eraser*** — Ластик, который предназначен для ручного стирания графических объектов. Если вы хотите стереть весь рисунок, то, наверное, удобнее выделить его и нажать клавишу ***DELETE***, но часто приходится удалять лишь некоторые ненужные части контуров и заливок.

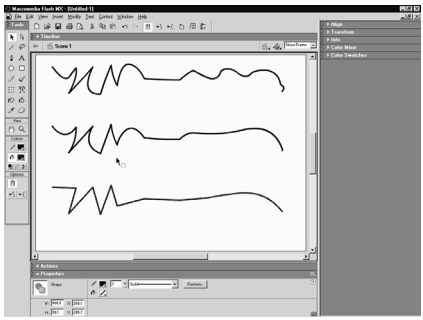
В этом случае пользоваться ластиком гораздо удобнее. Двойной щелчок на инструменте приведет к удалению всей графики на сцене, это довольно удобный способ быстрой очистки экрана. Если вы произвели эту операцию случайно, то не отчаивайтесь и воспользуйтесь кнопкой ***Undo*** на основной панели. Работать с инструментом Ластик очень просто. С помощью раскрывающегося списка ***Shape*** выберите форму и размер вашего ластика и просто проведите по рисунку, нажав левую кнопку мыши. Изображение под ластиком исчезнет, правда, это произойдет лишь в том случае, если инструмент работает в нормальном режиме, который, впрочем, установлен по умолчанию. Режимы работы ластика можно выбрать из меню ***Eraser Mode***.

С первым из них — ***NORMAL*** — мы уже познакомились, пункт — Стирание заливок — устанавливает режим, при котором под ластиком

стираются только заливки, а контуры остаются нетронутыми, режим ***LINES***, напротив, позволяет стирать только контуры, не затрагивая заливок. Щелкните на пункте включающем режим ластика — ***Eraser Lines*** Стирание контура Пункт ***Selected Fills*** предназначен для стирания только заливок или их частей, которые заранее были выделены, например, следующим образом. И, наконец, пятый пункт. ***Eraser Inside*** — позволяет стирать только заливки внутри замкнутого контура. Кроме того, при выборе инструмента Ластик, в разделе ***Options*** появляется кнопка ***Faucet*** Смывание.

Если активизировать эту кнопку, то вид указателя мыши изменится и станет похож на вентиль. Щелкните таким указателем по любому контуру или по заливке и попрощайтесь с ними. В принципе, действие этого модификатора эквивалентно выделению стрелкой с последующим нажатием клавиши ***DELETE***. Ну вот, пожалуй, и все, что я хотел вам рассказать на этом уроке.

5.1.1.8. Инструмент "Стрелка"



Уважаемые слушатели, на этом уроке мы продолжим знакомиться с панелью инструментов и более подробно изучим инструмент ***Arrow*** Стрелка, а также научимся работать с инструментом ***FREE TRANSFORM*** (Свободная трансформация), который впервые появился только в последней версии программы ***Flash***. Инструментом Стрелка мы уже неоднократно пользовались на предыдущих уроках для выделения и перемещения графических объектов. Однако этот инструмент редактора имеет ряд уникальных особенностей, позволяющих также редактировать нарисованные объекты. Рассмотрим эти особенности.

Нарисуем с помощью инструмента Карандаш контуры произвольной формы и нажмем клавишу ***V*** для быстрого выбора инструмента Стрелка. Теперь редактор находится в режиме выделения и перемещения объектов, о чем говорит маркер в виде пунктирного прямоугольника рядом с курсором. Подведем курсор к одному из концов нарисованного контура, маркер рядом с курсором примет вид прямого угла, это свидетельствует о том, что курсор переведен в режим перемещения точек, а точнее, узлов контура.

Теперь можно взять этот узел, нажав клавишу мыши, и перетащить в другое место. узлы находятся не только на концах, но и на изло MAX линии, так что потянув за излом возможно изменить положение сразу двух сегментов контура. Если же повести указатель мыши к линии контура, но не к узлу,

маркер примет вид дуги, теперь участок контура можно изменить, потянув его мышью, например, таким образом. Точно таким же способом можно модифицировать и заливки. Выберем, например, инструмент Кисть и сделаем мазок на полотне, снова активизируем инструмент Стрелка и подведем ее к краю заливки, теперь заливку можно изменять аналогичным образом, например, вот так. Но вернемся к нашим контурам. Чтобы склеить 2 контура, ухватитесь за узел одного и перетащите его к концу другого, курсор сам должен прыгнуть точно на конец другого контура, а точка привязки обозначится небольшой окружностью.

Отпустите кнопку мыши. Теперь два контура объединились и имеют общий узел. Необходимо помнить, что этот метод работает, если включен Режим объектной привязки ***Snap*** в разделе модификаторов — кнопка с изображением магнита. Этот модификатор имеет отдельную пиктограмму на основной панели и является переключателем, заставляющим перемещаемые объекты привязываться к определенным точкам других объектов. Если сделать двойной щелчок по контуру, то он выделится целиком, при одинарном щелчке будет выделен фрагмент от узла до узла.

Теперь имеется возможность контур разделить, перевести выделенный фрагмент на новое место. Отменим операцию. Если удерживая клавишу ***SHIFT*** щелкнуть еще на нескольких фрагментах контура, то можно выделить их одновременно. Снять выделение с какого-либо объекта можно не только щелчком на пустом месте, но и клавишей ***ESC*** или выполнив команду ***EDIT DESELECT ALL***. Вы, наверное, заметили, что при выборе инструмента ***Arrow*** в разделе ***Options*** появляются еще два модификатора: ***Smooth*** Сглаживание и ***Straighten*** — Спрямление. Они имеют дополнительные кнопки на основной панели и становятся активными только при выделении какого-либо объекта. давайте разберемся с принципами их работы.

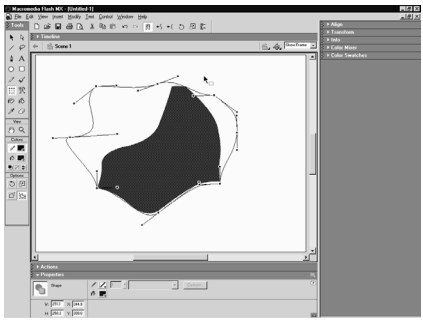
Выделим наш контур целиком и для наглядности сделаем 2 копии, к которым будем применять модификаторы. Для этого удерживая клавишу ***CTRL*** выполним перетаскивание вниз. Выделим первую копию и применим к ней модификатор ***Smooth***, просто щелкнув по нему несколько раз. Вы видите, что применение этой опции приводит к приданию кривым участкам контура более плавных очертаний. На отрезке прямых, если таковые имеются, модификатор никакого влияния не оказывает. Эта команда доступна также через меню ***Modify — Smooth***.

Теперь выделим вторую копию исходного контура и применим к ней модификатор ***Straghten***. Использование модификатора приводит к спрямлению кривых участков контура. Если применить его многократно, то все кривые участки превратятся в прямые отрезки. Команда меню ***Modify — Straighten*** выполняет те же действия. Замечу также, что рассмотренные

модификаторы можно применять не только к контурам, но и к заливкам, в чем предлагаю вам убедиться самостоятельно.

Ну и на закуску еще одна небольшая деталь: так, если вы работаете с каким-либо другим инструментом, например, Карандашом, то для временного выбора инструмента ***Arrow*** достаточно нажать клавишу ***CTRL***, как только клавиша будет отпущена, снова станет активным выбранный инструмент.

5.1.1.9. Инструмент "Свободная трансформация"

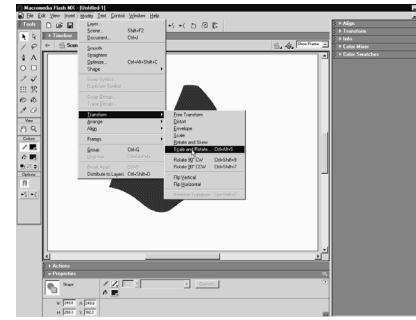


А сейчас поработаем с новым инструментом последней версии редактора — инструментом ***FREE TRANSFORM*** (Свободная трансформация). Судя по названию, он позволяет свободно преобразовывать графические объекты и текстовые блоки, а именно: с его помощью объекты можно перемещать, вращать, масштабировать, наклонять и искажать.

Посмотрим, как это делается. Для начала очистим экран, дважды щелкнув по инструменту Ластик, и нарисуем простую фигуру, например, прямоугольник без контура. Теперь выделим объект инструментом Стрелка и щелкнем мышью по кнопке ***FREE TRANSFORM*** или нажмем клавишу ***Q***. Фигура будет ограничена рамкой с маркерами. Теперь я подведу указатель к объекту, а вы посмотрите, как изменяется вид указателя в зависимости от того, к какой точке объекта я его подвожу. Каждый новый вид указателя говорит об определенной операции, которую можно производить в данный момент. Итак, разберемся по порядку: для перемещения объекта установите курсор поверх него,и когда рядом с указателем появится четырехнаправленная стрелка можно перемещать, только не перемещайте центральный круглый маркер — он предназначен для другой цели; для поворота объекта установите указатель за пределами любого углового маркера и потяните мышью.

Если при перетаскивании удерживать нажатой клавишу ***SHIFT***, то поворот будет происходить с шагом 45 градусов, а при нажатой клавише ***ALT*** объект будет вращаться вокруг противоположного маркера; для масштабирования объекта в двух измерениях (ширине и высоте) наведите указатель на угловой маркер и перетащите по диагонали, если при этом удерживать клавишу ***SHIFT***, то размеры будут изменяться пропорционально; для масштабирования только в одном направлении перетащите соответствующий маркер одной из сторон ограничивающей рамки; для наклона объекта поместите указатель на рамку между маркерами и потяните мышью.

Идем дальше. Для того, чтобы исказить форму объекта, наведите курсор на один из маркеров рамки и нажмите клавишу ***CTRL*** — вид указателя изменится, и теперь маркер можно перетаскивать в любом направлении, форма объекта будет при этом искажена. Если же навести указатель на угловой маркер и дополнительно нажать клавишу ***SHIFT*** (т.е. удерживать одновременно клавиши ***CTRL*** и ***SHIFT***), то объект можно как бы сузить или расширить, поскольку смежные углы переместятся на равных расстояниях в противоположных направлениях. Обратите также внимание на круглый маркер в центре рамки, который правильнее назвать трансформирующей точкой, так как все преобразования производятся относительно нее. Эту точку легко перетащить мышью в любое место рабочей области.



Теперь попробуем повернуть объект. Хорошо видно, что преобразования производятся относительно нового положения трансформирующей точки.Но и это еще не все возможности инструмента ***FREE TRANSFORM***. Посмотрите на раздел модификаторов панели инструментов. Здесь находятся 4 кнопки переключателя. Их назначение состоит в том, чтобы ограничить возможности инструмента конкретным видом преобразования. Например, если необходимо только отмасштабировать объект, то включите модификатор ***Scale*** — теперь мы сможем только изменять размеры объекта, возможности наклона, поворота или искажения формы будут отсутствовать. Тем самым мы предохраним себя от случайного ненужного преобразования. Включение кнопки ***Rotate and Scew*** обеспечит возможность поворота и наклона объекта, другие преобразования будут недоступны. При включенном модификаторе ***Distort*** можно будет только искажать форму объекта.

Ну и наконец модификатор ***Envelope*** (Огибающее) — он предоставляет совершенно новую уникальную возможность для преобразования формы объектов. При его активизации объект будет ограничен управляющими точками с касательными маркерами. Управляющие точки расположены по углам и в середине сторон рамки. Их допустимо перемещать совершенно произвольным образом. После этого форму огибающей можно изменить также с помощью, касательных маркеров, положение которых определяет направление изгиба и степень кривизны огибающей в данной точке. Короче говоря, модификатор ***Envelope*** дает практически абсолютную свободу редактирования выделенного объекта, которая может быть ограничена только

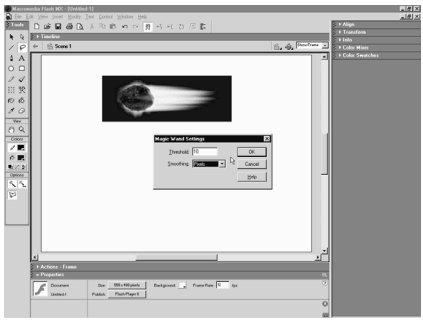
вашей фантазией. Все операции подключения модификаторов продублированы в основном меню. Откроем меню. ***Modify***, ***TRANSFORM***.

Команда ***FREE TRANSFORM*** активизирует одноименный инструмент, следующие 4 команды включат соответствующие модификаторы этого инструмента. Команда ***Scale and Rotate*** имеет некоторую специфику. При ее выполнении появится диалоговое окно, в котором в цифрах можно задать степень масштабирования и угол поворота выделенного объекта, например, если нужно уменьшить объект в два раза и одновременно повернуть его на 20 градусов по часовой стрелке, то введем в верхнее поле 50%, а в нижнее цифру 20. для поворота против часовой стрелки следует ввести отрицательное число.

Щелкнем по кнопке ОК и посмотрим результат. Вернемся к меню ***Modify*** и снова откроем подменю ***TRANSFORM***. Здесь имеются также команды для поворота объекта на 90 градусов по часовой стрелке и против часовой стрелки. Команды ***FLIP VERTICAL*** и ***FLIP GORIZONTAL*** создают соответственно вертикальное и горизонтальное зеркальные отображения объекта.

Весьма полезные команды для создания симметричных фигур. Ну и наконец команда ***REMOVE TRANSFORM*** отменяет последнее преобразование. Есть и другой, более быстрый, доступ к некоторым из перечисленных мною команд. Для этого щелкните правой кнопкой мыши на объекте, но не на пустом месте, и в третьем сверху разделе контекстного меню вы найдете уже знакомые опции. В заключение хотел бы отметить, что инструмент ***FREE TRANSFORM*** и его модификаторы не исчерпывают всех возможностей редактора по преобразованию объекта. для этого могут также применяться вспомогательные панели ***INFO*** и ***TRANSFORM***, о них я расскажу на отдельном уроке. Не знаю, как вы, но я слегка утомился перечислять бесчисленные возможности ***Flash*** по преобразованию объектов, поэтому объявляется перекур.

5.1.1.10. Инструмент "Лассо"



Уважаемые слушатели, на этом уроке мы продолжим знакомиться с панелью инструментов редактора ***Flash MX*** и научимся работать с инструментами Лассо, ***PEN*** (Перо) и ***SUBSELECTION*** (Частичное выделение).Инструмент Лассо является средством выделения и весьма похож на аналогичный инструмент выделения программы ***FOTOSHOP***. В отличие от инструмента ***Arrow*** он позволяет выделять области произвольной и неправильной формы и разрозненные участки вашего изображения, т.е. возможности по выделению у него

значительно выше, чем у инструмента Стрелка. Инструмент Лассо может работать как с векторной, так и с растровой графикой.

Для демонстрации его возможностей нарисуем что-нибудь или импортируем какой-либо рисунок из файла. Воспользуемся командой меню ***FILE***, ***IMPORT*** и выберем файл доступного графического формата. Пусть это будет растровый рисунок формата ***.JPEG***. После импорта изображение представляет собой единый объект, который можно лишь перемещать, вращать и масштабировать. Прежде чем работать инструментами выделения, его необходимо преобразовать в растровую заливку с помощью команды ***Modify — Break Apart*** (Разбить).

Теперь объект доступен для редактирования и выделения редакторами ***Flash*** и в частности инструментом Лассо. Пользоваться этим инструментом достаточно просто. Щелкнем на полотне и удерживая кнопку мыши выделим часть рисунка. Закончим выделение отпусканием кнопки мыши. После этого выделенный участок может быть перемещен, масштабирован и т.д., т.е. его можно использовать самостоятельно. С помощью кнопки Отменить вернем рисунок в исходное состояние. Если необходимо выделить одновременно несколько областей, то перед началом выделения нажмите и удерживайте клавишу ***SHIFT***, вот таким образом. Теперь обратите внимание на кнопки, которые появляются в разделе ***Options*** при выборе инструмента Лассо.

Модификатор ***Polygon Mode*** включает режим многоугольного выделения, при этом каждый щелчок мыши в рабочей области ролика порождает вершину будущего многоугольного выделения. Чтобы завершить выделение следует замкнуть многоугольник, выполнив двойной щелчок мышью. Если перед нами стоит конкретная задача аккуратно выделить нарисованный объект сложной формы, то использовать показанные выше методы весьма затруднительно. Можно, конечно, попробовать выделить нарисованный болид вручную, но при этом придется изрядно потрудиться и затратить значительное время. К счастью, инструмент Лассо имеет еще один модификатор, который может существенно облегчить решение поставленной задачи — это кнопка ***Magic Wand*** (Волшебная палочка).

Этот режим обычно встречается в редакторах растровой графики и предназначен для выделения по цвету. Другими словами, Волшебная палочка выделяет области одинакового цвета на растровом изображении. Выберем модификатор и подведем указатель к рисунку. Вид указателя изменится. Если теперь щелкнуть внутри изображения, то произойдет выделение пикселлов нужного цвета. Поскольку наш объект находится на фоне однородного черного фона, то в данном случае целесообразно выделить именно фон, щелкнув по нему.

Все черные пикселлы фона будут выделены. Теперь преспокойненько нажмем клавишу ***DELETE*** и на экране останется наш объект, аккуратно отделенный от фона. Если вас устраивает качество выделения, то теперь объект желательно сгруппировать, предварительно выделив его инструментом Стрелка и применив команду ***Modify Group***. Тем самым мы предохраним рисунок от случайного редактирования при наложении другой графики. О том, что объект сгруппирован, говорит голубой прямоугольник, который появится вокруг него при выделении.

Теперь можно создать новый фон и смело водрузить на него рисунок, не беспокоясь о нежелательном редактировании, например, вот так. Третья кнопка раздела модификаторов инструмента Лассо вызывает окно диалога Настройки параметров Волшебной палочки. Свойство ***Treshold*** (Порог) определяет ширину диапазона цветов, которые выделяются инструментом. Значение параметра находится в пределах от 0 до 200, более высокое число включает более широкий диапазон цветов. Если вы введете 0, то будут выбраны пикселлы только одного цвета. В поле ***Smoothing*** выбирается степень сглаживания границы выделения.

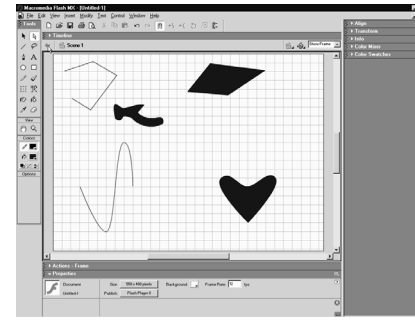
Пункт ***Smooth*** (Гладкая) означает сильное сглаживание, ***NORMAL*** — обычное, ***Rough*** — грубое сглаживание, и, наконец, пункту ***Pixels*** соответствует полное отсутствие сглаживания, т.е. сохраняется исходный пиксельный рисунок цветовой границы. Давать вам практические рекомендации по установке указанных параметров дело неблагодарное, все зависит от конкретной цели и рисунка, с которым идет работа. имея общее представление о назначении перечисленных опций вы всегда добьетесь нужного результата.

Заканчивая разговор об инструменте Лассо еще раз подчеркну, что режим Волшебная палочка применим только к растровым рисункам, для векторной графики придется ограничиться обычным режимом или режимом многоугольного выделения. Собственно говоря, именно по этой причине я демонстрировал возможности Волшебной палочки на импортированном растровом рисунке.

5.1.1.11. Инструменты "Перо", "Частичное выделение"

А теперь очистим экран и займемся изучением инструментов ***PEN*** (Перо) и ***SUBSELECTION*** (частичное выделение). Эти инструменты появились в пятой версии ***Flash*** и предназначены, главным образом, для рисования и редактирования контуров сложной формы с помощью кривых Безье. Их можно также использовать для редактирования контуров и заливок, созданных другими инструментами.

Для пользователей, кто уже имеет опыт работы в программах векторной графики, таких, как ***MACROMEDIA FREEHAND***, ***ADOBE ILLUSTRATOR***, ***CORAL DROW***, применение этих инструментов не вызовет затруднений. Инструмент ***PEN*** используется для точного рисования сложных траекторий, состоящих из прямых линий и сглаженных кривых, а инструмент ***SUBSELECTION*** применяется только для редактирования формы уже нарисованных контуров и заливок. Посмотрим, как они работают. Выберем инструмент ***PEN***. Курсор примет вид пера. Рисовать пером прямые линии очень просто. Для этого достаточно щелкнуть в первой точке, затем во второй и т.д. Для того, чтобы замкнуть многоугольник. Подведите указатель к начальной точке и когда рядом со значком пера появится петля щелкните, при этом замкнутая фигура автоматически закрасится установленным в разделе ***COLORS*** цветом заливки. Если вы хотите нарисовать незамкнутый контур, то дважды щелкните по последней точке или выберите другой инструмент, можно также нажать клавишу ***ESC***. А сейчас попробуем изобразить более сложную фигуру с использованием кривых Безье. Прежде чем рисовать настоятельно рекомендую сделать одну полезную настройку. Откроем меню ***EDIT*** и выполним команду ***PREFERENS***



Предпочтение. Перейдем во вкладку ***Editing*** и установим флажок на пункте ***Show Pen Preview*** (Предварительный просмотр линии). Подтвердим выбор. Для удобства рисования отобразим также вспомогательную сетку. Это можно сделать с помощью меню ***VIEW*** или же через контекстное меню, щелкнув правой кнопкой мыши и выполнив команду ***Grid***, ***Show Grid*** (Показать сетку). Кстати говоря, параметры сетки можно настраивать по своему вкусу. О том, как это делается, будет рассказано на следующих уроках. Щелкнем мышью на полотне для обозначения первого узла контура и переместим указатель. Вы видите, что теперь за указателем тянется линия, хотя кнопка мыши еще не нажата. Это происходит благодаря сделанной установке. Щелкнем в другом месте и не отпуская клавишу мыши выполним перетаскивание, формируя вид кривой с помощью исходящей от узла касательной линии. На концах касательной находятся маркеры кривых Безье.

Их расположение относительно узла кривой определяет направление изгиба и радиус кривизны контура в данном узле. Отпустим клавишу мыши и проделаем ту же операцию с третьим узлом контура, четвертым и, наконец, замкнем кривую, щелкнув мышью на первом узле. Приведем еще один пример

и нарисуем незамкнутую кривую, похожую на синусоиду. Щелкнем на полотне для получения первого узла, затем второго и сформируем в этом узле в помощью маркеров Безье одну из вершин синусоиды. Проделаем такие же операции с еще несколькими вершинами и закончим рисование двойным щелчком мыши. С помощью Пера можно также удалить существующий или добавить новый узел. Для удаления подведите указатель к узлу. При этом рядом с указателем появится маленький минус.

Теперь, если произвести щелчок, узел будет удален и вид контура изменится. Для добавления нового узла наведите инструмент на любое место кривой, но не на узел. Рядом с указателем появится знак "+". Это означает, что щелчком мыши в этом месте можно добавить новый узел. Правда, здесь есть ограничение: новый узел можно добавить только на участок сглаженной кривой, на отрезке прямой линии этого сделать нельзя. Теперь мы имеем возможность отредактировать нарисованную кривую с помощью других инструментов. Выберем инструмент ***SUBSELECTION*** и разберем его возможности для редактирования контуров. Щелкнув инструментом на любой кривой мы увидим ее узлы, обозначенные квадратиками.

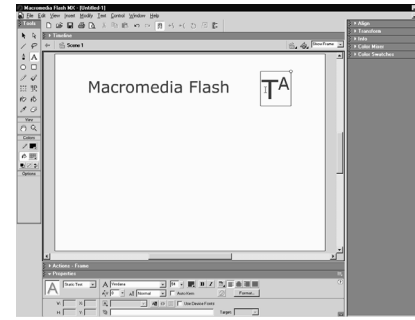
Выделив любой узел, можно его перемещать. Для более точного перемещения выделенного узла желательно использовать клавиши со стрелками на клавиатуре. Имеется также возможность изменять радиус кривизны и направление изгиба кривой в данном узле, используя маркеры Безье, например, вот таким образом. При перемещении одного из маркеров автоматически корректируется и другой так, что их противоположная направленность сохраняется. Но если необходимо расположить маркеры под углом друг к другу, то во время этой операции удерживайте нажатой клавишу ***ALT***. Это позволит расположить векторы искривления под произвольным углом. На контуре при этом образуется излом.

Выделив инструментом какой-либо узел и нажав клавишу ***DELETE*** можно этот узел удалить. Кроме того. Инструмент ***SUBSELECTION*** позволяет перемещать всю кривую целиком, потянув за нее в любом месте, где нет узла. Как я уже отмечал, этим инструментом можно редактировать и заливки. Выберем инструмент Кисть, сделаем мазок на полотне, а затем инструментом ***SUBSELECTION*** щелкнем по контуру заливки — вы увидите узлы, которые можно редактировать показанными ранее способами. Если вы никогда раньше не работали с подобными инструментами в других программах векторной графики и у вас на первых порах ничего не получается, не впадайте в отчаяние и не пытайтесь задушить свой компьютер сетевым шнуром. Применение рассмотренных инструментов требует определенных практических навыков, которые у вас скоро обязательно появятся. Итак, в рамках первой главы нам с вами осталось рассмотреть только один инструмент, а именно: инструмент

Текст. Поскольку это достаточно большая тема, то сейчас сделаем перерыв и продолжим изучение панели инструментов на следующем уроке.

5.1.1.12. Инструмент "Текст"

Уважаемые слушатели, на этом уроке мы завершаем изучение панели инструментов редактора ***Flash MX*** и рассмотрим принципы использования инструмента Текст, предназначенного для размещения текстовой информации внутри ролика. Следует сразу заметить, что программа ***Flash*** не предназначена для использования больших объемов текста. наличие значительного количества кривых в шрифтах может неоправданно увеличить размер файла фильма и, разумеется, приведет к проблемам при его размещении в сети.



Но, с другой стороны, если вы посмотрите на ***Web***-страницы, созданные с помощью ***Flash***, практически на каждой из них в том или ином виде будет присутствовать текст. Он незаменим в надписях меню, на кнопках, полях ввода, при использовании сппециальных эффектов и в ряде других случаев. Разумное применение текста во ***Flash***-роликах позволяет создавать по настоящему интерактивные интернет-приложения. Инструмент Текст используется совместно с панелью ***Properties***, на которой после его выбора появляется множество опций для настройки текстовых полей. При щелчке в рабочей области при выбранном инструменте

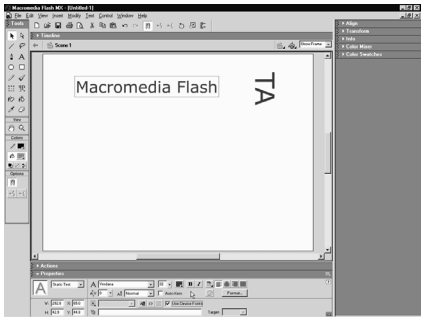
Текст будет создано текстовое поле, представляющее собой прямоугольную область. В редакторе существует 3 типа текстовых полей, определяемых на панели ***Properties*** во вкладке ***Text Tipe***: статические, динамические и поля ввода. Статические поля предназначены для отображения текста, который не изменяется во время проигрывания ролика, динамические поля отображают текст, который может изменяться по ходу демонстрации ролика (например, при щелчке пользователя по кнопке, перехода фильма в определенный кадр или других действиях), третий тип текстовых полей предназначен не столько для отображения какой-либо информации, сколько для ввода ее пользователем. На этом уроке мы подробно рассмотрим свойства статических текстовых полей, которые применяются чаще всего.

Давайте введем в наше поле пару слов, например, ***MACROMEDIA FLASH***. Текстовые поля в программе ***Flash*** могут иметь фиксированную ширину или быть расширяемыми, т.е. способными менять свою ширину в зависимости от содержимого. При создании текстового поля щелчком мыши оно изначально

оказывается расширяемым, о чем говорит круглый манипулятор в правом верхнем углу.

Если потянуть манипулятор мышью, то поле приобретает фиксированную ширину, и манипулятор в углу становится квадратным. Когда текстовый блок не укладывается в этот размер, будет происходить автоматический перенос слов, поле при этом будет расширяться вниз. Чтобы поле фиксированной ширины снова сделать расширяемым нужно дважды щелкнуть на его манипуляторе. Завершим ввод щелчком на инструменте Стрелка. Текст при этом будет выделен голубой рамкой. Теперь для изменения его параметров необходимо обратиться к панели ***Properties***. Список ***FONT*** позволяет изменить шрифт, выбрав любой из установленных на вашем компьютере, например, вот такой. Для изменения размеров шрифта введите нужное значение в поле ***FONT SIZE*** или сделайте это мышью, используя ползунок.

Правее расположено окно, щелкнув по которому мы получим возможность выбрать цвет текста. Помимо предложенных в палитре образцов, с помощью появившейся пипетки можно скопировать нужный цвет с любого элемента интерфейса программы. Давайте, например, возьмем образец с пиктограммы модификатора привязки, а затем отменим предыдущую операцию, щелкнув по кнопке Отменить. Правее окна палитры находятся кнопки, имеющиеся практически в любом текстовом редакторе и устанавливающие полужирное и курсивное начертание текста. В поле мы можем задать межбуквенный интервал вот таким образом. Это часто полезно делать в различных заголовках.



До сих пор я изменял параметры всего выделенного текста, но это можно также сделать для отдельных слов или символов. Нажмем клавишу T для быстрого выбора инструмента Текст и щелкнем внутри текстового поля. Затем нажмем левую кнопку мыши и выделим отдельные символы или слова. Двойной щелчок выделяет слово целиком. Теперь показанным ранее способом можно изменять начертание, размер и цвет только выделенных символов. Все очень похоже на работу в текстовом редакторе. Кроме того, с помощью панели ***M*** допустимо изменять прозрачность отдельных символов или всего текстового блока в целом, устанавливая нужное значение в поле ***Alfa***. Сделаем отмену последних операций. Идем дальше. Для демонстрации других опций создадим еще одно текстовое поле и введем туда две прописные буквы:

T и A. Выберем инструмент Стрелка и для наглядности немного увеличим размер шрифта. Список ***Corrector position*** (Положение символов) кроме варианта ***Normal*** (обычный) содержит два эффекта: ***Superscript*** (Верхний индекс) и ***Subccript*** (Нижний индекс). Выберем снова инструмент Текст, выделим одну букву в текстовом блоке и применим опцию верхнего индекса. Точно так же можно сделать и нижний индекс. Весьма полезная функция, особенно если вы применяете в тексте математические формулы. Отменим последнюю операцию. Включение флажка приводит к автоматической установке расстояния между отдельными буквами. Текст при этом выглядит более эстетично. Выберем инструмент Стрелка и выделим короткий текст.

Теперь пощелкаем в поле ***Kern***. Вы видите, как меняется расстояние между буквами. К сожалению, автоматический ***r***ернинг работает далеко не на всех шрифтах и практически не работает на кириллице, однако я не нахожу никакого резона снимать этот флажок, разве что в каких-то особых случаях. В редакторе ***Flash MX*** появилась новая кнопка, позволяющая размещать текст по вертикали. При щелчке по ней появляется меню из трех пунктов. По умолчанию установлено горизонтальное положение символов, но мы можем расположить символы и вертикально, выбрав другой пункт, при этом становится активной кнопка, позволяющая развернуть символы на 90 градусов, например, вот таким образом.

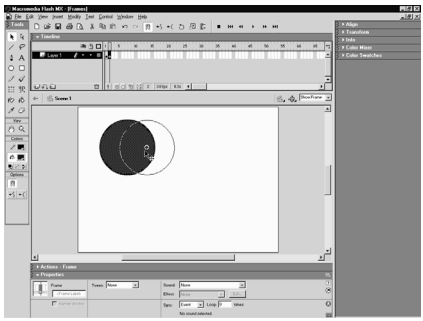
Если выделить какой-либо текстовый блок и включить кнопку ***Select table*** (выделяемый), то это дает возможность пользователю выделять текст вашего ролика для последующего копирования. Убедимся в этом, для чего выполним команду ***C***. Теперь мы можем выделить текст ролика и скопировать его в буфер обмена. Закроем тестовое окно. Обычно нет причин устанавливать опцию***Select table***, однако, если вы включили в свой ролик, например, большой кусок программного кода, и хотите распространить его среди пользователей, то можете облегчить им жизнь, включив вышеупомянутую кнопку.

Для дополнительного уменьшения размера файла фильма установите флажок ***USE DEVICE FONTS*** (использовать шрифты пользователя). При этом ***Flash*** будет сохранять в ролике не начертание букв, а только характеристики и название шрифта. На практике это всего несколько байт.

При воспроизведении будет использован указанный шрифт, если он окажется установлен в системе пользователя, в противном случае ***Flash*** использует ближайший по характеристикам шрифт, т.е. в этом случае нет гарантии, что текст на другом компьютере будет выглядеть так же, как на вашем экране. По моему мнению, устанавливать упомянутый флажок необходимо только в крайнем случае, когда ролик содержит большое количество текста — не стоит приносить в жертву дизайн фильма ради уменьшения размера файла.

5.1.2. Основы анимации во Flash, символы и экземпляры

5.1.2.1. Создание покадровой анимации, отображение кадров



Итак, на предыдущих занятиях мы с вами освоили основные приемы создания статической графики в редакторе ***Flash MX*** и познакомились с методами применения растровых изображений. Теперь на основе багажа полученных знаний, можно переходить к основной теме курса — Анимации. Именно, благодаря развитым и в тоже время интуитивно понятным средствам создания анимации, программа ***Flash*** и приобрела такую широкую популярность. Тема анимации весьма обширна. Трудно даже определить с чего начать. Самое разумное, наверное, идти от простого к сложному. Поэтому этот урок мы посвятим самому простому методу анимации — так называемой покадровой анимации.

Как известно смысл анимации заключается в быстром последовательном проигрывании кадров фильма, содержание которых меняется. Таким образом, создается иллюзия движения и т.п. Давайте посмотрим на шкалу времени или хронометрическую линейку ***TimeLine***, расположенную над рабочим полотном. В левой ее части находится перечень слоев, о которых мы поговорим поподробнее позже. Правую часть линейки занимает шкала кадров, где находятся пронумерованные кадры фильма.

Над шкалой кадров расположен маркер в виде красного прямоугольника, который находится над текущим кадром. Именно содержимое текущего кадра отображается в рабочей области, где мы до сих пор и проводили все эксперименты по рисованию. При запуске программы ***Flash*** создает новый ролик с одним единственным кадром под номером 1. Но мы можем добавлять в ролик практически любое количество кадров.

Чтобы двигаться дальше разберемся в двух терминах. Дело в том, что в программе ***Flash*** можно создавать кадры двух типов. Ключевые кадры ***KeyFrames*** и обычные кадры. Ключевые кадры независимы друг от друга и могут содержать любое изображение. Обычные кадры в общем случае содержат то же изображение, что и предыдущий ключевой кадр. Первый кадр ролика по умолчанию является ключевым.

Если в ключевом кадре нет никакой графики, как сейчас, то он обозначается небольшой окружностью. Если же кадр содержит какой-либо рисунок или объект, то он обозначится черным кружком. Нарисуем в рабочей области, например круг и убедимся в этом.

Теперь давайте создадим второй ключевой кадр и сделаем его пустым. Для этого выделим мышью вторую позицию шкалы. И выполним команду ***Insert — Blank Keyframe*** пустой ключевой кадр, то же самое можно сделать нажав клавишу ***F***7.

Вы увидите, что рабочая область стала пустой, а маркер переместился во второй кадр. Передвинув маркер обратно, можно снова посмотреть на то, что было нарисовано в первом кадре. Итак, в первом кадре у нас имеется изображение круга, а второй пуст. Вообще-то сейчас мы создали анимацию, хотя и самую примитивную. Если сейчас опубликовать ролик, то мы увидим в нем круг мигающий с частотой 12 кадров в секунду. К сожалению, формат наших видеоуроков не позволяет просматривать плавную анимацию с большой частотой кадров, поэтому для наглядности я сделаю на панели ***Properties*** скорость воспроизведения ролика равной 3 кадрам в секунду.

Теперь давайте опубликуем ролик, создав файл с расширением ****.swf***. Для этого выполним команду ***Control — Test Movie***. Или нажмем сочетание клавиш ***CTRL+Enter***. Вот вам пожалуйста мигающий круг. Закроем тестовое окно и вернемся к основной сцене. Выделим пустой ключевой кадр во второй позиции и удалим его. Выполнив команду ***Insert — Remove Frames*** или нажмем сочетание ***Shift+F***5. Снова выдели вторую позицию и проделаем несколько иную операцию. Выполним команду ***Insert-Keyframe*** вставить ключевой кадр или нажмем клавишу ***F***6. Теперь во втором кадре мы имеем точную копию первого кадра. Поскольку изображения в ключевых кадрах не зависят друг от друга, мы можем изменить рисунок во втором кадре. Например, передвинуть его.

Выделим теперь третью позицию шкалы и повторим операцию нажав клавишу ***F***6. Переместим также объект в третьем кадре. а теперь подвигаем маркером в пределах первых трех кадров. Вы видите, что объект движется по экрану. Таким образом, создается покадровая анимация. публиковать ролик не будем, я думаю и так все понятно. И наконец познакомимся еще с одной операцией с кадрами. Предварительно удалим третий и второй кадры, выделив их и нажав сочетание ***Shift F***5.

Выберем произвольную позицию на шкале, например десятую и выполним команду ***Insert-Frame*** вставить обычный кадр или нажмем клавишу ***F***5. Все пространство между первым и последним десятым заполнится обычными кадрами. Содержание обычных кадров дублируют содержание

предшествующего. В нашем случае первого ключевого кадра. Не пустые обычные кадры закрашиваются серым, а пустые остаются белыми.

Заключительный кадр в цепочке обычных кадров обозачается белым прямоугольником. Операцию по вставке обычных кадров полезно делать в тех слоях, где объекты не изменяются в течении анимации, например для фона. Можно, конечно, копировать и ключевые кадры, не меняя содержание в них, но большое количество ключевых кадров в ролике заметно увеличивает его размер. Итак настоятельно рекомендую запомнить три клавиши, которыми вы будете очень часто пользоваться. Клавиша ***F***7 вставляет в выделенную позицию пустой ключевой кадр. Клавиша ***F***6 создает копию предыдущего ключевого кадра. А клавиша ***F***5 заполняет шкалу обычными кадрами, в которых отображается содержание предшествующего ключевого кадра.

А теперь в целях закрепления пройденного материала, создадим простую покадровую анимацию, которая иллюстрирует растущую Луну. Удерживая клавишу ***Shift***, выделим все кадры со второго по 10 и удалим их комбинацией ***Shift***+***F***5. Перейдем в первый кадр. И удалим из него всю графику. На панели ***Properties*** установим синий цвет фона. Выделим вторую позицию шкалы и клавишей ***F***7 вставим туда пустой ключевой кадр. Выберем инструмент Овал. Определим светло-желтый цвет заливки. И нарисуем круг, котрый будет являться заготовкой нашей Луны. Нажмаем клавишу ***V***, для быстрого выбора инструмента Стрелка. И выделим объект двойным щелчком.

На панели ***Properties*** или с помощью панели ***Info*** установим диаметр фигуры, равный 100 пикселам. Создадим еще 5 копий второго ключевого кадра. Выделим третью позицию на шкале и нажмем клавишу ***F***6 — эта операция скопирует предыдущий ключевой кадр в выделенную позицию. Аналогичным образом вставим ключевые кадры в позиции, начиная с 4-ой по 6-ую. Таким образом, мы получили один пустой и пять ключевых кадров с одинаковым изображением. Передвинем маркер во второй кадр и выделим только контур фигуры. Затем переместим контур на 20 пикселов вправо. Это удобно делать с помощью клавиатуры. Удерживая клавишу ***Shift***, дважды нажмем на клавишу со стрелкой вправо. Каждое нажатие переместит выделенный объект на 10 пикселов в соответствующем направлении. Без клавиши ***Shift*** контур перемещался бы на один пиксел. Теперь снимем выделение. Вы уже знаете, что при этом контур разрежет нашу заливку на две части. Выделим весь контур и правую часть заливки, и удалим их.

В результате, мы получили заливку в виде узкого желтого полумесяца. Проделаем похожие операции в других ключевых кадрах. В третьем кадре выделим и переместим контур на 40 пикселов. После снятия выделения также удалим лишнюю часть графики. В четвертом кадре переместим контур на 60 пикселов. И затем проделаем те же операции. В 5 кадре сместим контур на 80

пикселов и повторим процедуру. Шестой кадр оставим практически без изменения. Удалим лишь контур фигуры. Теперь протестируем ролик, нажав ***CTRL+Enter***. Как видите, мы получили покадровую анимацию лунных фаз. Конечно, полученный ролик можно до бесконечности совершенствовать.

Так для получения более плавной анимации можно было бы смещать режущий контур не на 20 пикселов, а скажем на 10 или еще меньше. Но для этого нужно было бы добавлять еще ключевые кадры, что отняло бы слишком много времени. Для нас сейчас самое главное понять принцип покадровой анимации. Давайте сначала научимся пользоваться удочкой, а крупную рыбу попытайтесь ловить самостоятельно. А теперь приведем характерный пример, когда целесообразно использовать обычные, а не ключевые кадры. Напр. вы захотели, чтобы анимация происходила на фоне какого-либо статического пейзажа.

Вставлять этот пейзаж в каждый ключевой кадр, абсолютно неразумно. Это и трудоемко и заметно увеличит размер файла. Гораздо лучше воспользоваться другим слоем с обычными кадрами. Создадим новый слой, щелкнув по кнопке со знаком +. Программа автоматически создаст в нем 6 кадров для выравнивания слоев по длительности. Первый кадр нового слоя — ключевой, а остальные простые, которые повторяют содержимое ключевого кадра. Перетащим новый слой мышью в нижнюю часть списка.

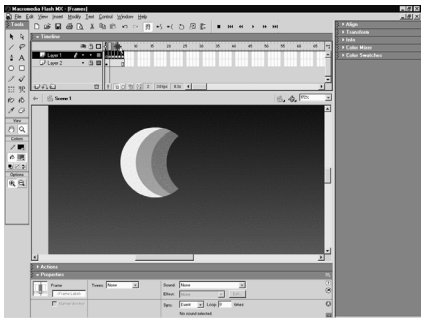
Теперь здесь можно создавать любой фон, который будет отображаться во всех кадрах анимации. Не мудрствую лукаво, изобразим закатное небо. Выберем инструмент прямоугольник и нарисуем соответствующую фигуру. Выделим и удалим его контур. Затем выберем инструмент Заливка и откроем панель ***Mixer***. Установим линейный градиент и настроим переход цвета от светло-розового до темно-синего. Заполнм фигуру полученной заливкой. И повернем градиент таким образом, чтобы розовый цвет находился внизу. Эту операцию можно сделать и с помощью инструмента ***Fill Transform***.

Затем выделим и с помощью модификатора ***Scale***, отмасштабируем полученный прямоугольник таким образом, чтобы он занимал всю рабочую область. При желании здесь же можно нарисовать любую фигуру, здания, деревья, короче все, что душе угодно. Протестируем полученный фильм. Заканчивая разговор о покадровой анимации, следует отметить, что этот вид анимации — процесс весьма трудоемкий, поэтому применяется обычно в тех случаях, когда он действительно необходим. Наприме, если нужно отобразить много быстрых мелких движений, в частности, анимацию мимики лица, игру на пианино или томку подобное. В большинстве случаев в программе ***Flash*** используются другие методы анимации: анимация формы или морфинг и анимация движения, о которых речь пойдет на следующих уроках.

В заключении урока познакомимся с кнопками, расположенными под шкалой кадров, которые могут помочь нам при создании анимации. До сих пор мы предполагали, что в рабочей области отображается только текущий кадр. В большинстве случаев это действительно так, но иногда необходимо посмотреть сразу несколько кадров анимации.

Вы можете сделать это щелкнув по кнопке ***Onion Skin***. Дословно луковая кожура, но мы лучше назовем эту кнопку шлейфом. При этом вокруг маркера появятся границы шлейфа, имеющие вид квадратных скобок. Все кадры внутри скобок отобразятся в области рисования. Текущий кадр будет показан четко, а остальные выглядят полупрозрачными. Для наглядности выберем инструмент ***Zoom*** и увеличим масштаб. Теперь эффект хорошо заметен. Скобки, ограничивающие шлейф, можно перемещать мышью. И таким образом настраивать область отображения.

При перемещении маркера, перемещаются и границы шлейфа, оставаясь на том же расстоянии от маркера. Щелчок по кнопке ***Onion Skin Outlines*** — контурный шлейф, дает похожий эффект, только все кадры кроме текущего будут отображаться в контурном виде. Это уменьшает нагрузку на компьютер и увеличивает скорость работы редактора на слабых машинах. Кнопка ***Edit Multiple Frames***, включает режим одновременного редактирования нескольких кадров.



Такой режим удобен при работе с покадровой анимацией, когда необходимо модифицировать изображение сразу в нескольких соседних кадрах. Например, после создания ролика вы вдруг захотели изменить размер анимированной фигуры, в данном случае Луны. В обычном случае пришлось бы в каждом ключевом кадре производить масштабирование фигуры и при этом следить, чтобы коэффициент изменения масштаба был одинаков. Упомянутая кнопка поможет избавиться от этой рутинной работы.

Включим режим ***Edit Multiple Frames***и настроим границы шлейфа так, чтобы они охватывали все кадры анимации. Перетаскиванием инструмента ***Arrow***, произведем выделение объекта. При этом проследим, чтобы черным цветом были отмечены все кадры, которые мы хотим одновременно редактировать. Затем выберем модификатор ***Scale***.

И самым обычным способом отмасштабируем фигуру. Аналогичным образом можно одновременно изменить координаты объекта во всех кадрах. Цвет и т.д. Протестируем ролик. Как видите, никаких недоразумений не

произошло и синхронность анимации не нарушена. Закроем тестовое окно и выключим режим ***Edit Multiple Frames***. Щелчок по самой правой кнопке данного ряда вызывает меню настроек шлейфа. Его нижняя секция позволяет устанавливать определенное расстояние от маркера до границы шлейфа. Так выбор пункта ***Onion*** 2. Определит расстояние в 2 кадра от маркера до границы шлейфа. А ***Onion All*** распространит шлейф на все созданные кадры. В этом меню имеется также установка, позволяющая показывать маркеры шлейфа всегда, даже когда режим отображения шлейфа отключен.

И установка закрепляющая границы шлейфа. И, наконец, самая левая кнопка этого ряда ***Center Frame*** — центрировать кадр, будет полезна только в том случае, если вы работает с анимацией содержащей большое количество кадров, которые не помещаются в окно шкалы времени. Ну, например, переместим слайдер шкалы и отметим 120 позицию. Клавишей ***F***5 вставим туда обычный кадр. Снова переместим слайдер в исходное положение. Теперь для того, чтобы увидеть текущий кадр с маркером достаточно щелкнуть по вышеупомянутой кнопке. Тем самым вы даете команду редактору передвинуть слайдер так, чтобы текущий кадр находился в середине шкалы. На этом закончим урок и сделаем перерыв.

5.1.2.2. Раскадровка формы (морфинг)

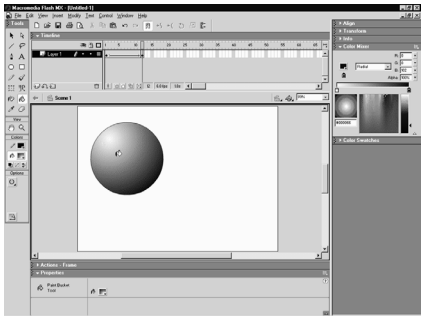
Рассмотрев основные принципы покадровой анимации, можно переходить к так называемой промежуточной анимации — или раскадровке. Смысл этого названия заключается в следующем: вместо того, чтобы последовательно рисовать все кадры подряд, пользователь рисует только первый и последний кадр анимации, а все промежуточные рассчитывает программа.

В редакторе ***Flash*** существуют два вида промежуточной анимации — это анимация формы и анимация движения. На этом уроке мы рассмотрим анимацию формы, или как принято ее называть морфинг, которая позволяет плавно изменять форму объектов. Кому совсем незнаком термин морфинг советую вспомнить или пересмотреть еще раз фильм Терминатор 2 Ссудный день, где многие спецэффекты построены на этом принципе. А теперь на простых примерах начнем разбираться как создается такой вид анимации. Для начала рассмотрим простейший пример, в котором круг плавно превращается в квадрат. Нарисуем в первом кадре круг. Выделим, например, 12 позицию на шкале времени и нажмем клавишу ***F***7, вставив туда пустой ключевой кадр. Теперь выберем инструмент прямоугольник. И изобразим в этом кадре квадрат. снова выделим первый ключевой кадр. И посмотрим на панель ***Properties***. Сейчас нас здесь интересует раскрывающийся список ***Tween*** — тип анимации, в котором по умолчанию выбран пункт ***None*** — нет.

Ранее в предыдущих версиях редактора, для создания промежуточной анимации использовалась отдельная вспомогательная панель ***Frame*** — кадр.

Морфинг в программе ***Flash*** называют анимацией формы. Поэтому выберем в отмеченном списке пункт ***Shape*** — форма. Если все сделано правильно, то на шкале времени вся цепочка обычных кадров окрасится в салатовый цвет и по ней пройдет сплошная стрелка, указывающая на следующий ключевой кадр.

Теперь, нажав клавишу ***CTRL+Enter***, можно посмотреть как круг плавно превращается в квадрат. Закроем тестовое окно. Перетаскивая маркер по шкале времени можно рассмотреть все промежуточные кадры анимации, не тестируя ролик. Правда, изменить их программа не позволит, поскольку рисовать можно только в ключевых кадрах. Сразу хочу предупредить о типичной ошибке, которую часто допускают начинающие пользователи при создании анимации формы. Хорошенько запомните, что морфинг принципиально невозможен при работе с группами, символами или текстом.



В обоих ключевых кадрах должна находится только обычная несгруппированная векторная графика, а именно линии и заливки. Сгруппируем, например, фигуру в первом кадре, выделив ее и выполнив команду ***Modify-Group***. Теперь можно убедиться, что анимация формы пропала. Если некорректный объект будет находиться в последнем кадре, то эффект будет тот же самый. Поэтому если у вас морфинг не получается проверьте прежде всего какие объекты находятся в ключевых кадрах анимации. Разгруппируем объект в первом кадре комбинацией ***CTRL+Shift+G*** и немного усложним анимацию. Выделим последний кадр. Передвинем прямоугольник. Изменим его масштаб. А также цвет заливки.

Протестируем ролик, нажав ***CTRL+Enter***. Как видите, анимация идет нормально. Объект перемещается, плавно изменяя свою форму и цвет. Закроем тестовое окно и посмотри какие настройки допускает анимация формы. Вернемся в первый кадр. И обратим внимание на еще два списка на панели ***Properties*** ниже параметра ***Tween***. Раскрывающийся список ***Blend*** — перетекание, отвечает за гладкость промежуточных форм, получающихся при морфинге.

При выборе пункта ***Distributive*** — распределенная, формы получаются более сглаженными, а при выборе пункта ***Angular*** — более угловатые. Поле ***Easing*** управляет равномерностью анимации. Сюда можно ввести любое число от –100 до +100. При нулевом значении превращение одной фигуры в другую происходит равномерно. при отрицательных значениях анимация начинается

как бы постепенно, разгоняясь к концу. Давайте введем максимальное отрицательное число -100 и опубликуем ролик. Заметно, что анимация идет с ускорением. Закроем тестовое окно и введем в это поле максимальное положительное число. Т.е. +100. И снова протестируем ролик. Теперь наблюдается противоположный эффект. Анимация к концу замедляется. С помощью этой настройки удобно создавать такие эффекты как движение с ускорением, торможение и т.д. А теперь поэкспериментируем немного с заливками. Выделим последний кадр.

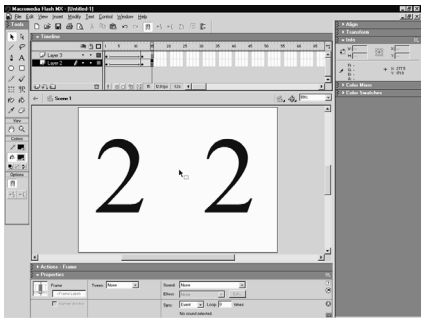
Обратимся к панели ***Mixer*** и установим линейную градиентную заливку. Настроим, например, красно-белый переход. Затем выполним команду ***Control-Test Movie*** и посмотрим, что получилось. Анимация идет корректно. Сплошная заливка плавно перетекает в градиентную. Закроем тестовое окно. Если у нас в обоих кадрах градиентная заливка, то и в этом случае все будет работать нормально. Единственно, что редактор в этом плане не умеет делать, так это плавно превращать линейный градиент в круговой и наоборот. Поэтому следите, чтобы в начальном и конечных кадрах был одинаковый тип градиентной заливки, либо линейная, либо радиальная. Используя свойства плавного перетекания градиентных заливок, можно получить ряд интересных эффектов. Создадим, например, иллюзию перемещения источника освещения объекта. Перейдем к первому кадру, выберем инструмент Заливка, на панели ***Mixer*** установим круговой градиент, и настроим переход от белого цвета к темно-синему.

Затем щелкнем по кругу так, чтобы центр градиента находился где-нибудь слева вверху. Скопируем объект в буфер, выделив его инструментом ***Arrow*** и кликнув по кнопке ***Copy***. Теперь перейдем в последний кадр, удалим прямоугольник. И вставим из буфера скопированный объект командой ***Edit - Paste in Place*** вставить на место. При этом вставка произойдет точно в те же координаты, откуда объект был скопирован. Выберем инструмент ***Paint Bucket*** и сместим центр круговой заливки в противоположную сторону. Протестируем ролик и посмотрим результат. Центр градиента плавно смщается и создается впечатление перемещения источника освещения. Похожим способом несложно сделать также пульсирующие градиенты, которые хорошо подходят для изображения фонарей, мигалок, светильников, звезд и т.п. Раскрепостите свою фантазию и успех будет вам обеспечен.

Поработав с морфингом фигур сложной формы, вы непременно заметите, что иногда превращение фигур в друг друга происходит не совсем так, как этого хотелось бы. Для подобных случаев в программе предусмотрено дополнительное средство управления анимацией формы. Приведем простой пример. Предположим, нам нужно сделать превращение цифры 1 в цифру 2. Создадим новый слой, используя кнопку со знаком +, затем перейдем в

нижний слой и удалим его, щелкнув по значку Корзины. Выберем инструмент Текст, щелкнем на полотне и введем цифру 1. Убедимся, что мы работем со статическим текстом. Выберем инструмент Стрелка и на панели ***Proprties*** увеличим для наглядности размер шрифта примерно до трехсот пунктов. Отметим на шкале 12 позицию и скопируем туда ключевой кадр, нажав клавищу ***F***6. В новом ключевом кадре, с помощью инструмента Текст, заменим цифру 1 на цифру 2. Снова выберем инструмент ***Arrow*** и выполним команду ***Modify-Break Apart*** — Разбить или нажмем комбинацию ***CTRL+B***. Тем самым мы преобразовали текстовый символ в обычную графику. В противном случае анимация формы у нас работать не будет. Перейдем к первому кадру и проделаем ту же операцию с цифрой 1. На панели ***Properties*** назначим анимацию формы.

Вроде бы анимация готова. Нажмем ***CTRL+Enter*** и увидим, что анимация происходит, но фигура при этом будто выворачивается наизнанку, чуть ли не исчезая в некоторых кадрах. Как же получить более убедительное превращение? Для таких случаев в программе предусмотрен механизм меток морфинга. Покажем как он реализуется. Оставим сделанную анимацию без изменений для того, чтобы иметь возможность сравнивать. Создадим новый слой, щелкнув по кнопке с + внизу списка слоев. Программа автоматически добавит в новом слое 12 пустых кадров для выравнивания слоев по длительности. Удерживая клавишу ***Shift***, выделим все кадры нижнего слоя с анимацией, щелкнем правой кнопкой мыши и в появившемся контекстном меню выберем команду ***Copy Frames***, скопировав кадры в буфер обмена.



Выделим теперь первый кадр верхнего слоя, отобразим снова контекстное меню и выполним команду ***Paste Frames*** — вставить кадры. Кстати говоря, операции по копированию и вставке кадров можно было также выполнить с помощью соответствующих команд меню ***Edit***. Уберем лишние кадры из верхнего слоя. Для чего выделим их, удерживая клавишу ***Shift***, и выполним команду ***Insert Remove Frames*** или нажмем комбинацию ***Shift+F***5. Теперь мы получили две абсолютно одинаковые анимации, расположенные друг над другом в разных слоях. Чтобы посмотреть их отдельно, давайте сместим объекты верхнего слоя вправо. Выделим первый кадр и обратимся к панели ***Info***. В поле, где отображаются координаты объекта на оси ***X***, увелим значение, скажем, на 300 пикселов. Добавим тройку впереди текущих цифр и нажмем клавишу ***Enter***. Затем выделим последний кадр и выполним такие же действия.

Теперь обе анимации можно увидеть одновременно. Выделим снова первый кадр верхнего слоя и выполним команду ***Modify-Shape-Add Shape Hint*** добавить метку формы. При этом где-то в середине фигуры появится красный кружок, помеченный маленькой буквой. Если у вас на экране ничего не появилось, то откройте меню ***View*** и установите флажок ***Show shape Hints*** — показывать метки формы. Переместим метку в какую-нибудь характерную точку рисунка, проследим, чтобы она легла на край заливки или на контур, иначе программа нас не поймет. Затем перейдем в заключительный кадр и установим аналогичную метку в точку, куда, по вашему мнению, должна перейти метка в первом кадре. Если все сделано корректно, то кружок станет зеленым, а в первом кадре — желтым. В данном случае одной метки морфинга недостаточно. Для удовлетворительного результата установим еще две метки. Теперь это можно сделать следующим образом. Установим курсор на первой метке и щелкнем правой кнопкой мыши.

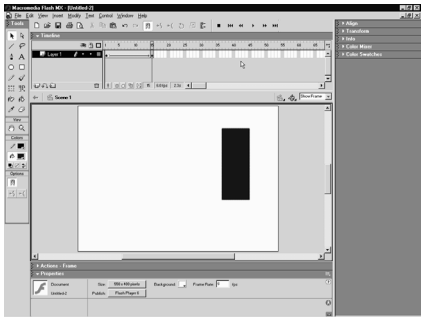
В контекстном меню выберем команду ***Add Hint*** — добавить метку. Переместим вторую метку в другую характерную точку. Перейдем в заключительный кадр и выполним аналогичное действие. И, наконец, проделаем похожие операции по установке третьей метки морфинга. Для улучшения визуального восприятия введем небольшую задержку в заключительный кадр анимации. Для этого выделим 15 позицию на шкале и нажмем клавишу ***F***5. То же самое проделаем в нижнем слое. Теперь посмотрим результат нашей работы, нажав клавиши ***CTRL+Enter***. Вы видите, что анимация с использованием меток морфинга происходит более естественно.

В заключении этого урока хотел бы дать несколько общих рекомендаций, касающихся анимации формы. Если вам нужно одновременно показать морфинг нескольких объектов, то используйте для этого отдельные слои. В противном случае, программа вряд ли вас поймет и выдаст полную абракадабру. Метки морфинга ставьте последовательно, по принципу: добавил, переместил в первом кадре, затем переместил в заключительном кадре. После этого переходите к установке следующей метки. Кроме того, специалисты ***Macromedia*** рекомендуют устанавливать метки последовательно по контуру фигуры против часовой стрелки. Не увлекаетесь установкой большого количества меток морфинга. Этим вы сможете полностью заморочить редактору голову и результат будет непредсказуем. И, наконец, при морфинге фигур сложной формы, не старайтесь сразу сделать анимацию исходного изображения в конечное, разбейте анимацию на два три этапа. Для этого вставьте дополнительные ключевые кадры между первым и последним и нарисуйте там промежуточные формы, какими вы хотели бы их видеть. Тогда мешанина форм в середине анимации будет не столь заметной. Вот, пожалуй, и все что я хотел вам рассказать на этом уроке.

5.1.2.3. Раскадровка движения

На предыдущем уроке мы освоили один из методов промежуточной анимации, а именно, анимацию формы или морфинг. Другим распространненым видом промежуточной анимации, применяемом во ***Flash***, является анимация движения. В отличие от морфинга, котрый работает с обычной векторной графикой, в ключевых кадрах анимации движения должны находится так называемые символы, которые представляют собой особый самостоятельный вид ***Flash***-объектов.

Можно, правда, использовать и сгруппированные объекты, но в этом случае программа сама автоматически конвертирует их в символы. Символы и особенности их применения — это весьма большая и важная тема раскрывать, которую я начну постепенно на следующем уроке. А пока, для того, чтобы разобраться с анимацией движения, я покажу вам лишь один из способов создания символов. В символ можно конвертировать любой графический символ. Нарисуем, например, прямоугольник. Инструментом стрелка выделим его. И выполним команду ***Insert*** — ***Convert to Symbol*** или нажмем клавишу ***F***8. Появится диалоговое окно, в котором в поле ***Name*** предлагается задать Имя, под которым символ будет храниться в Библиотеке ролика.



А в списке ***Behavior*** — Поведение выбрать один из трех возможных типов символа. Не вдаваясь в подробности, выберем наиболее простой тип ***Graphic*** и щелкнем по кнопке ОК. Теперь у нас на сцене находится экземпляр символа и можно переходить к созданию анимации движения. Выделим на шкале нужную позицию, например 15-ую и нажмем клавишу ***F***6, скопировав туда ключевой кадр. Переместим объект по горизонтали и снова выберем первый кадр анимации. На панели ***Properties*** в списке тип анимации выберем пункт ***Motion*** — Движение. При этом на шкале времени последовательность кадров окрасится в сиреневый цвет и сквозь нее пройдет сплошная линия со стрелкой.

Если вместо сплошной линии появится пунктирная, значит в ключевых кадрах присутствуют обычные формы, а не символ, и анимация движения будет невозможна. Нажав клавишу ***CTRL+Enter***, посмотрим как объект постепенно перемещается по сцене. Закроем тестове окно. Выделим последний кадр и с помощью модификатора ***Scale*** отмасштабируем объект. Снова протестируем ролик. Вы видите, что одновременно с движением объект изменяет размеры. Вернемся к основной сцене и разберемся с настройками

анимации движения. Выделим первый кадр и посмотрим на панель ***Properties***. Если снять флажок ***Scale*** — Масштаб и протестировать ролик, то размер объекта не будет изменяться в процессе анимации, вплоть до последнего кадра, где прямоугольник внезапно увеличится. Закроем тестовое окно и снова включим опцию ***Scale***. Параметр ***Ease*** выполняет ту же самую функцию, что и при морфинге, и служит для ускорения или замедления движения в процессе анимации. С его помощью удобно создавать такие эффекты, как движение под действием гравитации, торможения, движения в вязких средах и многое другое.

Раскрывающийся список ***Rotate*** предназначен для управления вращением объекта. По умолчанию, в нем выбран режим ***Auto*** — Автоматически. При этом объект вращается только в том случае, если мы вручную повернем его в одном из ключевых кадров. Направление поворота выбирается так, чтобы угол у поворота был минимальным. Если, например, развернуть объект на 270 градусов по часовой стрелке, то фигура повернется в процессе анимации на 90 градусов против часовой стрелки. Проверим это. Перейдем в заключительный кадр. Выделим объект. откроем панель ***Transform***. И в поле ***Rotate*** введем значение 270 градусов. Теперь протестируем ролик. Вы видите, что в процессе анимации объект поворачивается не на 270 градусов по часовой стрелке, а на 90 градусов против часовой стрелки. Что и требовалось доказать.

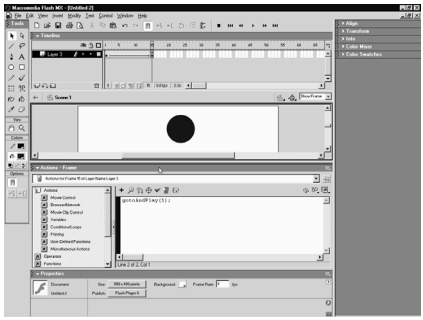
Закроем тестовое окно. Если же в одном из ключевых кадров сделать полный разворот на 360 градусов, то такой поворот восприниматься редактором вообще не будет, поскольку ориентация объекта не изменилась. Но это вовсе не значит, что в процессе анимации движения, объект нельзя заставить вращаться на угол больше чем 180 градусов. Перейдем в первый кадр и откроем список ***Rotate***. Пункты ***Clockwise*** — по часовой стрелке или ***Counteck Clockwise*** — против часовой стрелки, как раз и предназначены для того, чтобы заставить объект принудительно вращаться в нужном направлении. При этом в поле справа необходимо указать количество полных оборотов.

Введем в это поле, например, цифру 1 и протестируем ролик. Вы видите, что объект в процессе движения делает дополнительный полный разворот против часовой стрелки. Закроем тест. Установка флажков ***Orient to Path*** и ***Snap*** имеет значение лишь в том случае, если мы создаем анимацию движения с помощью так называемого специального ***Guide***-слоя, в котором нарисована траектория движения объекта. Флажок ***Synchronize*** позволяет синхронизировать анимацию символов уже содержащих в себе анимацию. Об этих параметрах мы поговорим подробнее на соответствующих уроках. Кроме того в процессе анимации движения можно заставить символ, а точнее его

экземпляр изменять свой цвет и прозрачность. О том как это сделать, я расскажу на следующем уроке.

А сейчас рассмотри еще один пример и создадим анимацию, в которой объект движется по окружности. Создадим новый слой, а старый удалим. в первом ключевом кадре с помощью инструмента овал нарисуем небольшой кружок в верхней части полотна. Выделим его инструментом ***Arrow*** и преобразуем в символ нажав клавишу ***F***8. Отметим тип ***Movie Clip*** или ***Graphic*** и щелкнем кнопку Ок. Выберем инструмент ***Free Transform*** и переместим мышью точку регистрации объекта примерно в середину рабочей сцены. Теперь выделим заключительный кадр анимации и скопируем туда первый ключевой кадр с помощью клавиши ***F***6.

Перейдем в первый ключевой кадр и на панели ***Properties*** в списке ***Tween*** назначим анимацию движения. В списке ***Rotate*** выберем пункт вращение по часовой стрелке и введем в поле справа цифру 1. Теперь наш круг должен в процессе анимации делать один оборот по часовой стрелке вокруг точки регистрации. Убедимся в этом, протестировав ролик. У этой анимации имеется небольшой недостаток. Объект немного задерживается в верхней точке, хотя при данной скорости воспроизведения нашего фильма отмеченный эффект почти не заметен. Задержка происходит потому что при зацикливании анимации после последнего кадра проигрывается первый, а в обоих этих кадрах находится одинаковое исходное изображение, что и приводит к небольшой визуальной остановке движения. Для устраненния этого эффекта достаточно в последний ключевой кадр вставить команду перехода в первый ключевой кадр. Поскольку нам все равно не обойтись без применения ***Flash***-кода, то давайте посмотрим как это сделать, не вдаваясь пока в подробности. Выделим последний кадр.



И откроем панель ***Actions Frame*** — Действия кадра, щелкнув по ее заголовку. Если эта панель вообще отстутствует на экране, то ее можно вызвать командой ***Window — Actions*** или нажав клавишу ***F***9. Щелкнем по кнопке с + и в подменю ***Movie Control*** выберем команду ***GoTo and Play***. В круглых скобках установим переход к первому кадру. При этом в последнем кадре будет добавлена команда перехода в первый ключевой кадр. Свернем панель ***Actions***. Обратите внимание, что в кадре, в котором вставлен какой-либо код появится маленькая буква альфа. Вы можете вообще спросить

зачем нужна эта команда, если при зациклинной анимации, программа сама автоматически переходит из последнего кадра анимации в первый. Дело в том, что при выполнении команды перехода, стодержимое кадра в который вставлена эта команда не прорисовывается.

Подчеркну еще раз, в этом случае последний кадр прорисовываться не будет и тем самым мы избавляемся от так называемого краевого эффекта. Теперь наш круг будет вращаться на сцене равномерно без каких-либо задержек. Убедимся в этом, еще раз протестировав ролик. Если вы раньше никогда не программировали, то можете поздравить себя с почином. В заключении урока приведем еще один пример и создадим анимацию движения, содержащую несколько ключевых кадров.

Изобразим, например, прыгающий мяч. Создадим новый ролик, щелкнув по кнопке ***New***. При необходимости отмасштабируем рабочее полотно командой ***Show Frame***. В нижней части полотна нарисуем прямоугольник однородного цвета, который будет изображать какую-либо твердую поверхность. Анимация будет расчитана на 20 кадров, поэтому выделим двадцатую позицию на шкале и клавишей ***F***5, вставим цепочку обычных кадров.

Теперь создадим еще один слой, в котором будем делать анимацию движения. Выберем инструмент овал, сбросим цвет контура, и в первом ключевом кадре нарисуем круг. Затем выберем инструмент ***Paint Bucket***, в стандартном наборе отметим радиальную градиентноую заливку и заполним нашу фигуру для придания ей объемности. Выберем инструмент Стрелка, выделим объект, и нажмем клавишу ***F***8. В поле ***Name*** введем имя ***Ball*** — мяч. В списке ***Behavior*** отметим пункт ***Graphic***, и щелкнем по кнопке ОК. Отметим восьмую позицию и нажмем клавишу ***F***6. Опустим наш объект в этом ключевом кадре под срез поверхности. Эту операцию проделаем с помощью клавиатуры, чтобы случайно не завалить мяч куда-нибудь в сторону.

Теперь выделим 10 позицию и клавишей ***F***6 скопируем туда восьмой ключевой кадр. Выберем модификатор ***Scale*** отмасштабируем объект по высоте и ширине так, как он должен выглядеть после удара о твердую поверхность. Затем снова опустим фигуру под срез прямоугольника. В 12 кадре объект должен выглядеть также как и в 8, поэтому выделим 8 кадр, щелкнем правой кнопкой мыши, и в появившемся контекстном меню выберем команду ***Copy Frames*** — Копировать кадры. Теперь выделим 12 позицию, вызовем контекстное меню. Выберем пункт ***Paste Frames*** — вставить кадры. Затем выделим 14 позицию и снова нажмем клавишу ***F***6.

В этом кадре отмасштабируем объект так, как он должен выглядеть в момент отскока от поверхности, например, следующим образом. И, наконец, сделаем заключительный ключевой кадр в 20 позиции. Расположим фигуру в том же месте, что и в первом кадре. Поэтому выделим первый кадр и

скопируем его с помощью контекстного меню. Отметим 20 позицию и вставим содержимое буфера обмена.

Итак, ключевые кадры анимации готовы, теперь нужно назначить им анимацию движения. Перейдем в первый кадр и установим на панели ***Properties*** пункт ***Motion***. Проделаем ту же операцию с другими ключевыми кадрами анимации за исключением последнего. Кстати, это можно сделать быстрее, щелкнув правой кнопкой мыши на соответствующем кадре и выбрать в появившемся контекстном меню — ***create Motion Tween*** Создать анимацию движения. Проделаем эту операцию для всех кадров. Теперь протестируем ролик, нажав ***CTRL+Enter***. Закроем окно и немного усложним анимацию. Для придания большей реалистичности движению объекта выполним еще ряд операций. Выделим первый кадр и в поле ***Ease*** установим отрицательное значение -80.

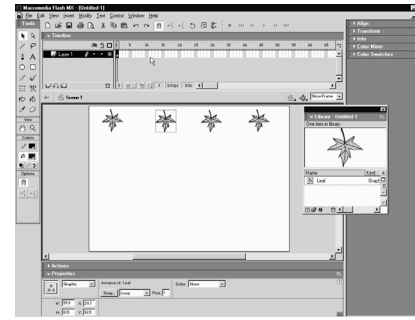
Тем самым, мы заставим двигаться мяч с ускорением, как будто на него воздействует сила тяжести. В предпоследнем ключевом кадре введем в это же поле положительное значение. Это приведет к тому, что по мере подъема, скорость движения мяча будет уменьшаться, как и должно быть в реальных условиях. Протестируем ролик и посмотрим результаты нашей работы. Если просмотреть этот фильм в реальном режиме ***Flash***, скажем со скоростью 12 кадров в секунду, то вы увидите вполне правдоподобную картинку. Закроем тестовое окно и на этом закончим текущий урок.

5.1.2.4. Символы и экземпляры, изменение свойств экземпляров символов

На этом уроке мы начнем подробно разбираться в том, что представляют собой символы в программе ***Flash*** и научимся изменять свойства экземпляров символов. Символы являются особыми типами ***Flash***-объектов и ключевыми символами, которые позволяют на полную мощность использовать эффективность редактора. Применение символов дает в руки пользователей чрезвычайно гибкий инструмент для создания интерактивных роликов, также существенно сокращает размер файлов и экономит ресурсы программы. Поэтому тема этого занятия очень важная. На мой взгляд лучше раскрыть эту тему, опираясь на конкретный пример.

Допустим, необходимо изобразить листопад. И для начала вы нарисовали вот такой листок, который я вставлю из буфера обмена. В обычной графической программе для изображения листопада необходимо было бы создать множество копий нарисованного объекта, например, вот таким образом. В этом случае размер файла ролика значительно возрастет, согласитесь, что это не очень то рационально, особенно если учесть при использовании в Интернете размер файла часто имеет критическое значение. При необходимости размещения на сцене нескольких, пусть даже двух

похожих графических объектов, опытные пользователи поступают иным образом. А именно, конвертируют объекты в символы. Один из способов как это делается вы уже знаете — с помощью клавиши ***F***8. Удалим все копии рисунка, выделим исходный объект и нажмем клавишу ***F***8.



В диалоговом окне зададим имя символа ***Leaf*** — Лист и выберем один из трех типов символа, например, ***Graphic***. Об особенностях каждого типа мы поговорим отдельно несколько позже. Щелкнем по кнопке ОК. Теперь описание или эталон нашего рисунка, который называется символом, будет помещен в Библиотеку ролика. На рабочем же столе остался не сам символ, а его, так называемый, экземпляр — ***Instance***. Этот экземпляр при необходимости можно даже удалить из рабочей области, его описание потеряно не будет. Для отображения Библиотеки выполним команду ***Window-Library***, библиотека или нажмем клавишу ***F***11. В окне Библиотеки находится список всех имеющихся в файле проекта ***Flash***-символов. В данном случае у нас есть только один графический символ с именем ***Leaf***. При щелчке мышью на названии объекта, он отобразится в области предварительного просмотра в верхней части Библиотеки. Чтобы разместить новый экземпляр символа на сцене, достаточно просто перетащить его значок из окна Библиотеки в рабочую область. Перетаскивание можно выполнить также из области предварительного просмотра. Давайте разместим на полотне 4 экземпляра нашего символа.

Таким образом, мы можем поместить на сцене любое количество экземпляров объекта, практически без увеличения размера файла ролика. Так, например, при размещении очередного экземпляра символа любой сложности на сцену, размер файла ролика увеличится всего на 12 байт, которые необходим для запоминания только новых координат объекта. Описывать программе объект заново не нужно, поскольку вся информация о его линиях и заливках уже хранится в Библиотеке. Использует только ссылка на эталон.

Образно выражаясь проигрыватель ***Flash*** в этом случае размышляет следующим образом: взять из Библиотеки содержимое с таким-то именем и показать его. А теперь представьте себе какую экономию ресурсов мы будем иметь при размещении в фильме десятка или более экземпляров одного символа. Не следует, однако, думать, что все экземпляры символа должны быть точной копией эталона. С помощью модификаторов ***Scale*** и ***Rotate*** можно изменять форму объекта, т.е. произвольным способом масштабировать,

поворачивать и наклонять его в разных плоскостях и т.д. Размер файла при этом будет конечно возрастать, но очень ненамного.

Кроме того, к каждому экземпляру символа можно применить различные цветовые эффекты и сделать его практически уникальным. Для этого необходимо выделить како-нибудь экземпляр и обратиться к раскрывающемуся списку ***Color*** на панели ***Properties***. В пятой версии ***Flash*** для этого использовалась отдельная вспомогательная панель Эффект. Для более наглядной демонстрации цветовых эффектов давайте установим на панели ***Properties*** другой цвет полотна. Например, светло-сиреневый.

Теперь выделим какой-либо экземпляр. Например, первый слева и обратимся к списку ***Color***. По умолчанию здесь установлен пункт ***None*** — нет эффекта. Следующий пункт раскрывающегося списка ***Brightness*** — яркость, позволяет регулировать относительную яркость объекта. Положительные значения осветляют объект, вплоть до чисто белого цвета, а отрицательный значения затемняют его. Пункт ***Tint*** — оттенок, позволяет выбрать любой цвет экземпляра. Это можно сделать с помощью палитры, либо ввести значения цветовых составляющих Красный-Зеленый-Синий в полях, расположенных ниже. После установки цвета в самом правом поле можно задать процентное соотношение данного цвета в пределах от 0 до 100%.

Следующий пункт списка ***Alpha*** позволяет изменять прозрачность экземпляра. Значения варьируются от 0% — полностью прозрачный, до 100% — непрозрачный. При выборе по***с***леднего пункта ***Advanced*** — расширенный, появится дополнительная кнопка ***Settings***, для настройки данного цветового эффекта. При щелчке по ней откроется панель, где можно произвольно регулировать и цвет и прозрачность объекта. Вы можете спросить зачем вообще нужен этот эффект, если есть отдельные регулировки яркости, оттенка и прозрачности.

Дело в том, что для экземпляра ***Flash***-символа может быть использован только один из пунктов списка ***Color***. Если, например, сначала настроить прозрачность, а затем выбрать для этого же экземпляра пункт ***Tint***, то установки прозрачности будут сброшены. Пункт ***Advanced*** позволяет независимо и одновременно настраивать цветовые составляющие и коээфициент прозрачности экземпляра символа. Здесь имеются 8 полей, левые из которых используютс для регулировки параметров цвета и прозрачности по относительной шкале. А правые для корректировки абсолютных значений цветовых составляющих.

Если применять их одновременно, то это дает большие возможности, но поначалу может смутить своей сложностью. Поэтому я предлагаю некоторе время поэкспериментровать с данными настройками самостоятельно. Эти усилия будут вознаграждены красочностью ваших роликов. Хочу сразу

отметить, что установки списка ***Color*** действуют также на все виды символов, которые вы будете создавать, включая растровые изображения, конвертированные в символы.

5.1.2.5. Графические символы и символы видеоклипов

На этом уроке мы подробно рассмотрим применение графических символов и символов видеоклипов. Определим их основыне различия, а также научимся создавать символы видеоклипов из готовой анимации расположенной на основной сцене. Графические символы, которые мы и применяли на прошлых занятиях — это самый простой тип символов и сипользуется в соновном для хранения статической графики.

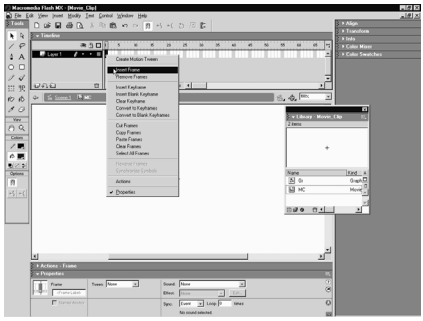
Тем не менее графические символы имеют собственную временную шкалу и мугут внутри себя содержать анимацию. Однако анимация внутри таких символов привязана к основной временной шкале. Это означет, что елси отсановиь воспроизведение фильма на основной сцене, то анимация внутри графического символоа тоже остановится. Покажем это на простом примере. Создадим новый графический символ. До сих пор мы это делали, рисую что-либо на рабочем полотне, а затем конвертирвали изображение в символ, с помощью команды ***Insert-Convert to Symbol*** или клавишей ***F***8. Но можно сделать и по-другому. Выполним команду ***Insert-New Symbol***. Новый символ или нажмем сочетание ***CTRL+F***8. Появится знакомое диалоговое окно. Выберем тип ***Graphic*** и дадим символу какое-нибудь коротенькое имя, например, ***Gr*** — сокращенное от ***Graphic*** и щелкнем по кнопке ОК. Появится пустая область редактирования символа с маленьким крестиком по середине. Здесь находится центр координат нашего символа. О том, что мы находимся в режиме редактирования символа, говорит пиктограмма с его именем.

Давайте создадим простенькую раскадровку формы на временной шкале нового символа. Выберем инструмент прямоугольник и нарисуем соответствующую фигуру. Отметим 10 позицию шкалы и вставим туда пустой ключевой кадр клавишей ***F***7. Теперь с помощью инструмента овал, где-нибудь правее нарисуем круг. Перейдем к первому кадру и на панели ***Properties*** установим раскадровку формы. Подвигаем маркер вдоль шкалы времени, чтобы убедиться в работоспособности анимации. Теперь вернемся к основной сцене. Это можно сделать выполнив команду ***Edit-Edit Document***. Однако проще и быстрее щелкнуть по пиктограмме с надписью ***Scena***1. Перед нами снова чистое рабочее полотно. Куда же делся наш символ? Конечно, он находится в Библиотеке ролика. Для вызова окна Библиотеки, нажмем клавишу ***F***11. Выделим наш символ в окне Библиотеки. Обратите внимание, что в окне просмотра появились дополнительные кнопки контроллера.

Поскольку символ содержит анимацию. Теперь ее можно посмотреть прямо в Библиотеке.

Перетащим мышью символ на рабочее полотно, тем самым создав его экземпляр. Здесь отобразится первый кадр анимации символа. Впрочем мы можем отобразить и любой другой кадр анимации. Для этого в списке опций на панели инстпектора свойств выберем пункт ***Single Frame*** — один кадр и определим его номер, например, пятый. Если теперь протестировать ролик, то кроме статичного пятого кадра, мы ничего не увидим. Закроем окно теста.

Это происходит потому что основная шкала времени содержит только один кадр. Т.е. анимация основного фильма остановлена. Следовательно анимация внутри графического символа, также не будет. Теперь на панели ***Properties*** вернемся к пункту ***Loop*** — Цикл и добавим к основному фильму несколько простых кадров. Выберем 10 позицию шкалы и нажмем клавишу ***F***5.



Снова протестируем ролик. Как видите, анимация ***Flash***-символа теперь производится в полном объеме. Поскольку как вы помните эта анимация состоит также из 10 кадров. Закроем тестовое окно и сократим количество кадров основной шкалы. Для чего, удерживая клавишу ***CTRL***, переместим 10 кадр в пятую позицию. Протестируем ролик и увидим, что теперь происходит анимация только половины графического символа. Я думаю теперь вам понятно, почему это происходит. Итак, вопроизведение анимации внутри графического ***Flash***-символа напрямую зависит от хода анимации основного фильма.

А теперь рассмотрим особенности символа типа ***Movie Clip*** и сравним их с графическими символами. Создадим точно такую же анимацию формы, но поместим ее внутри символа ***Movie Clip***. Чтобы сделать это по-быстрее воспользуемся возможностью программы копировать кадры. Щелкнем дважды по значку символа в окне Библиотеки. Тем самым мы вошли в режим его редактирования. Удерживая клавишу ***Shift***, выделим все кадры анимации. Щелкнем правой кнопкой мыши и в контекстном меню вберем команду ***Copy Frames***. Веренемся к основной сцене. Убедимся, что ничего не выделено и нажмем сочетание ***CTRL***+***F***8. В диалоге введем имя ***Mc***, а в разделе ***Behavior*** — Поведение определим тип ***Movie Clip***. Подтвердим установки. Выделим первый кадр шкалы ***Movie Clip***а, вызовем контекстное меню и выполним команду ***Paste Frames*** — Вставить кадры. Теперь мы имеем здесь точно такую же анимацию, что и в графическом символе. Снова вернемся к

основной сцене. Теперь в Библиотеке ролика у нас появился новый ***Flash***-символ ***Mc***, типа ***Movie Clip***. Обратите внимание, что он имеет другой значок.

Среди флэшеров этот тип символов часто называют мувиками, по созвучию с английским названием ***Movie Clip***. Я тоже иногда буду использовать этот термин. На основной сцене оставим только один кадр, для чего выделим кадры со второго по пятый и удалим их командой ***Insert-Remove Frames***.

Или нажмем комбинацию ***Shift***+***F***5. Перетащим наш мувик в рабочую область и протестируем фильм. Вы видите, что анимация экземпляра графического символа не происходит, а анимация мувика идет как ни в чем не бывало. Закроем тест. Вывод следующий: ***Movie Clip***, по сути представляет собой отдельный фильм, живущий своей жизнью и остановка проигрывания онсовной сцены никак не влияет на его воспроизведение.

5.1.2.6. Символы кнопок и их особенности, создание пульсирующей кнопки

На этом уроке мы подробно познакомимся еще с одним из трех типов символов, которые очень часто применяются в программе ***Flash*** — символоами кнопок. Кнопка представляет собой специальный объект, ориентированный на интерактивность. Т.е. обычно предполагается, что пользователь щелкает мышью по кнопке и при этом что-нибудь происходит. Подобно графическим символам и мувиклипам, кнопки имеют собственную временную шкалу, которая воспроизводится независимо от основной шкалы времени.

Но на этом все сходства заканчиваются, основное отличие между кнопками и видеоклипами заключается в том, что временная шкала кнопок всегда состоит только из четырех кадров, каждый из которых имеет свое название. Давайте сначала научимся создавать и редактировать кнопки. Кнопки как и другие типы символов можно создавать нажав сочетание ***CTRL***+***F***8 или нрисовать что-нибудь на полотне, а затем конвертировать в символ кнопки с помощью клавиши ***F***8. Итак, нажмем ***CTRL***+***F***8, дадим имя символу ***Button*** 1, назначим также тип ***Button*** и щелкнем по кнопке ОК. Теперь мы вошли в режим редактирования кнопки.

Посмотрим на временную шкалу первый кадр имеет название ***Up*** и ему соответствует свободное состояние кнопки. Т.е. так она будет выглядеть, когда курсор пользователя находится за ее пределами. Давайте что-нибудь нарисуем. Можно использовать любое изображение, текст, а также и анимированный

экземпляр другого типа. Но чаще всего для кнопок используются прямоугольные или округлые формы. Выберем инструмент прямоугольник, сбросим цвет контура, щелкнем по модификатору и в появившемся окне установим радиус скругления углов равным 10. Теперь рисуем. Выберем инструмент Стрелка, выделим объект и с помощью панели ***Align*** при активном переключателе ***To Stage***, отцентрируем его относительно начала координат.

Затем займемся кадром ***Over***, которому соответствует состояние кнопки при наведении на нее курсора. Можно клавишей ***F***7 вставит пустой ключевой кадр и нарисовать в нем любое изображение. но проще вставить ключевой кадр клавишей ***F***6. И изменить в этом кадре форму или, скажем, цвет фигуры, нпример на зеленый. Теперь при наведении курсора кнопка должна выглядеть именно так. Аналогичным образом поступим с кадром ***Down***, который отображается, когда пользователь нажимает в области кнопки левую кнопку мыши. Скопируем клавишей ***F***6 предыдущий кадр и изменим цвет прямоугольника на красный.

Кадр ***Hit*** — Область наведения никогда не отображается, но часто он очень важен, поскольку именно этот кадр определяет область действия кнопки. Если оставить кадр ***Hit*** пустым, то по умолчанию его содержимое считается совпадающим с предыдущим непустым ключевым кадром. Пока не будем ничего рисоват в кадре ***Hit***. В учебных целях изобразим также надписи на кнопке. Перейдем в первый кадр и выберем инструмент Текст. На пнели ***Properties*** установим черный цвет, полужирное начертание и выравнивание по центру. Щелкнем на полотне и напишем название кадра.

Закончим ввод выбором инструмента Стрелка. При необходимости подберем подходящий размер шрифта и отцентрируем текст относительно начала координат. Чтобы больше не заниматься выравниванием надписи в других кадрах, поступим следующим образом. Скопируем текст в буфер обмена, затем перейдем во второй кадр и выполним команду ***Edit Paste in Place*** — вставить на место. Теперь выберем снова инструмент Текст, щелкнем по надписи и заменим название кадра. Перейдем в третий кадр, вставим сюда скопированное текстовое поле и проделаем аналогичные манипуляции с заменой имени кадра. Все простейшая кнопка готова и можно вернуться к основной сцене. Клавишей ***F***11 отобразим окно Библиотеки и претащим символ кнопки на рабочее полотно.

Теперь протестируем ролик. Когда курсор находится вне кнопки он имеет голубой цвет, т.е. отображается кадр ***Up***. Наведем указатель на кнопку. Вы видите, что она стала зеленой, т.е. отобразился кадр ***Over***. Обратите внимание, что курсор при наведении на кнопку, приобретает вид руки. Если теперь нажать клавишу мыши, то отобразится кадр ***Down*** и кнопка станет красной. Закроем тестовое окно. Кстати, говоря, если в меню ***Control*** включить опцию

Enable Simple Button, то можно посмотреть как работает кнопка прямо на рабочей сцене. Убедимся, что это так. Довольно удобно. Однако в этом случае объект нельзя выделить и следовательно премещать, масштабировать и вообще производить какие-либо модификации.

Кроме того, эта установка работает, если в кадрах кнопки, содержится только статическая графика. Теперь разберемся подробно с назначением 4 кадра кнопки кадра ***Hit***. Его особенно важно определять если мы используем кнопку сложной формы, например, в виде текста. Выберем инструмент Текст, щелкнем на полотне и введем буквы ОК. Выберем инструмент стрелка и на панели ***Properties*** увеличим размер кегля, нажмем клавишу ***F***8, зададим имя, например такое, сокращенное от ***ButtonText*** и конвертируем текст в символ кнопки.

Дважды щелкнем по значку нового символа в Библиотеке, скопируем клавишей ***F***6 в кадр ***Over***, содержимое кадра ***Up*** и изменим цвет букв на зеленый. Определим также кадр ***Down*** и сделаем цвет красным. Кадр ***Hit*** также пока определять не будем. Перейдем к основной шкале и протестируем кнопку. В видите, что кнопка реагирует на указател мыши, только в том случае, когда последний находится точно на контуре символов. Центральная область буквы О, а также промежутки между буквами не являются активной областью. Согласитесь, что это не очень то удобно, поскольку при работе с такой кнопкой потребуется хирургическая точность наведения. Давайте исправим положение и создадим область ***Hit***. Войдем в режим редактирования текстовой кнопки, клавишей ***F***6 создадим ключевой кадр ***Hit***.

Затем выберем инструмент прямоугольник, сбросим цвет контура и нарисуем фигуру, размером чуть больше текста. Цвет фигуры абсолютно никакой роли не играет. Важен лишь размер заливки, ее форма и расположение относительно объектов в других кадрах. Перейдем к основной сцене и протестируем кнопку. Теперь она работает как положено. Чтобы измежать показанных выше недоразумений считается хорошим тоном всегда определять область действия кнопки.

5.1.3. Принципы работы со специальными слоями.Применение сцен

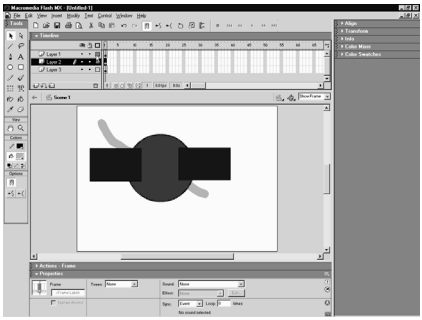
5.1.3.1. Общие принципы работы со слоями, опорные слои

Как вы уже знаете из предыдущих уроков, при наложении графические объекты взаимно редактируют друг друга, чтобы этого не происходило объекты можно группировать или создавать ***Flash***-символы. Еще один простой и удобный способ раположить в ролике множество независимых объектов — это использование слоев. Техника слоев применяется во многих популярных

графических программах, например в ***Photoshop***'***e***. Однако, в отличие от этой прогарммы увеличение слоев во ***Flash*** на влияет на размер опубликованного ролика. Поэтому вы можете создавать столько слоев сколько считаете необходимым. Фактически верхним пределом является 16000 слоев, н не думаю, чтобы вам может столько понадобиться. Традиционно слои принято сравнивать с листами прозрачно кальки, наложенных друг на друга. Рисунки расположенные в верхних слоях, закрывают собой графику в нижних. Там, где изображения нет будут просвечиваться рисунки нижних слоев.

На прошлых занятиях мы уже неоднократно использовали слои и теперь самое время разобраться с тонкостями работы с нимив редакторе ***Flash MX***. Кроме того, в программе могут применяться специальные слои, которые мы начнем изучать на этом уроке. Выберем инструмент прямоугольник и нарисуем две фигуры в первом и пока единственном слое.

А теперь создадим новый слой, щелкнув по кнопке ***Insert Layer*** со значком +. Будет создан новый слой, который автоматически станет активным. О том, что слой является активным, говорит его выделение черным цветом и изображение карандаша. Выберем инструмент овал, изменим цвет заливки и нарисуем круг. Его изображение будет помещено в текущий активный слой.



Поскольку рисунок расположен в верхнем слое, то он будет перекрывать графику нижнего слоя. Создадим еще один слой, разумеется это можно сделать и с помощью меню, выполнив команду ***Insert — Layer***. Новый слой окажется сверху текущего. Выберем инструмент ***Brush***, изменим цвет заливки и сделаем мазок кистью поверх других фигур. Порядок расположения слоев, можно легко изменить, просто перетаскивая их мышью. Чтобы сделать слой активным, т.е. доступным для редактирования. щелкните по нему мышью. При этом произойдет выделение всех объектов, расположенных в этом слое.

Для редактирования какого-либо отдельного объекта нет необходимости сначала делать активным соответствующий слой. Это произойдет автоматически при выделении заданного объекта. Очень удобное свойство редактора ***Flash***. Для облегчения работы слоям можно давать любые имена, в том числе и русские, для этого дважды щелкните на имени слоя и введите новое название. Всегда используйте при этом осмысленные имена. Теперь обратите внимание на три пиктограммы в верхней части списка слоев. Если

щелкнуть в каком-либо слое на точке под изображением глаза, то графика этого слоя станет невидима, а вместо точки появится красный крестик.

Повторный щелчок по этому же месту сделает слой снова видимым. При щелчке непосредственно по пиктограмме глаза, невидимыми или снова видимыми станут все слои. Если же удерживать клавишу ***Alt*** и щелкнуть по соответствующей точке, то графика всех остальных слоев станет невидима, теперь можно спокойно работать с изображением только одного слоя. Сделаем снова видимыми все слои.

Второй столбец точек, расположенный под пиктограммой замка, позволяет защитить графику слоев от редактирования. Когда на слое висит замочек, то выделить и редактировать расположенные в нем объекты, становится невозможным. Щелчок по самой пиктограмме замка ставит или снимает запрет на редактирование всех имеющихся слоев. И, наконец, третий столбец, обозначенный квадратиком, предназначен для отображения содержимого слоев в виде контуров. Принцип использования этой опции такой же как и у двух предыдущих.

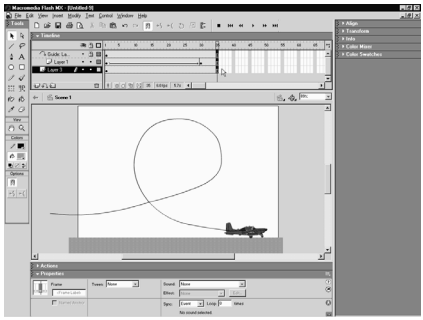
При щелчке на квадрате в соответствующем столбце, содержимое текущего слоя будет просматриваться в виде контуров одного цвета, в данном случае синего. Повторный щелчок вернет графике обычный вид. Контурная прорисовка объектов слоя бывает весьма полезна при работе со сложными изображениями и снижает нагрузку на процессор. Для каждого самостоятельного слоя можно выбрать свой цвет отображения контуров, но об этом чуть позже. Во ***Flash MX*** появилась новая возможность организовывать слои в папки. Перейдем в верхний слой и щелкним по кнопке ***Insert Layer Folder***. Будет создана папка, которой также можно дать любое имя. Чтобы поместить наши слои в папку, необходимо мышью перетащить слой вверх и немного вправо. Пиктограмма слоя сместится вправо и теперь он находится в папке. Аналогичным образом разместим в папке два других слоя.

Вытащить слой из папки, можно переместив его пиктограмму вниз и влево. Папку легко свернуть щелкнув по треугольнику слева от ее пиктограммы, и тем самым экономя драгоценное экранное пространство. Повторный щелчок вновь откроет папку. Если применить к папке опцию блокировки или другие, то эти операции будут применены ко всем слоям, размещенным в папке.

В п***а***пку можно помещать другие папки и таким образом организовывать слои подобно дереву файлов на компьютере. Как то мне на глаза попался исходник, сделанный во ***Flash*** 5, который содержал более 30 слоев. Разобраться, что и где было крайне сложно. Представляю как обрадовался его автор, узнав о новой возможности редактора ***MX*** создавать папки для слоев. Таким образом, с помощью слоев и папок можно просто и эффективно расположить множество объектов на сцене, не беспокоясь о возможности их

взаимного редактирования. Однако не будем забывать, что ***Flash***, главным образом предназначена для анимации. Именно при создании анимации слои помогают больше всего и позволяет добиться того, чтобы объекты двигались одновременно в различных направлениях и были по-разному согласованы по времени. При создании банера мы уже использовали этот принцип, но я не акцентировал на нем внимание.

5.1.3.2. Слои траекторий. Принципы их использования



На этом уроке мы познакомимся со слоями траекторий, который позволяют задавать маршруты движения объектов графически, просто рисуя их на экране. Специальные слои траекторий очень мощное средство создание аниамции в программе ***Flash***. когда в процессе анимации необходимо заставить объект вращаться, двигаться прямолинейно или по окружности, то можно применить обычную раскадровку движения, указав в ключевых кадрах начальное и конечное положение объекта. Однако, что делать для реализации движения объекта по произвольному пути? Вот тут то нам и не обойтись без слоев-траекторий. Выберем инструмент Овал и нарисуем с его помощью соответствующую фигуру.

Выделим рисунок и, нажав клавишу ***F***8 конвертируем в графический символ или мувиклип. Отметим на шкале 30 позицию и клавишей ***F***6 вставим туда ключевой кадр. Вернемся в первый кадр и с помощью контекстного меню назначим анимацию движения. Теперь попробуем заставить двигаться наш объект по произвольному пути. Для этого необходимо создать спциальный слой и нарисоввать там нужную траекторию. Самы простой способ создания слоя траектории — это щелчок по кнопке ***Add Motion Guide***.

Появится новый слой траектории с 30 пустыми кадрами. Одновременно нижний слой с объектом будет прикреплен к ***Guide***-слою и изменит свой тип. Убедимся в этом, открыв диалог Свойства слоя. Кстати говоря, это можно сделать еще одним способом. Достаточно дважды щелкнуть по пиктограмме слоя. Как видите, теперь слой имеет тип ***Guided*** — Ведомый или направляемый. Кроме того, о том что первый слой прикреплен к слою траектории, говорит пунктирная линия на границе между ними.

Закроем окно свойств. Перейдем в ***Guide***-слой и нарисуем здесь произвольную траекторию. Траектория должна быть незамкнутой и

представлять собой контур или заливку. Рекомендуется использовать только контур, поскольку применение заливки, иногда приводит к неожиданным результатам. Выберем инструмент Карандаш, включим модификатор ***Smooth*** для получения более плавной кривой и рисуем. Толщина и цвет контура значения не имеют, так как сама траектория в опубликованном ролике просматриваться не будет. Заблокируем слой траектории и перейдем в слой объекта. Выберем инструмент Стрелка и привяжем фигуру к начальной точке траектории.

При этом должен быть включен режим ***Snap*** на основной панели. Привязку можно было бы сделать и несколько иначе. Сделаем отмену последней операции перемещения, выделим првый кадр анимации и включим опцию ***Snap*** на панели ***Properties***. Объект самостоятельно привяжется к начальной точке траектории. Перейдем в заключительный кадр анимации и разместим объект на другом конце кривой. Строго говоря не обязательно привязывать объект именно к концам траектории, но нет смысла и рисовать контур так, чтобы исползовать его не полностью.

Нажмем кнопку ***Play*** и убедимся, что наш объект движется точно по нарисованному пути. Затем нажмем сочетание ***CTRL***+***Enter*** и еще раз посмотрим, что результат достигнут, а сама траектория в опубликованном ролике не видна. Закроем тестовое окно и снимем на панели Инспектора флажок ***Snap***, чтобы программа не лишала нас самостоятельности при выборе точек привязки. Слои-траектории также как и опорные слои не экспортируются вместе с другим содержимым фильма и следовательно не влияют на размер окончательного файла в формате ***.swf***.

Поэтому можно использовать их без ограничения. Если вы хотите, чтобы траектория движения объекта была видна, то используйте обычный слой с копией маршрута. К одному слою траектории можно привязывать несколько ведомых слоев с объектами. Покажем это. Сделаем текущим нижний слой и щелкнем по кнопке Добавить обычный слой. Появится новый слой, который будет автоматически прикреплен к нашему ***Guide***-слою. Нарисуем в новом слое какой-нибудь объект, например, прямоугольник. инструментом Стрелка выделим его и, используя клавишу ***F***8, конвертируем в символ. перейдем в 30 кадр и нажмем клавишу ***F***6, создав там ключевой кадр. Выделим первый кадр и с помощью контекстного меню назначим анимацию движения.

Сделаем текущим ***Guide***-слой, разблокируем его и с помощью Карандаша нарисуем в нем новую траекторию движения для прямоугольника. Выберем инструмент стрелка, выделим прямоугольник и привяжем его в первом кадре к начальной точке пути. Перейдем к заключительному кадру анимации и разместим объект в конечной точке траектории. Дополнительно, с помощью модификатора ***Scale*** изменим масштаб прямоугольника. Если мы хотим чтобы

дополнительно объект при движении еще и вращался, то снова перейдем в первый кадр и на панели Инспектора Свойств установим в поле ***Rotate*** поворот по часовой или против часовой стрелки. А также введем полное количество оборотов объекта в процессе анимации.

Здесь же на панели можно при желании сделать установку, которая заставит двигаться объект с положительным или отрицательным ускорением. Тепь протестируем ролик. Мы видим, что обе фигуры движутся независимо и одновременно по разным траекториям. При этом прямоугольник в процессе движения вращается и изменяет свой размер. Закроем тестовое окно. А сейчас приведем еще один пример и покажем как можно управлять ориентацией объекта в процессе его движения по заданной траектории.

Создадим новый ролик, щелкнув по кнопке ***New***. И как обычно отмасштабируем рабочее полотно. Нарисуем в единственном слое или вставим из буфера обмена готовое изображение самолета времен второй мировой войны, созданное в другой программе. Теперь попробуем заставить его лететь по произвольной кривой, но сначала создадим анимацию движения. Нажмем клавишу ***F***8 и конвертируем рисунок в символ, выделим 30 позицию на шкале и нажмем клавишу ***F***6. В контекстном меню первого кадра установим анимацию движения. Теперь создадим слой траектории щелкнув по соответствующей кнопке.

Выберем инструмент Карандаш и нарисуем траекторию полета самолета. Рисовать начнем слева за пределами полотна, чтобы наш объект влетал слева в рабочую область ролика. Попробуем изобразить траекторию так, чтобы самолет при движении делал мертвую петлю. Инструментом стрелка выделим нарисованный путь и щелкнем несколько раз по модификатору ***Smoth***, чтобы придать контуру более плавное очертание. При необходимости иснтрументом Стрелка можно подкорректировать форму кривой. Теперь заблокируем ***Guide***-слой, повесив на него замок. Выделяя объект поочередно в первом и последнем кадрах анимации, привяжем его к началу и к концу траектории соответственно. Чтобы сымитировать посадку самолета в конуе анимации, нарисуем земную поверхность.

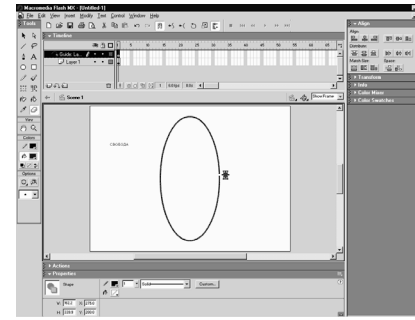
Для этого создадим новый обычный слой и переместим его мышью в нижнюю часть списка. Передвинем маркер в заключительный кадр, чтобы видеть самолет в конце анимации. Выберем инструмент Прямоугольник, сбросим цвет контура, а цвет заливки установим серым. Нарисуем внизу полотна соответствующую фигуру, постараемся сделать так, чтобы нарисованная поверхность касалась шасси самолета. Для лучшего визуального восприятия будущего ролика, удерживая клавишу ***Shift***, отметим 35 позицию каждого слоя и нажмем клавишу ***F***5.

Теперь посмотрим анимацию, нажав ***CTRL+Enter***. Все бы ничего, но наш самолет летит все время направив нос в одну сторону, что выглядит совершенно неестественно. А хотелось бы, чтобы он поворачивался в процессе движения паралельно траектории. Оказывается в программе ***Flash*** сделать это не трудно. Выделим первый кадр анимации и на панели ***Properties*** установим галочку в пункте ***Orient To Path*** ориентировать по траектории.

В поле ***Easing*** установим положительное значение около 90%. анимация в этом случае пойдет с замедлением и наш самолет будет постепенно останавливаться при посадке. Щелкнем на пустом месте и на панели Инспектора Свойств установим светло-голубой цвет имитирующий небо. Посмотрим еще раз анимацию. Теперь наш объект двигается правильно. В заключении хотел бы отметить, что с помощью слоев траекторий и масштабирования можно добиться интересных эффектов, имитирующих движение объектов в трехмерном пространстве. Примеры таких эффектов будут рассмотрены на следующем уроке.

5.1.3.3. Примеры создания трехмерных эффектов с помощью слоев траекторий

На этом занятии с помощью слоев траекторий, масштабирования и некотрых других приемов мы создадим два ролика, в которых имитируется движение объектов в трехмерном пространстве. Сначала создадим объемный текстовый эффект. Выберем инструмент Текст и на панели Инспектора свойств сделаем следующие настройки: шрифт ***Arial*** кириллица, размер кегля девятый, цвет синий. Собственно говоря шрифт и цвет вы можете выбрать на свое усмотрение.



Переключимся на русскую раскладку клавиатуры, щелкнем на полотне и введем слово Свобода, используя при этом только прописные буквы. Закончим ввод выбором инструмента Стрелка. Конвертируем текст в символ, для чего нажмем клавишу ***F***8. Дадим имя символу ***text***, выберем тип ***Graphic*** или ***Movie Clip*** и подтвердим установки. Теперь для управления движением клипа, нужно создать для него траекторию в соответствующем слое. Щелкнем по кнопке Добавить слой траектории. Выберем иснтумент Овал, сбросим цвет заливки и нарисуем на сцене незалитый эллипс. Высота фигуры должна быть несколько меньше ширины рабочего полотна, впрочем, это всегда можно подкорректировать модификатором ***Scale***. Нажмем клавишу ***V***, для быстрого выбора инструмента Стрелка, выделим эллипс и с помощью

панели ***Align*** разместим его строго по центру сцены. Затем выберем инструмент Ластик, определим в разделе модификаторов его самый маленький размер и сделаем небольшой разрыв в центральной правой части нарисованной фигуры. Снова выберем инструмент Стрелка и заблокируем слой траектории.

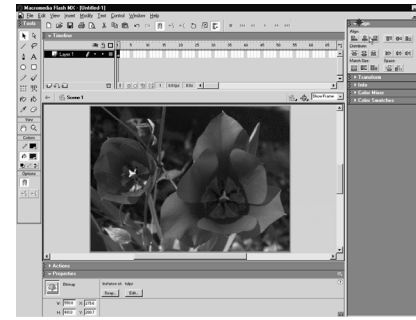
Выделим экземпляр текстового символа перетащим его к эллипсу и привяжем к верхней точке разрыва. Клавишей ***f***6 создадим ключевые кадры в 5, 10, 15 и 20 кадрах. Отметим также 20 позицию слоя траектории и клавишей ***F***5 вставим цепочку простых кадров. Перейдем в 5 кадр нижнего слоя и привяжем объект к самой верхней части эллипса. В 10 кадре переместим символ в крайнюю левую точку траектории. В 15 кадре осуществим привязку к нижней точке эллипса.

И, наконец, в 20 кадре разместим объект справа, но уже в нижней точке разрыва. Итак ключевые кадры готовы и теперь нужно назначить для них раскадровку движения. Сделаем это с помощью контекстного меню, последовательно щелкая правой кнопкой мыши в каждом ключевом кадре и устанавливая опцию ***Create Motion Tween*** — Создать анимацию движения. Протестируем предварительный результат нашего фильма и убедимся, что объект правильно движется по эллиптической траектории.

Закроем тестовое окно и приступим ко второй части создания ролика. А именно, в каждом ключевом кадре изменим параметры масштабирования и прозрачности, чтобы добиться визуального восприятия объемности. Перейдем в первый ключевой кадр, выделим объект. В списке цветовых эффектов на панели ***Properties*** выберем пункт ***Alpha*** и установим прозрачность клипа равно 50%. В пятом кадре выполним аналогичные действия, определим прозрачность 75%, а затем с помощью панели ***Transform***, при установленном флажке ***Constrain***, введем масштаб 300%. В 10 кадре прозрачность изменять не будем, а масштаб определим равным 600%.

В 15 кадре сделаем такие же установки, что и 5, т.е. установим прозрачность 75%, а масштаб 300%. И, наконец в последнем 20 кадре сделаем прозрачность равной 50%, а опции масштаба трогать не будем. Снова протестируем наш фильм и убедимся, что изменив параметры масштабирования и прозрачности, удалось добиться достаточно убедительного эффекта объемности. Вы можете также поэкспериментировать с цветом фона и получить более выразительный эффект. Закроем тестовое окно.

5.1.3.4. Принципы работы с маскирующими слоями. Примеры их использования



Кроме вспомогательных опорных слоев и слоев траекторий в редакторе ***Flash*** очень часто применяются так называемые слои-маски, с помощью которых можно создавать очень интересные эффекты. Применение масок подразумевает наличие как минимума двух слоев, а именно маскируемого слоя, в котором может находиться максируемый объект и слоя маски. Что же такое маска? Маска используется для сокрытия определенной части изображения, расположенного в нижележащем слое. Графика, находящаяся в слое маске не просматривается в опубликованном ролике, но она выполняет роль отверствия или окна, через которое видно то, что нарисовано в прикрепленных маскируемых слоях.

Если применить к слою-маске или к маскируемому слою анимацию, то можно получить множество замечательных эффектов. Прежде всего вы должны знать какие объекты должны быть использованы в качестве масок. Это простые формы, т.е. заливки, статический текст и графические символы размещенный в одном слое и содержащие простые формы.

Линии или контуры, градиентные или растровые заливки, а также динамический текст не могут быть масками. В принципе в качестве масок можно применять и мувиклипы, но в этом случае есть большое но. Так ***Flash*** использует в качестве маскирующего элемента, только первый кадр нижнего слоя клипа. Поэтому в общем случае мувиклипы не являются лучшим выбором для создания масок. Зато они прекрасно подходят в качестве маскируемых элементов. а теперь перейдем к практическим примерам. Создадим эффект прожектора — это классический пример, котрый демонстрирует, когда хотят познакомит с маскированием.

Импортируем в проект подходящее изображение, выполнив команду ***File Import***. Выберем нужный файл и щелкнем по кнопке Открыть. При необходимости с помощью панели ***Align*** разместим рисунок в центре полотна. Нажмем клавишу ***F***8 и конвертируем его в графический символ или в мувиклип. Эту операцию мы сделали для того, чтобы в последующем применить к изображению цветовые эффекты. Теперь нам нужно создать слой маску. К сожалению разработчики программы не предусмотрели для этого специальной кнопки под списком слоев. Поэтому мы сначала создадим обычный слой и дадим ему имя ***Mask***. Далее дважды щелкнув по пиктограмме

слоя или через контекстное меню откроем окно его свойств. Здесь установим тип ***Mask*** и щелкнем по кнопке ОК. Обратите внимание как изменилась пиктограмма слоя. Сделать из слоя маску можно и быстрее, установив в контекстном меню флажок на пункте ***Mask***. Далее необходимо прикрепить нижний слой к слою маске, т.е. сделать его маскируемым.

Это можно реализовать двумя путями. Щелкнем правой кнопкой мыши по слою и откроем окно его свойств. Если установить сейчас тип ***Masked***, то слой станет маскируемым, но мы пока этого делать не будем, поэтому щелкнем по кнопке ***Cancel*** — Отмена. И покажем более быстрый способ прикрепления слоя к маске. Просто перетащим нижний слой под слой маску.

Пиктограмма слоя изменится и он станет маскируемым. Заблокируем нижний слой и перейдем в слой маску. Затем с помощью иснтрумента Овал нарисуем круг. Цвет заливки круга значения не имеет. Инструментом ***Arrow*** выделим и удалим контур фигуры, поскольку линии в масках не используются. Если сейчас протестировать ролик, то на экране мы увидим только часть изображения нижнего слоя, которое находится под маской. Закроем тест. Кстати говоря принцип дефствия маски можно посмотреть и на основной сцене, для этого нужно заблокировать оба слоя. Снова разблокируем слой маски и создадим анимацию движения. Выделим круг, нажмем клавишу ***F***8 и конвертируем фигуру в графический символ. Перетащим экземпляр символа за левую часть полотна, затем выдели 15 позицию слоя и клавишей ***F***6 скопируем туда первый ключевой кадр. В этом кадре переместим объект вправо за пределы сцены. Используя клавишу ***F***5 увеличим длительность нижнего слоя также до 15 кадров.

Щелкнем правой кнопкой мыши на первом кадре слоя Маски и назначим анимацию движения. Теперь протестируем фильм. Мы получили достаточно инстересный эффект, но это еще не эффект прожектора. Закроем тестовое окно и добавим к фильму небольшое усложнение. Создадим новый обычный слой и перетащим его вниз пакета слоев. Убедимся, что новый слой не является маскируемым. Сделаем невидимыми два верхних слоя, клавишей ***F***11 откроем окно Библиотеки и поместим еще один экземпляр символа изображения на сцену. Используя панель Выравнивание отцентрируем объект. Не снимая выделения в списке цветовых эффектов панели ***Properties*** установи пункт яркость и установим ее значение — 60%.

Протестируем созданную анимацию еще раз. Теперь получился эффект настоящего прожектора. Закроем тестовое окно и приведем еще один пример, где с помощью маски попробуем создать анимацию вращающегося глобуса, которую часто можно увидеть на ***Web***-страницах ***Internet***. Создадим новый проект и отмасштабируем рабочую область. На этот раз маску сделаем статичной, а анимировать будем маскируемое изображение. Импортируем из

файла или вставим из буфера обмена рисунок карты в виде прямоугольной области. Нажмем клавишу ***F***8 и конвертируем рисунок в символ, что необходимо для создания анимации движения. Щелкнем дважды на названии нового слоя и дадим ему имя ***Map*** -карта. Теперь создадим новый слой, который в дальнейшем будем использовать в качестве маски. Выберем инструмент Овал, сбросим цвет контура и нарисуем круг любого цвета диаметром чуть меньшем ширины нашей карты.

Выделим его инструментом ***Arrow*** и разместим относительно карты в какой-нибудь характерной точке. Например, так, чтобы левый край фигуры касался крайней правой точки Африки на карте. Перейдем в слой карты, выделим 18 позицию слоя и создадим клавишей ***F***6 ключевой кадр. Чтобы выровнять слои по длительности, отметим 18 позицию верхнего слоя и нажмем клавишу ***F***5. Опять перейдем в 18 кадр нижнего слоя и спомощью клавиатуры сместим карту влево настолько, чтобы Африка оказалась снова у левого края круга. Этим мы добъемся того, что в первом и последнем кадрах анимации, под маской будет одинаковое изображение. Перейдем в первый кадр нижнего слоя и с помощью контекстного меню, назначим анимацию движения. Щелкнем правой кнопкой мыши по верхнему слою и в контекстном меню установим тип.

Mask. Теперь протестируем ролик, нажав ***CTRL+Enter***. Анимация происходит но выглядит абсолютно плоской и мало похожа на глобус. Закроем тестовое окно и попробуем устранить отмеченный недостаток. Придадим нашей фигуре объемный вид. При активном слое Маске создадим новый обычный слой. Присвоим ему имя ***Shade*** — Тень. Здесь изобразим круг с полупрозрачной заливкой, которая будет выполнять роль тени. Чтобы не рисоват фигуру заново и совмещать ее с маской, поступим следующим образом. Снимем блокировку со слоя Маски, выделим круг, скопируем его в буфер обмена, перейдем в верхний слой и вставим из буфера копию фигуры командой ***Edit-Paste in Place*** — Вставить на место.

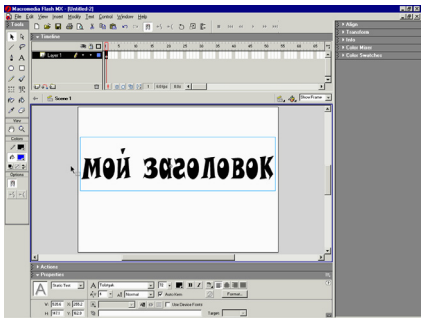
Снова заблокируем слой маски, чтобы случайно его не изменить. Зделаем также нижние слои временно невидимыми и поработем с фигурой в верхнем слое. Теперь нам нужно залить круг полупрозрачной градиентной заливкой. Выберем инструмент ***Paint Bucket*** и с помощью панели ***Mixer*** настроим нужную заливку. Установим на панели радиальную градиентную заливку, щелкнем по левому маркеру и введем непрозрачность цвета 0%. Добавим еще один маркер, щелкнув под линейкой перехода цвета. Выберем черный цвет, а непрозрачность установим равной 60%. Для третьего маркера также назначим черный цвет и сделаем непрозрачность равной 90%.

Щелкнем инструментом заливка, где-нибудь левее и выше центра круга. Фигура будет залита круговым полупрозрачным градиентом. Включим

видимость всех слоев и заблокируем их, щелкнув по пиктограмме замка. Осталось только протестировать ролик, если сделать анимацию более плавной, т.е. увеличить скорость воспроизведения, то она будет неплохо смотреться на какой-нибудь ***WEB***-страничке. А сейчас сделаем перерыв.

*5.1.3.5. Примеры применения текста при маскировании (**II** уровень)*

На этом уроке мы продолжим изучение приемов маскирования и создадим несколько роликов, где в качестве маски используется текст. Создадим, например текстовый заголовок, в котором буквы будут переливаться всеми цветами радуги. Выберем инструмент Текст и на панели ***Properties*** подберем подходящий шрифт, размер, начертание и напишем что-нибудь на сцене. В целях экономии времени я вставлю заготовленный текстовый блок из буфера обмена. Цвет текста в данном случае роли не играет, поскольку мы планируем использовать заголовок в качестве маски.



Дадим единственному пока слоя имя ***Mask***. Анимация расчитана примерно на 40 кадров, поэтому выделим 40 позицию шкалы и клавишей ***F***5 вставим цепочку простых кадров. Создадим новый обычный слой и сразу перетащим его вниз, так как маска должна быть расположена выше маскируемого слоя. Заблокируем слой маски.Выберем инструмент Прямоугольник, сбросим цвет контура, и определим линейную многоцветную заливку из стандартного набора ***WEB***-палитры. Нарисуем прямоугольник такого размера, чтобы он перекрывал наш текст. Напомню, что градиентную заливку нельзя использовать как маску, но с успехом можно применить в качестве маскируемого элемента. Нажмем клавишу ***V*** и инструментом Стрелка выделим фигуру. Сочетание ***CTRL+G*** сгруппируем ее.

Далее скопируем заливку в буфер обмена и выполним команду ***Edit-Paste in Place***, создав ее копию в том же месте. Не снимая выделение, переместим копию влево настолько, чтобы два прямоугольника оказались состыкованы друг с другом. Затем выполним команду ***Modify-Transform-Flip Gorizontal*** — Отразить по горизонтали. Удерживая клавишу ***Shift*** выделим оба прямоугольника и нажмем клавишу ***F***8. Установим тип ***Graphic*** и щелкнем по кнопке ОК. Теперь в нижнем слое можно создавать раскадровку движения. Выделим 20 кадр и клавишей ***F***6, создадим здесь ключевой кадр. То же самое сделаем в последнем 40 кадре анимации. Снова перейдем в 20 кадр и при

помощи клавиши со стрелкой сместим символ заливки максимально вправо, но только, чтобы он не вышел за рамки текста.

Теперь щелкнем правой кнопкой мыши и установим в 20 кадре раскадровку движения. Ту же операцию проделаем в первом ключевом кадре. Подвигаем маркером вдоль шкалы и посмотрим как перемещается градиентная заливка относительно текста. И последнее, что нам осталось, так это сделать верхний слой Маской. Щелкнем по нему правой кнопкой мыши и в контекстном меню выберем пункт ***Mask***. Протестируем ролик и посмотрим результаты работы. Буквы заголовка непрерывно меняют свой цвет, чего мы собственно и добивались. Закроем тестовое окно. Хочу таже отметить,что в качестве маскируемого элемента, вы с таким же успехом можете использовать и любое растровое изображение. При этом картинку можно оставить неподвижной, а анимировать сам текст, предварительно преобразовав его в символ.

Давайте коротко проанализируем готовый подобный файл, который у меня уже открыт в программе. Для его отображения воспользуемся командой меню ***Window*** и сделаем активным файл ***Garden*** — Сад. Здесь также имеются два слоя: слой маска и маскируемый слой. В нижнем слое находится неподвижное растровое изображение, отображаемое цепочкой простых кадров.

В слое маске расположена текстовая строка, преобразованная в графический символ. Экземпляру символа применена раскадровка движения с изменением масштаба и поворотом объекта на один поворот по часовой стрелке. В результате протестировав ролик, мы увидим следующий эффект. Закроем тестовое окно и в заключении этого урока приведем еще один, чуть более сложный пример текстового эффекта с использованием маски. Создадим новый ролик и отмасштабируем рабочую область сочетанием ***CTRL***+2. С помощью инструмента Текст создадим какую-нибудь надпись подходящего размера и цвета. Я же вставлю заготовленный текстовый блок из буфера. Для улучшения контрастности, с помощью панели Инспектора Свойств изменим цвет полотна на черный.

Создадим новый слой, который в дальнейшем будем использовать в качестве маски. Выделим текст в нижнем слое и скопируем его в буфер обмена. Сделаем текущим верхний слой и вставим в него содержимое буфера командой ***Edit-Paste in Place*** Вставить на место. Заблокируем нижний слой. В маске будем применять раскадровку формы, поэтому необходимо

преобразовать строку в заливку в виде текстовых символов. Для этого выполним команду ***Modify-Break Apart*** — Разбить. Первоначально редактор разбивает текстовую строку на отдельные символы, поэтому не снимая выделения, выполним ту же самую команду еще раз. Теперь мы получили то, что нужно для реализации морфинга. Выделим на шкале 12 позицию и клавишей ***F***6 скопируем туда первый ключевой кадр. Перейдем в первый кадр и с помощью модификатора ***Scale*** Масштаб сильно сожмем графику по вертикали, а затем с помощью клавиатуры, немного подымем ее так, чтобы она не касалась текста в нижнем слое.

На панели ***Properties*** установим анимацию формы. Далее выделим 18 позицию на шкале и нажмем клавишу ***F***5. Для выравнивания слоев по длительности выделим также 18 позицию нижнего слоя и вставим туда обычные кадры клавишей ***F***5. Теперь осталось только сделать верхний слой маской. Для этого щелкнем по нему правой кнопкой мыши и выберем пункт ***Mask***. Протестируем ролик и посмотрим, что у нас получилось. Создается впечатление, что пустая текстовая форма, постепенно заполняется цветной жидкостью.

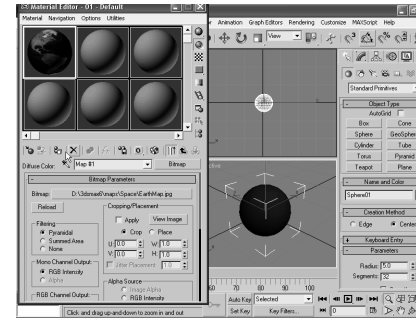
В заключении хотел бы отметить, что показанные в данной главе примеры, далеко не исчерпываю возможности использования слоев масок. Маски например часто применяются для реализации эффектов печатной машинки, линзы и многих других. Разобравшись с принципом действия масок, вы сможете теперь самостоятельно воплотить в жизнь всои собственные идеи. Вообще говоря самый лучший способ найти интересные идеи, путешествие по ***Flash***-сайтам с целью увидеть, понять как это делается и взять на вооружение интересный прием. При этом никто н обвинит вас в плагиате. Что касается текстовых эффектов, то иногда не стоит повторно изобретать велосипед.

Если вам понадобиться какой-либо творческий текстовый эффект, то можно воспользоваться программами сторонних компаний, живущих за счет таких именно разработок. Из наиболее известных продуктов подобного рода следует отметить программу ***SWFx***, позволяющую создавать, в настоящее время, более 250 текстовых эффектов. И программу ***FLAx*** имеющую 25 типов различных текстовых эффектов, которые к тому же поддаются самым разнообразным видам пользовательской настройки.

Эти программы позволяют создавать файлы в формате *.***SWF**, которые вы можете использовать самостоятельно или импортировать в среду ***Flash***. Давайте коротко взглянем на интерфейс программы ***FLAx***. В указанном списке выбирается нужный эффект, который в реальном масштабе времени отображается на экране. Остальные опции, расположенные на трех панелях, позволяют настроить каждый эффект индивидуальным образом и сделать его уникальным. Например, вот так. На этом текущий урок закончим.

5.2. Трехмерная графика. Создание рекламного ролика в 3*D Max* 6

Рассмотрим теперь такое интересное применение 3***D Max***, как создание рекламных роликов. Конечно же, надо сказать, что рекламные ролики иногда приводят нам просто-напросто в ярость. Особенно, когда прерывают показ по телевизору какого-нибудь интересного фильма или напряженного матча, спортивного состязания. Но, тем не менее, реклама основательно и безоговорочно вошла в нашу жизнь и прочно заняла в ней свое место. И без нее теперь уже невозможно представить себе наше телевидение. Попробуем мы тоже идти в ногу со временем и создать какой-либо рекламный ролик, который будет представлять собой не очень сложную конструкцию, например, пусть вокруг земли вращается логотип какого-либо текста, который будет являться, конечно же, название нашей фирмы.

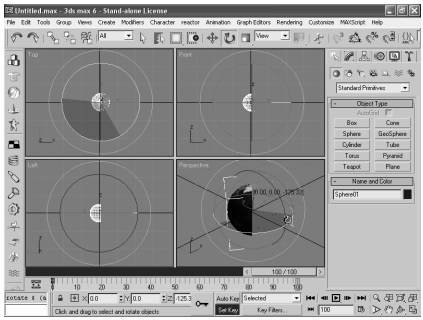


Попробуем непосредственно начать создание этого ролика. Ну, конечно же, первое, что нам необходимо сделать, это создать изображение самой наши земли. Для этого достаточно просто-напросто нарисовать обычную сферу. Для этого переключимся на разряд стандартных примитивов и возьмем вот эту сферу. Нарисуем ее. В принципе это достаточно стандартный процесс. Зададим радиус для земли. Пусть будет, например, 5 единиц. Конечно же, в таком виде это изображение будет малым, поэтому немного увеличим изображение при помощи кнопки ***Zoom***. Воспользуемся, например, кнопкой, чтобы одновременно увеличивалось изображение во всех наших проекциях. Приведем, например, на нашей сцене увеличение, вот к такому виду. Конечно же, в таком виде оставлять нашу землю абсолютно не логично, надо, чтобы она выглядела, как земля. Для этого надо воспользоваться стандартными текстурами из библиотеки или загрузить стандартный материал. Для этого щелкнем на кнопку ***M***, чтобы вызвать редактор материалов. Вот он появился перед нами. Выберем первый из них. Далее щелкнем на свитке карт и в качестве диффузной карты, щелкнув на кнопку ***None***, выберем материал, который по нашему мнению наиболее подходит к земле. Щелкнем теперь на кнопку, чтобы загрузить какую-либо картинку, соответствующей земле. Щелкнем дважды. Выберем в качестве фильтра два наших рисунка, рисунки ***JPGA*** и откроем папку ***Maps***. Щелкнем дважды. Далее откроем папку ***Space*** и здесь выберем такую карту, как — карту

земли. Вот можно ее видеть на нашей сцене. Щелкнем теперь на кнопку ***Open***, и можно увидеть, что мы ее применили на нашем материале. Вот он появился здесь. А теперь присвоим этот материал нашему шару, нашей земле. Для этого можно просто-напросто перетащить или же щелкнуть на вот эту кнопку. Щелкнем на этот раз сейчас на вот эту кнопку.

Как мы видим, материал применился, но, правда, он не отобразился у нас на окне перспективы. В окне проекции. Для того чтобы это исправить, достаточно щелкнуть на эту кнопку и в этом случае, мы нашу землю как раз и можем видеть на нашей сцене уже совершенно в нормальном виде. Закроем теперь это окно. Оно нам уже не нужно. И теперь попробуем сделать так, чтобы во время анимации наша земля вращалась вокруг своей оси. Для этого нам, конечно же, нужно создать ключи анимации. Щелкнем на кнопку и далее щелкнем на вот этом ключе анимации, для того чтобы зафиксировать первое положение в нулевом фрейме.

Далее передвинем наш бегунок времени в последний 100 кадр. И теперь повернем наш земной шар на 360 градусов, сделав полный оборот. Для этого, конечно же, нам нужно выбрать пункт поворота, и далее включим режим поворота. Вот, начнем поворачивать. Можно увидеть, насколько градусов мы при этом поворачиваем наш объект. Вот, в результате, после нескольких попыток, можно достичь числа 360 градусов. Щелкнем теперь на кнопку, чтобы зафиксировать и этот второй ключ анимации, и отпустим вот эту кнопку. Теперь у нас уже два ключа анимации есть. В нулевом и в 100 фрейме. Щелкнем теперь на кнопку ***Play***, чтобы посмотреть, что у нас получилось. В принципе, как мы видим, земля вращается вокруг своей оси, правда, вращается она не очень хорошо. Так сказать рывками, щелкнем на кнопку ***Stop***, и посмотрим, в чем же дело. Дело в том, что вращение нашей земли происходит не равномерно, а рывками. В начале, она так сказать при старте разгоняется и в конце, около 100 фрейма она замедляет свое вращение.



Это, конечно же, неправильно. Вращаться она должна с постоянной скоростью, как мы знаем из практики. Посему нам придется отредактировать соответствующий трек нашей анимации, в котором у нас зафиксированы повороты. Поэтому щелкнем на этой кнопке и вызовем окно редактора треков. Появляется вот это стандартное окно, в котором мы можем видеть, траектории движения нашей земли. Его вращение. Как мы видим, все наши

предположения оправдались. У нас начинается все с разгона и кончается торможением. Придется это исправить. Для этого выделим первую из этих точек. Щелкнем на ней. И в качестве потенциональной составляющей выберем линейный вариант. Щелкнем. Вот, как мы видим, около этой точки у нас получилось именно то, что мы хотели, линейное нарастание поворота.

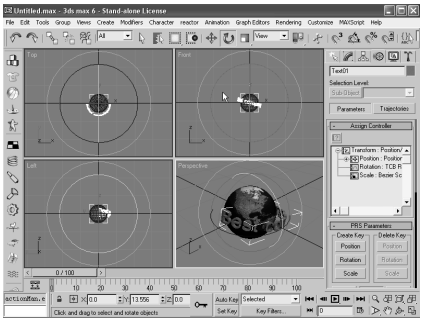
Ну и то же самое нам придется сделать относительно второй точки. Щелкнем здесь тоже на линейном варианте и вот такую прямую линию, нам как раз и нужно для наших поворотов. Исправим еще одну неточность, которая у нас закралась. Конечно же, нам нужно 360 градусов, а не 360 с хвостиком. Уберем, так сказать вот этот хвостик, и закроем окно. Оно теперь нам уже не нужно. Попробуем теперь посмотреть, что у нас получилось. Щелкнем на кнопку ***Play*** и, как мы видим, сейчас вращение у нас равномерное. Остановим нашу анимацию. И попробуем провести дальнейшие шаги по улучшению внешнего вида нашего ролика.

Теперь создадим некий текст, который будет вращаться вокруг нашей земли. Для этого перейдем в вид ***Front***, далее переключимся на пункт ***Shapes***, и здесь у нас есть такая кнопка, как ***Text***, которая и поможет нам для написания текста. Вот мы на ней щелкнули. Поместим его на нашей сцене. Вот, например, вот таким образом. В принципе, конечно же, размеры текста нас совершенно не устраивают. Поэтому сдвинем всю эту панель чуть выше и здесь в качестве размера возьмем не 100 единиц, а гораздо меньше. Например, возьмем 3 единицы. В этом случае, конечно же, размер текста уже больше похож на то, что мы хотели. Отредактируем сам текст. Конечно же, наша компания самая лучшая. Добавим теперь этому тексту объем. Для этого нам понадобится использовать модификатор. Переключимся на режим модифицирования. Щелкнем на выпадающем списке модифицирования. И здесь нам нужен, будет модификатор вытягивание. Вот он появился перед нами. Щелкнем на нем. И теперь нам нужно задать размер вытягивание.

Сейчас, как мы видим, здесь у нас стоит 0. Но, конечно же, 0 нас совершенно не устраивает, Возьмем, например, 0.5. В результате чего, наш текст приобретает некий объем. Теперь же нам необходимо обернуть вот этот текст вокруг нашего земного шара. Для этого воспользуемся еще одним модификаторов, модификатором согнуть. Щелкнем на вот этом выпадающем списке выбора модификатора и найдем этот модификатор. Вот он появился перед нами.

Щелкнем на нем. А теперь, первое, что зададим, это угол поворота. Зададим его в 180 градусов, половину полного круга. И поворот создадим относительно оси ***X***. Теперь же совместим наш вот этот текст с нашей землей. Для этого включим режим перемещения и далее в этой фронтальной проекции щелкнем на кнопку перемещения и вот таким образом выровняем его по горизонтали.

Теперь выровняем еще и в виде сверху. В принципе, вроде все в порядке. Хотя надо сказать, что мы слишком уж прижали наш текст к нашей земле. Лучше его сделать немного больше. Поэтому выделим вот этот текст. И размер его зададим не 3 единицы, а, например, 3.5, немного больше. Далее теперь придется нам опять немного его скорректировать, сдвинув его вот сюда вниз.



Теперь же нам необходимо совместить базовую точку нашего текста с центром сферы. Щелкнем для этого на вот этой вкладке Иерархия и далее щелкнем на вот этой кнопке. Вот появляется точка, которая и является его базовой. Конечно же, нам ее необходимо переместить в центр сферы. Вот таким образом. Попробуем теперь выровнять вот эти наши объекты. Для этого щелкнем на кнопку ***Tools*** и далее ***Align***. Далее щелкнем вот в этом месте. Появилась возможность выровнять и здесь. Проведем следующие настройки. Выберем выравнивание по всем трем осям. И выравнивание по базовым точкам. Щелкнем теперь на кнопку ОК. Теперь же нам необходимо сделать так, чтобы вот этот текст вращался бы вокруг вот этой земли. Для этого проделаем несколько шагов. Первое, что сделаем, повернем вот этот текст, чтобы он вращался не так, горизонтально, а немного наклонно. Для этого достаточно выбрать пункт поворота.

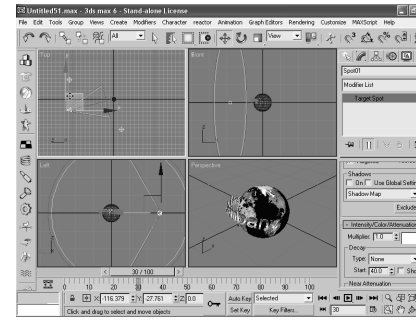
Выберем вот этот текст. И немного его наклоним в соответствующую нужную нам сторону. Например, вот таким образом. Не очень сильно. Теперь же, перейдем в контрольную панель движение и выберем, конечно же, пункт — вращение. Щелкнем теперь на вот этом свитке, выберем вот этот пункт и щелкнем на вот эту кнопку. Выберем такой контроллер. Щелкнем на ОК. Вот, как мы видим, вот этот поворот зафиксировался у нас в качестве контроллера для нашего текста.

Теперь же передвинем бегунок времени на начальный, нулевой кадр и попробуем повернуть так, вот этот текст, чтобы он в начальном кадре оказался бы позади нашей сферы, нашей земли. Щелкнем теперь на пункте поворота, и выберем еще и локальную систему координат. Для этого щелкнем на вот этом выпадающем списке и вместо системы координат, связанной с нашим видом, возьмем локальную. Вот, как мы видим, по вот этим наклонным осям, эта система координат совпадает с плоскостью нашего текста. А теперь будем его поворачивать. Первое, что нам необходимо сделать, как мы сказали, это убрать

вот этот текст, чтобы в начале нашей анимации он находился за нашей землей. Повернем вот таким образом. И зададим первый из ключей анимации.

Для этого щелкнем на кнопку ***set Key*** и на ключе анимации, который у нас создался вот в этом нулевом фрейме. А теперь же сдвинем вот этот свиток вверх и изменим параметр ***Continuity***. Вместо 25 установим 0, для того чтобы у нас течение шло с постоянной скоростью, а не с ускорением и замедлением. Вот, как мы видим, у нас получился здесь линейный график. Теперь же перейдем здесь на следующий кадр, 50, и создадим здесь еще один ключ анимации. Для этого повернем теперь наш текст вот таким образом. Он у нас должен появиться. Будем тащить, пока не достигнем чуть меньше, чем 1280 градусов. Вот, приблизительно до такого состояния.

И щелкнем на ключе анимации, чтобы его создать. Вот в этом фрейме. Щелкнем. Точно также изменим и вот этот параметр, и опять для равномерности движения установим его в 0. И теперь осталось сделать последний ключ анимации, в последнем 100 фрейме. Передвинем наш бегунок времени дальше, к последнему кадру. Опять будем двигать и поворачивать. И тоже еще на приблизительно 179 градусов, чуть меньше, чем 180, нам нужно добиться. Ну и, конечно же, щелкнем на ключе анимации, создавая соответствующий ключ. Теперь изменим еще и равномерность вращения. Установим здесь 0 и теперь у нас вращение вот этого текста, уже должно быть готово.



Посмотрим, как все это будет выглядеть. Выключим этот режим, и щелкнем на кнопку ***Play***. Как мы видим, у нас все более или менее совпадает с тем, что мы задумали. Текст вращается вокруг вот этой земли, которая вращается в другую сторону. Остановим анимацию. Щелкнем на кнопку ***Stop***. Добавим теперь на нашу сцену пару источников света. Для этого перейдем в пункт ***Create*** и выберем, конечно же, кнопку. Ну и, нам нужен источник света. Щелкнем на нем. Теперь, прежде чем установим его на нашей сцене, немного изменим масштаб изображения вот в этом виде Top. Для этого выберем пункт ***Zoom***. Щелкнем. Далее немного вот таким образом изменим и поместим соответствующие осветители. Пусть один из них находится вот в этом месте, а второй сзади и левее. Сдвинем чуть повыше вот эти источники света.

Для этого щелкнем на кнопке ***Move*** и установим ее координату ***Z***. Например, возьмем 50 единиц. То же самое сделаем со вторым нашим

источником света. Тоже его приподнимем. Вот таким образом, получая приемлемое для нас расположение наших объектов и источников света. Зададим теперь нашей сцене фоновое изображение. Для этого щелкнем на пункте меню ***Rendering*** и далее ***Environment***. Появляется вот это стандартное окно, в котором мы можем установить цвет для фона или же какое-либо изображение. Щелкнем на кнопку, чтобы сейчас мы могли использовать какую-либо фотографию для фона. Щелкнем. В результате появляется вот такое стандартное окно, в котором щелкнем на кнопку, чтобы выбрать какую-либо из фотографии.

Щелкнем на ней дважды. Появляется вот это окно, в котором мы можем выбрать какую-либо из картинок. Выберем, например, вот эту. Щелкнем на кнопку ***Open***, чтобы зафиксировать. И вот эта фотография появилась у нас в качестве фона. Закроем теперь это окно. И зададим еще и материал для вот этих наших букв. Поэтому щелкнем на клавише M, чтобы вызвать опять окно редактора материалов. Выберем вот этот второй материал. Далее щелкнем на свитке карт и выберем диффузную карту. Щелкнем на этой кнопке, и выберем один из стандартных библиотек. Сдвинем теперь вот этот весь список и найдем такую карту, как, например, этот. Щелкнем на ней дважды и далее применим этот материал к нашему тексту.

Щелкнем теперь еще на эту кнопку, для того чтобы мы могли видеть, что у нас получилось. Как мы видим, картина стала гораздо более живописной. Отметим еще одно обстоятельство. Если приглядеться к нашей сцене. Особенно, если передвинуть наш бегунок времени, то можно увидеть, что тот материал, который мы наложили на наш шар, который и должен был играть роль земли, имеет некий недостаток. А именно, он не очень уже хорошо аппроксимирует рисунок нашей земли. И поэтому вот в этом месте, в месте соединения образовался вот такой, видимый на глаз, шов. В принципе, это плохо выглядит. Поэтому попробуем с этим что-то сделать.

Конечно, лучше было бы, если мы имели рисунок лучше. Но сейчас поступим несколько по-другому, используя тот рисунок, который у нас есть. Для этого перейдем в первый материал. Далее сдвинем вот этот весь свиток вверх и используем несколько другое расположение нашего рисунка по нашему объекту. Выберем флажок на этом пункте. В этом случае, каждый следующий рисунок, при раскладывании их на объекте, будет разворачиваться. Щелкнем. И посмотрим, что в результате получилось. Как можно видеть, на этот раз шва не видно. Правда, в результате мы получили не совсем правильное изображение нашей земли. Но, в принципе, для наших целей, это вполне пойдем. Закроем теперь это окно.

Попробуем теперь дальше усовершенствовать нашу сцену, добавляя различного рода эффекты. Первый из них, что мы можем сделать, это создать

яркий прожектор, луч которого будет освещать каждую букву нашего текста, от чего, конечно же, наша сцена достаточно сильно выиграет. Выберем направленный теперь прожектор. Щелкнем, и расположим его на нашей сцене. Расположим его, например, вот в этом месте и направим его луч света, конечно же, в сторону нашего текста и нашей земли. Переключимся на вкладку ***Modify*** и далее, первое, что сделаем, попробуем лучше нацелить наш прожектор на вот этот наш текст, найти буквы.

Поэтому сдвинем весь этот свиток наверх и здесь установим вот эти параметры поменьше. Возьмем, например, 1. А в качестве наружного радиуса возьмем число 3. А теперь еще схватим вот эту цель нашего прожектора и лучше направим его на наши буквы, вот таким образом. Теперь же переключимся опять на наш прожектор и включим отображение тени. Поставим флажок на тенях. Посмотрим теперь, что в результате всего этого у нас получилось при визуализации.

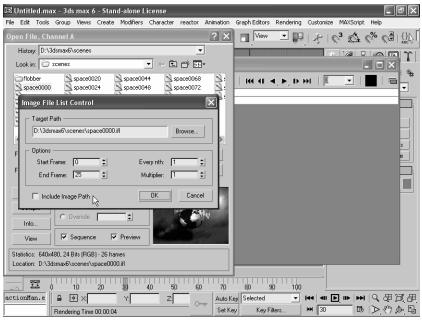
Для этого щелкнем на кнопку ***Rendering***, далее ***Render***. Конечно же, весь промежуток сейчас нам не нужен, мы хотим посмотреть всего лишь одиночный кадр, чтобы получить быстрее результат. Поэтому щелкнем на кнопку ***Render***. И вот можно увидеть, что у нас в результате всего этого получилось. Как мы видим, в принципе все достаточно хорошо. Все те эффекты, которые мы задавали, у нас как раз пока и получились. Закроем теперь это окно. Закроем и окно Визуализации тоже. И вот мы опять попали в наше стандартное окно 3***D Max***.

Мы теперь до конца исполнили все наши желания на нашей сцене. Осталось в принципе добавить еще какую-нибудь изюминку, в качестве которого добавим на нашу сцену, эффект вспышки. Это у нас будет, как бы солнце на нашей сцене. Для этого, сначала создадим вспомогательный объект пустышку, для того чтобы в дальнейшем нам было, куда привязывать наш спецэффект. Поэтому переключимся на вкладку ***Create*** и далее щелкнем на кнопку ***Helpers***.

И здесь как раз и находится наша пустышка. Щелкнем на ней и разместим ее на нашей сцене. Например, вот в этом, в правом углу нашей сцены. Теперь откроем окно видеомонтажа. Для этого щелкнем на пункте меню ***Rendering***. И далее ***Video Post***. Щелкнем. Появилось вот такое стандартное окно, в котором нам нужно добавить сначала визуальный вид. Для этого щелкнем на вот этой кнопке. Щелкнем. В принципе, ничего здесь менять не будем. Щелкнем на ОК. Добавим еще и фильтр изображения. Щелкнем. Появляется вот такое окно, в котором выберем как раз линзовый эффект. Вот этот — вспышка. Щелкнем. И теперь щелкнем на кнопку ***Set Up***, чтобы настроить его внешний вид. Вот, появилось вот такое окно настройки линзовых эффектов. Здесь, как мы видим, очень много параметров, настраивая которые можно получить различные эффекты. Сейчас настраивать особо ничего не будем. Загрузим при помощи

кнопки ***Load*** какой-либо из заранее заготовленных линзовых эффектов, которые приготовили для нас разработчики 3***D Max***. Щелкнем на кнопку ***Load***. В принципе появилось вот такое стандартное окно, в котором выберем один из линзовых эффектов.

Например, вот этот 6. Щелкнем на кнопке ОК. Вот в принципе, его настройки появились на нашей сцене. Чтобы увидеть. как же все это будет выглядеть у нас на экране, можем щелкнуть на кнопку ***Preview***. И вот можно увидеть, что у нас должно получиться в добавок к нашей сцене на экране, на картинке. Конечно же, в принципе, все достаточно симпатично. Теперь же, щелкнем на кнопку, чтобы привязать этот линзовый эффект к нашей пустышке, который мы уже создали для этого эффекта на нашей сцене. Щелкнем. Вот появляется вот такое окно выбора, окно ***Selection***, в котором, конечно же, нам нужен вот этот объект.



Щелкнем на кнопку ОК. Он появился здесь у нас в соответствующем месте. Щелкнем теперь на кнопку ОК. Посмотрим, как все это будет выглядеть при визуализации. Щелкнем на вот этой кнопке. Появилось вот такое стандартное окно, в котором для простоты сначала возьмем простой кадр. И возьмем, например, 22 фрейм. Щелкнем на кнопку ***Render*** и можно увидеть, что в результате у нас получается на нашей сцене. В принципе, все достаточно эффектно выглядит. Закроем теперь это окно. Теперь же настроим файл, в котором мы будем сохранять нашу анимацию. Для этого щелкнем на вот этой кнопке. Появляется вот такое стандартное окно, в котором, щелкнув на кнопку ***File***, мы можем настраивать файл для вывода, в котором мы будем сохранять всю нашу информацию.

Поскольку сейчас, как мы видим, у нас здесь стоит эта надпись, что означает, что пока мы, даже если проведем ***Rendering***, никуда это записываться не будет. Щелкнем на этой кнопке. Вот появилось вот это стандартное окно, в котором можем выбрать на какой-либо из форматов для вывода нашей сцены, нашей анимации. Можем, конечно же, все это записать в виде ***avi***. файла, но попробуем сейчас сделать по-другому. А именно, записать все это, как совокупность ***BMP*** изображения. Выберем для них какое-либо имя. Например, назовем это так. Щелкнем на кнопку ***Save***. Возьмем, конечно, 24 бита на одну точку для цвета, 256 цветов это, конечно же, очень мало. Щелкнем на кнопку ОК. Щелкнем теперь на кнопку ОК.

А теперь попробуем провести визуализацию нашей сцены. Для этого щелкнем на вот этой кнопке и выберем на этот раз, конечно же, не одиночный кадр, а совокупность всей нашей анимации от 0 до 100 фрейма. Ну и, возьмем, конечно же, не все фреймы, а, например, через каждые 4 фрейма. Щелкнем теперь на кнопку ***Render***. Визуализация прошла успешно. Закроем теперь вот это окно видеомонтажа. Запустим ***Ram Player***. Щелкнем на кнопку ***Rendering***, далее ***Render***. Щелкнем на кнопку ***Open***, открыть. Вот появились вот эти файлы до последнего кадра. Как мы видим, вся наша анимация записалась в виде отдельных рисунков, но в принципе, каждый из них мы можем посмотреть, как будет выглядеть. Щелкнув, можно увидеть, как он выглядит.

В принципе можно щелкнуть на кнопку ***View*** и посмотреть все это в увеличенном нормальном виде. Но, конечно же, нас сейчас интересует посмотреть в анимации. Для этого поставим флажок, для того чтобы нам компьютер собрал все вот эти файлы в один видеоряд. Как мы видим, сейчас у нас 26 фреймов. Щелкнем на кнопку ***Open***.

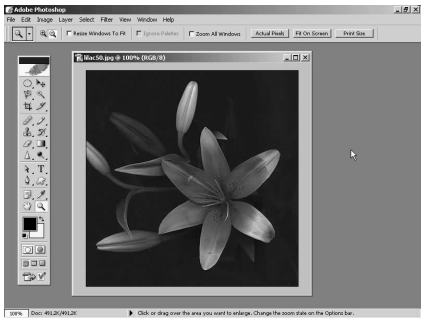
Появилось вот такое стандартное окно, в котором нам говорится, что мы хотим собрать от 0 до 25 фрейма. Согласимся. ОК. Согласимся и со всеми этими стандартными установками. ОК. Вот, перед нами появился стартовый первый файл нашей анимации. Щелкнем теперь на кнопку ***Play***. И можно увидеть, что у нас в результате получилось. Конечно же, наша земля вращается и вокруг нее вращается вот эта вся надпись. Но получилось как раз то, что мы хотели.В принципе, конечно же, у нас есть и линзовый эффект и фоновое изображение, и все наши эффекты. Все достаточно хорошо и красиво. Закроем теперь это окно. Щелкнем на вот этом крестике и вот мы опять попали в наше стандартное окно 3***D Max***.

5.3. Создание изображений для Web в программе Adobe Photoshop CS

5.3.1. Форматы графических файлов для Web

Если вы занимаетесь ***WEB*** дизайном, или просто решили создать свою ***WEB*** страницу, и хотите разместить на ней какие-то изображения, то вам придется руководствоваться следующим правилом: размер каждого размещаемого на ***WEB*** странице изображения должен быть как можно меньше. Как можно меньше не в смысле размера картинки, которая будет показана на экране монитора, а в смысле объема файла, содержащего эту картинку. Если объем файла с картинкой слишком большой, то вывод на экран ***WEB*** страницы, содержащей эту картинку, пользователем Интернет, может потребовать слишком большого времени, особенно если у пользователя модемное соединение с сетью Интернет. И такой пользователь может, не дождавшись

вывода на экран вашего замечательного изображения, просто покинуть вашу страницу, переключившись на что-либо другое.



Разумеется, делая размер файла с изображением, как можно меньше, нужно, при этом, следить за качеством изображения. То есть, чтобы при уменьшении размера изображения, все, что вам бы хотелось в нем представить на экране, осталось бы различимым. И художественные достоинства исходного варианта картинки не сильно бы пострадали. Создание из изображения его варианта для размещения на ***WEB*** странице, с соблюдением приемлемого баланса между качеством нового изображения и размером файла с ним, называется оптимизацией изображения для ***WEB***. Набор форматов графических файлов, которые можно поместить на ***WEB*** страницу, довольно ограничен. Для растровой графики, это, по большому счету, всего два формата: формат ***JPEG***, и формат ***GIF***.

Для файла в формате ***JPEG*** расширение имени этого файла, может быть одним из трех перечисленных здесь видов. При сохранении изображения в формате ***JPEG***, как правило, происходит потеря графической информации, содержащейся в исходном изображении. Потому что при сохранении в этом формате, сохраняются, как правило, не все пиксели изображения, а только, так сказать, опорные. И при последующем выводе этого изображения на экран из файла, цвета промежуточных точек между этими опорными, получаются интерполяцией.

Чем меньше этих опорных точек, при сохранении картинки в формате ***JPEG***, тем меньше получается объем файла с изображением, но и тем дальше, получающаяся картинка от исходной. И тем больше в ней потеря исходной графической информации, и, как следствие, тем хуже получающаяся картинка. При сохранении изображения в формате ***GIF***, который тоже во многих случаях позволяет существенно уменьшить объем файла, потеря информации о цветах пикселей исходной картинки, не происходит. Но, только если этих цветов, в исходной картинке, не более 256. А вот формат ***JPEG*** позволяет сохранять свыше шестнадцати миллионов цветов. Формат ***JPEG*** больше подходит для оптимизации фотографий. А формат ***GIF*** для схематических изображений. Или для изображений, содержащих ограниченное количество цветов. В частности, его можно использовать для сохранения черно-белых

фотографий. Формат ***GIF***, помимо уменьшения размера файла, имеет еще три замечательных возможности.

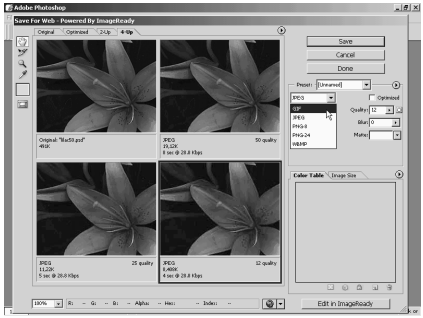
Изображение, сохраненное в формате ***GIF***, может содержать области прозрачности. Сквозь эти области прозрачности в ***GIF*** изображении будут видны другие элементы ***WEB*** страницы. Сохраняя изображение в формате ***GIF*** можно сохранить его с последующим поэтапным выводом его на экран. При таком поэтапном выводе ***GIF*** изображения на экран, при просмотре ***WEB*** страницы, на экран выводится сначала грубый эскиз изображения, который поэтапно все более и более уточняется. При этом, тот, кто просматривает эту ***WEB*** страницу, уже может, не дожидаясь конца загрузки этого изображения, получить о нем приблизительное представление, и решить, стоит ли ждать окончательного вывода этого изображения на экран, или отключить для этой страницы вывод изображения, и ознакомится с другой информацией на этой странице.

И третий бонус формата ***GIF***. Этот формат позволяет сохранять простейшие анимации изображений. Все перечисленные возможности оптимизации изображений для ***WEB*** страниц, реализованы в программе ***PHOTOSHOP***. Кроме, создания анимированных изображений в формате ***GIF***. Сам ***PHOTOSHOP*** анимировать изображение не умеет. Но подобные изображения можно подготовить в программе ***PHOTOSHOP***, в виде отдельных кадров, расположенных на разных слоях изображения. А затем, для автоматической смены этих кадров, переключится в программу ***IMAGE READY***, которая всегда поставляется вместе с программой ***PHOTOSHOP***.

Переключение в программу ***IMAGE READY*** может происходить прямо в сеансе работы с программой ***ADOBE PHOTOSHOP***, с помощью вот этой пиктограммы в палитре инструментов. Переключаемся. Делаем в программе ***ADOBE IMAGE READY*** то, что нам надо. И можно возвращаться обратно в программу ***PHOTOSHOP***.

5.3.2. Команда Save For Web. Инструменты и режимы просмотра команды

Чтобы оптимизировать изображение для ***WEB***, нужно в меню ***FILE*** выбрать команду ***SAVE FOR WEB***. Появляется диалоговая панель команды ***SAVE FOR WEB***. С помощью которой можно оптимизировать изображение для ***WEB***, и одновременно рассматривать результат этой оптимизации в окне предварительного просмотра. Окно предварительного просмотра может работать в четырех режиMAX. Каждому из этих режимов соответствует специальная закладка, в диалоговой панели команды ***SAVE FOR WEB***.



По умолчанию, активна закладка ***OPTIMIZED***. Что означает, что в окне предварительного просмотра выводится результат оптимизации и его параметры. Результат оптимизации определяется значениями параметров команды ***SAVE FOR WEB***, которые устанавливаются в правой части диалоговой панели этой команды. Если нужно сравнить оптимизированные изображения с оригиналом, то можно вывести в окне предварительного просмотра этот оригинал, переключившись на соответствующую закладку — ***ORIGINAL***. Если, переключаясь между окнами ***ORIGINAL*** и ***OPTIMIZED*** трудно заметить разницу между оптимизированным и исходным изображениями, то можно вывести на экран их оба, рядышком. С помощью закладки 2-***UP***.

После включения этой закладки окно предварительного просмотра разбивается на два кадра. В левом выводится исходное изображение. В правом — результат оптимизации. Если, при этом, изображение выводится на экран не целиком, то для просмотра нужной части изображения, можно воспользоваться инструментом рука диалоговой панели команды сохранить для ***WEB***, подвинув с помощью этого инструмента изображение в любом из кадров. Для вывода на экран нужны его части. Заметьте, что при этом, изображения в обоих кадрах, перемещаются синхронно. И на экран в обоих кадрах, выводится один и тот же фрагмент изображения. Слева, исходного, справа — оптимизированного.

При переключении на закладку 4-***UP***, на экран выводится четыре варианта исходного изображения: оригинал, само исходное изображение, оптимизированное в соответствии с выбранными пользователем установками оптимизации. А так же еще два варианта оптимизации. По умолчанию, с в два раза худшим качеством, по сравнению с установленным пользователем, и в четыре раза с худшим качеством, по сравнению с установками пользователя. Впрочем, в каждом из этих двух дополнительных окнах, в третьем и четвертом, пользователь может выбрать другие установки оптимизации, в отличие от установок по умолчанию. Например, сейчас для четвертого окна поменяем формат файла

JPEG на ***GIF***. И можно сравнить, что нам даст оптимизация в формате ***JPEG*** с тем, что нам даст оптимизация в формате ***GIF***. При переключении на закладку по умолчанию, закладку ***OPTIMIZED***, из режима 4-***UP***, в окне

предварительного просмотра будет выводиться та оптимизация, которая была активна при закладке 4-***UP***.

Сейчас активна ***GIF*** оптимизация. Переключаемся на закладку ***OPTIMIZED***. И именно результат и параметры ***GIF*** оптимизации сейчас выводятся в диалоговой панели команды сохранение для ***WEB***. И можно продолжать работать именно с этим способом оптимизации. Например, уменьшить размер файл, с оптимизированным таким способом изображением, за счет уменьшения количества цветов с 256, например, до 64. Изображение пострадало не сильно. Это оригинал. Это оптимизация. А посмотрите, на сколько уменьшился размер файла с этим изображением. Здесь 73 килобайта. А здесь 491 килобайт.

По умолчанию, в окне предварительного просмотра команды ***SAVE FOR WEB***, изображение выводится в своем реальном размере. То есть, когда каждый пиксель изображения, представляется на экране одним экранным пикселем. Именно такого физического размера и будет это изображение на ***WEB*** странице, по умолчанию, если автор ***WEB*** страницы не укажет для этого изображения другие размеры. Можно непосредственно из диалоговой панели команды сохранить для ***WEB*** посмотреть, как будет выглядеть эта картинка на ***WEB*** странице, включив команду ***PREVIEW IN DEFAULT BROWSER***, просмотр в браузере по умолчанию.

Вот, как-то так, в браузере ***MICROSOFT INTERNET EXPLORER***. Для такого просмотра программа ***PHOTOSHOP*** сама создает ***HTML*** код этой страницы. Закрываем окно просмотра оптимизированной картинки в браузере, возвращаясь в диалоговую панель команды ***SAVE FOR WEB***. О том, что сейчас картинка на экране в реальном размере можно судить по значению масштабного коэффициента ***ZOOM LEVEL***, равному сейчас 100%.

Установить иной масштабный коэффициент просмотра, можно с помощью этой команды ***ZOOM LEVEL*** в левом нижнем углу диалоговой панели команды ***SAVE FOR WEB***. Установим, например, сейчас масштаб предварительного просмотра в 200%.Такое увеличение масштаба предварительного просмотра может понадобится для того, чтобы более детально рассмотреть изменения в изображении при его оптимизации.

Это исходное. А вот такая зернистость появляется при оптимизации в формате ***GIF***. Уменьшение же масштаба предварительного просмотра может понадобиться, например, для того, чтобы обозреть все изображение целиком, наиболее подходящим значением параметра ***ZOOM LEVEL***, является значение ***FIT ON SCREEN*** — уместить на экране.

Для изменения размера, выводимой в окне предварительного просмотра, картинки, можно в поле выбора ***ZOOM LEVEL***, воспользоваться также инструментом лупа в диалоговой панели команды ***SAVE FOR WEB***. А для

уменьшения масштаба просмотра с помощью инструмента лупа, нужно, как обычно в ***PHOTOSHOP***, нажать на клавиатуре клавишу ***ALT***, при этом курсор инструмента лупа меняет свой знак "плюс" на знак "минус". И удерживая клавишу ***ALT*** нажатой, щелкнуть таким инструментом лупа по изображению.

Слева, под картинкой, в окне предварительного просмотра выводятся характеристики файла с этой картинки. Это характеристики, графический формат файла, в данном случае, это формат ***GIF***, размер файла, с оптимизированной картинкой. В данном случае, это примерно 73 килобайта. А также, сколько примерно времени, будет выводиться данная картинка на экран при указанной скорости модемного соединения. Почти пол минуты. Это многовато будет. А вот размер файла с исходным изображением. 491 килобайт. Это безумно много. Оптимизация позволила, почти в семь раз уменьшить размер файла. Но и этот размер можно еще уменьшить, выбрав в качестве графического формата не ***GIF***, который и вообще не очень подходит для данного изображения, а формат ***JPEG***.

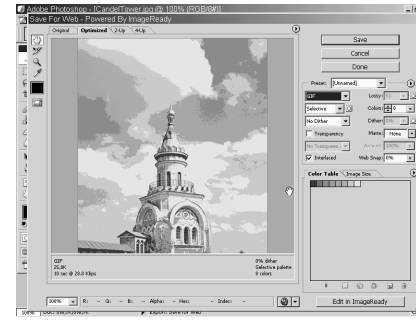
И даже с таким низким качеством сохранение в формате ***JPEG***, картинка выглядит не так уж очень плохо, а размер файла всего восемь с половиной килобайт. И выводиться такая картинка на экран будет всего лишь четыре секунды. Переключимся на закладку 4-***UP***.

Здесь на каждом из четырех вариантов, слева внизу выводятся те же самые характеристики: графический формат, размер файла, скорость загрузки на ***WEB*** странице. Сохраняя результат оптимизации в режиме 4-***UP*** или в режиме 2-***UP***, а это сохранение происходит по нажатию кнопки ***SAVE*** в диалоговой панели команды ***SAVE FOR WEB***. Так вот при сохранении нужно следить за тем, какой именно из кадров активен. Именно активный кадр и сохраняется, как результат оптимизации.

5.3.3. Сохранение для Web в формате jpg

Сохраним это изображение свечной башни Свято-Борисоглебского монастыря в г. Торжке, для размещения его на ***Web***-странице, посвященной этому городу. Выбираем в меню ***File*** команду ***Save for Web***. Поскольку исходное изображение являлось отсканированной фотографией, а значит, имело непрерывные плавные переходы цветов, То сохранять это изображение для ***Web*** нужно не в формате ***GIF***, который сейчас установлен в диалоговом окне команды ***Save for Web***, и позволяющий сохранять в изображении не более 256 цветов, а в формате ***JPEG***.

Формат ***JPEG*** позволяет сохранять в изображении широкий диапазон цветов с тончайшими градациями их яркости и насыщенности. Прежде всего, посмотрим, какой размер будет занимать эта картинка в окне браузера. Для



этого нажмем на пиктограмму ***Preview Interfold Browser*** — предварительный просмотр в браузере, установленном по умолчанию внизу окна команды ***Save for Web***. Вот такой величины будет это изображение в окне браузера ***Internet Explorer*** при разрешении экрана 800*600 пикселей. Видно, что изображение целиком не умещается в окне браузера. Это нехорошо. В большинстве случаев, картинка на ***Web***-странице должна быть видна целиком. Значит, нужно немножко уменьшить ее размер. Закрываем окно браузера ***Internet Explorer*** и уменьшим размеры этого изображения в диалоге с командой ***Save for Web***. Для уменьшения размера в пикселях изображения вовсе не обязательно закрывать диалоговое окно команды ***Save for Web*** и возвращаться в программу ***Internet Explorer***, а можно просто выбрать закладку ***Image Size*** — размер изображения.

В верхней части этой закладки выводится размер в пикселях исходного изображения. Если помните, то оно не уместилось на ***Web***-страницу по вертикали. Но для данного изображения нельзя уменьшить только вертикальный размер ***Height*** — высоту, так как оно исказится. Размеры изображения нужно уменьшать пропорционально. Значит, параметр ***Constricted Proportion*** должен быть включен.

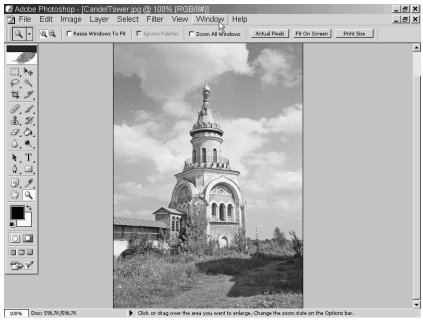
И в этом случае, достаточно указать новый размер в процентах. Введем это новое значение размеров изображения в поле параметра ***Percent*** — процент. И качество — ***Quality*** преобразования ресемплирования, изменения размера выбираем высокое, бикубическая интерполяция. Применяем к изображению преобразование ресемплирование. нажимаем кнопку ***Apply***. Снова посмотрим, как будем выглядеть картинка на ***Web***- странице. Вот теперь она полностью уместилась в окне браузера ***Internet Explorer***.

Параметр ***Quality*** определяет качество сжатого изображения. Сейчас это качество среднее, ***Medium*** — или 51 % от максимально возможного. Размер файла с картинкой при этом 37.69 Кбайт, что в пределах допустимого для ***Web***-изображений. Если выбрать максимально возможное качество, сохраняя новое изображение, то размер картинки будет 189.9 Кбайт. Это очень много. При минимально возможном качестве, размер оптимизированного изображения будет меньше 9 Кбайт. Это замечательно. Но качество изображения неважное. Появилась куча артефактов. Выберем инструмент лупаи увеличим изображение. А также, выведем в окно предварительного

просмотра рядышком, чтобы сравнить исходное и оптимизированное изображения, выбрав для этого закладку 2A.

Вот это исходное изображение, а вот тот же его кусок у оптимизированного изображения. Просмотрите, какие на нем гадкие артефакты! Вы, наверное, часто встречались с такими, разглядывая, или заимствуя чужие изображения их ***Web***. Качество такого изображения низкое, ***Low***, хотя размер его файла очень даже подходящий параметр ***Blur*** — размытие, позволяет сгладить эти артефакты. Просто путем их размытия.

Причем, подобное размытие приводит и к уменьшению размера файла с картиной. Ну и естественно, размытое изображение гораздо менее четкое, чем не размытое. Но если Вы хотите представить в ***Web*** подобную фотографию с более-менее приличным качеством, то придется найти разумный баланс между качеством изображения и размером файла, содержащего это изображение. В данном случае, это где-то 50-% качество. Посмотрите, разница почти не заметна. Особенно при выводе изображения на экран в реальном размере. Нажимаем на клавиатуре клавишу ***Alt*** и щелкаем по любому из представленных изображений. Именно так оно и будет выглядеть в ***Web***. Артефакты почти не заметны и размер меньше 32 Кбайт. Сохраняя это изображение для ***Web***, я бы не включала параметры ***Optimized*** и ***Progressive***.



Включение какого-либо из этих параметров ***Progressive*** или ***Optimized*** позволяет несколько уменьшить размер файла с изображением за счет, так сказать, более интеллектуального алгоритма сжатия ***JPEG***. Но некоторые браузеры (или старые версии известных браузеров) не смогут вывести сжатое под таким алгоритмом изображение на экран. Эти копейки, которые мы выиграем в размерах файла, не стоят того, чтобы часть посетителей ***Web***, не увидела наше замечательное изображение. Как известно, одна и та же картинка на разных экранах может выглядеть совершенно по-разному. Это зависит и от типа монитора и от его настроек. Если Вы долго трудились над цветами своего изображения и Вам хочется донести их как можно более точно до любого посетителя Вашей страницы, то включите параметр ***Icc Profile***, профиль изображения — цветовой профиль

Вашего монитора. И тогда браузер пользователя, просматривающего Ваше изображение, в соответствии с встроенным в это изображение цветовым профилем, постарается подстроить эту картинку к тому, что Вы видите у себя

на экране в программе ***Photoshop***. Но за встроенный цветовой профиль придется платить. Платить увеличением размера файла с изображением. Вот сейчас, этот размер равен 35 Кбайт. Это размер со встроенным профилем. А без профиля — 31.5 Кбайт.

Сохраняем оптимизированное изображение. Нажимает кнопку ***Save*** в диалоговой панели команды ***Save for Web***. Обратите внимание, что нажатие кнопки ***Save*** не означает автоматического сохранения оптимизированного изображения вместо исходного. Команда ***Save*** в диалоговом окне команды ***Save for Web***-это фактически команда ***Save as*** — сохранит как. Потому, что она позволяет сохранить оптимизированное изображение под именем, отличным от исходного имени и/или в другую папку, или на другой диск. Сохраним это оптимизированное изображение в ту же папку, в которой находится исходное изображение, но под другим именем. ***Candletower***1.

При таком сохранении, окно команды ***Save for Web*** закрывается, и мы возвращаемся в программу ***Photoshop***, в которой открыто и активно исходное изображение, которое мы оптимизировали. А оптимизированного изображения в списке открытых в данном сеансе работы с ***Photoshop*** изображений нет. Поэтому, если Вы хотите вывести его на экран, например, чтобы сопоставить с исходным изображение, то нужно его открыть как любое другое изображение. Меню ***File***, команда ***Open*** — открыть. Появляется диалоговое окно команды ***Open*** с содержимым папки ***Web***3, в которой помимо прочего, содержатся оба изображения свечной башни в Торжке.

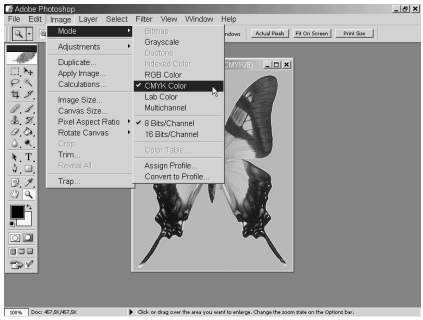
Исходное ***Candletower.JPEG*** — обратите внимание на его размер — 138.3 Кбайта и оптимизированное, его размер меньше 32 Кбайт. Открываем оптимизированное изображение. Исходное уже и так на экране. Нажимаем кнопку ***Open***. Вот это исходное изображение, а вот это — оптимизированное. Оба изображения выведены на экран в реальном размере. При этом, оптимизированное меньше. Вы еще не забыли, что мы при оптимизации уменьшали размер изображения, чтобы оно целиком помещалось в окне браузера?

5.3.4. Цветовые модели изображения в формате jpg и Web

Продолжим рассмотрение вопросов, связанных с созданием в фотошоп изображений, которые затем должны быть размещены на ***Web***-страницах. Поместим вот это изображение бабочки на эту ***Web***-страницу. Перед вами на экране ***HTML*** код этой страницы. Чтобы вставить на эту страницу изображение бабочки, нужно в той строке где мигает курсор написать следующую команду языка ***HTML***, вот такую команду сейчас она выведена на экран в инверсных цветах. Вот так выглядела страница без изображения

бабочки, а так она будет выглядеть с бабочкой. Нажимаем для обновления экрана клавишу ***F***5 на клавиатуре, и где бабочка? Почему браузер ***Internet Explorer*** не смог ее показать?

Набор форматов графических файлов, которые могут отображаться на ***Web***-страницах, достаточно ограничен, но уж формат-то ***JPEG***, в котором сохранено изображение бабочки уж точно может отображаться на ***Web-страница***х, в чем же дело? А дело в том, что цветовая модель этого изображения ***CMYK***. Изображение формата ***JPEG*** в цветовой модели ***CMYK*** не отображаются на ***Web-страница***х, точно так же, как в цветовой модели ***LAB color***. Чтобы бабочка вылилась на ***Web***-страницу, достаточно поменять ее цветовую модель на ***RGB*** и затем сохранить это изображение в цветовой модели ***RGB***.



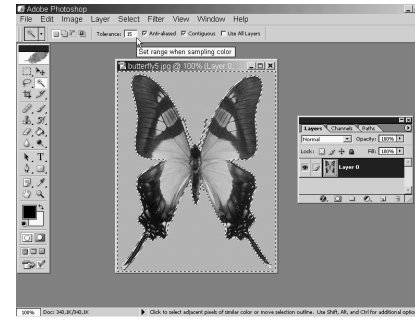
Вот бабочка теперь в цветовой модели ***RGB***, и в меню файл выбираем команду ***SAVE***. Обновляем ***Web***-страницу, нажимаем клавишу ***F***5 на клавиатуре, бабочка на месте. Если же воспользоваться для сохранения изображения в цветовой модели ***CMYK***, командой ***SAVE for WEB*** (сохранить для ***Web***), то помимо всех прочих действий, которые можно проделать с изображением в диалоговом окне этой команды при сохранении этого изображения оно будет автоматически конвертировано в цветовую модель ***RGB***. Сохраняем это изображение под именем исходного, появляется диалоговая панель с предупреждением, о том, что оптимизированное изображение затрёт исходное, можно отказаться от этого, нажав в диалоговой панели кнопку ***CANCEL***, а можно и затереть, нажав кнопку ***REPLACE***. Мы вернулись в основное окно программы фотошоп.

Перед нами та же бабочка, но почему-то все равно в цветовой модели ***CMYK***, хотя в одноимённом файле той же самой папки, откуда изначально была взята бабочка она в цветовой модели ***RGB***, а это изображение сейчас находится только в оперативной памяти компьютера. Программа фотошоп просто дает нам возможность сохранить исходное изображение, но уже желательно под другим именем. Попробуем тупо сохранить это изображение. В меню файл команда ***SAVE***. Обычно при вызове этой команды происходит без всяких предупреждений и дополнительных вопросов, запись обработанного изображения на место исходного под тем же именем.

Ну что ж, сохраняем изображение. Нажимаем на команду ***SAVE***. И вот тут-то умница программа выводит предупреждение, то есть работает в данном

случае не как команда ***SAVE***, а как команда ***SAVE AS*** (сохранить как), переспрашивая действительно ли мы желаем затереть сохраненную ранее дисковую копию изображения, которая, напоминаю в цветовой модели ***RGB***, копией изображения из оперативной памяти, которая в цветовой модели ***CMYK***. Нет, не хотим. Нажимаем кнопку ***CANCEL***, а изображение сохраняем под другим именем с помощью команды ***SAVE AS*** в той же самой папке.

5.3.5. Изображения в формате jpg и прозрачность



Посмотрим еще раз на ***Web***-страницу с изображением бабочки. Фон страницы одного цвета, а фон у бабочки другого. Можно ли в фотошопе сделать так чтоб фон бабочки был прозрачным, прозрачным именно на ***Web***-странице, потому что в фотошопе его, конечно, можно сделать прозрачным. Вернёмся в фотошоп и сделаем у бабочки прозрачный фон. Выводим на экран палитру слоев. В меню ***WINDOW*** команда ***Layers***. Изображение состоит всего из одного слоя ***BACKGROUND***. На слое ***BACKGROUND*** — основа — областей прозрачности не бывает, но это дело легко поправить, просто сконвертировав слой основы в обычный слой двойным щелчком по нему в палитре слоев. Здесь ничего не меняем, просто нажимаем кнопку ***OK***. Слой основы превратился в обычный слой. Теперь выбираем инструмент волшебная палочка. Убеждаемся, что включен параметр ***Contiguous*** — это нужно чтобы при выделении пикселей фона не захватить пиксели такого же цвета на самой бабочке, и щёлкаем инструментом волшебная палочка по какой либо точке фона. Значение параметра ***tolerance***, равное 15, обеспечило такое точное выделение фона. Обычно это значение приходится подгонять.

Теперь нажимаем на клавиатуре клавишу ***DELETE***. Фон стал прозрачным. Снимаем выделение. Нажимаем на клавиатуре комбинацию клавиш ***CTR-D*** и попробуем сохранить это изображение с прозрачным фоном для ***Web***. В меню ФАЙЛ команда ***SAVE for WEB***. Увы, это невозможно. Изображение в формате ***JPEG*** нельзя сохранить с областями прозрачности, которые понимались бы как таковые браузерами. Команда ***SAVE for WEB*** залила эту область прозрачности белым цветом. Правда, этот белый цвет можно изменить на другой, и, в частности, подогнать его к цвету фона ***Web***-страницы, сымитировав тем самым прозрачную область в изображении с бабочкой.

Правда, это будет всего лишь имитация, потому что при изменении фона ***Web***-страницы фон бабочки снова будет на нем выделяться. Всё же давайте это проделаем. Сделаем фон бабочки такого же цвета, как фон ***Web***-страницы.

Это снова перед нами ***HTML***-код этой страницы. Инверсными цветами выделен цвет фона. К сожалению, он задан именем лимоншифон, таких имен фотошоп не понимает. Поступим так: захватим с экрана окно ***Web***-страницы, нажав на клавиатуре комбинацию клавиш ***ALT-PRINT SCREEN***. При этом окно браузера помещается во временный буфер обмена. Сворачиваем окно браузера. Выходим из команды ***SAVE for WEB***, нажав кнопку ***CANCEL***. Создаем новое изображение в меню Файл командой ***NEW***. Ничего не меняя, нажимаем кнопку ОК и достаём из временного буфера обмена помещённое туда изображение.

Нажимаем на клавиатуре комбинацию клавиш ***CTRL-V***. Это не ***Web***-страница, это всего лишь картинка. Теперь нужно определить, какой это цвет, в терминах, которые понимает программа фотошоп, и передать его в диалоговое окно команды ***SAVE for WEB*** для заливки там им фона бабочки. Выбираем инструмент пипетка для взятия, так сказать, пробы этого цвета. К сожалению, инструмент пипетка может взять пробу цвета только с изображения, открытого в программе фотошоп. А вот, например, из окна ***INTERNET EXPLORER*** — не может. Видите, внутри фотошоп курсор имеет вид пипетки а здесь теряет этот вид. Поэтому нам и понадобились манипуляции с буфером обмена. Итак, берём инструментом пипетка пробу этого цвета — лимоншифон. Просто щёлкнем левой кнопкой мыши по пикселю этого цвета.

Цвет лимоншифон установился как основной цвет. Посмотрим, какие у него координаты. Щёлкаем левой кнопкой мыши по пиктограмме основного цвета на палитре инструментов. Появляется окно выбора цвета ***COLOR PICKER***, текущим в котором является цвет лимоншифон, а также в этом окне представлены координаты цвета лимоншифон в различных цветовых моделях: в модели ***HSB***, в модели ***RGB***, в модели ***LAP***, в модели ***CMYK***. Но проще всего передать в панель команд ***SAVE for WEB*** шестнадцатеричное представление цвета лимоншифон.

В каждой из перечисленных цветовых моделей цвет представляется в виде набора чисел, который невозможно за один прием передать через временный буфер обмена. А шестнадцатеричное представление цвета — можно. Подсвечиваем его двойным щелчком мыши, копируем во временный буфер обмена. Комбинация клавиш ***CTRL-C***. Закрываем окно ***COLOR PICKER***, активизируем изображение с бабочкой. Вызываем команду ***SAVE for WEB*** — в меню файл команда ***SAVE for WEB***. Вызываем команду ***COLOR PICKER*** в панели команд ***SAVE for WEB*** щелчком левой кнопки мыши по пиктограмме

цвета. Появляется окно команды ***COLOR PICKER***. Подсвечиваем шестнадцатеричное представление основного цвета двойным щелчком левой кнопки мыши по нему и заменяем это шестнадцатеричное представление основного цвета на шестнадцатеричное представление цвета лимоншифон, которое находится во временном буфере обмена. Нажимаем на клавиатуре комбинацию клавиш ***CTRL-V***.

Текущим цветом становится цвет лимоншифон. Нажимаем кнопку ОК в диалоговой панели команды ***COLOR PICKER***. Теперь заливаем область прозрачности исходного изображения, которое сейчас залито белым цветом, текущим цветом — цветом лимоншифон. Это позволяет сделать команда ***MATTE*** — матовость. Нажимаем кнопку выбора различных вариантов команды ***MATTE*** и выбираем вариант ***EYEDROPER COLOR*** — текущий цвет, теперь бывшая прозрачка залита цветом фона ***Web***-страницы. Сохраняем такой вариант бабочки. Нажимаем кнопку ***SAVE***. Сохраняем под именем ***BATTERFLY***7. Теперь вставляем новый файл с бабочкой на ***Web***-страницу, сохраняем измененный код ***Web***-страницы — в меню файл команда ***SAVE***.

Сворачиваем окно с ***HTML***-кодом страницы и обновляем окно с бабочкой. Нажимаем на клавиатуре клавишу ***F***5. Вот так теперь выглядит ***Web***-страница. Мы сымитировали прозрачность фона бабочки, но, подчеркиваю, это всего лишь имитация прозрачности. Изображение в формате ***JPEG*** не может содержать областей прозрачности, и, если мы поменяем в коде страницы цвет фона лимоншифон на какой нибудь другой фон, например, на оранжевый цвет, то на нем фон бабочки опять будет выделяться.

5.3.6. Формат GIF

Если изображение содержит непрерывные переходы цветов, то для размещения его на ***Web***-странице лучше сохранять это изображение в формате ***JPEG***. Для подобных изображений именно формат ***JPEG*** позволит получить приличное качество изображения при приемлемом размере файла с этим изображением, но если картинка содержит так называемые плашечные цвета, то есть большие области, залитые однородным цветом, содержат текстовые фрагменты и вообще являются высококонтрастными, то такое изображение лучше сохранять в формате ***GIF***.

GIF иногда произносят, как ***GIF*** — сокращение от ***Graphic Interchange Format*** — формат обмена графическими данными. Это стандартный формат для сжатия изображения с плашечными цветами, четкими деталями типа чертежей, схем, логотипов или изображений с текстовыми фрагментами. Сжатие графической информации в формате ***GIF*** происходит без потерь, в отличие от формата ***JPEG***, однако сохранить в этом формате можно не более 256 цветов изображения, потому то он и не очень подходит для сохранения изображений с непрерывными тональными переходами. Но вот такое изображение выгоднее сохранить в формате ***GIF***, а не в формате ***JPEG***. Сохраним его для размещения на ***Web***-станице в меню ФАЙЛ команда ***SAVE FOR WEB***. Вот примерно такой вид будет иметь эта картинка на ***Web***-странице при сохранении в формате ***GIF***. Вполне приличное качество, и размер файла тоже подходящий — 12,5 ***Kb***.

Посмотрим, какое качество мы получим, сохраняя эту картинку в формате ***JPEG*** с примерно таким же размером файла. Щелкаем по полю выбора графического формата или по кнопке выбора этого формата и выбираем формат ***JPEG*** — изображение выглядит на первый взгляд чуточку лучше, по крайней мере, глаза быка имеют градиентную раскраску, как и в исходном изображении. Вот это исходное изображение. Вот это ***GIF***. Градиентной раскраски глаз нет, а в ***JPEG***е есть.

Но размер ***JPEG*** файла при данном значении качества ***Quality***, равном 38, существенно больше, чем размер аналогичного файла в формате ***GIF***. В формате ***JPEG*** — 30,5 кбайт, а в формате ***GIF*** — всего 12,5 кбайт Можно, конечно, сохраняя картинку в формате ***JPEG***, уменьшить значение параметра ***Quality*** (качество), уменьшив тем самым и размер файла с изображением. Уменьшим значение параметра ***Quality*** до минимально возможного и посмотрим, каким при этом станет размер файла. Нажимаем на кнопку установки значения параметра ***Quality***, и на появившейся шкале установки параметра качества перемещаем ползунок до крайнего левого положения.

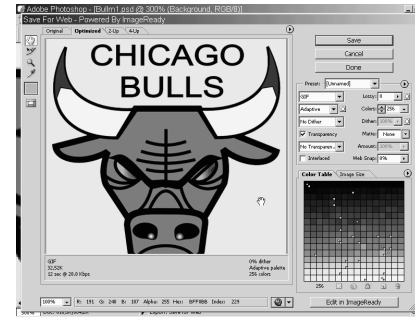
Значение параметра ***Quality*** сейчас минимальное, равное 0. Размер файла с изображением существенно уменьшился, уменьшился в 2 раза и равен сейчас 15,5 Кбайт, что всё-таки несколько больше, чем размер файла с этой картинкой в формате ***GIF***, но визуальное качество картинки существенно ухудшилось. Вокруг границ контрастных областей появились характерные для формата ***JPEG*** с большой степенью сжатия артефакты: гадкие разводы вокруг границ контрастных областей, которых нет в формате ***GIF***. Посмотрите, какое чистое здесь изображение. А можно ли в формате ***GIF*** сделать так, чтобы раскраска глаз была градиентной, как в исходном изображении? Можно добиться желаемых цветовых переходов на глазах двумя способами. Первый способ: увеличить количество цветов в изображении.

Сейчас этих цветов 16. Все используемые в изображении цвета представлены в так называемой цветовой таблице (***color table***) в правом нижнем углу диалоговой панели команды ***SAVE FOR WEB***. Попробуем увеличить количество цветов. Нажимаем на кнопку выбора количества цветов, и выбираем значение 32. Похоже, что этого значения недостаточно. Выбираем значение, равное 64.

Каждое такое увеличение количества цветов в изображении приводит к увеличению размера файла в формате ***GIF*** с этим изображением. А нужного результата — качественной раскатки на глазах — мы не достигли. Посмотрите на цветовую таблицу. В ней сейчас 64 цвета. При увеличении количества цветов в цветовой таблице и в изображении появилась целая куча ненужных нам цветов — оттенков чёрного цвета, а необходимые для качественной раскатки глаз цвета не появились. И только максимальное количество цветов — 256 позволяет закрасить глаза, как нам хочется, но ценой увеличения размера файла, содержащего картинку, до 32,5 кбайт.

Посмотрим на соответствующую цветовую таблицу. В ней сейчас 256 цветов, но отнюдь не все из них используются в изображении. Удалим некоторые цвета из цветовой таблицы, которые заведомо не используются в изображении, вручную. Тем самым мы уменьшим размер файла, и постараемся сохранить в изображении нужные цвета.

5.3.7. Цветовая таблица (Color Table)



Посмотрим на соответствующую цветовую таблицу. В ней сейчас 256 цветов, но отнюдь не все из них используются в изображении. Удалим некоторые цвета из цветовой таблицы, которые заведомо не используются в изображении, вручную. Тем самым мы уменьшим размер файла и постараемся сохранить в изображении нужные цвета. Выбираем инструмент Пипетка в диалоговой панели команды ***SAVE FOR WEB*** и щелкаем по тому цвету, который нужно сохранить. Вот этот цвет в цветовой таблице. Он обведен дополнительной рамочкой. Он сейчас выделен. Щелкнем левой кнопкой мыши по пиктограмме, изображающей замочек, внизу цветовой таблицы, предохраняя, таким образом, этот цвет от последующего удаления. В правом нижнем углу квадратика с эти цветом в цветовой таблице появился маленький белый кружок, признак блокировки этого цвета от последующего удаления.

Щелкаем по оранжевому цвету и блокируем его. По чёрному. Блокируем. Видите: у заблокированных цветов появляется маленький белый кружок, подчеркиваю, внизу квадрата с этим цветом в цветовой таблице. На кружок в центре квадрата с цветом сейчас не обращаем внимания.

Он к нашему процессу не имеет никакого отношения. Это признак так называемого безопасного цвета для веб. Аналогичным образом я заблокировала и некоторые другие плашечные цвета, встречающиеся в этом изображении. С цветами на глазах поступим так, для ускорения процесса выбора нужных цветов. Щёлкнем в этом месте глаза (был выбран зеленый цвет), затем нажмем на клавиатуре клавишу ***Shift***, и пощелкаем вдоль всего глаза. Посмотрите: выделяются другие цвета. Клавишу ***Shift*** нужно удерживать во время всего процесса выделения цветов. Затем отпускаем клавишу ***Shift*** и блокируем сразу всю группу выделенных цветов, нажимая на пиктограмму с замочком.

А что там у нас с раскладкой ноздрей? Нужно сохранить цвета и этой раскладки. Повторное нажатие на пиктограмму с замочком снимает блокировку выделенных цветов, поэтому сначала снимем выделение с только что заблокированных цветов. Нажимаем на кнопку вызова меню цветовой таблицы и выбираем там команду ***Deselect All Colors*** (Снять выделение со всех цветов). Снова выбираем инструмент Пипетка и опять пройдемся, но уже по носу, так же, как мы только что прошлись по глазу. Посмотрите снова на цветовую таблицу. Все цвета, которые имеют в правом нижнем углу белый кружочек, заведомо используются в изображении. Я надеюсь, что таким способом я выбрала все цвета, которые присутствуют в изображении, и поэтому остальные сейчас удалю.

Вот этот цвет у меня заблокирован. Его удалять не надо. Выбираю ближайший к нему в цветовой таблице. Следующий используемый в картинке цвет — вот этот. Всё, что между ними, можно удалить. Нажимаю на клавиатуре клавишу ***Shift*** и, удерживая ее, щелкаю по вот этому цвету. Отпускаю клавишу ***Shift***. Таким образом, я выделила в цветовой таблице целый непрерывный промежуток цветов, который можно удалить. Удаляю их, нажав на пиктограмму мусорной корзины в правом нижнем углу цветовой таблицы.

Картинка, похоже, не изменилась. Видимо, я и вправду не удалила ни одного нужного цвета. Вообще, должна заметить, что работа в диалоговом окне команды Сохранить для веб требует повышенного внимания, потому как отменить неправильно произведенные действия невозможно. По крайней мере, я этой возможности здесь не нашла. Повторяем процесс удаления ненужных цветов. Щелчок.

Нажимаем клавишу ***Shift***. Щелчок. Затем щелчок по пиктограмме мусорной корзины. Обратите внимание: количество цветов уменьшается, их уже 222, а вместе с этим количеством цветов уменьшается и размер файла. Изображение же при этом визуально не меняется. И вот что мы имеем в итоге. Цветов всего 24. Размер файла чуть больше… чуть больше 15 килобайт, глаза раскрашены так, как я хотела, и нос не пострадал.

Посмотрим, как этот бык будет выглядеть на веб-странице. Вполне прилично. И посмотрите, какая большая картинка, и какой небольшой размер файла: всего 15 килобайт. Подобный способ визуального представления цветов, отсутствующих в цветовой таблице, требует довольно кропотливой работы, и хуже всего то, что, ошибившись, приходится начинать всё сначала. Существует и другой способ представления цветов, отсутствующих в цветовой таблице. Это так называемый Дизаринг.

5.3.8. Дизеринг. Поэтапный вывод изображения на экран

Существует и другой способ представления цветов, отсутствующих в цветовой таблице. Это так называемый дизаринг. Дизарингом называется метод имитации цветов, которых нет в таблице, с помощью тех цветов, которые в таблице есть. При дизаринге отсутствующий цвет представляется россыпью пикселей, каждый из которых имеет цвет, содержащийся в таблице. Изображения, содержащие изначально только плашечные цвета, могут выглядеть хорошо и без дизаринга.

Изображения, содержащие фрагменты с непрерывно меняющимся цветом, как в данном случае глаза или ноздри, требуют дизаринга хотя бы для предотвращения цветных полос при сохранении этих изображений в формате ***GIF***. Дизаринг может производиться разными способами, по разным алгоритмам. Включим какой-нибудь алгоритм дизаринга.

Диффузия. Посмотрите, как при включенном дизаринге представляются промежуточные цвета между зеленым и черным. Включение алгоритма дизаринга приводит к небольшому увеличению размера файла. А вот так будут выглядеть глаза, если установить в изображении таблицу из 32 цветов. Это по алгоритму Диффузия. Выберем алгоритм ***Pattern*** (шаблон). По-моему, так смотрится лучше, но размер файла чуть-чуть увеличился. А вот это алгоритм ***Noise*** (случайный шум).

Этот алгоритм еще чуть-чуть увеличивает размер файла. Лучше ли стало изображение, сказать не могу. Вообще, тут трудно дать какие-либо рекомендации. К каждому изображению нужно подходить индивидуально и экспериментировать в диалоговой панели команды Сохранить для веб. Я не буду останавливаться на выборе алгоритма сжатия при сохранении в формате

GIF. Про эти алгоритмы можно прочитать в любом подробном справочном руководстве по программе Фотошоп.

Это заняло бы слишком много времени. Тем более что на практике большинство людей, сохраняющих изображения в формате ***GIF***, просто экспериментируют с этими алгоритмами, пытаясь добиться наиболее качественного изображения. Включение параметра ***Interlace*** приводит, опять же, к некоторому увеличению размера файла, но зато позволяет при выводе веб-страницы, содержащей данное изображение, на экран выводить его не сразу в окончательном виде, а поэтапно. Сначала — грубый эскиз изображения, потом — всё более и более точные приближения, и, наконец, окончательный вариант.

Считается, что если изображение грузится долго, то посетитель страницы может не дождаться конца загрузки этого изображения, заскучать и покинуть страницу. Тем более что время, в данном случае загрузки, — деньги. А вот, увидев быстро выведенный грубый эскиз изображения, посетитель вашей страницы страшно заинтересуется, и дождется окончательной загрузки.

Утверждение, я считаю, достаточно спорное. Ну, разве что на вашей странице представлена схема какого-то великого изобретения. Но большинство изображений, особенно в формате ***GIF***, являются только элементами декора вашей страницы. И, может быть, в этом случае и не стоит предъявлять посетителю страницы грубый некрасивый предварительный эскиз этого декора.

А вот для формата ***JPEG*** существует ли возможность поэтапного вывода изображения на экран при просмотре его в веб-браузере? Да. И этот поэтапный вывод на экран обеспечивается включением параметра ***Progressive***. Причем в данном случае — в случае формата ***JPEG*** — вопрос о целесообразности сжатия по прогрессивному алгоритму ***JPEG*** должен решаться, еще и учитывая, что не все браузеры, а также другие программы, графические редакторы или текстовые, поддерживают этот прогрессивный формат.

5.3.9. Изображения в формате GIF и прозрачность

Посмотрите, пожалуйста, на эту веб-страницу. Я хочу вставить на нее портрет Пушкина, а также написать заголовок этой страницы — фамилию Пушкина с инициалами — шрифтом, имитирующим почерк Пушкина. Подобный шрифт установлен на моем компьютере. Вставим на страницу этот автопортрет Пушкина. Я отсканировала его с первой страницы «Литературной газеты» и затем слегка почистила в Фотошопе. Вот так будет выглядеть эта же веб-страница после вставки на нее этого изображения с автопортретом

Пушкина. Эта же страница будет выглядеть гораздо привлекательнее, если убрать белый фон у автопортрета, то есть сделать его прозрачным, чтобы сквозь него просвечивал фон самой веб-страницы. Сделать это можно, если сохранить данное изображение, которое сейчас в формате ***JPEG***, сохранить его в формате ***GIF***. Формат ***GIF***, в отличие от формата ***JPEG***, позволяет сохранять изображения с областями прозрачности. Сохраним это изображение для веб в формате ***GIF*** с областями прозрачности. Меню Файл, команда ***SAVE FOR WEB***.

Выбираем формат сохранения ***GIF***. Включаем параметр ***Transparency*** (прозрачность). Размер этой картинки в формате ***GIF*** больше 47 килобайт. Это практически предельно допустимый размер для файла с картинкой, которая должна быть помещена на веб-страницу. Но этот объем файла можно сильно уменьшить. Действительно, изображение у нас практически двуцветное, черно-белое, и поддержка 256 цветов для этой картинки совершенно ни к чему. Уменьшим количество цветов до двух. Посмотрите, размер файла уменьшился почти в 10 раз. Но правильное ли это решение: ограничиться для этого изображения всего двумя цветами? Выберем инструмент Лупа и увеличим изображение.

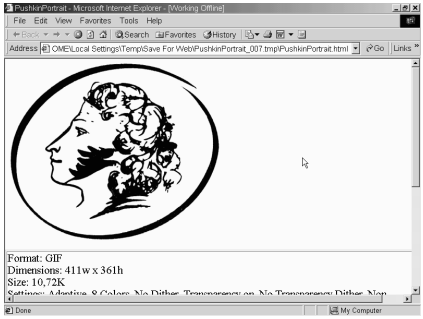
Посмотрите, какие у него рваные ломаные края в двуцветном режиме, и какие они были в режиме, содержащем 256 цветов. Посмотрите, какое гладкое, чистенькое изображение. А такое оно потому, что в 256-цветном режиме неровности краев, контуров изображения, сглаживаются путем добавления к этим краям пикселей промежуточных цветов между черным и белым. Поэтому для качественного представления этого изображения на веб-странице двух цветов мало. Выберем 8 цветов. Картинка выглядит вполне прилично, и размер файла небольшой — меньше 11 килобайт. Выведем это изображение в окне предварительного просмотра команды Сохранение для веб в том размере, в каком оно будет представлено на веб-странице. Для этого делаем двойной щелчок по инструменту Лупа.

Теперь сделаем область белого цвета на этом изображении областью прозрачности. Выбираем инструмент Пипетка и щелкаем им по какой-либо точке белого цвета на изображении. После этой операции белый цвет в таблице ***Color Table*** выделен. Делаем его прозрачным. Нажимаем на пиктограмму с шашечками, которыми по умолчанию изображаются в Фотошоп области прозрачности, на пиктограмму Транспейренси

(прозрачность) внизу таблицы ***Color Table***. Теперь выделенный ранее цвет, в данном случае, белый, в таблице ***Color Table***, отображается следующим образом. Квадратик с этим цветом в таблице теперь разделен на две части диагональной линией. В верхней части тот цвет, который был до редактирования цветовой таблицы, в данном случае, белый цвет. В нижней же части этого квадратика цвет, полученный редактированием вместо белого, в данном случае, это прозрачный цвет. Посмотрим, как будет выглядеть это изображение на веб-странице.

Мы на самом деле ничего не увидели. Не увидели, была ли создана область прозрачности, потому что цвет фона веб-страницы белый, и цвет фона у автопортрета Пушкина тоже был белый. По умолчанию команда ***SAVE FOR WEB*** устанавливает для такого предварительного просмотра редактируемого в ней изображения белый цвет фона для веб-страницы. Закрываем окно предварительного просмотра в браузере изображения и возвращаемся в окно команды ***SAVE FOR WEB***.

Если бы цвет фона веб-страницы был бы не белым, а каким-нибудь цветным, то на этом цветном фоне мы бы разглядели, создались ли области прозрачности, и как в этом случае выглядит изображение на веб-странице. Можно ли в команде Сохранение для веб поменять цвет фона веб-страницы предварительного просмотра? Да, можно. Закрываем окно предварительного просмотра в браузере изображения и возвращаемся в окно команды ***SAVE FOR WEB***. Нажимаем на кнопку вызова скрытого в данный момент меню параметров веб-страницы для просмотра изображения и выбираем в нем команду ***Edit Output Settings*** (редактировать установки вывода).



Появляется диалоговая панель команды ***Output Settings*** (установки вывода). Все ее параметры я сейчас описывать не буду, их слишком много. Меня сейчас интересует только цвет фона веб-страницы. Диалоговая панель команды ***Output Settings*** на самом деле состоит из нескольких страниц. Сейчас перед нами на экране первая из этих страниц, и на ней установки цвета фона нет. Поэтому нажимаю кнопку ***Next*** (следующая страница). На ней тоже нет цвета фона. Поэтому снова нажимаю кнопку ***Next***. А вот и он, параметр ***Color***. Нажимаю на кнопку выбора цвета фона веб-страницы и выбираю значение ***Other*** (установка произвольного цвета). Появляется диалоговая панель выбора цвета ***Color Piker***. Выберу для цвета фона какой-нибудь цвет, но не близкий к черному и не близкий к

белому. Мне нужно, чтобы картинка на веб-странице резко контрастировала с цветом ее фона. Выбираю, например, зеленый цвет.

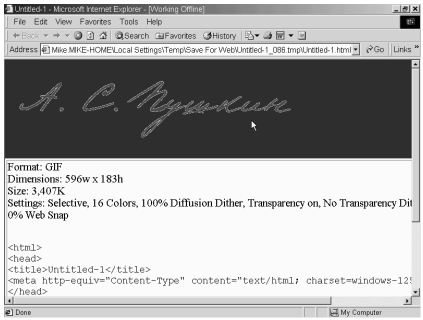
Вот такой зеленый. Выхожу из панели команды ***Output Settings***, нажимая кнопку ОК, и снова смотрю, как будет выглядеть это изображение в окне браузера. Да… Нельзя сказать, что это красота неописуемая. Область прозрачности вместо белого цвета действительно создана, но пиксели промежуточных серых цветов резко выделяются на зеленом фоне страницы и создают ощущение неряшливости изображения.

Попробуем сделать прозрачным и этот светло-серый цвет. Выделяем его в таблице ***Color Table***, щелкаем на пиктограмму прозрачности, и снова просматриваем изображение на веб-странице. Изображение стало лучше, но недостаточно. Повторим операцию, сделаем прозрачным и этот цвет. Почти идеально. Попробуем сделать прозрачным и этот цвет. А вот сейчас уже то, что надо. Размер файла уменьшился, но цветов-то осталось всего четыре. Не проще ли было выбрать их с самого начала? Нет. Потому что, во-первых, у нас сейчас не четыре, а пять цветов. Пятый — это прозрачный цвет. Изображение, конечно, пострадало: оно стало более рваным, чем исходное. Попробуем чуть-чуть сгладить контуры. Выберем в цветовой таблице самый темный из ныне прозрачных цветов и восстановим его непрозрачность.

Контуры сгладились. Но мы помним, что на веб-странице этот цвет выделялся на цвете фона. Выделялся как более светлый, чем цвет фона. Значит, сделаем его чуть более темным. Двойной щелчок по этому цвету в цветовой таблице. Появляется диалоговая панель команды ***Color Picker*** (выбор цвета). Выбранный цвет представлен в диалоговой панели ***Color Picker*** вот здесь и вот здесь. Сделаем его чуть темнее. Щелкаем чуть ниже в этом окне.

Вот таким этот цвет был, вот таким стал. Нажимаем кнопку ОК. Квадратик этого цвета в цветовой таблице снова разделен диагональю пополам. Сверху — цвет, который был, снизу — который стал. Правда, разглядеть здесь что-либо почти невозможно. Смотрим в браузере. Неплохо. А на моей странице будет еще лучше, потому что там фон светлее. Сохраняем. И вот я его вставила на свою веб-страницу. Красота? Правда, лучше с прозрачным фоном, чем без оного? И размер этого файла в формате ***GIF*** примерно 9 килобайт, что гораздо меньше, чем размер файла с изображением в формате ***JPEG***. Этот размер 58 килобайт.

5.3.10. Создание надписей для Web-страниц



На моём компьютере установлен шрифт, имитирующий почерк Пушкина. Я хочу написать этим шрифтом заголовок, инициалы и фамилию Пушкина на данной ***Web***-странице. Разумеется этого нельзя делать, то есть я не получу нужного результата, если пропишу этот шрифт непосредственно в ***HTML*** этой страницы, так как на компьютерах подавляющего большинства просматривающих мою страницу этот шрифт не установлен, так как не является стандартным шрифтом, и эта надпись будет выведена в соответствии с установкой браузера пользователя. И единственное решение в подобной ситуации заменить эту текстовую надпись картинкой. Создадим такую картинку в программе ***Adobe Photoshop***. Выберем инструмент создания текстовых фрагментов на палитре инструментов в программе. Установим курсор, просто щелкнув левой кнопкой мыши, и создадим соответствующую надпись. В качестве цвета надписи я выбрала фиолетовый цвет чернил. В качестве гарнитура шрифта выбрала шрифт Пушкин.

Подобрала желаемый размер и выбрала максимальную гладкость, отображение букв на экране — ***smooth***. Переключаюсь в режим ввода русских букв, с клавиатуры. В моём случае это комбинация ***Ctrl***+***Shift***. И печатаю надпись. Текстовый фрагмент вводится на отдельный слой. В данном случае это слой ***Layer***1. Завершаем ввод текста, нажатием на пиктограмму ***OK***, имеющую вид галочки, панели параметров инструмента ***text***. Созданный текстовый слой с надписью имеет прозрачный фон. Я отключила видимость слоя ***Background***, подложки текста и фон надписи теперь отображается характерными шашечками, то есть является прозрачной. Удалим из палитры слоев слой ***Background***, просто перетянув его на пиктограмму мусорной корзины внизу этой палитры.

Он мне не нужен. Надпись я буду создавать в прозрачном фоне. Это ещё один способ создания изображения в формате ***GIF***, с областями прозрачности. А именно не делать некоторые цвета прозрачными непосредственно диалоговой поныли команды «Сохранение для ***WEB***», а создать эти области прозрачности непосредственно в главном окне программ ***AbobePhotoshop***, ещё до вызова программы «Сохранение для ***WEB***». В данном случае это 2 способ

самый предпочтительный, так как текстовая надпись создаётся на прозрачном фоне. И единственное, что нужно сделать сейчас до вызова команды «***Save for WEB***», это обрезать данное изображение, оставив только его существенную часть, саму надпись. Для уменьшения размера файла с этой надписи разверну окно с изображением на все рабочее поле программ ***Photoshop***.

Выберу на палитре инструментов, инструмент ***Crop*** — обрезка, и обрежу данное изображение, просто протягиванием этого инструмента по диагонали через надпись. Более светлая часть изображения остаётся, затемненная обрезается. Завершаю процесс обрезки нажатием клавиши ***Enter*** на клавиатуре. А вот теперь вызываю программу «Сохранение для ***WEB***». Меню ***File*** (Файл) команда «***Save for WEB***». Формат, разумеется, ***GIF***. Параметр ***Transparency*** (прозрачность), разумеется, включен. Посмотрите на световую таблицу — ***Color Table***. В ней среди цветов, которые используются в изображении, изначально присутствует прозрачный цвет.

Вот он. Потому что мы вызывали программу СОХРАНИТЬ ДЛЯ ***WEB***, уже имея в руках области прозрачности в изображении. Посмотрим, как это будет, выглядит на ***Web***-странице. Я специально сейчас установила на ***Web***-странице темный фон, чтобы была более заметной обводка букв белым цветом, которая мне не нужна и о которой я программу не просила. Откуда она взялась? А взялась эта белая обводка из значения параметра ***matte*** (матовость).

Сейчас это значение белый цвет. Цвет выбранный как значении параметра ***matte***, окрашивается в полу прозрачные пиксели изображения, при выводе этого изображения на ***Web***-страницу. А появляются эти полупрозрачные пиксели в изображении, при сглаживании контуров букв. Если в качестве значений параметра ***matte*** выбрать не белый цвет, а значение ***none*** (отсутствие цвета), то при выводе изображения на ***Web***-страницу все пиксели этого изображения прозрачность которых больше 50% будут полностью прозрачны. А все пиксели с прозрачностью равной или менее 50% будут полностью не прозрачны. Выберем значение ***none*** и посмотрим, что получится. Ну, гораздо лучше. Посмотрите, какой размер файла, примерно 1,5 килобайта. Ради имитации почерка Пушкина можно и вставить такую маленькую картинку. Сохраним её под таким именем в формате ***GIF***. И вот так это выглядит на ***Web***-странице.

5.3.11. Надпись с тенью и объемная надпись

Такое ощущение, что чего-то здесь не хватает. Автограф, несомненно, лучше выглядит в черном цвете, но все равно он тонковат и наверно лучше будет смотреться, если толщина линии в нём будет побольше. Конечно, средствами ***Photoshop*** можно сделать линии букв более толстыми, но это уже

не будет почерк Пушкина. Поступим иначе. Рисунок букв менять не будем, а добавим к этой надписи, какие-нибудь изобразительные эффекты типа тень или трёхмерность.

Вернемся в основное окно программы ***Adobe Photoshop*** и добавим эффекты к исходной надписи. Сначала поменяем цвет надписи. Выбираем снова инструмент текст. Цвет надписи отображается на палитре параметров, инструмента текст делаем по этой пиктограмме щелчок левой кнопкой мыши. Появившейся панель команды ***Color Piker***, устанавливаем для надписи новый цвет. Пусть это будет что-то близкое к цвету фона ***Web***-странице. Сначала цветовой тон, потом сам цвет. На ***Web***-странице всё будет видно, потому что мы добавим сейчас эффект к текстовому слою. Для добавления эффекта к текстовому слою делаем двойной щелчок левой кнопкой мыши по изображению этого слоя в палитре слоёв. Включим обводку надписи, ***Stroke***. Меняем цвет обводки на черный.

Толщину обводки делаем равную двум пикселям, затем включаем падающую тень, ***Drop shadow***. Увеличиваем густоту этой тени параметр ***Spread***. Все пиксели тени будут отображаться черным цветом в той или иной степени её прозрачности. Нажимаем кнопку ОК, в диалоговой панели эффектов слоев. Снова вызываю команду «Сохранить для ***WEB***». Смотрим, как это будет выглядеть на ***Web***-странице. Надо бы поменять цвет фона, на таком темном фоне теневые эффекты не заметны. Меню режимов вывода команда ***edit Output setting*** нажимаем кнопку ***next*** и ещё раз кнопку ***next***. Меняем цвет фона ***Web***-страницы. Параметр ***Other***, установка произвольного цвета, выбираем цветовой тон, выбираем цвет. Чтобы был примерно таким же, как цвет букв надписи. Ок, и ещё раз Ок. И просматриваем на ***Web***-станице.

По-моему, неплохо. Но тень выглядит довольно неряшливо. Тень, правда, выглядит немного неряшливо. Можно ли её сделать более аккуратной? Попробуем поменять алгоритм ***Dither***. То есть аппроксимации полупрозрачных цветов. Поменяем представления прозрачных цветов со случайного шума, ***Noise Transparency***, на диффузный ***Diffusion Transparency Dither***. Заметили, что тень стала более аккуратной? Вот эта диффузия, вот это шум, вот это снова диффузия. А вот как на ***Web***-странице. Чуть- чуть бы ещё убрать вот эти точки, которые слегка вываливаются из тени.

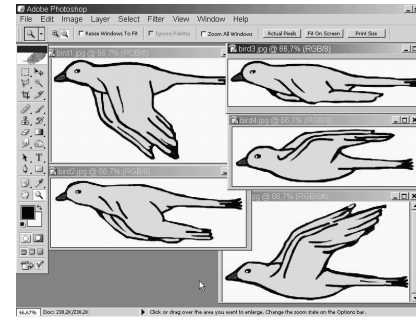
Но наверняка это можно сделать с помощью параметра ***matte***, матовость, которая как-то классифицирует полупрозрачные точки по степени их прозрачности. Сейчас значения параметров ***matte*** мало. Параметр ***matte*** никак не влияет на цвет полупрозрачных точек на ***Web***-странице. Дадим параметру ***Matte*** значение ***white*** — белый. Теперь цвета полупрозрачных пикселей смешиваются. Часть из них будет выводиться изначальным цветом теней, а остальные цветом, установленным компьютером значений параметров ***matte***.

Смотрим, как это будет на ***Web***-странице. Сохраняем. Под именем Пушкин сайт три. И вот как теперь выглядит моя ***Web-страница***.

А вот так она будет выглядеть, если сделать цвет букв совпадающим с цветом фона ***Web***-страницы. По-моему вот так лучше всего. Теперь давайте сделаем вот что. Отключим вообще ***Dithering*** полупрозрачных цветов, параметр ***No Transparency dither***. Смотрим на страницу. Вот так есть некое ощущение объемности букв, правда мешает белая обводка. Откуда она? Ну конечно же от параметра ***matte***. Его значения сейчас белый цвет. Поменяем его на какой-нибудь средне серый.

Например, вот такой. И снова посмотрим, как это будет выглядеть на ***Web***-странице. Вполне пристойная объемная надпись. А вот так эта надпись выглядит на моей ***Web***-странице. Несколько грубовато конечно, но это можно легко исправить. Делаем в исходном изображении цвет падающей тени ***Drop shadow***, менее черным. Делаем двойной щелчок левой кнопкой мыши по этому эффекту, ***Drop shadow***, в палитре слоев. Меняем черный цвет тени на серый. Команда «Сохранить для ***WEB***». И в диалоговой панели этой команды делаем более светлым значение параметра ***matte***, вот теперь объемная надпись выглядит не так грубо.

5.3.12. Анимации в формате GIF



Замечательной особенностью графического формата ***GIF*** является его способность сохранять анимированные изображения. Так называемые ***GIF***-анимации поддерживаются практически всеми браузерами ***Internet***. Если вставить такую ***GIF***-анимацию на веб-страницу, то при ее просмотре можно будет увидеть движущееся изображение, нечто подобное тому, что мы видим сейчас. На эту страницу вставлены два анимированных ***GIF***'а — две птички. А вот на этой — один анимированный ***GIF***. Создадим такую ***GIF*** -анимацию в программе ***Adobe Photoshop***. Первым шагом в создании ***GIF***-анимации является создание всех отдельных кадров этой анимации.

В данном случае для птички, машущей крыльями в полете, были нарисованы пять отдельных фаз этого полета. Каждая такая фаза была сохранена в отдельном файле с расширением ***jpg***. Вообще-то для создания ***GIF***-анимации совершенно необязательно сохранять ее кадры в виде отдельных файлов. Я их сохраняла потому, что процесс рисования отдельных

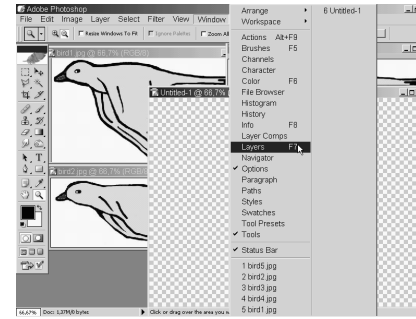
кадров занял у меня достаточно много времени, и никто мне не гарантировал, что в процессе этого рисования не повиснет компьютер или не отключат электричество. Итак, перед нами на экране последовательность кадров для ***GIF***-анимации. Это первый кадр. Это второй. Это третий. Это четвертый. И это, соответственно, пятый. Заготовкой для ***GIF***-анимации будет служить файл в родном для Фотошопа формате ***PSD***, состоящий из нескольких слоев. Каждый слой будет содержать один кадр, и порядок слоев в стопке слоев этого изображения будет совпадать с порядком фаз движения птички. Создадим эту заготовку для анимированного ***GIF***'а. Создадим его как новое изображение в Фотошоп. Меню ***File***, команда ***New***. Появляется диалоговое окно создания нового файла. Введем в нем необходимые значения параметров. Значение ширины изображения ***Width*** делаю равным 800 пикселям, высота изображения ***Height*** будет равна 600 пикселям.

Я не вычисляю сейчас необходимые размеры, а ввожу такие, которые заведомо позволят мне разместить каждый кадр изображения. Результирующее изображение потом обрежу для минимизации размера файла с изображением. Значением параметра ***Resolution*** (Разрешение) можно пренебречь. Он не играет никакой роли при создании изображений для ***Web***. Его значение важно только при выводе изображения на печать. А при выводе изображения в окне браузера каждый пиксель этого изображения отображается на экране одним экранным пикселем независимо от того, какое значение дать параметру ***Resolution***, а также от того, какой у пользователя монитор и в каком режиме он работает. Дальше. Важно, чтобы цветовая модель была ***RGB***.

Так как я хочу, чтобы птичка была цветная, а не черно-белая, поэтому мне не годятся режимы ***Bitmap*** и ***Grayscale***, а цветовые модели ***CMYK*** и ***Lab*** браузерами не поддерживаются, поэтому только ***RGB***. Цвет фона ***Background Contents*** выбираю ***Transparent*** (прозрачный), чтобы фоном для летящей птички был фон веб-страницы. ОК. Вот на этих шашечках, на этом прозрачном фоне и будем размещать все фазы полета, только выведем сначала палитру слоев. Меню ***Window***, команда ***Layers*** (Слои). Начинаем компоновать из кадров стопку слоев.

Активизируем изображение с первым кадром. Удаляем у него белый фон. Единственный слой первого кадра — это слой ***Background*** (Основа). На слое ***Background*** областей прозрачности не бывает, поэтому конвертируем слой ***Background*** в обыкновенный слой. Делаем по нему в палитре слоев двойной щелчок левой кнопкой мыши. В появившемся диалоговом окне оставляем значения параметров, которые выводятся по умолчанию, и нажимаем кнопку ОК. Теперь можно удалить белый фон из первого кадра, превратить его в область прозрачности. Выбираем на палитре инструментов инструмент

выделения Волшебная палочка. Параметр ***Contiguous*** (Связность) на панели параметров инструмента Волшебная палочка должен быть включен. Это обеспечит нам выделение только фона, без захвата белых пикселей глаза. Выделяем фон. Удаляем фон. Нажимаем клавишу ***Delete*** на клавиатуре.



Снимаем выделение. Нажимаем на клавиатуре комбинацию клавиш ***Ctrl-D***. Выбираем на палитре инструментов инструмент Перемещение и перетаскиваем первый кадр из его окна во вновь созданное окно. При перетаскивании фрагмента из одного окна изображения в другое происходит копирование этого фрагмента. В результате такого перетаскивания в новом окне появляется новый слой ***Layer*** 2. На нем-то и находится первый кадр с птичкой. Сворачиваем окно с первым кадром, активизируем второй кадр, конвертируем его слой ***Background*** в обычный слой, выбираем инструмент

Волшебная палочка, щелкаем по белому фону второго кадра, удаляем белый фон, нажав на клавиатуре клавишу ***Delete***, снимаем выделение (***Ctrl-D*** на клавиатуре), выбираем инструмент Перемещение и перетаскиваем вторую птичку во вновь созданное окно, где уже имеется кадр с первой птичкой. Вторая птичка тоже лежит на своем собственном слое ***Layer*** 3. Чтобы в полете птичка не дергалась и не проваливалась, выравниваем в этом окне вторую птичку по первой. Ориентиром при таком выравнивании будет служить голова и клюв птички. Инструмент Перемещение уже выбран, подгоняем с его помощью вторую птичку к первой. Сворачиваем окно с вторым кадром и повторяем процедуру для кадра номер 3. Активизируем.

Преобразуем слой ***Background*** в обычный слой. Волшебная палочка. Щелчок по белому фону. Клавиша ***Delete*** на клавиатуре. Комбинация клавиш ***Ctrl-D*** на клавиатуре. Инструмент Перемещение. И перетаскиваем птичку. Выравниваем третью птичку с первыми двумя. Теперь в создаваемом изображении четыре слоя: три слоя с кадрами и первый исходный слой. Он пустой, поэтому его можно безболезненно удалить. Перетаскиваем этот слой на мусорную корзину в палитре слоев. Теперь птичка номер 4. Активизируем. Конвертируем слой ***Background*** в обычный слой. Выбираем инструмент Волшебная палочка. Щелкаем где-нибудь по белому фону.

Удаляем этот фон клавишей ***Delete*** на клавиатуре. Снимаем выделение (***Ctrl-D***). Выбираем инструмент Перемещение. Перетаскиваем птичку. Выравниваем четвертую птичку по первым трем. Сворачиваем изображение с

четвертой птичкой, сначала активизировав его. Активизируем изображение с пятой птичкой и повторяем ту же процедуру. Конвертация фонового слоя. Инструмент Волшебная палочка. Щелчок по фону. Удаление его клавишей ***Delete***. Снятие выделения ***Ctrl-D***. Инструмент Перемещение. Перемещаем птицу. Выравниваем птицу. Сворачиваем птицу номер 5 и смотрим на палитру слоев. Было пять кадров, стало пять слоев. Теперь нужно убедиться, что кадры в стопке слоев расставлены в правильном порядке.

Даже если они расставлены в неправильном порядке, в этом нет ничего страшного, потому что отдельные слои можно перемещать в стопке слоев. Каждый слой в палитре слоев содержит миниатюрку, уменьшенное изображение того, что находится на этом слое, чтобы можно было судить, что на каком слое. К сожалению, сейчас эти миниатюры настолько маленькие, что, глядя на них, трудно понять, правильно ли мы расставили кадры. Но эти миниатюры можно увеличить.

Выводим на экран скрытое меню палитры ***Layers*** и выбираем команду ***Palette Options*** (Параметры палитры). Появляется диалоговая панель параметров палитры слоев. Выбираем в ней максимально возможное значение параметра ***Thumbnail Size*** (Размер миниатюры) и закрываем это диалоговое окно. Посмотрите теперь на палитру слоев. Сейчас размер миниатюр вполне достаточен, чтобы понять, что кадры расположены в правильной последовательности.

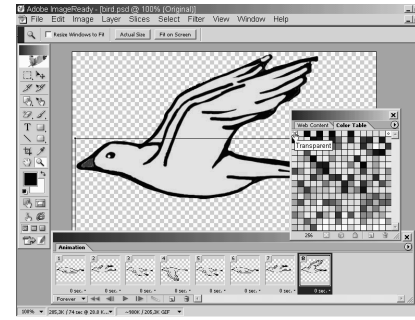
Но только этих кадров мало. Нужно добавить еще несколько для реалистичной имитации полета. Вот птица опускает крылья из самого верхнего положения до самого нижнего, а затем она должна их в обратной последовательности поднимать. Поэтому новые кадры не придется рисовать, достаточно будет отдублировать уже существующие и расположить их в стопке слоев в нужном порядке. После этого кадра должен идти этот. Дублируем этот слой в палитре слоев. Выводим меню палитры слоев и выбираем команду Дублирование слоя.

Здесь, ничего не меняя, нажимаем кнопку ОК. Вот в палитре слоев появился дубликат вот этого слоя. Перетаскиваем его в самый низ стопки слоев. Теперь активизируем слой ***Layer*** 4, дублируем его и тоже перетаскиваем его в самый низ стопки слоев. Слой ***Layer*** 5. Дублируем. Перетаскиваем. Слой ***Layer*** 6, самое верхнее положение крыльев, дублировать не будем, так же, как не нужно дублировать слой ***Layer*** 2, самое нижнее положение крыльев.Если их продублировать, то в этих крайних положениях птица будет при движении как бы зависать. Свернем палитру слоев, развернем созданное изображение до размеров рабочего стола и обрежем изображение, оставив только ту его часть, в которой есть графическая информация. Выбираем инструмент ***Crop***

(Обрезка), протягиваем его по диагонали от левого верхнего угла изображения до правого нижнего и сдвигаем границы изображения вплотную к птице.

Затемненная часть изображения будет обрезана. Для завершения процесса нажимаем клавишу ***Enter*** на клавиатуре. А вот таков реальный размер изображения. Сохраняем его вместе со всеми его слоями. Это можно сделать только в родном формате Фотошопа — формате ***PSD***. Никакой другой формат раскладку по слоям не поддерживает. Вообще говоря, для создания ***GIF***-анимации вовсе не обязательно на промежуточном этапе этого создания сохранять подобное расслоенное в формате ***PSD***. Я это делаю только в целях безопасности. Меню ***File***, команда ***Save*** (Сохранить). Сохраняю под именем ***Bird*** в родном фотошоповском формате ***PSD*** со всеми слоями.

Теперь создаем собственно анимированный ***GIF***. В самой программе Фотошоп этого сделать нельзя. Команда ***Save For Web*** не позволяет сохранять анимированные ***GIF***ы. Воспользуемся для этой цели программой ***Image Ready***, которая поставляется вместе с программой Фотошоп и переключиться в которую можно прямо в сеансе работы с Фотошоп. И при таком переключении программа Фотошоп не будет закрыта. Кнопка переключения в программу ***Adobe Image Ready*** находится в самом низу палитры инструментов. Переключаемся в программу ***Image Ready***. Мы в программе ***Adobe Image Ready***. А вот и кое-что отличное от программы ***Adobe Photoshop*** — палитра Анимация. Сейчас в ней один кадр, и этот кадр — совмещенное изображение со всех слоев. Это не то, что нам надо. Выводим на экран меню палитры Анимация и выбираем команду ***Make Frames From Layers*** (Сделать кадры из слоев).



Теперь на палитре Анимация представлены все слои исходного изображения в виде отдельных кадров. Выбираем число повторов этой последовательности кадров при открытий веб-страницы, на которую будет вставлено это изображение с анимацией. Выбираем ***Forever*** (Бесконечный повтор). Посмотрим, как это будет выглядеть. Нажимаем кнопку ***Play*** на палитре Анимация. Ну, вообще-то на веб-странице будет не совсем так. Картинка сейчас не в реальном размере, а уменьшенная в 2 раза. Нажимаем кнопку ***Stop*** на палитре Анимация. Браузеры всегда показывают картинку в реальном размере. Разворачиваем окно изображения и увеличиваем его до реального размера двойным щелчком по инструменту Лупа на палитре параметров.

Снова запускаем предварительный просмотр анимации. Останавливаем предварительный просмотр анимации. Теперь сохраняем изображение как анимированный ***GIF***. Меню ***File***, команда ***Save Optimized***. Сохраняем под именем ***Bird*** 2. Интересно, какой получился размер у файла с ***GIF***-анимацией? Ничего себе: 206 килобайт. Это безумие. Подобная простенькая анимация не стоит такого гигантского размера файла. Можно ли как-то уменьшить этот размер? Наверняка тут дело в слишком большой цветовой таблице, которую можно урезать, ведь в изображении совсем немного цветов. Выведем на экран палитру с цветовой таблицей. Меню ***Window***, команда ***Color Table***.

Делаем щелчок левой кнопкой мыши по пиктограмме Генерация цветовой таблицы. И вот перед нами цветовая таблица, состоящая из 256 цветов. Конечно, такое гигантское количество цветов не используется в этом простеньком изображении. Сократим количество цветов в цветовой таблице, оставив только самые необходимые, те, что используются в изображении птички. Упорядочим цвета в цветовой таблице по их использованию в данном изображении, то есть по тому, сколько пикселей каждого конкретного цвета встречается в этом изображении. Нажимаем на пиктограмму вызова скрытого меню палитры ***Color Table*** и выбираем команду ***Sort by Popularity*** (Упорядочить по количеству пикселей данного цвета в изображении).

Посмотрите: в этом изображении больше всего прозрачных пикселей, и после упорядочивания по популярности прозрачный цвет стоит на первом месте в цветовой таблице. Не буду сейчас разбираться с конкретными цветами, содержащимися в изображении. Ясно, что у меня должен остаться в цветовой таблице прозрачный цвет, пшеничный цвет, черный цвет, красный цвет и белый цвет. Точные их оттенки мне сейчас не важны. Это всё-таки не фотография, а схематичное изображение. Поэтому оставлю в цветовой таблице только вот этот диапазон цветов: от прозрачного до первого встретившегося красного, а все остальные удалю. Щелкаю по следующему за красным цветом в цветовой таблице левой кнопкой мыши, нажимаю на клавиатуре клавишу ***Shift***, и, удерживая ее, щелкаю по последнему цвету в цветовой таблице.

С помощью клавиши ***Shift*** выделяется непрерывный диапазон цветов, который я сейчас удалю. Нестрашно, если среди удаляемых цветов попадется цвет, который встречается в изображении. Он будет просто заменен на похожий из оставшихся, не удаленных, цветов. Щелчок по пиктограмме мусорной корзины внизу таблицы цветов. Тем самым удаляются выделенные цвета в цветовой таблице. Таблица урезана. Конечно, ее можно еще подсократить, но посмотрим, какой сейчас будет размер файла у изображения с анимированной птицей. Меню ***File***, команда ***Save Optimized***. Сохраняем изображение под именем ***Bird*** 3. Размер 122 килобайта, а было 206 килобайт. Но 122 килобайта — это тоже слишком много для размещения такой картинки

на веб-странице. Придется еще сокращать цветовую таблицу. Посмотрим, какой объем файла будет у данного изображения вот в такой, урезанной до предела, цветовой таблице. Меню ***File***, команда ***Save Optimized As***.

Сохраняем это изображение под именем ***Bird*** 4. И его размер уже вполне приемлемый: всего 45 килобайт. Чувствуете разницу? 206 килобайт, 122 килобайта и, наконец, 45 килобайт. Вот какого уменьшения размера файла можно добиться редуцированием цветовой таблицы изображения в формате ***GIF***. И вот наша птичка полетела. Вторая, меньшая, птичка, была сделана из первой просто уменьшением ее размера в 2 раза и затем тоже вставлена на эту веб-страницу.

Подобный файл с анимированным ***GIF***-изображением можно вставить на веб-страницу, открыть в программе ***Image Ready*** для дальнейшей обработки, а вот при попытке открыть подобный анимированный ***GIF*** в программе ***Adobe Photoshop*** появляется предупреждающее сообщение, что выведен будет только один кадр. И при последующем сохранении этого файла, простом сохранении, то есть под тем же именем и в том же месте, произойдет потеря информации, а именно, утеряны будут все остальные кадры этого изображения.

Но файл откроется как простой ***GIF***, состоящий из одного кадра. Аналогично можно вставить эту картинку в документ, созданный в программе ***Microsoft Word***, и картинка будет представлять собой один кадр из анимированного ***GIF***'а, а именно, первый кадр. А вот области прозрачности в ***GIF*** -файле, неважно, простом или анимированном, текстовым редактором ***Microsoft Word*** воспринимаются. Текст просвечивает сквозь области прозрачности изображения, что производит впечатление обтекания картинки текстом.

5.4. Разработка электронных документов в программе Adobe Illustrator CS

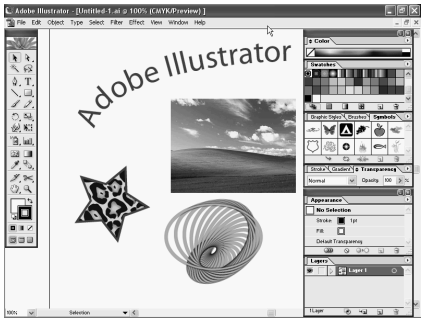
5.4.1. Создание Web страницы

Программа ***Adobe Illustrator*** позволяет создавать не только обычные графические документы и распечатывать их на принтере, но и создавать их электронные версии, создавать ***Web*** страницы. Из исходного документа ***Adobe Illustrator*** цветового формата ***RGB***, можно создавать различного типа электронные документы. Такие, как просто графические ***Web*** страницы, причем управляющие информацией, которая хранится в файле формата ***HTML***. А графическая, в файлах, выбранного формата для ***Web***. Таких, как растровые форматы ***GIF***, ***JPEG***, ***PNG***, ***WBMP*** и т.д. Или же векторных

форматах ***Flash*** и ***SVG***. Или же мы можем создать видеоклип с анимацией, хранящийся в специальном файле формата ***Flash***. Или же видеоклип с интерактивными эффектами.

Как понятно из всего вышеперечисленного, для создания ***Web*** страницы лучше всего подходит документ в формате ***RGB***, а не в формате ***CMYK***, как сейчас у нас. Поэтому, конечно же, прежде чем мы начнем создавать нашу ***Web*** страницу, лучше преобразовать наш документ к виду цветового формата ***RGB***. Для этого щелкнем на пункте меню ***File***, далее пункт ***Document Color Mode*** и затем выберем ***RGB Color***. Щелкнем. И вот теперь уже наш документ именно в том формате цветовом, который нам и нужен, ***RGB***.

Теперь же попробуем преобразовать наш документ к виду ***HTML*** файла. Это достаточно просто сделать. Для этого, достаточно после того, как мы скомпоновали нашу страницу и все элементы на нем, рисунки, надписи и т.д., щелкнем на пункте меню ***File*** и далее ***Save for Web***. Щелкнем на вот этом пункте меню. И в результате появляется вот такое стандартное окошко, в котором мы можем настраивать различные параметры для нашей будущей ***HTML*** страницы. Это окно состоит из нескольких основных частей.



Это панель инструментов, которая находятся слева из состоит из 6 инструментов, из центральной части рабочей области с видом нашей будущей ***HTML*** страницы, из строки состояния, которая находится в нижней части нашего окна и панели управления с различного рода настройками, которые находятся в правой части нашего окна настройки ***Web*** страницы. И наверху у нас находятся 4 вкладки для различного вида показа нашей ***HTML*** страницы. Рассмотрим сначала, какие у нас есть возможность вот в этой панели инструментов. Первая из них, это ***Hand Tool*** рука, при помощи которой мы можем перемещать ***HTML*** страницу по рабочей области. Сейчас, как мы видим, у нас включен именно этот режим, в результате чего мы свободно можем двигать нашу ***HTML*** страницу вверх-вниз.

Далее у нас находится ***Slice Select Tool*** выделение вырезки, при помощи которой мы можем в документе выделять вырезки, для задания для них личных файловых параметров. Сейчас, у нас пока вырезок на нашем документе нет, поэтому рассмотрим следующий пункт, пункт ***Zoom Tool***, при помощи которого мы можем задавать увеличение нашей ***HTML*** страницы и рассматривать ее с увеличением.

Например, мы задали более крупный вид на некую область нашей ***HTML*** страницы. Далее мы можем, например, переключиться на режим передвижения, при помощи вот такой руки и рассматривать именно те части нашей ***HTML*** страницы, которые нас больше всего интересуют в данном конкретном случае. Далее находится пункт ***Eyedropper Tool*** пипетка, при помощи которого мы можем в активном изображении образца выбрать какой-либо цвет. Например, выберем цвет вот в этой области. И сам цвет отображается в соответствующем прямоугольнике. И вот в этой области задания цвета можем видеть, какой цвет соответствует выбранной нами точке в нашей ***HTML*** странице.

Можем выбрать, например, другой, вот этот белый цвет, вот такой темно-зеленый, зеленый, в общем, где бы мы ни щелкали нашей пипеткой, соответствующий цвет и его выбор из нашей палитры, отображается в соответствующих местах нашей страницы, которая перед нами. И ниже, мы уже видели, вот этот кусочек, кусочек прямоугольника выбора цвета. Мы можем, конечно же, щелкнуть на нем непосредственно, и, так сказать, в ручном режиме выбрать какой-либо цвет для дальнейшего использования. Здесь абсолютно стандартные параметры, так же, как и в любом другом диалоге выбора цвета.

Мы можем выбрать, как в формате ***RGB***, задавая, красный, зеленый и голубой, составляющий цвета, можем выбрать в формате ***CMYK***, задавая соответствующие процентные составляющие оттенков. Или из любых других параметров. Можно непосредственно выбрать цвет, можно непосредственно выбрать из вот этой радуги цветов. Можем поставить галочку на ***Only Web Colors***, для того чтобы мы могли выбрать только из цветов, которые обычно используются на ***Web*** страницах, для более точного отображения нашей ***Web*** страницы на различного рода браузерах.

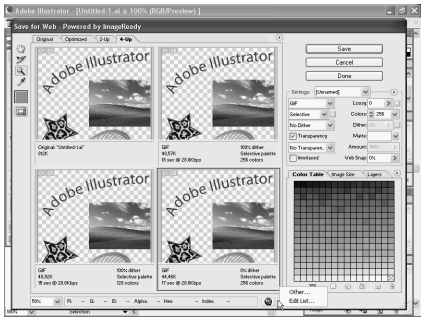
И после выбора можем щелкнуть на кнопке ОК, и этот выбор зафиксируется у нас на вот этом прямоугольнике цвета. И далее, ***Toggle Slices Visibility*** управление видимостью вырезок, задает управление отображения вырезок, доступных для выделения. И, после всего, что мы могли выбрать, отобразить мы можем, конечно же, при помощи нашего увеличения масштаба, изменить масштаб изображения. Вот, например, если мы сейчас просто щелкнем на каком-либо месте нашего экрана, то произойдет дискретное увеличение масштаба. А если мы нажмем на кнопку ***Alt***, и так щелкнем где-либо мышкой, то произойдет, наоборот, уменьшение масштаба изображения.

Вот, например, мы уже целиком поместили наш документ внутрь нашей рабочей области. А теперь чуть-чуть опять увеличим, в общем вот таким образом, мы можем регулировать масштаб и различного рода параметры, при

помощи инструментов, которые находятся в левой части этого окна сохранения документов в виде ***Web*** страницы.

Наверху у нас над рабочей областью находятся 4 вкладки. Первая из них, эта вкладка ***Original*** оригинал. При этом в рабочей области окна будет находиться лишь исходное изображение нашего документа. Далее находится вкладка ***Optimized***, при которой в рабочей области будет расположено результирующее изображение, файловые параметры которых оптимизированы программой или заданной нами. Далее находится вкладка 2-***up***, при которой у нас будут отображаться два варианта. Рабочая область, как мы видим, у нас разбивается на две половинку, правую и левую. В левой, из которых будет находиться исходное изображение, а в правой, будет находиться оптимизированное изображение, которое будет получено при помощи настройки различного рода параметров.

И последняя вкладка, вкладка 4-***up***, 4 варианта. При этом рабочая область, как мы видим, разбивается на 4 области. Наверху, слева будет находиться исходное изображение, справа оптимизированное. А в нижней части будут расположены два дополнительных обработанных изображения, файловые параметры которых отличаются от оптимизированного. И, в нижней части каждого из изображения, можно вкратце видеть те параметры, при которых и создано это изображение. Вот, в частности, например, вот это изображение у нас является рисунком ***GIF***, с 256 цветов. Время загрузки будет составлять 18 секунд при модеме 28,8 Кбайт в секунду. А размер файла будет составлять 48 с хвостиком Кбайтов.



И в нижней части остальных вариантов, тоже можно видеть соответствующие настройки, которые использованы для их создания. Далее, в нижней части нашего окна расположена строка состояния, которая состоит, во-первых, из окошка ***Zoom Level***, при помощи которого мы можем задавать увеличение для нашей рабочей области. Например, зададим 100% или любое другое значение. Например, можем задать 50%, что сразу же отражается на наших рабочих видах ***HTML*** страницы. И далее, находится информационное поле, в котором выводится информация о цветовых параметрах текущего пикселя, текущей точки изображения. Вот, по тому, как быстро мелькают цифры, можно увидеть, как меняется составляющая цвета, при перемещении нашей мыши по нашему документу. Конечно же, если мы вводим мышь по

прозрачным областям, то в этом случае, все составляющие цвета будут нулевыми.

Вернее, они будут просто-напросто отсутствовать. И далее, вот здесь находится, во-первых, кнопка запуска обозревателя по умолчанию, или же мы можем выбрать какой-либо другой браузер, для отображения этой ***HTML*** страницы, ее результирующего вида. Вот, например, щелкнем на ***Preview in Default Browser***, и можно видеть наш документ уже в стандартном обозревателе ***Microsoft Internet Explorer***. И в нижней части можно видеть описание ***HTML*** файла, который управляет выводом вот этого изображения. И все настройки, которые использованы для создания этого документа в ***HTML*** виде. Закроем теперь этот браузер, щелкнем на вот этом крестике, и мы опять попали в нашу программу ***Adobe Illustrator***.

Рассмотрим теперь те возможности, которые у нас сконцентрированы в правой части этого окошка. Наверху у нас находится кнопка ***Save***, при помощи которой можно сохранить изображение и наш документ в ***HTML*** файле уже непосредственным образом. Далее находится кнопка ***Cancel***, которая просто-напросто отменяет выполнение всех наших действий и возвращает нас к основному виду ***Adobe Illustrator***. И кнопка ***Done***, сохранение текущих параметров настроек с закрытием окна и без выполнения сохранения файлов в ***HTML*** виде. Или же, если мы нажмем на кнопку ***Alt***, то кнопка ***Cancel*** меняется на кнопку ***Reset***, при котором происходит восстановление исходных параметров настроек без его закрытия.

Или же следующая кнопка, кнопка ***Done*** меняется на кнопку ***Remember***, при которой происходит сохранение текущих параметров без закрытия данного окна. И ниже находится у нас раскрывающийся список ***Settings***, в котором мы можем выбрать стиль сохранения изображения. Здесь у нас представлены практически все виды графических файлов, такие распространенные, как ***GIF*** с различным количеством цветов. Далее ***JPEG*** вид с различного рода настройками по сжатию изображения и качества выходного документа. Или же файл типа ***PNG***. И далее, для каждого из выбранных нами форматов, можно настроить различного рода параметры.

Например, для файла типа ***GIF*** или ***JPEG*** и т.д. Выбрав, какой-либо из них, далее можем выбрать его настройки. Первый из этих параметров, например, это параметр ***Lossy*** регулировка допустимого уровня потери графической информации в растровом файле. Это можно регулировать при помощи вот этого ползунка. Далее список ***Colors***, при помощи которого можно задавать количество цветов в нашем ***GIF*** файле. Далее ***Specify algorithm***, в котором, мы можем при помощи ползунка ***Dither*** задавать уровень обработки изображения.

При помощи флажка ***Transparency*** мы можем подключить режим сохранения прозрачных участков, задавая соответствующий флажок и, задавая

тип прозрачности. При помощи пункта ***Matte*** мы можем задавать выбор цвета раскраски прозрачных участков исходного изображения. Если, конечно же, мы при этом отменяем флажок ***Transparency***. Далее у нас находится пункт ***Amount*** количество регулировки степени случайного перемешивания полностью прозрачных и непрозрачных точек. И флажок ***Interlaced***, задает подключение режима постепенного повышения, в качестве изображения в процессе его загрузки по сети, что направлено на снижение времени задержки появления изображения на экране.

Особенно это существенно для пользователей, у которых достаточно низкая скорость связи с Интернетом. В этом случае, конечно же, этот пункт позволяет ему получить хоть какое-то изображение достаточно быстро, которое затем будет повышаться в своем качестве. И пункт ***Web Snap*** со своим ползунком, позволяет регулировать диапазон цветовых оттенков, при попадании в которое, будет происходить замена исходных цветовых образцов изображения меньшим количеством, относящихся к таблице стандартной цветовой палитры ***Web***. Конечно же, если мы возьмем вместо формата ***GIF***, формат ***JPEG***, то соответствующие настройки уже совсем другие. В частности мы можем выбрать один из стандартных настроек по степени сжатия изображения. Мы можем выбрать флажок ***Progressive***, при помощи которого может задавать подключение режима постепенного повышения качества изображения в процессе загрузки по сети, что аналогично практически с тем же параметром для случая ***GIF*** изображения.

Далее качество в процентах, сейчас, как мы видим, 60. Можем задавать пункт ***Blur*** регулировка уровня незначительного размытия изображения. И последний пункт ***Matte*** выбор цвета, в которые будут раскрашены прозрачные участки исходного изображения. Можем выбрать формат ***PNG***, в этом случае соответствующие настройки тоже можно видеть перед нами. Можно выбрать файл ***WBMP*** или же файл формата ***SWF*** или ***SVG***. Сейчас выберем обратно файл с расширением ***GIF***, что достаточно часто используется в ***HTML*** страницах. Внизу этой части нашей страницы находятся три вкладки. Первая из них, это ***Color Table*** цветовая таблица, предназначена для цветовой обработки результирующего изображения, которое представлено в табличном цветовом формате. Что, достаточно четко видно на вот этом поле.

Далее находится ***Image Size*** размер изображения. Эта вкладка используется для изменения экранных размеров формируемого изображения, которая задается в количестве точек по горизонтали и по вертикали. И последняя вкладка, вкладка ***Layers***, позволяет представить слои исходного документа в качестве слоев каскадной стилевой таблицы будущей ***Web*** страницы, которая будет доступна для управления отображения на экране. В этой же части этого окна у нас находятся пункты меню. Если мы щелкнем на вот этой кнопке, то

появляется вот такое меню, в котором у нас есть несколько параметров. Первый из них, это ***Save Settings*** сохранить установки, используются для формирования нового стиля сохранения изображения из текущих файлов их параметров. Далее находится ***Delete Settings*** удалить установки. ***Optimize to File Size*** оптимизировать под файловый размер, при котором оптимизируются файловые параметры по заданному размеру файла.

Repopulate Views обновить виды, автоматически обновляет все файловые параметры для двух других образцов результирующих изображения, отличных от выбранного на вкладке 4-***up***, на которой мы сейчас как раз находимся. Далее у нас три пункта меню, связанных с ***Slices*** вырезками. Это связать вырезки, разорвать связи для вырезок, и разорвать связи для всех вырезок ***Unlink All Slices***. И последний пункт ***Edit Output Settings*** редактировать выходные настройки, при помощи которого мы можем раскрыть диалоговое окно выходных настроек с 4 вкладками, для настроек в нем выходных параметров, создаваемой ***Web*** страницы. Вот та страница, которая при нажатии на данный пункт меню, появляется перед нами. Как мы видим, здесь тоже у нас достаточно много параметров для различного рода настроек нашей ***HTML*** страницы. Щелкнем сейчас на кнопку ***Cancel*** и вернемся к нашему окну ***Save for Web*** сохранение ***Web*** страницы.

Отметим, что меню также появляется и при нажатии на вот эту кнопку, кнопку вызова соответствующего контекстного меню, в котором у нас есть еще достаточно много кнопок, связанных с настройками цветов. И пункты меню появляются при нажатии и на вот такую кнопку. Здесь у нас появляется размер результирующего файла, который мы можем выбрать и его время загрузки различного рода настройками модема или иной связи с Интернетом, что задается с соответствующей скоростью в пунктах меню. После того, как мы настроили все наши параметры и выбрали тип рисунка, его настройки, цветовую гамму и т.д., можем щелкнуть на кнопку ***Save***, после которого появляется вот такое стандартное окно, для сохранения нашего файла.

Здесь мы можем выбрать тип нашего файла, ***HTML*** и картинки в нем или только картинку в ***GIF*** файле, или же только ***HTML*** страницу, без сохранения рисунка. Выберем сейчас ***HTML and Images***, и его управляющие ***HTML*** файл, далее можем выбрать имя для нашего файла. Например, пусть будет ***My HTML***. И щелкаем на кнопке ***Save***. После чего наша страница успешно сохранена на нашем жестком диске, а мы вернулись к нашему основному виду ***Adobe Illustrator***. Теперь же, загрузив вот этот стандартный браузер и раскрыв ***My Documents***, Мои документы, которые находятся у нас на жестком диске, можем видеть вот эту ***HTML*** страницу, которую мы только что создали. Далее щелкнув на нем дважды, мы можем видеть, как будет выглядеть вот эта страница уже в браузере ***Microsoft Internet Explorer***. Как мы видим, она

представляет собой просто-напросто одну страницу, состоящую из просто-напросто одной картинки, одного ***GIF*** файла. Закроем теперь эту ***HTML*** страницу. Щелкнем на вот этом крестике, и мы опять вернулись в ***Adobe Illustrator***

5.4.2. Создание Web страницы с вырезками и гиперссылками

Рассмотрим теперь, как в ***Adobe Illustrator*** можно создавать ***Web*** страницы с вырезками, которые представляют собой статическую ***Web*** страницу, отображаемое содержание которых находятся в несколько графических файлах, а не в одном, как мы рассматривали ранее. Информацию, которых можем выбирать из приведенных в ***Adobe Illustrator***. Таких как ***GIF***, ***JPEG*** и т.д. Для этого, конечно же, перво-наперво, надо создать вырезки в нашем документе. Для этого выделим, например, соответствующий объект, например, вот эту звездочку. Щелкнем на пункте меню ***Object***, далее ***Slice*** и далее щелкнем на кнопке ***Make***.

И, как мы видим, у нас создалась вырезка, внутри которой находится наша звезда. Далее мы можем сделать то же самое и для других частей нашего изображения. Например, выделим вот этот текст ***Adobe Illustrator***. Далее щелкнем на кнопке ***Object***, ***Slice***, ***Make*** и, как мы видим, создали точно такую же вырезки и для вот этого текста. Выделим теперь вот этот рисунок. Далее ***Object***, ***Slice***, ***Make***. Вот, теперь у нас сформированы три вырезки по вот этим трем объектам. И для полноты картины, сформируем вырезку и для последнего нашего объекта.

Для этого щелкнем опять на пункте меню ***Object***, ***Slice***, ***Make*** и теперь попробуем создать управляющую информацию, для каких-то из наших вырезок. Например, сначала выделим вот этот текстовый объект, далее щелкнем на пункте меню ***Object***, ***Slice*** и далее ***Slice Options***. Щелкнем. Появляется вот такое стандартное окошко, в котором мы можем настроить, например, ссылку, куда мы будем переходить и, щелкнув на этом объекте. Например, введем какую-либо ссылку, ***http*** и далее введем, например, такую распространенную ссылку, как ***www.Google.com***. Далее то сообщение, которое будет появляться при наведении мышки на этот объект. Например, просто-напросто ***It is text***. И далее, тот текст, который будет появляться в виде подсказки внутри нашей

HTML страницы. Например, напишем здесь просто-напросто ***text*** и щелкнем на кнопку ОК, чтобы зафиксировать все наши изменения, связанные с вот этим текстом. Теперь введем, например, управляющую информацию, связанную с вот этой картинкой. Для этого щелкнем на пункте меню ***Object***, далее ***Slice*** и ***Slice Options***. Здесь введем, например, другую ссылку ***http***://***www.Adobe.com***.

Далее введем сообщение. Например, ***It is picture*** и введем тот текст, который будет появляться при наведении мышки. Например, просто-напросто ***Picture***. Щелкнем на кнопке ОК, и теперь попробуем сформировать и ссылку при щелканье на вот этой звездочке. Для этого поступим немножко по-другому. А именно, вызовем окно ***Attributes*** атрибуты, щелкнем на вот этом пункте. Далее выберем тип границы, в которой мы хотим, чтобы срабатывала наша гипер ссылка. Выберем ***Polygon***, для этого и зададим ссылку, по которой будем переходить. Например, выберем ***http***://***www.yandex.ru***. Вот теперь у нас эта ссылка тоже сформирована. А теперь, после всех этих настроек попробуем сохранить этот файл в ***Web*** странице. Для этого щелкнем на пункте меню ***File***, далее ***Save for Web***. Вот появился наш файл в окне ***Save for Web***, в котором сейчас ничего не будем менять в настройках, а щелкнем просто-напросто на кнопке ***Save***- сохранить.

Вот, появилось это стандартное окошко с сохранением файла. Выберем название ***MyHTML***2. Сохраним, как и в предыдущем случае ***HTML and Images***, т.е. и картинки и сам ***HTML*** файл, и щелкнем на кнопке ***Save***-сохранить. И вот, после этого мы вернулись обратно в наше стандартное окно ***Adobe Illustrator***. Открыв директорию ***My Documents*** — Мои документы, можем увидеть тот документ, который мы только что создали ***MyHTML***2. Попробуем его открыть. Для этого щелкнем на нем дважды. Вот эта ***HTML*** страница появилась перед нами и сейчас, если мы наведем на каждый из объектов, которым мы создали соответствующие вырезки и гиперссылки, то можно увидеть тот текст, который появляется в качестве подсказки.

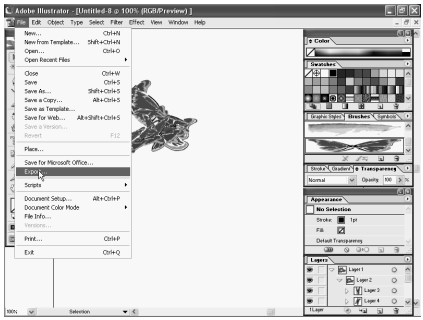
То, что у нас появляется в нижней части, в качестве статусной строки, можем видеть ту гипер ссылку, которая появится, если мы щелкнем на соответствующем объекте. Например, на вот этой звездочке. Как мы видим, у нас все наши настройки, которые мы создавали для наших объектов, исправно действуют. Закроем теперь вот это окошко. Щелкнем на вот этом крестике и вот мы опять вернулись в наше стандартное окно ***Adobe Illustrator***

5.4.3. Создание анимационного клипа. Движение объекта

В программе ***Adobe Illustrator*** можно создавать различные анимационные эффекты, причем для их сохранения используются файлы формата ***Flash***.

Каждый из таких анимационных эффектов может быть сохранен и в отдельном файле с расширением ***SWF*** и далее проигран при помощи ***Flash*** проигрывателя, или же может быть включен в состав ***Web*** страницы. Надо отметить, что для анимационного клипа, который мы можем создавать при помощи ***Adobe Illustrator***, есть некие ограничения. А именно, в этом клипе может быть использован всего лишь один анимационный эффект. А также мы не можем в этот клип включить интерактивный и звуковые эффекты. При этом надо отметить, что сам формат ***Flash*** такие эффекты в принципе поддерживает.

Рассмотрим теперь, как формируется в ***Adobe Illustrator*** такого типа клипы. Для этого существует три способа создания видеоклипа. Первый, это создание серии объектов любых типов, которые затем вручную распределяются по отдельным слоям, из которых далее собирается видеоклип. Причем каждый ***frame*** этого видеоклипа, будет представлять собой, и содержать один слой нашего документа.



Второй способ, это создание серии объектов произвольных типов с плавно изменяющимися параметрами, объединенных затем в групповой объект, который затем автоматически разлагается по отдельным слоям документа, который опять-таки является далее отдельными фреймами видеоклипа. И последний способ, это создание из двух объектов векторной графики перехода, который затем, точно также автоматически распределяется по слоям. Попробуем использовать каждую из этих возможностей в ***Adobe Illustrator***. Первый, создадим, так сказать, видео клип, вручную распределяя объекты по слоям.

Конечно, в принципе, это наиболее трудоемкий путь. Выберем, например, какой-либо объект. Проще всего взять какой-либо символ. Например, возьмем вот эту бабочку. Поместим ее на нашем документе. Далее нам нужно создать несколько копий вот этого объекта. Проще всего это сделать, используя инструментарии ***Brush***. Щелкнем на вот этом инструменте. Далее создадим новый ***Brush***. Щелкнем на вот этой кнопке ***New Brush***. Появляется вот это стандартное окошко, в котором выберем пункт ***New Art Brush***. Щелкнем на ОК. Появляется вот этот выделенный объект, в нашем случае, вот эта бабочка, из которой мы и будем создавать новую кисть. В принципе, согласимся со всеми установками по умолчанию. Здесь сейчас мы ничего не будем менять.

И щелкнем на кнопке ОК. Вот, новая кисть у нас зарегистрировалась и появилась в соответствующей палитре. Теперь выберем инструмент рисования, карандаш ***Pencil Tool*** и нарисуем, так сказать, эту бабочку несколько раз на нашем документе. Прежде, конечно же, нам нужно выбрать вот эту кисть, что у нас зарегистрировалась в соответствующем окошке ***Appearance***. И будем теперь рисовать эти бабочки, и их различного рода трансформации. Вот, таким образом, мы создаем все новые и новые бабочки на нашем документе. Теперь же нам нужно эти бабочки распределить по слоям. Для этого, конечно же, необходимо создать сами слои.

Конечно, можно щелкнуть, например, на вот этом меню и далее выбрать ***New Layer*** или же ***Duplicate Layer***, создать еще десяток слоев. Далее вручную переместить каждую из бабочек на соответствующий слой и в дальнейшем создать анимационный эффект. Это, конечно, в принципе, задача вполне достижимая теоретически, но практически достаточно сложно будет это реализовать. Особенно, если у нас вот таких промежуточных объектов достаточно много. Именно для этих целей в ***Adobe Illustrator*** существует, так сказать, другой путь. А именно, мы можем автоматизировать этот процесс. Для этого выделим все те бабочки, которые мы только что создали, все эти объекты. Далее сгруппируем их. Для этого щелкнем на пункте меню ***Object***, далее ***Group*** и, после того, как мы их сгруппировали, выделим вот эту группу в палитре ***Layers***.

Далее щелкнем на выпадающем меню и выполним пункт ***Release to Layers*** (***Sequence***) распределить по слоям. Если мы щелкнем, то можно увидеть, что все наши бабочки у нас распределились по соответствующим слоям. Как мы видим, их у нас 14 слоев, на каждом из которых расположена своя бабочка.

А теперь мы можем непосредственно создать наш видео клип. У нас уже все для этого готово. Для этого щелкнем на пункте меню ***File***, далее ***Export***. Щелкнем. Появляется вот такое стандартное окошко, для экспорта нашего документа. Здесь мы можем выбрать тип, в котором мы хотим экспортировать наш файл. Здесь у нас достаточно много возможностей, от файла ***AutoCAD*** до ***Windows Metafile*** (****.WMF***), но нас сейчас больше всего интересует ***Macromedia Flash***, файлы ***SWF***, файлы видео клипов.

Щелкнем. И выберем какое-либо имя для нашего видео клипа. Например, возьмем просто ***clip***. Далее щелкнем на ***Save***. Теперь появляется вот такое стандартное окошко, в котором мы можем настроить формат ***SWF*** файла. Первый из параметров, который здесь есть, это, как мы хотим экспортировать наш файл. Включить ли все так сказать составляющие части в один файл. Хотим ли мы сохранить каждый из слоев в отдельный фрейм, или же сохранить все слои в ***SWF*** файл.

Сейчас выберем вот этот средний вариант, сохранение каждого слоя в отдельном фрейме, при помощи которого и как раз происходит анимационный эффект. В этом случае мы должны задать еще и количество кадров в секунду. Сейчас, как мы видим, у нас выбрано три кадра в секунду. Мы можем взять, конечно же, большее значение, если мы хотим добиться более плавного перехода между отдельными кадрами нашей анимации.

Выберем пункт ***Looping*** для непрерывности нашего анимационного эффекта, чтобы он выполнялся бы так сказать циклически. Далее, как мы видим, мы можем или же сформировать отдельный ***SWF*** файл, или ж сгенерировать ***HTML*** страницу, в которую и будет включен наш анимационный клип. Далее хотим ли мы создать файл с атрибутом ***Read Only***, но пропустим этот пункт. Далее качество нашей кривой. Далее у нас несколько параметров, связанных с качеством каждого из изображения, входящих во фрейм. Согласимся с установкой по умолчанию, которая здесь у нас есть и щелкнем на кнопке ОК. Вот мы смогли преобразовать наш файл в ***HTML*** страницу с видео клипом и вернулись в наш ***Adobe Illustrator***. Вот мы запустили браузер, в котором отображена папка ***My Documents***, в которой и находится только что созданный нами ***HTML*** файл.

Вот этот файл с названием ***clip***. Попробуем его запустить. Для этого щелкнем на нем дважды. И можно увидеть, как при запуске нашей ***HTML*** страницы, на ней начинает прыгать вот эта бабочка и машет крылышками. Т.е. анимационный эффект нам вполне удался. Закроем теперь эту страницу. Щелкнем вот на этом крестике и вот мы опять попали в наше стандартное окно ***Adobe Illustrator***.

5.4.4. Создание анимационного клипа. Преобразование объектов и их появление

Создадим теперь анимационный эффект в программе ***Adobe Illustrator*** путем создания из двух объектов векторной графики, объекта перехода. Причем при этом составные части автоматически будут распределяться по отдельным слоям нашего документа. Выберем для этого какие-либо из символов, которые у нас есть в палитре символов. Возьмем вот эту бабочку с одной стороны и выберем, например, вот эту рыбку. Т.е. попробуем сделать так, чтобы вот эта бабочка при нашем переходе превратилась за несколько шагов, при помощи анимационного эффекта, вот в эту рыбку.

Для этого выделим вот эти оба объекта. Далее щелкнем на пункте ***Object***, далее ***Blend*** и щелкнем на пункте ***Make***. Вот можно увидеть, как при этом образовалась целая цепочка промежуточных объектов от нашей бабочки к нашей рыбке. Конечно же, мы можем выбрать пункт ***Object***, ***Blend*** и далее ***Blend Options***, задавая различного рода параметры. В частности, количество

шагов, за которые мы хотим превратить нашу бабочку в нашу рыбку. Сейчас, согласимся с той установкой, которая у нас есть, 10 шагов перехода, и щелкнем на кнопке ОК.

Теперь же разобьем этот объект перехода, так сказать, который получается превращением из бабочки вот в эту рыбку на отдельные его составляющие. Для этого щелкнем на пункте меню ***Object*** и далее ***Expand***. Оставим здесь все установки по умолчанию. Щелкнем на кнопке ОК. И мы превратили наш один объект перехода, который у нас был, в серию вот таких последовательных объектов. А теперь же выделим их все опять, и сгруппируем их в одну группу. Для этого щелкнем на пункте меню ***Object***, далее ***Group***. Щелкнем. А теперь это у нас один объект. Если мы щелкнем на любом месте вот в этом объекте, как мы видим, выделяются сразу все объекты одновременно.

А теперь щелкнем на нашем слое ***Layer***1, выделим вот эту группу, которую мы только что создали. Далее щелкнем на всплывающем меню и выполним пункт ***Release to Layers*** (***Sequence***) — распределить по слоям. Щелкнем. И как мы видим, наши все бабочки распределились по слоям вместе с переходом в рыбку, по вот этим слоям ***Layers***. Как мы видим, их у нас 14 штук, что можно убедиться, посмотрев на палитру слоев ***Layers***. А теперь мы можем приступить непосредственно к созданию нашего анимационного эффекта. Для этого щелкнем на пункте меню ***File***, далее, конечно же, ***Export***. Появляется вот это стандартное окно экспорта, в котором, конечно же, нам нужно выбрать файлы типа ***Macromedia Flash*** с расширением ***SWF***. Щелкнем. Теперь выберем название для нашего файла.

Пусть это у нас будет просто-напросто ***clip***2. Щелкнем на кнопке ***Save***. Конечно же, опять появляется вот это стандартное окошко с настройками нашего видео клипа ***Macromedia Flash***. В принципе можем сохранить все эти установки, которые здесь есть. Сохранить каждый из слоев в отдельном фрейме. Взять три фрейма в секунду, частота смена кадров. Установить циклическое отображение эффектов нашего видео клипа. Сформировать ***HTML*** файл, и т.д. Теперь же щелкнем на кнопке ОК.

И после этого наш ***HTML*** файл с вот этим эффектом перехода полностью готов. Вот мы запустили опять наш браузер обозреватель, в котором мы отобразили сейчас в папку ***My documents***, и в ней можно увидеть наш ***HTML*** файл ***clip***2. В принципе, здесь же можно увидеть наш ***Flash*** объект с расширением ***SWF***. Сейчас раскроем просто-напросто ***HTML*** документ. Щелкнем для этого на нем дважды. И после того, как наш ***HTML*** файл загружается, можно увидеть, как вот эта бабочка, последовательно двигаясь вниз, превращается при помощи соответствующего перехода в рыбку. Т.е. наш анимационный эффект достаточно успешно был реализован.

Закроем теперь это окошко. Щелкнем вот на этом крестике и мы вернулись в наше стандартное окно ***Adobe Illustrator***. Еще один вид эффекта, который мы можем реализовать на нашей ***HTML*** странице в ***Adobe Illustrator***, так сказать, в виде анимационного клипа, это создание клипа так сказать с наполнением. При котором каждый из последующих фреймов, будет так сказать дополнять предыдущую, а не только целиком его заменять. Попробуем это реализовать. Для этого создадим новый документ, чтобы так сказать очистить наше поле для работы. Закроем сначала вот этот. Щелкнем на вот этом крестике. Сохранять не будем. Щелкнем на кнопке ***No***. Далее щелкнем на пункте меню ***File***. Далее ***New***, новый документ. Согласимся со всеми установками по умолчанию ***RGB***, файл цветового формата, все размеры, которые есть, ориентацию, имя файла, пусть все будет, как и есть в этом окне.

Щелкнем на кнопке ОК. А теперь на нашем чистом документе, который у нас перед глазами последовательно, разбросаем на нем символы. Возьмем, какие-либо из них. В принципе, достаточно все равно, в каком порядке и как мы будем брать. Далее они будут так и появляться на нашем ***HTML*** файле в ***Web*** документе. Можем в принципе, повторить какие-либо из элементов. В принципе, нам это совсем даже не помешает. А теперь выделим все эти объекты одновременно, чтобы их сгруппировать. Это мы сделали при помощи вот такого прямоугольника выбора. Далее щелкнем на пункте меню ***Object***, далее ***Group*** сгруппировать. И после этого, у нас все эти объекты сгруппировались в одну группу. Можем это проверить, щелкнув на каком-либо из этих элементов. Как мы видим, выделились все объекты одновременно. Что как раз и означает, что у нас все объекты сгруппировались в одно целое.

А теперь же выделим эту группу в нашем списке слоев. Для этого щелкнем на вот этом треугольнике в ***Layers***. Далее щелкнем на вот эту группу. Затем щелкнем на кнопку появления контекстного меню и выберем на этот раз не ***Release to Layers*** (***Sequence***), распределить по слоям, а ***Release to Layers*** (***Build***) распределить по слоям с накоплением. Щелкнем. Вот появился список всех наших слоев с соответствующим расположением объектов в них. А теперь сформируем нашу ***HTML*** страницу с видео клипом. Для этого щелкнем на пункте меню ***File***, далее ***Export***, выберем имя для нашего файла. Пусть это будет ***clip***3. Щелкнем на кнопке ***Save***, сохранить.

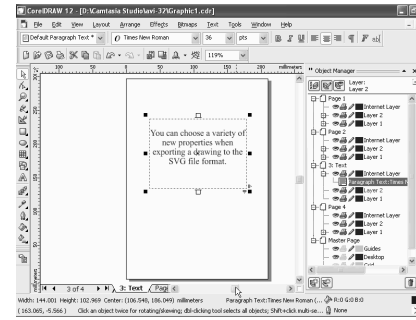
Конечно же, ***Macromedia Flash*** у нас уже выбран вид файла. Вот появилось перед нами окошко с настройками нашего ***Flash*** файла. Согласимся со всеми его установками, согласимся с тремя кадрами в секунду. В принципе этого вполне достаточно для наших целей и, конечно же, все файлы мы хотим сохранить, каждый из слоев в отдельном фрейме. Щелкнем сейчас на кнопке ОК. И наша цель успешно достигнута. ***HTML*** файл создан, и мы вернулись в

наш ***Adobe Illustrator***. Вот теперь мы опять открыли наш браузер папку ***My Documents***, в которой можно увидеть наш ***HTML*** документ ***clip***3, вместе, конечно же, с соответствующим сопутствующим ему ***Flash*** файлом, который мы можем в принципе проиграть на ***Flash*** проигрывателе, если у нас он установлен на нашем компьютере. Сейчас опустим нашу ***HTML*** страницу.

Щелкнем дважды. И можно увидеть, как на нашей ***HTML*** странице последовательно появляются все те объекты, которые мы и зарегистрировали для появления. В принципе, наш эффект полностью соответствует тому, что мы и ожидали. При этом, как мы видим, каждый из вот этих объектов, при появлении не затеряет так сказать предыдущий. Они появляются последовательно, накопляясь на нашей ***HTML*** странице. Закроем теперь этот файл. Щелкнем вот на этом крестике и вот мы опять вернулись в наше стандартное окошко ***Adobe Illustrator***.

5.5. Создание электронных документов в CorelDRAW 12

5.5.1. Обработка исходного документа. Ввод гиперссылок и закладок



Рассмотрим раздел Создание электронных документов. В программе ***Corel DRAW*** 12 можно создавать не только документы, предназначенные для печати, но и их электронные версии, допускающие использование двух файловых форматов векторного типа ***HTML*** с расширением ***Htm*** и ***PDF***. Первый формат используется при публикации документа в Интернете, а второй при любых видах электронной публикации. К исходному документу допускается подключать лишь файлы видеоклипов и аудиоклипов, которые будут воспроизводиться в электронном документе ***HTML*** формата. Никакие другие исполняемые файлы работать не будут. Это можно сделать с помощью команды из главного меню ***Edit*** / ***Insert Internet Object*** / ***Embedded file*** (Правка / Вставить объект Интернета / Внедренный файл).

Кроме этого существует еще два ограничения, которые следует принимать во внимание при форматировании файлов электронного документа; нельзя использовать русскоязычные имена для этих файлов и исходный документ может содержать связанные растровые файлы только тех форматов, которые

используются в ***Web***; ***GIF***, ***JPEG*** и ***png***. В противном случае изображения в электронном документе будут теряться. При обработке исходного документа и для сохранения его в формате ***HTML*** вам может понадобиться его дополнительная обработка, включающая; формирование в документе интерактивных состояний (обратитесь к главе 2) и обработку текстовых объектов содержащих текст обычного типа командой ***Text*** / ***Make Text HTML Compatible*** в данном случае это команда имеет вид ***Make Text Web Compatible*** (Текст / Сделать текст совместимым с Интернет).

Данная обработка позволит этот текст выделять в электронном документе с целью привлечения к нему внимания или копирования через буфер обмена ***Windows***. В результате обработки текстовых объектов командой ***Text*** / ***Make Text Web Compatible*** эти объекты будут преобразованы в объекты Интернета, располагаемые в отдельном верхнем слое исходного документа. Чтобы это было наглядно, вызовем докер ***Object Manager*** одноименной командой из главного меню ***Window*** / ***Dockers***. И выделим текст. Он у нас уже выделен. В главном меню ***Text***, как видите, рядом с пунктом ***Make Text Web Compatible*** стоит галочка. Уберем ее. Вот теперь из рабочего окна докера видно, что объект третьей страницы ***Paragraph Text***, т. е. вот этот объект, который расположен на текущем листе документа не будет располагаться в верхнем слое исходного документа.

А если снова поставим галочку рядом с данной командой, то вот этот объект ***Paragraph Text***; ***Times New Roman*** будет располагаться в отдельном верхнем слое исходного документа. Чуть сдвинем влево страницу, чтобы отобразить ее целиком на экране и рассмотрим ввод гиперссылок и закладок. Существуют два способа ввода в объекты документа гиперссылок и закладок, а также другой управляющей и текстовой информации. Первый из них базируется на использовании панели инструментов ***Internet***, а второй — докера ***Object Properties*** (Свойства объекта). Рассмотрим эти способы по порядку.

Рассмотрим порядок ввода информации первым способом. Откройте контекстное меню панели инструментов. В появившемся контекстном меню выбираем пункт ***Internet***. Чуть сдвинем панель ***Internet*** вверх и оставим ее в плавающем виде. Теперь выделите рабочим инструментом ***Pick*** (Выбор), в данном случае он активизирован, объект документа в который необходимо вставить управляющую и текстовую информацию. Допустим, выделим вот этот объект. Для вставки в объект закладки откройте список ***Behavior*** (Поведение) и выберите в нем пункт ***Bookmark*** (Закладка) и введите закладку в поле ***Internet Bookmark***. А для вставки гиперссылки выберите в списке ***Behavior*** пункт ***URL***. Затем выполните один из двух вариантов действий. Выберите в списке ***Internet Address*** (Адрес Интернета) одну из гиперссылок,

вводимые ранее в этот или другие документы, которые обрабатывались в программе. Пускай останется эта гиперссылка. Или введите с клавиатуры в редактированное поле данного списка новую гиперссылку. Здесь вы можете с клавиатуры ввести новый адрес.

Выберите в списке ***Target Frame*** (Целевой раздел) режим отображения в окне ***Web*** обозревателя новой страницы электронного документа связанного с его текущей страницей через гиперссылку. Щелчком на кнопке ***ALT Comments*** (Контекстные комментарии) откройте дополнительную панель и введите в ее поле контекстное сообщение, которое будет отображаться в электронном документе рядом с указателем. Щелчком на кнопке ***Hotspot*** (Горячая зона) откройте дополнительную панель и задайте на ней следующие параметры. Форма области срабатывания совпадающая с формой объекта, задается кнопкой ***Object shape***, а прямоугольной формой с помощью кнопки ***Bounding box of object***. Для настройки параметров оформления области срабатывания используются два флажка и два раскрывающихся списка с образцами цветов.

Повторите все эти шаги, которые мы только что продемонстрировали для всех объектов документа, в которые необходимо ввести управляющую и текстовую информацию, если они у вас есть. Закройте панель Интернет, щелкнув на кнопке с перекрестием находящейся в правом верхнем углу. Сейчас рассмотрим порядок ввода информации вторым способом. Для этого нужно открыть докер ***Object Properties*** (Свойства объекта), выполнив для этого команду ***Properties*** из главного меню ***Edit***.

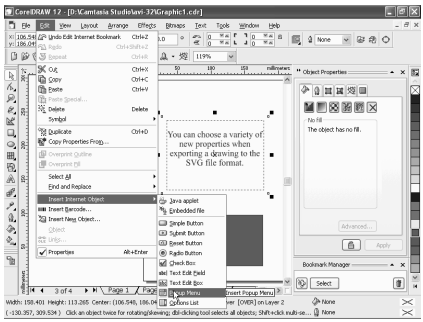
В результате на вашем экране появляется докер ***Object Properties*** в плавающем виде. Двойным щелчком на заголовке докера закрепим его в правой части экрана. С помощью инструмента ***Pick***, в данном случае он активизирован, выделим текстовой объект и перейдем в докере ***Object Properties*** на вкладку ***Internet***, щелкнув на ней. Теперь для вставки в объект закладки откройте список ***Behavior*** и выберите пункт ***URL***. А затем в списке ***URL*** выберите гиперссылку. Вы можете выбрать гиперссылку из данного списка или ввести с клавиатуры новый адрес. Выберите в списке ***Target Frame*** режим отображения в окне ***Web*** обозревателя новой страницы электронного документа связанной с его текущей страницей через гиперссылку.

Введите в поле ***ALT Comments*** контекстное сообщение, отображаемое в данном документе рядом с указателем. Откройте список, расположенный под полем ***ALT Comments*** и задайте форму области срабатывания, выбрав один двух пунктов; ***Object`s Shape*** — в форме объекта или ***Object`s Bounding Box*** — в форме прямоугольника описанного вокруг объекта. Используя два флажка и два раскрывающихся списка, вот они, с образцами цветов находящимся внизу докера, задайте параметры оформления области срабатывания. Если режим автоматического обновления параметров объекта не был задан — кнопка с

изображением замка внизу докера отжата, то щелкните на кнопке ***Apply*** (Применить) введя управляющую информацию в выбранный объект. В данном случае кнопка ***Apply*** не активизирована. Чтобы активизировать кнопку ***Apply***, в поле ***ALT Comments*** введем контекстное сообщение, допустим, ***break*** и нажмем на кнопку ***Apply***.

Повторите выше указанные шаги инструкции для всех объектов документа, если они у вас имеются в которые необходимо ввести управляющую текстовую информации. Закройте докер ***Object Properties***, щелкнув на кнопке с перекрестьем находящейся в правом верхнем углу. Также закроем докер ***Object Manager***, который мы использовали в предыдущем уроке. Расположим страницу документа в центре по его высоте и рассмотрим докер ***Internet Bookmark Manager***. С помощью докера ***Internet Bookmark Manager*** (Диспетчер закладок Интернета) открываемого одноименной командой подменю ***Dockers*** меню ***Windows*** вы также можете выполнять некоторые операции с закладками. В частности можно изменять связи между объектами документа или создавать новые.

5.5.2. Ввод гиперссылок и закладок. Вставка объектов Интернета. Формирование изображения для Web



В рабочем окне докера ***Bookmark Manager*** должен содержаться список закладок — поле ***Bookmark*** с указанием страниц, на которых находятся соответствующие закладки — поле ***Page***. В данном случае у нас нет созданных закладок. Давайте создадим закладку. Для этого сначала активизируем панель ***Internet*** из контекстного меню панели инструментов. Немного сдвинем ее вверх и из раскрывающегося списка ***Behavior*** выберем вариант ***Bookmark***, а в списке ***Internet Bookmark*** введем название закладки для данного выделенного объекта. Допустим, ***Break***. И нажмем клавишу ***Enter***. Можно также создать объект прямоугольной формы. Вот так. Для наглядности зальем его красным цветом и в выделенном его положении в поле ***Internet Bookmark*** введем закладку под названием, допустим, ***Rectangle***. После ввода названия нажимаем клавишу ***Enter***. Давайте уберем плавающую панель ***Internet*** с экрана.

Вот теперь у нас есть две закладки для верхнего объекта и для нижнего объекта — в рабочем окне докера ***Bookmark Manager*** содержится список

закладок — ***Break*** и ***Rectangle*** с указанием страниц — обе закладки расположены на третьей странице. Внизу докера расположены три кнопки управления, имеющие следующие назначения в порядке слева направо. Кнопка ***Link*** устанавливает связь между объектом, выделенным в окне документа и объектом с закладкой, отмеченным в окне докера. Кнопка ***Select*** открывает ту страницу документа, на которой находится объект с закладкой, выбранный в окне докера. В данном случае оба объекта находятся на третьей странице. И кнопка ***Remove*** удаляет закладку, выделенную в докере. Допустим, удалим ***Break***.

А сейчас рассмотрим вставку объектов Интернета. Откройте докер ***Object Properties*** (Свойства объекта) выполнив для этого команду ***Properties*** (Свойства) из главного меню ***Edit*** (Правка) или команду ***Properties*** (Свойства) из контекстного меню рабочего окна документа. В данном случае здесь галочка стоит. Откройте документ на требуемой странице и вставьте в нужном месте объект Интернета, используя для этого ту команду подменю ***Insert Internet Object*** (Вставить объект Интернета) из меню ***Edit*** (Правка) в названии которой указывается требуемый тип создаваемого объекта. В это подменю входят следующие одиннадцать команд; ***Java Applet*** (***Java*** приложение), ***Embedded file*** (Внедренный файл), ***Simple Button*** (Простоя кнопка), ***Submit Button*** (Кнопка инициации), ***Reset Button*** (Кнопка сброса), ***Radio Button*** (Переключатель), ***Check Box*** (Флажок), ***Text Edit Field*** (Текстовое поле с редактированием), ***Text Edit Box*** (Текстовая рамка с редактированием), ***Popup Menu*** (Всплывающее меню) и ***Options List*** (Список параметров).

При вставке в документ объектов Интернета все они будут располагаться в специально отведенном для них верхнем слое документа, положение которого изменять нельзя. Выделите объект Интернета рабочим инструментом ***Pick***, в данном случае у нас выделен вот этот текстовый объект, и перейдите на последнюю вкладку докера ***Object Properties***. На этой вкладке будет выполняться ввод в объект управляющей и текстовой информации. Если режим автоматического обновления параметров объекта не был задан — кнопка с изображением замка внизу докера отжата, то щелкните на кнопке ***Apply*** (Применить). Повторите вышеуказанные шаги инструкции для всех объектов Интернета, вставляемых в документ, и закройте докер ***Object Properties***. Можно также закрыть докер ***Bookmark Manager***, который мы использовали в начале урока.

Сейчас рассмотрим следующий раздел Формирование изображения для ***Web***. ***C*** помощью команды ***Web Image Optimizer*** (Оптимизатор изображения для ***Web***) из подменю ***Publish To The Web*** (Публикация на ***Web***) меню ***File*** вы можете решать две задачи; выполнять настройку файловых параметров изображения открытой страницы документа сохраняемого в одном из

растровых форматов для ***Web***; ***GIF***, ***JPEG***, ***png*** 8 или ***png*** 24, и формировать стили сохранения изображений с целью их последующего использования при создании электронных документов в формате ***HTML***.

Команда ***Web Image Optimizer*** обладает широкими возможностями по настройке файловых параметров результирующего изображения исходя из требуемого его качества и заданных ограничений на размер файла. Она позволяет выполнять цветовую обработку изображений имеющих табличный цветовой формат, что характерно для двух используемых файловых форматов ***GIF*** и ***png*** 8. Сначала перейдем на четвертую страницу документа, выделим растровое изображение с помощью инструмента ***Pick***, немного увеличим его размеры, подправим месторасположения и активизируем команду ***File*** / ***Publish to the Web*** / ***Web Image Optimizer***. Рассмотрим устройство данного окна. На верхней его панели находятся следующие элементы управления;

Список стандартных скоростей передачи модума в килобитах на секунду, предназначенный для выбора той скорости, для которой будет рассчитываться время загрузки по сети файла изображения. Список масштабов представления в данном окне образцов изображений документа. Немного увеличим данный масштаб, допустим, до 200 процентов и сдвинем чуть ниже. Дальше следуют четыре кнопки управления, задающие количество и расположение разделов в окне, в каждом из которых будет находиться один обрабатываемый образец изображения. Схема расположения разделов представлена на каждой кнопке.

5.5.3. Порядок формирования изображения для Web

Рассмотрим первую кнопку управления. При нажатии на нее растровое изображение располагается целиком в рабочем окне. При нажатии следующих кнопок, рабочая область разбивается на две вертикальные части, на две горизонтальные и на четыре части. В данном случае можно немного уменьшить масштаб отображения изображения. При этом достаточно подправить расположение изображения одного из разделов и щелкнуть в другом разделе, чтобы изображения внутри остальных разделов автоматически исправились.

Верхняя часть раздела представляет собой рабочую область, в которой находится текущий образец изображения. Изображение можно масштабировать и перемещать лишь в том разделе окна, в котором указатель принимает вид кисти руки, вот обратите внимание на левое верхнее окно. При этом изображения находящиеся в других разделах будут иметь такие же положение и масштаб, что и регулируемое изображение после щелчка на них.

В нижней части каждого раздела, допустим, в нижней части вот этого раздела находится управляющая панель раздела, выполняющая три функции. Первая — настройка файловых параметров текущего образца изображения, вторая — активизация изображения щелчком мыши в области панели с целью его сохранения в файле и третья — отображение параметров изображения характеризующих его файловые размеры и качество. На данной панели расположены следующие элементы настройки в порядке слева направо. Первый элемент это список файловых форматов для текущего образца изображения, включающий семь пунктов; ***Original*** — исходное изображение документа не обрабатывается, в файловом формате ***GIF***, в формате ***JPEG***, в формате ***JP*** 2 ***Standard*** и ***JP*** 2 ***Codestream***, в формате ***Png*** 8 и в формате ***Png*** 24.

Следующий элемент это список стилей сохранения изображения, который в данном случае не активизирован. Далее следуют кнопка создания нового стиля со значком плюс, кнопка удаления текущего стиля со значком минус и кнопка ***Advanced***, расположенная внизу справа, открывающая дополнительное диалоговое окно, предназначенное для настройки файловых параметров текущего изображения.

Внизу диалогового окна находятся пять кнопок выполняющих общие функции. Эти кнопки ***Preview*** (Просмотр), ***Reset*** (Сброс), ***Ok***, ***Cancel*** (Отменить) и ***Help*** (Помощь). В частности кнопка ***Preview*** задает режим автоматического обновления изображения в окне при изменении их параметров, а кнопка ***Reset*** выводит во всех разделах окна образцы исходного изображения документа. А назначение остальных кнопок понятно из их названий. А сейчас вкратце еще раз рассмотрим порядок работы с данным диалоговым окном.

В первую очередь нужно открыть документ на той странице, изображение которой будет обрабатываться командой, с целью его использования в ***Web***. В данном случае у нас открыта страница, на которой изображено вот это растровое изображение. После этого необходимо выполнить команду ***File*** / ***Publish to the Web*** / ***Web Image Optimizer*** (Файл / Публикация на ***Web*** / Оптимизатор изображения для ***Web***), открыв одноименное диалоговое окно. Вот это окно перед вами. Задайте требуемое количество и порядок расположения изображений в окне, которых должно быть не менее двух. Давайте выберем вариант с двумя горизонтальными разделами.

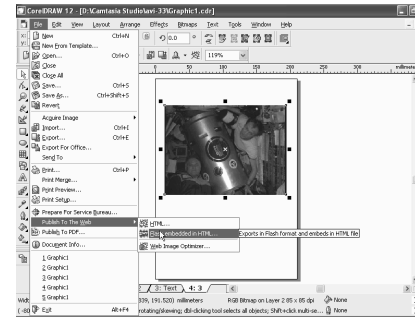
Подключите режим автоматического обновления изображения в окне, нажав кнопку ***Preview***. В одном из разделов окна выведите исходное изображение выбором пункта ***Original***, чтобы визуально сравнивать с ним образцы обрабатываемого изображения. В остальных разделах в данном случае во втором разделе окна, настройте файловые параметры результирующих образцов изображений. К примеру, выберем файловый формат ***GIF***. А из списка стилей сохранения изображений выберем вариант, допустим, 16 ***Color GIF Adaptive***. При необходимости вы можете выполнить настройки файловых параметров изображения с помощью кнопки ***Advanced***.

И чтобы улучшить качество результирующего изображения на вкладке ***Options*** увеличим параметр ползунка ***Dither Intensity***, допустим до максимального значения — до 100. И нажмем кнопку ***Preview***. Вы видите, что качество изображения немного улучшилось. Параметры остальных вкладок вы можете менять самостоятельно. И после достижения желаемого результат нажмите кнопку ***Ok*** два раза. Если вы выбрали, допустим, порядок расположения изображения в четырех окнах и имеется три варианта обработанных образцов, выберите тот обработанный образец изображения, который больше всего вас устраивает.

В данном случае выбираем изображение в нижнем окне, выделив панель в правой нижней части диалогового окна ***Web Image Optimizer*** красной рамкой. Щелкните на кнопке ***Ok*** и в открывшемся диалоговом окне ***Save Web Image to Disk*** (Сохранить на диске изображение для ***Web***) введите название файла изображения в поле ***File name*** (Имя файла). Откроем папку ***Sample Pictures*** и введем в поле имени файла название ***cosmos***. И для сохранения нажмем кнопку ***Save***. Таким образом, мы рассмотрели раздел Формирование изображения для ***Web***. В следующем разделе рассмотрим создание документов в формате ***HTML***.

5.5.4. Создание документов в формате HTML

На этом уроке рассмотрим создание документов в формате ***HTML***. В ***Corel DRAW*** 12 формирование электронного документа формата ***HTML*** может быть выполнено с помощью двух команд из подменю ***Publish to the Web*** главного меню ***File***. Эти команды ***HTML*** и ***Flash embedded in HTML*** (***Flash*** внедренный в ***HTML***). Рассмотрим их. Начнем с команды ***HTML***. Существует ряд особенностей создания электронного документа командой ***HTML***, которые состоят в следующем. Первая. Каждая страница исходного документа представляется в окне ***Web*** обозревателя отдельной страницей состоящей из основного файла формата ***HTML*** с расширением ***htm*** и набора файлов с растровыми изображениями. Между страницами электронного документа в



исходном документе могут устанавливаться связи через объекты, которые на них находятся. Вторая. Все графические объекты исходного документа, а также образцы художественного текста сохраняются в трех растровых файловых форматах для ***Web***, выбираемых пользователем; ***GIF***, ***JPEG*** и ***png***. Третья. Интерактивные состояния созданные в исходном документе переносятся в электронный документ, а файлы аудио клипов связанные с этими состояниями теряются. Четвертая особенность. Для объекта с гиперссылкой контекстное сообщение, задаваемое в поле ***Alt Comments*** панели инструментов ***Internet***, будет отображаться рядом с указателем только при условии, что этот объект не является растровым изображением.

В данном случае контекстное сообщение отображаться не будет, так как на данное изображение является растровым. Уберем панель ***Internet***. Пятая. Блоки обычного текста, один такой блок у нас находится на третьей странице, вот он, могут быть представлены в электронном документе в качестве текста доступного для выделения и копирования, либо в виде изображений. Необходимым условием сохранения текстовой информации в ***HTML*** файле является обработка исходного текстового объекта командой меню ***Text*** / ***Make Text Web Compatible*** (Текст / Сделать текст совместимым с ***HTML***).

Шестая. Русский текст обычного типа, обработанный командой ***Make Text HTML Compatible***, в окне ***Web*** обозревателя ***Internet Explorer*** 6 будет искажаться даже при выборе правильной кодировки с ключевым словом Кириллица в ее названии. Чтобы этого не произошло, вы должны задать в обозревателе пользовательский вид кодировки — пункт, определенный пользователем. Седьмая особенность. Русский текст, относящийся к объектам Интернета, в окне ***Web*** обозревателя ***Internet Explorer*** 6 будет отображаться без искажений при выборе кодировки, название которой содержит слово Кириллица. Перейдем опять таки на четвертую страницу и рассмотрим порядок создания документа. Выполните команду ***File*** / ***Publish to The Web*** / ***HTML*** (Файл / Публикация на ***Web*** / ***HTML***), открыв диалоговое окно ***Publish to The Web*** на вкладке ***General***.

Настройте параметры команды, сгруппированные по функциональному признаку на первых четырех вкладках окна. Последние две вкладки являются информационными. К числу наиболее важных параметров относятся следующие; Список ***HTML Layout Method*** (Метод макетирования ***HTML***

файла) на вкладке ***General***, в котором рекомендуется выбрать пункт ***HTML Table*** (***most compatible***). Имя и адрес папки, в которой будут находиться ***HTML*** файлы ***Web*** страниц документа, а также имя вложенной в нее папки с растровыми изображениями указываемые в области ***Destination*** вкладки ***General***. Диапазон сохраняемых страниц документа, а также режим сохранения выделенных объектов определяемые в области ***Export Range*** вкладки ***General***.

Названия ***Web*** страниц и их файлов указываемые на вкладке ***Details***, названия растровых изображений электронного документа, а также используемые стили их сохранения, определяемые на вкладке ***Images***. В данном случае в открытом документе у нас имеются следующие объекты.

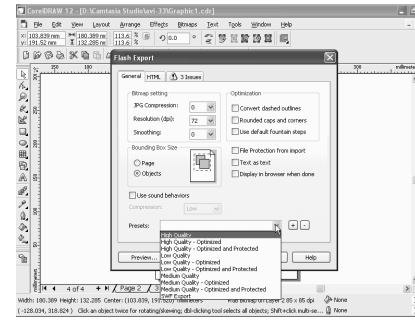
Вот они все. Если перед формированием файлов электронного документа вы захотите просмотреть его в окне ***Web*** обозревателя, щелкните на кнопке ***Browser Preview*** (Просмотр в Обозревателе) в нижней части окна ***Publish to The Web***. А теперь выйдем из обозревателя ***Microsoft Internet Explorer*** и для подтверждения в диалоговом окне ***Publish to The Web*** нажмем кнопку ***Ok***.

А сейчас рассмотрим команду ***Flash embedded in HTML***. Существуют некоторые особенности формирования электронных документов ***HTML*** формата командой ***Flash embedded in HTML*** (***Flash*** внедренный в ***HTML***). Рассмотрим их.

Первая. Создаваемый электронный документ состоит всего из одной ***Web*** страницы, являющейся копией текущей страницы исходного документа. При этом связи, устанавливаемые между страницами исходного документа, теряются. Вторая. Все содержимое исходного документа сохраняется в одном векторном файле ***Flash*** с расширением ***SWF***. Поэтому текстовая информация будет представлена в ***Web*** странице в виде изображения и ее нельзя выделять. И третья. Интерактивные состояния и связанные с ними файлы аудио клипов заданные в исходном документе переносятся в электронный документ.

5.5.5. Создание документов в формате PDF

Рассмотрим порядок создания электронных документов ***HTML*** командой ***Flash embedded in HTML***. Выполним команду ***Flash Embedded in HTML***, открыв диалоговое окно ***Export***. Выберите папку, в которой будут помещены два файла формируемого одностраничного электронного документа. В данном случае открыта папка ***export***. Если вы хотите поместить файлы формируемого электронного документа в другое место, используйте кнопку возврата и откройте желаемую директорию, где вы будете сохранять файл. Откроем опять таки директорию ***export***. Укажите название файла с изображением электронного документа имеющего векторный формат ***Flash***. Назовем его опять таки ***cosmos***. А теперь щелчком на кнопке ***Export*** закройте одноименное



диалоговое окно. Открывается окно ***Flash Export*** на вкладке ***General*** с параметрами сохранения документа в форматах ***HTML*** и ***Flash***. Откройте внизу вкладки список ***Presets*** (Предустановки) и выберите в нем стиль сохранения документа. В его названии указывается качество изображения и другие основные параметры. Выберем, допустим, ***High Quality — Optimized***. Если этот стиль вас не устраивает полностью, выполните настройку параметров команды сгруппированных по функциональному признаку на двух вкладках окна. Третья вкладка используется для выдачи предупреждающих сообщений о пробле MAX, которые могут возникнуть при формировании файлов электронного документа. В данном случае параметры менять не будем.

А если вы поменяли параметры и при этом выбранные параметры предполагается использовать в дальнейшем при работе с другими документами, то сохраните их в качестве нового стиля с помощью кнопки со знаком плюс, находящейся справа от списка ***Presets***. В поле ***Setting name*** появившегося окна введите имя для нового стиля, давайте введем имя ***cosmos***, и для подтверждения нажмите кнопку ***Ok***. Аналогично нажмите кнопку ***Ok*** в диалоговом окне ***Flash Export***. Таким образом, мы рассмотрели создание документов в формате ***HTML***.

Следующая тема создание документа в формате ***PDF***. Рассмотрим вопрос создания документа в формате ***PDF***. Формирование электронной версии документа ***Corel DRAW*** 12 в векторном файловом формате ***PDF*** основанном на языке ***PostScript*** выполняется с помощью команды ***Publish To PDF*** (Публикация в формате ***PDF***) меню ***File***. Существуют следующие особенности публикации документа в формате ***PDF***. Первая. Файлы видеоклипов и аудиоклипов подключаемые в ***Web*** обозревателе, через программу ***Quick Time*** загружаться не будут. Вторая. Интерактивные состояния и связанные с ними файлы аудиоклипов заданные в исходном документе, теряются в электронном документе. Третья.

Контекстное сообщение, задаваемое в поле или кнопкой ***Alt Comments*** (Альтернативные комментарии) будет отображаться в окне примечания свернутого в значок желтого цвета, расположенном рядом с соответствующим объектом. При этом русские шрифты для этого сообщения будут искажены. Четвертая. Русскоязычная текстовая информация, относящаяся к объектам

Интернета, будет искажаться из-за отсутствия необходимых шрифтов в программе чтения электронных документов.

Рассмотрим порядок создания документа. Активизируем команду ***Publish To PDF***. При этом на экране появляется диалоговое окно ***Save As PDF***. Выберите папку, в которую будет помещен файл сохраняемого документа, используя для этого кнопку возврата и список предложенных директорий. К примеру, выберем папку ***export***. И сохраним документ одним из двух способов. Первый способ заключается в выделении исходного названия в рабочей области диалогового окна с его последующей коррекцией в поле ***File name*** (Имя файла). В данном случае в рабочей области у нас нет файла.

А согласно второму способу, наименование файла сохраняемого документа в формате ***PDF*** вводится в поле ***File name*** с клавиатуры. Введем в данное поле название ***cosmos***. И для сохранения нажмите кнопку ***Save***. Но пока рассмотрим кнопку ***Settings*** (Настройки). Щелкнем ее. Открывается диалоговое окно ***Publish To PDF*** (Публикация в формате ***PDF***) на вкладке ***General*** (Общие) с параметрами сохранения документа в указанном формате. Откройте список ***PDF style***, находящийся внизу данного ока и выберите в нем стиль сохранения документа в формате ***PDF***.

Допустим, ***PDF for Editing***. И если данный стиль вас не устраивает полностью, то настройте параметры команды, сгруппированные по функциональному признаку на пяти вкладках окна; ***Objects***, ***Document***, ***Prepress***, ***Advanced*** и ***General***. Шестая вкладка используется для выдачи предупреждающих сообщений о тех проблеMAX, которые могут возникнуть при сохранении документа в данном формате. Если выбранные параметры предполагается использовать в дальнейшем при работе с другими документами, то сохраните их в качестве нового стиля сохранения документа в формате ***PDF*** с помощью кнопки со значком плюс, находящейся справа от списка ***PDF Style*** текущей вкладки. К примеру, сохраним выбранные параметры под именем ***PDF for Editing***.

И нажмем кнопку ***Ok***. Появившееся сообщение указывает, что у нас уже имеется такой стиль. Переименовать? Да переименовать. Нажимаем кнопку ***Yes***. Закройте окно ***Publish To PDF*** щелчком на кнопке ***Ok*** и щелкните в окне ***Save As PDF*** на кнопке ***Save*** (Сохранить). Теперь в директории ***export*** у нас имеется данное изображение, сохраненное в формате ***PDF***. Этим мы завершили шестую главу под названием Работа с документами и переходим к седьмой главе под названием Настройка параметров ***Corel DRAW*** 12.

5.6. Вставка мультимедиа при помощи программы FrontPage 2003

5.6.1. Вставка фильма в формате Flash

Macromedia Flash — это анимационная технология на основе векторной графики. Для вставки фильма в формате ***Flash*** нужно открыть страницу, на которую вы хотите вставить фильм. Например, пусть это будет страница ***details***. Страница открыта. Я закрою область Список папок. Фильм я вставлю в конце текста. Для этого нужно щелкнуть на полосе прокрутки. Поставьте курсор перед панелью ссылок. Для вставки фильма нужно открыть меню Вставка, навести указатель мыши на строку Рисунок.

В подменю выберите команду Фильм в формате ***Flash***. Открылось диалоговое окно Выбор файла. В этом окне нужно найти папку, в которой находится фильм в формате ***Flash***. Эта папка у меня открыта. Фильмы ***Flash***, готовые к просмотру имеют расширение ***SWF***. Такой фильм уже есть в списке, выберите его. Он взят из примеров, прилагаемых к программе ***Macromedia Flash***. После выбора файла нажмите на кнопку Вставить в данном диалоговом окне. В том месте, где находился курсор, появилась прямоугольная рамка. Такое место будет занимать фильм при просмотре. На рамке имеется надпись — это имя файла и расширение ***SWF***.

Для того, чтобы просмотреть фильм, нужно перейти к представлению Просмотр. Щелкните кнопку Просмотр. Одновременно с переключением в режим Просмотра. Появилось диалоговое окно Предупреждение о безопасности. В этом окне предлагается установить проигрыватель ***Macromedia Flash***. Без него фильм не будет отображаться, если на вашем компьютере не установлено соответствующее программное обеспечение. Подключитесь к Интернету. После того, как соединение установлено нажмите на кнопку Да. Когда проигрыватель будет установлен, щелкните на полосе прокрутки и здесь находится окно фильма в формате ***Flash***. Для того, чтобы лучше видеть, что здесь происходит, нужно увеличить это окно. Вернитесь в режим Конструктор.

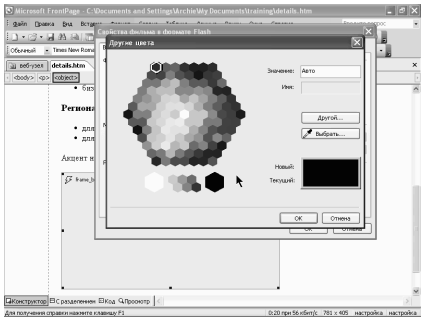
Щелкните прямоугольник. По краям его появились маркеры размеров. Можно увеличить это окно, перетаскивая маркер. Возьмемся за правый нижний маркер. Перетащите его по диагонали вниз. Теперь увеличим его вправо, взявшись за центральный маркер справа. Сохраните эти изменения. На экран выведено диалоговое окно Сохранение внедренных файлов. Чтобы полностью увидеть имя файла, увеличьте первый столбец. Переименовывать файл мы не будем. Нажмите на кнопку Сменить папку. На экран выведено

окно Смена папки. В списке мы видим те папки, которые присутствуют в веб-узле. Можно создать и новую папку.

Щелкните кнопку Создать папку в этом окне. Появилась новая папка и в ней рамка с текстом, который предлагает программа. Изменим этот текст, напишем, например, ***Flash***. Для подтверждения щелкните за пределами прямоугольника и нажмите на кнопку ОК в диалоговом окне Смена папки. Папка создана и файл будет помещен в нее, это папка ***Flash***. Нажмите на кнопку ОК в окне Сохранение внедренных файлов. Страница сохранена, перейдите в режим просмотра. Щелкните на полосе прокрутки. На этот раз окно стало больше и мы можем видеть, что здесь имеется несколько кнопок для управления фильмом, в частности кнопка остановки, запуска проигрывания и покадровой смены. Нажмите на кнопку для запуска проигрывания этого фильма. Вы видите, что на экране происходит действие. Это действие может происходить бесконечно, пока мы не остановим фильм. Перейдите обратно в режим Конструктор.

5.6.2. Изменение свойств фильма в формате Flash

Свойства вставленного на страницу файла в формате ***Flash*** можно изменять. Выделите окно фильма левой кнопкой мыши. По краям его появились маркеры размера, что указывает на выделение этого окна. Щелкните окно правой кнопкой мыши. В контекстном меню, которое сейчас появилось, выберите команду Свойства фильма в формате ***Flash***. На экран выведено окно Свойства фильма в формате ***Flash***. Здесь две вкладки: внешний вид и общие. Щелкните вкладку Внешний вид. Первая группа опций фильм, верхний список Качество, раскройте его.



Можно выбрать несколько вариантов: наилучшее, высокое, среднее, высокое авто, низкое авто и низкое. Чем более низкое качество мы выберем, тем быстрее будет загружаться фильм. Поэтому нужно выбирать оптимальный вариант между скоростью загрузки и качеством фильма. Я выберу высокое качество. Следующий список Масштаб. Сейчас выбран вариант по умолчанию все. Есть еще два варианта: без границы и подбор размера. Оставим здесь значение по умолчанию. Далее идет цвет фона. Раскройте этот список. Мы видим стандартные цвета: 16 цветов, которые имеются в этой программе, а также

выбранный нами ранее красный цвет, имеется ссылка на другие цвета. В результате щелчка на ней появляется диалоговое окно Другие цвета. Нажмите в нем кнопку Отмена. Если вы желаете, чтобы фон был прозрачным, установите флажок справа, который называется прозрачный. Ниже в той же группе имеется список выравнивание.

Сейчас выравнивание установлено по центру, имеются также такие варианты, как по левому краю, по правому и т.д. Следующая группа опций макет. Если выравнивание в группе фильм касалось выравнивания в веб-браузере, то здесь в Макете. Раскройте список Выравнивание в макете. Щелкнем, например, значение по середине. Установим толщину границы, пусть она будет равна двум точкам. Интервал по горизонтали предположим 20 точек., по вертикали то же самое. Далее идет группа опций Размер. Если установлен флажок задать размер, то мы можем изменять размеры окна. Если снять флажок, то изменять эти размеры мы не можем. Установите флажок. Значение по ширине установим равным 500 точкам. Для этого удалите то значение, которое сейчас там имеется и введите цифру 500. Нажмите на кнопку ОК для подтверждения изменений в диалоговом окне. Как видите, произошли изменения на странице.

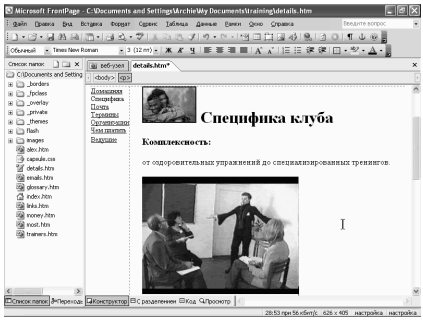
Окно фильма ***Flash*** изменилось, появилось некоторое расстояние между рамкой и внешней границей области ***Flash***. Я обозначил это расстояние стрелочками. Этот промежуток равен 20 точкам, как мы и задали в настройках. Щелкните окно ***Flash*** правой кнопкой мыши и выберите команду воспроизвести фильм в формате ***Flash***, изображение появилось уже в представлении Конструктор. Причем мы видим, что сам фильм занимает только часть окна, остальная часть — это просто рамка, опоясывающая его. Для проигрывания фильма нажмите на кнопку.

Как видите, проигрывание здесь не происходит. Нужно перейти в представление просмотр. Перейдя в это представление, щелкните на полосе прокрутки и уже здесь нажмите на кнопку Для проигрывания. Проигрывание началось. Мы можем при этом перемещать ползунок и нажимать на кнопки на полосе прокрутки. Вернитесь в режим конструктор. Произведем изменении настроек. Щелкните окно дважды.

Снова появилось диалоговое окно Свойства фильма в формате ***Flash***. Изменим выравнивание окна ***Flash*** в веб-браузере. Сейчас оно установлено по центру. Установите по правому краю. В группе макет выравнивание сделаем по нижнему краю. Интервал по горизонтали и вертикали удалим, пусть он будет равным нулю, а в группе Размер уменьшим величину окна, при этом сохранять пропорции не будем. Пусть например высота составляет 300 точек, а ширина 400. Нажмите кнопку ОК. В представлении конструктор мы видим, какие изменения произошли.

Перейдите в представление просмотр. Для того, что бы запустить проигрывание нажмите на соответствующую кнопку. Остановите проигрывание. Вернитесь в представление Конструктор. Выполним ещё некоторые изменения. Щёлкните дважды окно ***Flash*** фильма. В группе Фильм мы установили цвет фона как прозрачный. Снимите этот флажок. Раскройте список Цвет фона. В группе Стандартные цвета выберите Тёмно-синий цвет. Для подтверждения изменений нажмите кнопку ОК. Щёлкните кнопку Просмотр. Проигрываем фильм. С фильмом никаких изменений не произошло, но по краям, где был фон — появились синие полосы. Это и есть фон в окне ***Flash*** фильма. Вернитесь в представление Конструктор.

5.6.3. Вставка видеозаписи



На веб-странице можно вставлять видеофрагменты, видеофайлы. Откройте файл ***Details***. Поставьте курсор там, где должен быть клип. Например, щёлкните после слова тренингов, нажмите клавишу ***Enter***. Т. о. мы создали новую строку. Для вставки видео нужно раскрыть меню Вставка, навести указатель мыши на строку Рисунок. В появившемся подменю выберите команду Видеозапись. На экран выведено диалоговое окно Видео. В списке папок найдите нужную. Я знаю, что нужный мне файл находиться в папке Документы. Я раскрыл список папок и файлов, имеющихся в папке Мои документы и здесь мне нужен файл с именем ***Training*** и расширением ***avi***. Щёлкнув на этот файл, нажмите кнопку Открыть. Окно, содержащее первый кадр видео, появилось на странице. Для сохранения изменений нажмите на соответствующую кнопку.

На экран выведено окно Сохранение внедрённых файлов. Нажмите на кнопку Переименовать. Вместо ***T*** заглавной напишем ***t*** — строчную. Кроме того, создадим новую папку и поместим в неё этот файл. Для этого нажмите на кнопку Сменить папку. Появилось диалоговое окно Смена папки. В нём щёлкните кнопку Создать папку. Появилась новая папка, в которой нужно ввести новое имя. Пусть это имя будет ***video***. Теперь нажмите на кнопку ОК. Папка ***video*** создана, и файл будет помещён в неё. Для подтверждения изменений нажмите на кнопку ОК, в окне Сохранение внедрённых файлов. Перейдите в представление Просмотр.

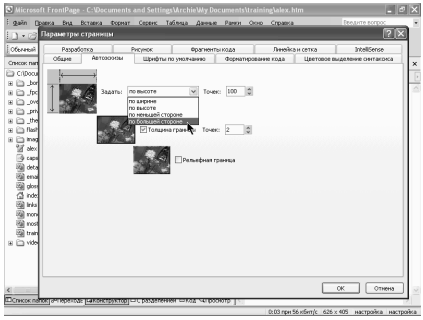
В этом представлении мы не видим видеофрагмента. Вместо него отображается только рамка и значок изображающий файл мультимедиа. Для того, что бы увидеть видеофрагмент, нужно открыть веб-обозреватель. Раскройте меню Файл. Наведите указатель на строку Просмотреть в обозревателе. Выберем, например такой веб-обозреватель как ***Opera***. Здесь мы видим этот видеофрагмент в действии, так сказать. Закройте этот веб-обозреватель. Параметры воспроизведения видеозаписи можно изменять по своему усмотрению. Щёлкните окно видеозаписи правой кнопкой мыши. В контекстном меню выберите команду Свойства рисунка. На экран выведено окно Свойства рисунка. Автоматически открылась вкладка Видеозапись. Здесь указан источник видеозаписи, это файл с именем ***training***, находящийся в папке ***video***. Ниже находится группа Повторить. Здесь указывается число повторов и задержка повтора. Увеличим значение повторов до 2-х.

Нажмите на кнопку ОК в окне Свойства рисунка. Сохраните изменения на веб-странице. Затем раскройте меню Файл, Просмотреть в обозревателе. Используем для просмотра программу ***Opera***. Программа открыта, и мы видим изображение, и слышим звук. Это был, так сказать, первый прогон и уже начался второй. Мы во второй раз видим то же самое, что видели в первый. После этого изображение просто исчезло в программе ***Opera***. В программе ***Microsoft Internet Explorer*** такого не будет. Изображение останется на экране. Закройте программу ***Opera***. Для того, чтобы изменить параметры воспроизведения видеозаписи, можно щёлкнуть данное окно дважды, левой кнопкой мыши. Сразу же появляется диалоговое окно Свойства рисунка, и вкладка Видеозапись уже открыта. Установите флажок Непрерывно.

В таком случае счетчик Число повторов становится недоступным, мы не можем менять здесь значения, и видеозапись будет проигрываться постоянно. Кроме того, её начало может зависеть от того, навели мы указатель мыши на окно или нет. Щёлкните переключатель При наведении указателя мыши. Подтвердите изменения нажатием кнопки ОК. Откройте ещё раз программу ***Opera***. Для этого нужно раскрыть меню Файл. Навести указатель мыши на строку Просмотреть в обозревателе и щёлкнуть в данном случае первый пункт. После чего программа требует сохранить изменения. Мы их не сохранили перед этим. Нажмите на кнопку Да.

Opera запустилась, при этом проигрывание началось сразу, без наведения указателя мыши. Т. е. в данной программе данная функция не работоспособна, но она будет работать в программе ***Microsoft Internet Explorer***. Зато проигрывание происходит здесь постоянно и это может продолжаться бесконечно, пока не закончится электроэнергия у компьютера. Закройте окно программы.

5.6.4. Настройки автоэскизов



Во вводной части курса Вы с автоэскизами. Раскройте двойным щелчком файл ***Alex***. Здесь имеется изображение, которое можно превратить в эскиз. При чём размер этого эскиза известен заранее. Для того, что бы увидеть, каким он будет — раскройте меню сервис, выберите команду Параметры страницы. На экран выведено диалоговое окно Параметры страницы. В этом окне имеется несколько вкладок. Из них выберите вкладку Автоэскизы. Верхняя опция на этой вкладке — это задание размера. Здесь имеется список вариантов. Сейчас выбран вариант По ширине, т. е. по ширине эскиз будет равен 100 точкам., высоту программа определит автоматически, так, что бы изображение изменилось пропорционально и не было искажено. Но мы можем задать расстояние не только по ширине для эскиза, но и например по высоте. В таком случая высота будет равна 100 точкам. Высота показана на схеме слева, ширина сверху. Какие ещё здесь есть варианты? По меньшей стороне и по большей стороне.

Если мы выберем вариант по меньшей стороне, то меньшая сторона изображения, которое мы превращаем в эскиз будет равна 100 точкам. Если по большей, то соответственно большая сторона будет равна 100 точкам, а меньшая ещё меньше. Установим значение По ширине и пусть, предположим оно будет равно не 100 точкам, а 80. Удаляем значение, которое сейчас расположено в счетчике и вводим 80. Следующая опция Толщина границы, флажок сейчас установлен. Давайте его снимем. В таком случае счётчик справа, указывающий сколько точек будет толщина границы — становится не доступным. Установите снова толщину границы. Значение это можно изменить. Вместо двух точек, например, напишем 10. Следующая опция Рельефная граница. Если Вы хотите, что бы граница была рельефной, установите флажок. Попробуем применить эти настройки и посмотреть, какой автоэскиз получится в результате. Нажмите на кнопку ОК.

Тем не менее окно Параметры страницы не исчезло. Программа вывела на экран сообщение: ***Please enter an integer between*** 0 ***and*** 8, что означает: Пожалуйста, введите целое число от 0 до 8. Нажмите кнопку ОК. Куда ввести это целое число, программа указала, выделив цифру 10 в счетчике точек. Введём цифру 8-ь, как она того требует. Теперь нажмите кнопку ОК в окне Параметры страницы. На этот раз диалоговое окно исчезло. Значит, настройки

приняты. Выделите изображение щелчком мыши на веб-странице. Откройте меню Сервис, в этом меню имеется команда Автоэскизы, щёлкните её. Изображение превратилось в эскиз. При чём вокруг этого эскиза появилась рамка, толщиной в 8 точек надо полагать. Проверять не будем. Я думаю в данном случае можно поверить программе. Попробуем произвести то же самое действие другим способом. Нажмите копку Отменить на панели инструментов.

Щёлкните изображение правой кнопкой мыши. В контекстном меню имеется команда Автоэскизы. Выберем её. Результат то же, изображение превратилось в эскиз. Отмените действие. И ещё один способ, это открыть меню Сервис снова и посмотреть, какое сочетание клавиш позволяет превратить изображение в эскиз, а это сочетание ***Ctrl*** + ***T***. Не будем здесь щёлкать. Щёлкните за пределами меню, выделите изображение и нажмите на клавиатуре сочетание клавиш ***Ctrl*** + ***T***.

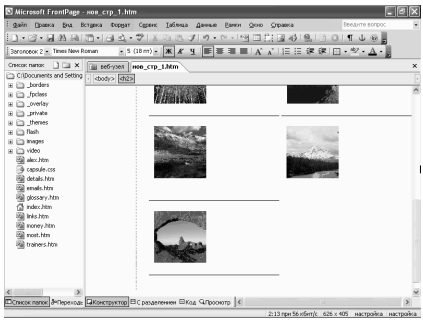
Как видите эффект то же самый, изображение превратилось в эскиз. Для сохранения изменений нажмите на кнопку Сохранить. На экран выведено окно Сохранение внедрённых файлов. Поскольку файл ***shura*** уже существует в папке ***images***, имеющейся на нашем веб-узле. Программа предлагает сохранить только то файл, который сейчас создали, т. е. этот эскиз и предлагает назвать его ***shura_small***. Согласимся с этим, но не согласимся с тем, какую папку следует выбрать. Нажмите на кнопку Сменить папку. Выберите в окне Смена папки папку ***images***, нажмите кнопку ОК. В столбце папка появилось название ***images***. В окне Сохранение внедрённых файлов нажмите на кнопку ОК.

Перейдите в представление Просмотр. Здесь так же хорошо видно и рамку толщиной 8 точек, которую мы создали, можно увидеть и рельефную рамку, которая находится внутри синей рамки и само изображение естественно. При наведении на него указателя мыши, указатель превращается в руку, щёлкните изображение. Мы вывели на экран полное изображение, которое было на странице ранее. Программа создала на него ссылку автоматически. Обновите страницу . После того, как был совершён переход по гиперссылке, вместо синего мы видим фиолетовый. Вернитесь в режим Конструктор.

Изменим настройки автоэскизов. Откройте меню Сервис, выберите команду Параметры страницы. Сделаем снова ширину равную 100 точкам. Толщину границы равной 2 точкам и снимем флажок Рельефная граница. Мы получаем те настройки, которые были по умолчанию. В самом начале. Нажмите на кнопку ОК. Изменение настроек будет применятся только к новым эскизам, которые мы создадим позже. Для того, что бы изменить границы сейчас на имеющемся эскизе, выделите его двойным щёлчком левой клавиши мыши.

Появилось диалоговое окно Свойства рисунка. В группе Положение на вкладке Вид, можно изменить толщину границы. Вместо 8 точек, которые мы определили в настройках, установить значение 2. Такое значение определяется по умолчанию. Нажмите кнопку ОК, для подтверждения изменений. Толщина границы изменилась.

5.6.5. Создание коллекции фотографий на основе шаблона



Коллекция фотографий — это веб-страница, содержащая фотографии, расположенные в соответствии с каким-то макетом. Для создания фото коллекции можно использовать шаблоны. Щёлкните стрелку рядом с кнопкой Создание новой обычной страницы. В меню выберите строку Страница. На экран выведено диалоговое окно Шаблоны веб-страниц. На вкладке Общие выделите шаблон Коллекция фотографий. Описание и изображение образца этого шаблона показаны справа. Для подтверждения выбора щёлкните кнопку ОК. На экране появилась новая страница, содержащая коллекцию фотографий. Пока это всего лишь образец. Сохраним эту страницу. После нажатия кнопки Сохранить, появилось диалоговое окно Сохранить как. Сейчас открыт тот веб-узел над которым мы работаем, ***training***. Следует задать имя файла. То имя, которое здесь предложено, удалим и напишем другое ***collection***, например.

Название, коллекция фотографий, которое предложила программа оставим. Нажмите на кнопку Сохранить. Программа вывела окно Сохранение внедрённых файлов, это те образцы, которые уже есть на странице. Эти образцы нам не нужны, поэтому я нажму кнопку Отмена в окне. Страница не была сохранена. Щёлкните на полосе прокрутки, щёлкните в области фотографий. Эта область выделена. Нажмите на кнопку ***Delete*** на клавиатуре. Группа фотографий удалена и теперь мы можем создавать коллекцию практически с самого начала. Но, можно ли было просто добавить новые фотографии в имеющуюся коллекцию. Нажмите кнопку Отменить на панели инструментов.

Щёлкните область фотографий дважды. На экран выведено диалоговое окно Свойства коллекции фотографий. Как только мы добавим что либо в коллекцию, образцы из неё исчезнут. На вкладке Рисунки нажмите на кнопку

Добавить. Появилось выпадающее меню состоящее из двух пунктов: Рисунки из файлов и Рисунки из сканера или камеры. Щёлкните строку Рисунки из файлов. На экране появилось диалоговое окно Открытие файла. Сейчас Вы видите открытой папку Коллекция картинок ***Microsoft***. Раскройте список. Щёлкните папку Документы. В списке найдите папку ***training***, это тот веб-узел, над которым мы работаем. Здесь откройте папку ***images***. Щёлкните фотографию под именем ***dance***, нажмите на кнопку Открыть, в окне Открытие файла. Имя данного файла появилось в окне ниже.

Здесь список тех имён, которые мы будем вносить в фотоколлекцию. Правее появился фотоэскиз этой фотографии, ещё правее размер эскиза — 100 точек по ширине и 70 по высоте. При этом программа сохраняет пропорции. Если мы изменим ширину или высоту, то соответственно изменится и второй параметр. Например, увеличьте ширину. По мере увеличения ширины, увеличивается и высота. Установим снова ширину равной 100 точкам. Высота автоматически стала равной 70. Под эскизом находится запись о том: сколько фотографий находится в коллекции. В настоящий момент 1-а. Кнопки Вверх и Вниз, находящиеся левее сейчас не доступны. Ещё ниже находится раздел, в котором Вы можете определить параметры шрифта.

Сейчас выбран переключатель использовать параметры шрифта страницы. Шрифт будет в подписи. Второй переключатель — это использовать другие параметры шрифта, щёлкните его. В этом случае становятся доступными Имя шрифта, Размер и цвет. А так же кнопки Полужирный, Курсив и Подчеркивание. Поставьте курсор в поле подпись, находящемся ниже. Здесь напишем подпись для фотографии, например, Танец. Ещё ниже можно ввести описание. Например «танцевальная терапия». Установим другой шрифт. Для этого нужно выделить текст. Затем раскрыть список шрифтов. Выберем например такой шрифт как ***Tahoma***. Представление текста изменилось. У него стало другое начертание. Можно изменить и размер. Выделите снова этот текст, щёлкните список размеров. Выберите второй к примеру.

Кроме того, можно изменить и цвет этого текста. Предположим, выберем малиновый. Нажмите кнопку ОК для подтверждения изменений. Щёлкните в том месте на странице, где пропадает значок с рукой, который мы сейчас видим. Таким образом мы ввели на страницу первую фотографию, подпись и описание к ней. Сохраните эту страницу. Сохранить как, пишем имя этой страницы ***collection***. Нажмите на кнопку Сохранить. Поскольку фотография уже была в папке ***images***, которая находится в нашем веб-узле, программа не требует сохранить это изображение как новое.

5.6.6. Создание коллекции фотографий как Web-компонента

Коллекцию фотографий можно создать не только на основе шаблона страницы, но и с помощью меню Вставка. Удалим файл ***collection*** из веб-узла. Закройте его. Выделите этот файл в области Список папок и нажмите клавишу ***Delete*** на клавиатуре. Появилось диалоговое окно Подтверждение удаления: «Вы действительно хотите удалить Коллекция фотографий ***collection***?» Нажмите на кнопку Да. Папку ***photo gallery***, которая была создана программой, тоже можно удалить. Выделив её нажмите на кнопку ***Delete*** на клавиатуре. Подтвердите удаление. Для создания коллекции фотографий щёлкните кнопку Создать страницу, в области Список папок или кнопку Создание новой обычной страницы. Я щёлкну первую кнопку.

Появилась Новая страница 1, такое имя предложила программа. Переименуем этот файл. Назовём его ***collection***. Подтвердите изменение имени, нажав клавишу ***Enter*** на клавиатуре. Теперь откройте этот файл. Курсор должен находится там, где будет коллекция фотографий. Сейчас он находится на странице в этом месте, пусть здесь и будет. Раскройте меню Вставка, выберите команду Веб-компонент. На экран выведено диалоговое окно Вставка компонента веб-узла.. В списке Тип компонента выберите строку Коллекция фотографий. Справа появилось 4-е варианта коллекций. Горизонтальное положение, так как сейчас, этот образец выбран, следующий Коллаж, группа фотографий на странице будет располагаться в виде коллажа).

Ещё один тип расположения, это в виде набора слайдов, когда эскизы расположены в верхней части и один из них отображается целиком. И последний вариант — это Вертикальная компоновка. Выберите первый вариант, нажмите на кнопку Готово. Появилось диалоговое окно Свойства коллекций фотографий. Нажмите на кнопку Добавить на вкладке Рисунки. Выберите строку Рисунки из файлов. Открылась папка, которую я открывал в прошлый раз — ***images***. Здесь я выберу те фотографии, которые хочу разместить в коллекции. Фотография ***dance***, ***desiccation***. Я выбираю их при нажатой клавише ***Ctrl***., а так же ещё несколько фотографий ***hands***, ***love***, ***portrait***, ***phone talk***, ***round***, Семенов, Вова и Таня. Нажмите на кнопку Открыть в диалоговом окне.

Сразу все эти фотографии помещены в список на вкладке рисунки. В списке фотографии распределены по алфавиту. В данном случае начинается с буквы ***D*** и заканчивается буквой ***V***. Порядок размещения фотографий можно изменять, нажатием кнопки вверх или вниз. Например, выберите фотографию ***round***, поместим её вверх. Для этого нужно нажимать кнопку вверх, столько раз, сколько необходимо, для перемещения фотографии. Сейчас она находится в самом верху списка. Для перемещения ниже нажмите кнопку Вниз. Число

рисунков в этой коллекции как указывает программа 10, так что считать их не нужно. Создадим подписи для фотографий. Для этого выберите фотографию ***dance***, например, и создайте подпись. Пусть она просто называется танец. Описание введём ниже. Пусть описание будет например таким. Переходим к следующей фотографии. Для неё так же нужно придумать подпись и описание. После ввода текста переходим к следующей фотографии и т. д.

Пока не будут заполнены поля для всех фотографий. После того как ввод подписей и описание фотографий завершён, нажмите на кнопку ОК в окне свойства коллекции фотографий. Коллекция создана на странице. Для сохранения изменений на странице, нажмите сочетание клавиш ***Ctrl*** + ***S***. Одновременно с сохранением страницы в области список папок появилась папка ***photo gallery***. Нажмите на знак «+» слева от изображения папки, здесь находятся ещё две папки. Щёлкните знак «+» рядом с первой из них. Папка не раскрылась, значит там ни чего нет.

Щёлкните следующий «+». В этой папке находятся изображения, а так же дополнительные файлы. Щёлкните знак минус рядом с папкой ***photo gallery***. Что бы лучше увидеть нашу коллекцию на странице закройте область Список папок и воспользуйтесь нижней кнопкой на полосе прокрутки. Сейчас мы видим первые пять фотографий и подписи к ним. Ниже находятся следующие пять фотографий, и там мы увидим подписи. Передвиньте вниз ползунок.

5.6.7. Вставка фотографии в коллекцию фотографий с помощью сканера

На странице мы сейчас видим две общие границы вверху и слева. В верхней границе имеется объявление, точнее место для него и место для панели ссылок. Для того, что бы можно было добавить объявление в автоматическом режиме нужно страницу поместить в структуру переходов. Для этого нужно открыть веб-узел, открыть Список папок. На вкладке Веб-узел откройте представление Переходы и из области Список папок перетащим в область Переходов нужную страницу. Это страница ***collection***, которая сейчас открыта. Пусть она будет связана со страницей Организации. Переименуем новую страницу. Для этого щёлкните в названии правой кнопкой мыши и выберите команду Переименовать в контекстном меню. Введите сюда текст.

Например, «коллекция фотографий». Нажмите клавишу ***Enter***, для подтверждения изменений названия. При этом имя файла никак не изменилось — ***collection***. Раскройте файл ***collection***, щёлкнув его вкладку и название, т. е. объявление на этой странице и панели ссылок в общих границах появились на данной странице. Однако файл ***Links***, страница Организации никак на панели ссылок не представлена. Для того, что бы он был

представлен, одну из панелей ссылок, которые мы сейчас видим, можно изменить.

Например, левую. Щёлкните её дважды левой кнопкой мыши. Появилось диалоговое окно Свойства панелей ссылок. На вкладке Общие для этой страницы выбран переключатель дочерней страницы домашней. Т. е. на панели ссылок отображаются страницы, которые являются подчинёнными по отношению к файлу ***index***. Кроме того, отображается домашняя и родительская страница. Изменим переключатель. Щёлкните например переключатель Тот же уровень.

При его выборе мы должны будем видеть ссылки на страницы находящиеся на том же уровне что и эта страница, а так же ссылку на домашнюю страницу и на родительскую, т. е. находящуюся на один уровень вверх. Нажмите на кнопку ОК. Панель ссылок слева изменилась. Сохраните изменения. Щёлкните гиперссылку Вверх при нажатой клавише ***Ctrl***. Открылась страница ***Links***. Именно она и является родительской по отношению к странице ***collection***. Настройка панели ссылок слева, как

Вы помните, это то же уровень. Для страницы Организации на том же уровне находятся такие файлы как Специфика, Почта, Термины и т. д. Но, файл с коллекции фотографии сюда не входит и мы не можем перейти к нему по ссылке. По этому следует опять изменить панель ссылок и сделать её не на том же уровне, а например на дочернем. Посмотрим, что у нас получится . Нажмите на кнопку ОК.

Страница изменилась, благодаря тому, что панель ссылок теперь выглядит иначе. Нажмите на кнопку Сохранить и вернитесь к странице ***collection***. Панель ссылок слева поменялась на всех страницах веб-узла. Щёлкните её дважды. Сейчас здесь выбран не переключатель тот же уровень, а дочерний уровень. Нажмите кнопку Отмена. Из коллекции фотографий мы можем перейти на один уровень вверх. Щёлкните ссылку Вверх при нажатой клавише ***Ctrl***. Открывается файл ***Links***. С этого файла мы можем вернуться обратно в коллекцию фотографий, щёлкнув ссылку, Коллекция фотографий. Кроме того с этой страницы с помощью левой панели ссылок мы можем перейти сразу на домашнюю страницу. Эта страница открылась, она имеет имя ***index***.

И со страницы ***collection*** мы можем переходить на все дочерние страницы домашней, с помощью верхней панели ссылок. Каждый эскиз фотографии представленный на странице является гиперссылкой на полное изображение. Перейдите в режим просмотр. Наведите указатель мыши на первое изображение. Указатель превратился в изображение руки. Щелкните фотографию. Мы по гиперссылке перешли к полному изображению.

Оно на столько большое, что для его просмотра требуется воспользоваться полосами прокрутки Нажмите на кнопку Обновить на панели инструментов.

Щёлкните следующую фотографию. Эта фотография несколько меньше и её можно просматривать целиком. Т. о. пользователь сможет просматривать любую фотографию, которая ему нужна. Уже созданную фотогаллерею можно изменять. Щёлкните в её области дважды.

Открылось окно Свойства коллекции фотографий. Фото в галерею можно добавлять не только с жесткого диска, но и со сканера или камеры. Нажмите на кнопку Добавить в этом окне. Появилось меню состоящее из двух пунктов: Рисунки из файлов и Рисунки со сканера или камеры. Щёлкните второй пункт. На экран выведено диалоговое окно Вставка рисунка со сканера или камеры. Выберите устройство в списке. Разрешение для передачи по Интернету или для Печати. Оставим для передачи по Интернету.

Кроме того, здесь же можно установить флажок Добавить рисунки в коллекцию картинок. Я установлю его. Что бы добавить рисунок в его исходной форме нужно нажать на кнопку Вставить в данном диалоговом окне, а что бы настроить его свойства — кнопку Специальная вставка, нажмите её. После нажатия кнопки появилось диалоговое окно Сканирование с помощью и далее указано название сканера. В окне расположено 4 переключателя: Цветной снимок; Черно-белое изображение; Черно-белый рисунок или текст; и Особые параметры. Оставьте выбранным переключатель Цветной снимок. Внизу находится гиперссылка Настроить качество сканированного изображения щёлкните эту ссылку. Появилось диалоговое окно Дополнительные свойства. В группе изображения можно настроить яркость и контрастность перемещением ползунка. Переместим, например, ползунок Яркость вправо. Изображение слева становится более ярким. Переместите так же правее ползунок Контрастность.

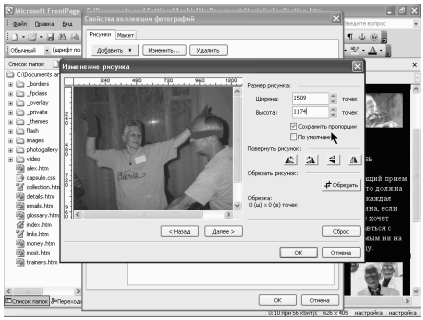
Это же изображение стало более контрастным. Ниже можно настроить разрешение сканирования. Сейчас установлено 300 точек на дюйм. Вы можете установить другое разрешение. Например, 200. Цифры можно вводить как с клавиатуры так и с помощью кнопок на счетчике. Если нажимать кнопку указывающую вверх, то число увеличивается. Если нажимать стрелку вниз, то уменьшается. Установим разрешение 300 точек. Справа имеется список Тип изображения, раскройте его. Здесь такие варианты как: Цветной снимок; Черно-белое изображение, оттенки серого; и Черно-белый рисунок или текст. Оставляем выбранным цветной снимок, а так же имеется кнопка Сброс. Нажмите на неё.

В этом случае все настройки остаются такими, какими определены по умолчанию. В частности ползунки Яркость и Контрастность снова установились на значение «0». Не будем здесь ничего менять, нажмите на кнопку Отмена, в окне Дополнительные свойства.

5.6.8. Изменение рисунка в коллекции фотографий

Прежде чем сканировать изображение, можно сделать предварительный просмотр его. Для этого нажмите на кнопку Просмотр в окне Сканирование с помощью. Начинается процесс работы сканера. Сейчас ожидание ответа сканера. Начинается предварительный просмотр изображения. Программа нашла фотографию в сканере и отметила её рамкой. Стороны этой рамки мы можем перемещать, для уменьшения или увеличения области захвата. Можно например уменьшить изображение, перетягивая стороны рамки таким образом или воспользоваться маркерами размера находящимися по углам этой рамки. После того, как Вы определили размер, можно начать сканирование. Нажмите на кнопку Сканировать. Сканирование началось. По окончании этого процесса диалоговое окно Сканирование с помощью исчезает. В списке изображений на вкладке Рисунки окна Свойства коллекции фотографий, появилось новое изображение. Эскиз его мы видим справа.

Каким будет размер самого изображения, мы пока не знаем. Видим только размеры эскиза, ещё правее, ширина 100 точек, а высота 77. Для того, что бы



увидеть, какой будет реальный размер фотографии, после того как мы перейдём по гиперссылке от эскиза, нажмите кнопку Изменить. На экран выведено диалоговое окно Изменение рисунка. Размер самого рисунка, а не эскиза будет 1509 точек по ширине и 1174 точки по высоте. Эти размеры можно изменять. Я бы даже сказал нужно изменять, потому что размер слишком большой. На обычный монитор такое изображение может не поместится. Установим, например по ширине значение 400 точек. Для этого удаляя имеющееся значение и вводим новое. Щёлкните поле Высота.

Программа автоматически преобразовала высоту. Теперь она равна 311 точкам. Если Вам показалось, что такое изображение будет слишком маленьким, можно увеличить его с помощью кнопок на счетчиках. Нажмите, например, верхнюю кнопку на счетчике Высота. Если её удерживать, фотография изменяется постепенно. Оставим такой размер 546 на 425 точек. Ниже находится группа опций Повернуть рисунок. Кнопки находящиеся здесь определяют поворот и отражение объекта. Щёлкните первую кнопку слева. Изображение повернулось на 90 градусов против часовой стрелки. Щёлкните следующую кнопку. Мы повернули изображение тем самым на 90 градусов по часовой стрелке. Следующая кнопка позволяет перевернуть изображение.

Щёлкните её ещё раз. Мы перевернули изображение снова. Теперь оно в нормальном виде. И последняя кнопка, фотография как бы отразилась в зеркале. Щёлкните её ещё раз. Произошёл обратный процесс. Ниже Вы видите группу Обрезать рисунок. Нажмите на кнопку Обрезать. На изображении появилась рамка с маркерами размера. Наведите указатель мыши на неё. Когда указатель оказывается над каким то маркером, он превращается в двунаправленную стрелку. Перетащите например этот маркер левее. Наведите указатель на центральный нижний маркер. Перетащите его вниз. Тоже самое сделайте с правым центральным маркером. После наведения на него указателя мыши, перетащите его в право. Для того, что бы обрезать лишние области, находящимися за пределами этой рамки, ещё раз нажмите на кнопку Обрезать.

В этом случае мы уменьшили рисунок, не с помощью перетаскивания его границ, а обрезав лишнюю часть. Если сделанные изменения Вас не устраивают можно нажать на кнопку Сброс. Я этого делать не буду. Для перехода к другим фотографиям воспользуйтесь кнопками Назад и Далее. Я нажму на кнопку Назад. Здесь хорошо видно, что размер рисунка, который сейчас на экране, слишком мал — 84 на 70 точек. Кроме того качество его меня не устраивает. Поэтому нужно будет заменить это изображение на странице. Для перехода к следующей фотографии можно нажать на кнопку Далее.

Для подтверждения всех изменений нажмите на кнопку ОК в окне Изменение рисунка. Если какое-либо изображение Вам не нужно, можете его выделить в списке и нажать на кнопку Удалить. Но, я этого делать не буду. Что нужно сделать действительно, так это внести подпись и описание под новым изображением. Подпись будет следующей, добавим так же описание. Выполнив все настройки нажмите ОК в диалоговом окне Свойства коллекции фотографий. В окне документа ***collection*** щёлкните на полосе прокрутки.

Новая фотография и подпись к ней оказались в конце списка. Так же как они были в конце списка в диалоговом окне. Страница претерпела изменения. О чём свидетельствует звёздочка на вкладке. Нужно сохранить эту страницу. Нажмите на кнопку Сохранить. Программа вывела на экран Сохранение внедрённых файлов. В этом диалоговом окне нужно определить имя и папку и действия. Переименуем файл. Пусть он называется, например, ***lady***. Щёлкните за пределами рамки. Папка, в которую будет сохранён файл пусть остаётся без изменений — ***images***. Нажмите на кнопку ОК в диалоговом окне. Проверим, как выглядит страница веб-обозревателя. Раскройте меню файл, просмотреть в обозревателе. Выберите первый пункт в списке.

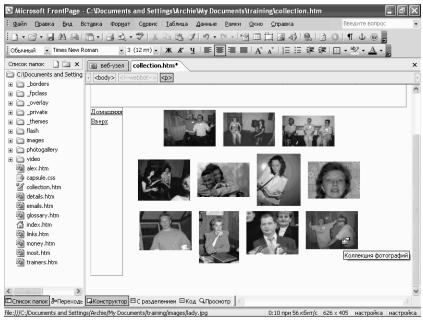
Страница Коллекция фотографий открыта. Мы можем её просматривать с помощью полосы прокрутки и переходить по гиперссылкам. В качестве которых служат эскизы. Щёлкните, например, 3-е изображение. Как видите

оно само то же маленькое. Поэтому нужно будет исправить недостаток и увеличить изображение. Щёлкните кнопку Назад. Выберите первую фотографию. Она достаточно большая, здесь всё нормально. Закройте окно программы ***Microsoft Internet Explorer***.

5.6.9. Макеты коллекции фотографий

Общий вид коллекции фотографий на веб-странице можно изменить как при создании такой коллекции, так и после того, как она сформирована. Щёлкните в области коллекции фотографий дважды левой кнопкой мыши. На экран выведено окно Свойства коллекции фотографий. С вкладкой Рисунки в этом окне мы уже познакомились. Перейдите на вкладку Макет щелчком мыши. Макет, которым мы всё время работали, называется Горизонтальная компоновка. Он сейчас выбран в списке Выберите макет. Образец этого макета находится справа.

Под образцом указано как создаётся и формируется макет. В частности здесь указано, что эскизы изображений создаются автоматически, упорядочиваются строками и снизу каждого эскиза указывается описание. Внизу имеется список Число рисунков в строке. Раскройте его. Сейчас указано значение пять. 5 фотографий на странице по горизонтали. Мы можем это значение изменить. Установите, например, значение 3. В образце никаких изменений не произошло. Посмотрим, будут ли изменения на веб-странице. Нажмите на кнопку ОК в диалоговом окне.



Щёлкните в том месте на странице, где указатель превращается в курсор(вертикальную черту). Т. о. мы сняли выделение области в которой изображены фотографии. Щёлкните на полосе прокрутки. Т.о. мы можем просматривать фотографии и видим, что расположено по 3-и фото в ряд. В последнем ряду две, одной не хватило просто. Выделите область фотографий двойным щелчком мыши, ещё раз. Мы снова видим диалоговое окно Свойства коллекции фотографий. Щёлкните вкладку Макет. В списке выберите макет. Щёлкните вторую строку Коллаж. В данном случае эскизы образуют узоры и эскизы изображений создаются автоматически. Обратите внимание, что список Число рисунков в строке теперь не доступен. Программа сама определяет сколько изображений нужно в строке. Нажмите кнопку ОК в диалоговом окне. Щёлкните за пределами области в которой находятся фотографии. Выделения мы сняли. Передвиньте

вниз ползунок. Ни подписей ни описаний мы не видим. Наведите указатель мыши на правое нижнее изображение. Всплывающая подсказка говорит только о том, что это коллекция фотографий. Перейдите в режим просмотра.

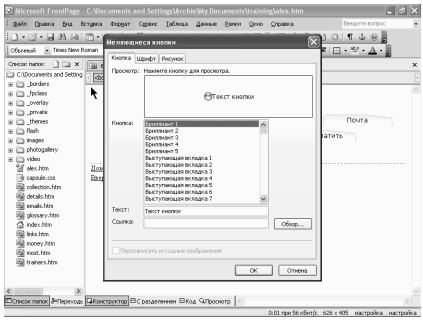
Переместите, ползунок вниз и снова наведите указатель мыши на правое нижнее изображение. На этот раз мы видим подпись: «Новый партнёр или партнёрша». Такая подпись стояла под этой фотографией. Наведите на фотографию, которая находится выше. Здесь другая подпись Портрет. Описание при этом не выводится. Следующая, это телефонный разговор. Вернитесь в представление Конструктор. Щёлкните дважды левой кнопкой мыши в области фотографий.

Окно свойства коллекции фотографий открыто. Щёлкните вкладку Макет. В списке щёлкните макет, выберите 3-ю строку, показ слайдов. В этом режиме число рисунков в строке мы так же определять не можем. Нажмите на кнопку ОК и рассмотрим, как отображаются фотографии в этом случае. Перетащите вниз ползунок. Что бы лучше было видно окно, закройте область Список папок. Щёлкните кнопку Просмотр внизу. Для того, что бы отображать крупным планом фотографии, нужно щёлкать их эскизы сверху. Щёлкните, например, эскиз Обсуждение. Фотография отобразилась внизу. При этом видно подпись и описание данной фотографии. Переместимся к следующей фотографии, это фотография Круг. Здесь так же мы видим и подпись и описание. Т.о. можно щёлкать все фотографии по очереди. Для того, что бы увидеть следующие изображения имеются стрелки справа и слева от ряда фотографий. Нажмите на правую. Т.о. мы можем перемещать ряд фотографий слева на право или наоборот. Щёлкните левую кнопку. Вернитесь в представление Конструктор.

Для открытия макета коллекций фотографий дважды щёлкните в области фотографий. Диалоговое окно открыто, щёлкните вкладку Макет. Последний вариант в списке Выберите макет — это вертикальная компоновка. В таком случае расписание размещается справа от каждого эскиза. Число рисунков в строке можно определять. Сейчас здесь установлено значение 3, пусть так и остаётся. Нажмите на кнопку ОК.

Мы изменили компоновку. Просмотрим, как это будет выглядеть. Из всех возможных вариантов выберите тот, который Вас больше всего устраивает. Мне кажется больше подходит первый в данном случае. Щёлкните дважды в области фотографий. На вкладке макет выберем первый вариант — горизонтальная компоновка. Число рисунков в строке пусть остаётся равным пяти. Нажмите на кнопку ОК. После того, как вариант компоновки выбран, сохраните веб-страницу.

5.6.10. Вставка меняющейся (интерактивной) кнопки



Для связи между страницами веб-узла можно использовать интерактивные кнопки. Меняющаяся кнопка позволяет связать одну страницу с другой с помощью гиперссылки. Откройте необходимый документ, например, файл ***alex***. Установите курсор там, где должна быть кнопка. Я щёлкну после слова «информация» и создам новую строку нажатием клавиши ***Enter*** на клавиатуре. Новая строка создана. Для вставки кнопки откройте меню Вставка. Выберите команду меняющаяся кнопка. На экран выведено диалоговое окно Меняющиеся кнопки. Это окно включает в себя три вкладки: Кнопка; Шрифт; и Рисунок. Щёлкните вкладку Кнопка. Вкладка Кнопка включает несколько полей, первая из них Просмотр, здесь находится образец кнопки и текста. Наведите указатель мыши на кнопку.

При наведении мыши она изменяет свой цвет. Если нажать левую кнопку мыши после наведения указателя — цвет будет иным. Т.о. мы видим все те изменения какие должны происходить на веб-странице с этой кнопкой. Ниже находится список Кнопка, в котором мы можем выбрать представления меняющейся кнопки. Выберите, например, Выступающая вкладка 1. Соответственно в образце представление кнопки изменилось. Для просмотра других вариантов кнопок щёлкните на полосе прокрутки. Выберите, например, такую кнопку как Граница сверху и снизу 1 или Граница сверху и снизу 2. Для просмотра различных вариантов можно использовать клавиши стрелка вверх и вниз на клавиатуре. Нажмите стрелку вниз.

Перейдите к следующей строке и к следующей и т. д. Для более быстрого перемещения щелкайте на полосе прокрутки. Выберите, например такую строку как Металлическая вкладка 1. Наведите указатель мыши на образец. Цвет этой кнопки изменился. Теперь наведите указатель и нажмите левую кнопку мыши, мы получили ещё один оттенок цвета на кнопке. Посмотрим, какие ещё есть варианты кнопок. Мягкая вкладка1 и т. д., меняются цифры рядом с этим вариантом. Мягкий прямоугольник 1, цифры могут быть другими. В зависимости от цифры меняется и представление и кнопки, её цвет в частности. Простая линия 1, 2 и т. д. Простой блок; Простой текст.

Варианты можно подбирать до тех пор пока Вы не получите тот, что Вам нужен. Щёлкните, например, строку Стеклянная капсула 5. Кнопка будет серого цвета. При наведении указателя мыши этот цвет становится более

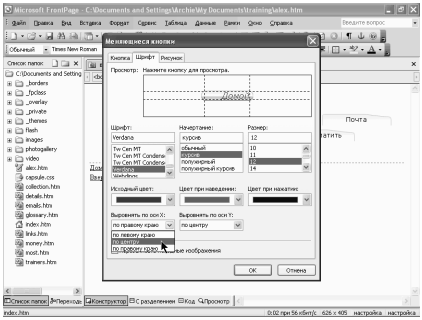
интенсивным, при нажатии ещё более интенсивным. Пусть такой вариант пока и останется. Следующее поле Текст, щёлкните в нём. Удалите слова «текст кнопки» и введите сюда тот текст, который должен быть на кнопке. Например, я введу слово «Домой». Следующее поле ссылка. Справа от неё находится кнопка Обзор, нажмите на эту кнопку.

На экран выведено диалоговое окно Изменение гиперссылки. Текст, который мы ввели, уже находится в текстовом поле. Это слово «Домой». Ниже находится папка. Это папка ***training***, т. е. тот веб-узел в котором мы сейчас работаем. Кнопка должна приводить к открытию страницы ***index***(это домашняя страница). Выберите этот файл в списке. Введём для него подсказку. Впишите текст подсказки «Главная страница». В диалоговом окне Подсказка для гиперссылки в которое мы ввели этот текст нажмите на кнопку ОК. В окне изменение гиперссылки тоже кнопку ОК. Ссылка введена. Мы также могли ввести её в ручную. Просто набрав символы с клавиатуры.

Прежде чем рассматривать назначение двух других вкладок в этом окне, проверим работу кнопки. Нажмите на кнопку ОК в окне. Диалоговое окно закрыто. На странице появилась кнопка. Щёлкните представление просмотр. Наведите указатель мыши на кнопку. Указатель превратился в изображение руки. Появилась всплывающая подсказка, нажмите эту кнопку. В этом же окне открылась домашняя страница веб-узла. Нажмите кнопку Обновить на панели инструментов. Обратите внимание на то, что в тексте кнопки была допущена ошибка. Каким образом исправить эту ошибку и изменить другие параметры, Вы узнаете на следующем уроке.

5.6.11. Изменение свойств меняющейся кнопки

Для изменения текста на кнопке и других её свойств наведите указатель мыши на эту кнопку и щёлкните правой клавишей мыши. Появилось контекстное меню, в котором выберите команду Свойства кнопки. Диалоговое окно Меняющейся кнопки открыто. Щёлкните в поле Текст и измените этот текст. Удалите в частности букву «и» и введите букву «м». Текст мы исправили. Теперь изменим цвет этой кнопки. Щёлкните строку Стеклянная капсула 1 в списке Кнопка. Каким будет цвет видно в образце. Перейдём на вкладку Шрифт. На данной вкладке так же имеется окно Просмотр. Для просмотра кнопки надо навести на неё указатель мыши и мы видим тот цвет, который будет при наведении указателя. Можно щёлкнуть эту кнопку. И опять мы видим тот цвет, который применяется для щелчка. Кроме того на вкладке Шрифт имеются такие окна как Шрифт (здесь указывается имя шрифта), Начертание, Размер шрифта, Исходный цвет.



Цвет при наведении и Цвет при нажатии. А так же списки Выровнять по оси ***X*** и Выровнять по оси ***Y***. Начнём с названия шрифта. Предположим вместо названия шрифта ***Tahoma***, который сейчас выбран, щёлкнем шрифт ***Times New Roman***. На образце сразу же отображаются изменения. Выберем другой шрифт. Например, такой шрифт как ***Arial***. Вместо обычного начертания применим курсив. А вместо 10 размера, который сейчас указан, 12-й. Все изменения видны в образце. Если шрифт не нравится, можно опять его изменить. Выберем, например такой шрифт как ***Verdana***. Раскройте список Исходный цвет. Для этого нужно нажать кнопку со стрелкой.

Появился список Стандартных цветов, других цветов, которые уже использовались. Кроме того можно щёлкнуть ссылку Другие цвета внизу. В этом случае появляется знакомое Вам диалоговое окно Другие цвета. Нажмите на кнопку Отмена. Ещё раз раскройте список Исходный цвет. Щёлкните, к примеру, Красный. Текст на кнопке сразу же изменился в цвете, он стал красным, естественно.

Но, только когда указатель мыши не подведён к этой кнопке. Если навести указатель, то цвет текста становится таким, каким и был раньше. Для того, что бы изменить этот параметр, раскройте список При наведении. Выберем, например, зелёный цвет. Тёмно-зелёный. Наведите теперь указатель мыши на образец. При наведении текст становится зелёного цвета. Нажмите на кнопку. Текст опять стал тёмным, как и ранее. Что бы изменить и данный параметр, раскройте список Цвет при нажатии. Выберите нужный цвет.

Пусть это будет, например, синий, тёмно-синий. Проведём ещё один эксперимент. Наведите указатель мыши на кнопку. Она стала зелёной, теперь нажмите и держите левую кнопку мыши. Текст стал синим. Отпустите. Текст опять зелёный и уберём указатель. Текст красный. Разберёмся со списками. Выровнять по оси. Начнём с оси ***X***. Раскройте список. Сейчас выбрано значение По центру. Щёлкните По левому краю, например. На кнопке текст выровнен по левому краю, т. е. он сместился левее. Раскройте список ещё раз. Щёлкните значение по правому краю. Соответственно текст сдвинулся вправо. Выберите значение По центру. Теперь можно выровнять по оси ***Y***. Сейчас он выровнен по центру, попробуем выровнять по нижнему краю. Текст сместился вниз на кнопке, но не выходит за её пределы. Щёлкните строку По центру. Оставим текст в таком виде. Щёлкните вкладку Рисунок.

На этой вкладке имеется окно для просмотра кнопки. При наведении изменяется эта кнопка. При нажатии тоже изменяется и текст и цвет кнопки. Если текст очень короткий, то можно уменьшить размеры кнопки по ширине. Сейчас указано значение 100 точек. Уменьшим, например до 80. А высоту наоборот увеличим. Используем для этого верхнюю кнопку на счётчике Высота. При этом мы опять получили ширину более 100 точек даже. Дело в том, что был установлен флажок Сохранить пропорции. Снимите его. Теперь удалите лишние цифры в ширине. Вставим цифру 8 и 0. Увеличим высоту. Можно ввести её в ручную.

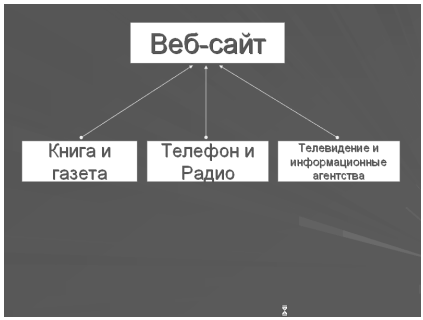
Вместо 21 точки ведём, например, 40. Получилась кнопка следующей величины. Ниже устанавливаются флажки. Такие как Создать изображение для выделенного состояния, Создать изображение для нажатого состояния и Предварительно загрузить изображение кнопок. Все эти флажки следует оставить. Ниже имеется два переключателя. Использовать для кнопки формат ***JPEG*** и фоновый цвет, этот переключатель сейчас выбран. И второй Использовать для кнопки формат ***GIF*** и прозрачный фон. Раскройте список первого переключателя. Здесь можно выбрать вполне конкретный цвет фона. Выберем, например, голубой.

В диалоговом окне Меняющиеся кнопки нажмите на кнопку ОК. Снимите выделение кнопки щелчком мыши. Кнопка занимает прямоугольное пространство. Однако там, где углы её скруглены, виден фон. Если бы фон всей страницы был голубой, тогда мы бы оставили такой же фон и для кнопки. Щёлкните дважды эту кнопку. Изменим фон. Для этого нужно щёлкнуть вкладку Рисунок. Если Вы не знаете, какой фон будет у страницы пока, то можно использовать второй переключатель. Использовать для кнопки формат ***GIF*** и прозрачный фон. Нажмите кнопку ОК в окне Меняющиеся кнопки. Фон кнопки в данном случае прозрачный и мы видим просто белый цвет. Попробуем изменить фон самой страницы. Для этого щёлкните её правой кнопкой мыши. Выберите команду Свойства страницы. В диалоговом окне Свойства страницы перейдите на вкладку форматирование.

Раскройте список фон в группе цвета. Выберем, например, голубой. Подтвердите изменения. Теперь мы видим, что вся страница имеет голубой фон. Тем не менее вокруг кнопки нет какой-то особенной прямоугольной рамки. Изменим ещё раз фон страницы, для эксперимента. Перейдём на вкладку форматирование в окне свойства страницы. Вместо голубого фона, выберите, например, красный. Нажмите на кнопку ОК. И в этом случае вокруг кнопки нет прямоугольной рамки, которая выдавала бы то, что здесь находятся очертания самой кнопки. Прозрачный фон позволяет отображать здесь красный цвет. Опять изменим свойства страницы. В диалоговом окне на вкладке Форматирование выберите фон Авто и подтвердите эти изменения.

Раздел 6. Web-сайты. Типы и разработка

6.1. Типы сайтов, назначение, цель, целевая аудитория



Тема этой лекции типы ***Web*** сайтов, их назначение и целевая аудитория. В принципе, ***Web*** сайт, это своего рода средство информации или медиа. Сайт, это своеобразный электронный журнал, открыв который, посетитель просматривает страницы в случайном порядке и останавливается на необходимой информации. Можно сказать, что ***Web*** сайт вобрал в себе качество таких привычных для нас носителей информации, как книга и газета. С газетой ***Web*** сайт роднит высокая степень оперативности обновления и сама тематика распространения горячей, новой и необходимой информации в кратчайшие строки. Также нужно отметить аналогию с телефоном и радио, заключающуюся в том, что Интернет предоставляет возможность общения, обмена мнениями и общественного обсуждения разных вопросов. Также ***Web*** сайт обладает качествами телевидения и информационных агентств, которые позволяют не только читать и видеть информацию, но и слышать и видеть картинку в виде изображения. И хотя передача видео через

Интернет продолжает оставаться достаточно редким явлением по причине технических ограничений, очевидно, в ближайшие годы телевизионная картинка в Интернете будет появляться все чаще. Интернет включает в себя огромное количество сайтов, удовлетворяющих практически любой запрос или интерес посетителей. По характеру размещенной информаций тематике или целевой аудитории, я разделил ***Web*** сайт на следующие условные группы.

Это коммерческие сайт, информационные сайт, развлекательные сайты, навигационные сайты, персональные сайты и комбинированные сайты. Теперь приступим к обсуждению каждой этой группы в отдельности. Начнем с коммерческих сайтов. Такие сайты еще называют бизнес сайтами. Они предназначены для поддержки бизнеса какой-либо фирмы или организации. Сайт используется для распространения информации о продуктах и услугах, предоставляемой организации и сведениях о способах связи с фирмой. Основная цель любого коммерческого сайта состоит в обслуживании

пользователей таким способом, который приносит компании выгоду, либо непосредственно, либо косвенно.

Целевая аудитория. Основной аудиторией такого сайта являются потенциальные и настоящие клиенты фирмы или организации. Дополнительная аудитория часто включает потенциальных и действительных инвесторов, заинтересованной организации. Такие как средства информации и фирмы конкуренты. Перейдем к обсуждению информационных сайтов. Сайты, ставящие перед собой задачу информирования пользователей в определенных сферах общественной жизни, правительственные, образовательные сайты, сайты новостей и сайты некоммерческих фирм, организаций, религиозных групп, а также различные общественные сайты. Посетители информационных сайтов, это люди, интересующиеся конкретным вопросом или просто узнающие новую информацию. Перейдем к обсуждению развлекательных сайтов. Развлекательные сайты ставят целью развлечь своих посетителей. Как правило, продаваемым товаром является само развлечение.

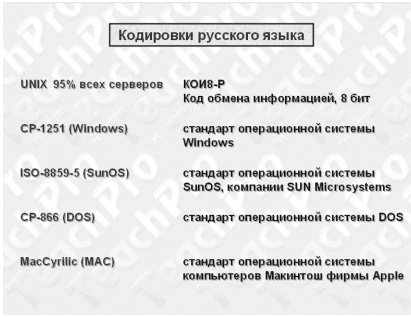
Целевая аудитория развлекательных сайтов, это менее деловая или занятая часть аудитории. Чаще это подростки или дети. Обсудим навигационные сайты. Навигационным сайтом является тот сайт, который помогает пользователем в поиске нужной информации через Интернет. Такие сайты часто называют Интернет порталами. Посетители навигационных сайтов, это чаще новички, разыскивающие ресурсы по интересующей их тематике. Обсудим персональные сайты. Их часто называют персональной домашней страничкой, и являются самовыражением его создателя.

Посетители персональных сайтов, это обычно более узкий круг друзей, знакомых, или просто расширяющих круг знакомств. И, наконец, комбинированные сайты. Они сочетают в себе два или более типов сайтов и их достоинства. Например, коммерческий Интернет портал. Или информационно-коммерческий сайт, и тому подобное. Аудитория комбинированных сайтов, это практически, весь контингент пользователей Интернета. Группы, на которые я разделил опубликованные в Интернете ***Web*** сайты, я повторяю, являются чисто условными, но охватывают практически весь смысловой диапазон существующих сайтов

6.2. Кодировка русского языка. Мировая паутина. Браузеры

На этом уроке мы с Вами поговорим о проблеМАХ русского языка. У всех русскоязычных пользователей Интернета извечно существовала головная боль с передачей русского алфавита. Дело в том, что по мировой компьютерной сети самым наилучшим образом передается семи битный код, который полностью покрывает все потребности английского языка. Но, вот для

передачи русского алфавита требуется восьми битный код. И самое грустное даже не в том, что многие, особенно американские серверы Интернета, не настроены на передачу восьми битного кода.



В результате чего происходит потеря этого самого восьмого бита, и вместо содержательного русского текста Вы видите на своем экране аббракодабру. Это еще пол беды. Ситуация ухудшается тем, что не существует единого стандарта для передачи русского текста в Интернете. В результате, русский текст, пропутешествовав по десятку другому серверов, столько же раз перекодируется из одной кодировки в другую. В конце концов, он превращается в такую же аббракодабру. Какие же существуют кодировки русского языка и как ими пользоваться? Изначально весь Интернет строился на серверах, работающих под различными разновидностями операционной системы ***Unix***. Да и сегодня процентов 95 всего Интернета — это все тот же мощный и сложный ***Unix***.

Поэтому, не удивительно, что первоначальным стандартом русской кодировки в Интернете явилась кодировка КОИ8-Р, принятая на русских машинах. КОИ8-Р расшифровывается, как Код Обмена Информацией (8 бит). Однако, сегодня, помимо КОИ8-Р, есть еще четыре русских кодировки. Давайте, перечислим их. Первая из них — ***CP***1251. Этот стандарт принят в операционных системах ***Windows***, которые, как известно, применяются подавляющим большинством конечных пользователей Интернета.

Следующая кодировка — ***ISO***- 8859- 5. Такая кодировка используется на компьютерах с операционной системой ***SunOS*** компании ***Sun MicroSystems***. Следующая кодировка — ***CP***-866. Это — кодировка русского языка операционной системы ***DOS***. Наконец, ***MacCyrillic*** — это кириллическая кодировка всех поклонников ***Macintosh***-а и компании ***Apple***. Можете теперь представить себе, какая путаница возникает со всеми этими кодировками. Тем не менее, не все так страшно.

Многие программы для работы в Интернете очень неплохо справляются с задачей перекодирования одного стандарта в другой. В подавляющем большинстве случаев приходится перекодировать КОИ8-Р в ***CP***1251 и обратно. Т.е. веб-браузеры, которые мы будем изучать в нашем курсе, делают это вполне прилично, причем, во всех аспектах Интернета: в мировой паутине, электронной почте и конференциях новостей. Например, отправляя

электронное письмо с помощью какого-либо ***Mailer***-а, Вы будете печатать в кодировке ***CP***1251, что естественно для операционной системы ***Windows***. Перед тем, как отправить письмо в сеть, сами почтовые клиенты автоматически перекодируют Ваш текст в КОИ8-Р и далее безболезненно отправят его на сервер провайдера, где, как Вы догадались, царит мир ***Unix***-ов.

Хуже обстоит дело с почтой, которая приходит к Вам из Интернета. Ведь отправители письма могут и не пользоваться ***Windows***, а, скажем, работать, по старинке, под ***DOS***. Бог его знает, по каким закоулкам сети пропутешествует его письмо на пути к Вашему компьютеру. Поэтому, велика вероятность, что Вы, все-таки, получите письмо в нечитаемом виде. Что мы можем посоветовать в этом случае? Одним выходом из такой ситуации является, например, приобретение программы ***Mail Reader***, которую создала российская компания "Агама".

Это очень полезная программка, которая может разобрать более восьми тысяч комбинаций, возникающих из последовательной обработки в сети письма самыми различными кодировками. Все, что Вам нужно сделать, так это, скопировать неразборчивое письмо в программу ***Mail Reader*** и запустить определение кодировки. Программа попытается применить все известные ей схемы и расшифровать сообщение. Если же и ***Mail Reader*** не поможет, то тут уж ничего не поделаешь: Вашему приятелю придется переслать письмо по второму разу. Мировая паутина. Мы будем подробно рассказывать о гипертекстовом протоколе ***HTTP*** и мировой паутине ***World Wide Web***-е, когда мы будем обучаться работе с различными браузерами.

А пока скажем несколько общих слов о компьютерных программах, которые предназначены для работы с ***Web***-ом. Эти программы называются браузерами (от английского слова ***to browse***, что означает блуждать). Существует десятка два браузеров. Однако бесспорными лидерами являются две программы: ***Netscape Navigator*** (который будем подробно изучать) и ***Microsoft Internet Explorer***. Имеются еще и такие браузеры, как ***Opera*** и другие. Мы с Вами подробно рассмотрим и браузер ***Opera***.

Но, надо отметить, что приблизительно более 90 процентов пользователей сети применяют один из браузеров ***Netscape Navigator*** или ***Microsoft Internet Explorer***. По большому счету, Вы не ощутите большой разницы при работе в различных веб-браузерах. На этом, давайте, завершим наше вводное занятие и приступим к изучению различных веб-браузеров, то есть, ***Netscape Navigator***-а, ***Microsoft Internet Explorer***-а и ***Opera***.

6.3. Обозреватели Интернет

6.3.1. Знакомство с браузерами

6.3.1.1. Что такое обозреватели?

Для того чтобы начать путешествие по веб нам необходим браузер. Браузер — это программа, которая находит нужные веб-страницы и отображает их на экране. К счастью, так как на нашем компьютере установлена операционная система ***Windows***, то, вероятнее всего, браузер у нас уже есть. Если же — нет, то его можно скачать с Интернета и установить дополнительно. На сегодняшний день распространены три вида браузера. Первый — это ***Internet Explorer***. Он обычно поставляется вместе с ***Windows***, начиная с версии ***Windows*** 98.

Давайте, попробуем его запустить. Вот — у нас обозреватель ***Internet Explorer***. Сейчас он пытается загрузить страницу, которая установлена в нем по умолчанию. А по умолчанию установлена главная страница ***MSN***. Вот это — браузер. Здесь у него имеется меню, имеется еще вот такая Панель навигации и строка состояния. Это — ***Internet Explorer***. Здесь в настоящий момент присутствует ***Internet Explorer*** шестой версии, то есть, мы работаем с ***Internet Explorer*** версии 6.0. Все предыдущие версии ***Internet Explorer*** выглядят достаточно, похоже. Правда, имеются внутренние отличия. Следующий обозреватель, то есть браузер — это... Иногда на русском языке говорят — обозреватель, мы можем употреблять оба этих слова: и обозреватель, и браузер. Следующий, браузер, который мы с Вами хотим рассмотреть — это ***Netscape Navigator***.

Давайте, запустим его (мы его заранее установили). Скачивать и устанавливать браузер ***Netscape Navigator*** мы с Вами изучим позже. Итак, перед нами ***Netscape*.*Navigator***. Запустим его. Он пытается открыть страницу, которая также в нем установлена по умолчанию, то есть, он должен открывать эту страницу сразу после запуска. Это — страница компании ***Netscape***. Сейчас она будет загружена полностью. На сегодняшний день доступна седьмая версия ***Netscape Navigator***. Правда, к сожалению, она не локализована, то есть, она существует на английском языке, на русском языке ее пока нет. Но это — не беда: мы с Вами научимся управлять ею на английском языке. После этого Вам будет легче работать с русской ее версией.

Сейчас нам сообщают, что произошло разъединение от нашего провайдера. Иначе говоря, нам нужно дозвониться до него заново. Совершим повторный звонок. Как Вы знаете, для того, чтобы иметь возможность входить в Интернет, нам надо дозваниваться до нашего провайдера. Это необходимо для осуществления подключения к Интернету.

Мы с Вами настраивали такое подключение на предыдущих уроках. А вот сейчас подключение уже произошло. Давайте, нажмем здесь ОК, перейдем на наш ***Netscape*** и с помощью нажатия вот на эту кнопку мы начнем загрузку данной страницы заново. Как Вы помните, загрузка у нас прервалась по причине разъединения связи с провайдером, и нам пришлось дозваниваться до него заново. Теперь с помощью этой кнопки мы сможем продолжить нашу работу.

Короче говоря, ***Netscape*** продолжает загружать страницу, которую он пытался загрузить до разъединения от линии. Надо сказать, что ***Netscape Navigator*** существует в виде нескольких разновидностей. Он существует для ***Windows***, для ***Macintosh***, а также для множества рабочих станций ***Unix***. ***Netscape***, поставляется, как часть большого комплекса, в который входит и ***Composer***. То есть, программа для составления своих веб-страниц, и почтовый клиент, и ***Messenger***.

Все это мы подробно изучим на последующих уроках курса. А сейчас мы с Вами просто путешествуем по веб-страницам. Главная страница фирмы ***Netscape***, наконец-то, загрузилась, и мы сейчас попробуем запустить следующий браузер, который также достаточно распространен. Этот браузер называется ***Opera***. Давайте, запустим его. Его мы также установили ранее. А как это делается, мы с Вами изучим впоследствии.

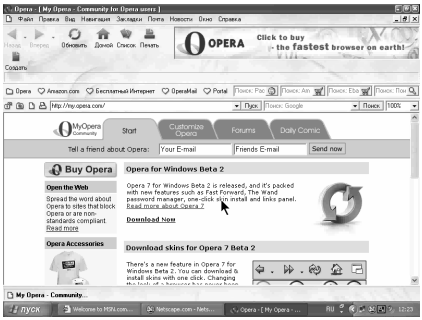
Запустим. ***Opera*** — это небольшой, но достаточно быстро работающий браузер. Он создан компанией из Норвегии. Существуют две его разновидности: бесплатный (в которой имеется небольшое количество рекламы) и платный. Платный стоит около сорока долларов и никакой рекламы не содержит. Сейчас нам ***Windows*** предложил использовать ***Opera***, как браузер по умолчанию для просмотра веб-страниц.

Иначе говоря, каждая веб-страница будет открыта именно браузером ***Opera***, если мы заранее не определили, каким браузером ее открывать. Это нам, в общем-то, не нужно, поэтому установим здесь флажок Больше не показывать и нажмем на кнопку Нет. То есть, мы не устанавливаем ***Opera***, как браузер по умолчанию. Полезные советы нас тоже не интересуют, и показывать их нам также не нужно. Сейчас перед нами открыта страница компании, которая создала ***Opera***. Нам сообщают, что уже существует его седьмая версия, правда, бета-версия. Бета — значит не до конца оттестированная и не до конца созданная, так называемая, промежуточная версия.

В данном случае у нас установлена версия 6.5. Это — окончательная версия. Вся панель навигации окончательно переведена. Здесь имеется небольшая реклама, именно та, о которой говорилось. Мы скачали бесплатную версию, и на ней имеется реклама. Сейчас, как Вы видите, перед нами три браузера: ***Internet Explorer***, ***Netscape Navigator*** и ***Opera***. Каждый пользователь подбирает браузер по своему вкусу. ***Opera*** — достаточно легкий браузер.

Правда, он имеет меньше возможностей. Если вы обратите внимание, здесь и количество меню меньше, и меньше возможностей, но зато он работает быстрее и легче, соответственно. То есть, на более слабых компьютерах предпочтительнее использовать именно его. Ну, имеется еще ***Netscape*** и ***Internet Explorer***. Как говорится, "на любой вкус". Сейчас мы с Вами попробуем начать путешествие по веб.

Для того чтобы грамотно путешествовать по веб, необходимо ввести два понятия: Адрес каждой страницы и Гипертекст. Начнем с адреса. Надо сказать, что каждая веб-страница имеет свое имя, по которому ее можно найти с помощью браузера. Теперь, обратите внимание, что во всех наших трех обозревателях имеется вот такое поле адреса. Именно здесь указывается адрес веб-страницы. Адрес веб-страницы еще называется ***URL***.



На русском языке это звучит примерно как Унифицированный указатель информационного ресурса, то есть, ***uniform resource locator***. Каждая веб-страница имеет свой ***URL***. Эта последовательность адреса обычно начинается вот с этой части — ***http***. Она может содержать три ***www***, иногда говорят "ввв", хотя это не очень грамотно. А дальше идет сам адрес страницы. Если мы переключимся на ***Netscape***, то здесь мы увидим примерно то же самое. Три ***www*** в данном случае уже отсутствует, а ***http*** присутствует. А дальше идет адрес вот этой загруженной веб-страницы.

Это — домашняя страница ***Home*** компании ***Netscape***. А теперь посмотрим, что мы имеем в ***Opera***. Здесь, опять-таки, присутствует ***http***, а затем уже — адрес страницы ***My Opera***. Исключительно каждая веб-страница имеет свой адрес. Теперь перейдем к понятию Гипертекст. Надо сказать, что практически на каждой странице, которую отображает для нас браузер, представлены дополнительные ссылки, которые указывают на другую страницу. В большинстве случаев они подчеркнуты вот таким образом. Если мы наведем сюда курсор мыши, он превратится вот в такую ладошку. Это означает, что

при нажатии на клавишу мыши мы сможем перейти на следующую веб-страницу. Получается, что наши веб-страницы, как бы, связаны. Если мы сейчас нажмем... Давайте, попробуем это сделать. Допустим, что мы хотим подробнее узнать об ***Opera*** 7. Вот мы и перешли на следующую веб-страницу, где подробно эта бета версия ***Opera*** 7 рассматривается. Здесь предлагается приобрести ее. Как мы уже говорили, версия без рекламы стоит примерно 40 долларов. Или же предлагается бесплатно скачать.

Вот такие кнопки в большинстве случаев также представляют собой гиперссылки. Кроме того, что гиперссылка может представлять собой подчеркнутый текст, она может представлять собой и не подчеркнутый текст. Поэтому надо быть очень внимательным. Например, в данном случае вот здесь нет никаких подчеркиваний. Только при наведении сюда курсора мыши он превращается в ладошку, и текст становится подчеркнутым. В данном случае мы здесь читаем информацию о версии ***Opera*** для ***Windows***.

Но мы можем прочитать и информацию о версии ***Opera*** для ***Linux***. Обратите внимание на то, что вот здесь ***Opera*** имеет свою строку состояния, и при наведении курсора мыши на гиперссылку там появляется ***URL*** данной страницы, куда мы и перейдем. Нажмем на ***Linux***. Здесь рассказывается про версию браузера для ***Linux***, и, соответственно, здесь уже изменился ***URL***. Давайте, перейдем в ***Netscape*** и попробуем проделать нечто подобное вот здесь. Что мы здесь имеем?

Вы можете убедиться в том, что, если мы наведем сюда курсор мыши на всю эту картинку, то он также превратится в ладошку. Иначе говоря, гиперссылкой может быть не только текст, но и картинка. В данном случае строка состояния у нас находится вот здесь внизу. Если мы наводим сюда курсор мыши, то видим, что здесь появляется ***URL*** той страницы, куда мы перейдем при нажатии клавиши мыши. Попробуем сделать это и посмотрим, поменяется ли здесь наш адрес.

Начинает выполняться переход... Перед нами в ***Netscape Navigator***-е начинает открываться та страница, на которую мы перешли с помощью гиперссылки. Вы можете видеть, что здесь несколько изменился адрес, так называемый, ***URL***: здесь дополнительно появились ***tv*** и \ (слеш). Точно так же мы можем переходить по гиперссылкам и в ***Internet Explorer***-е. Давайте, попробуем это сделать. Ну, что нас в данный момент здесь может интересовать? Допустим, что нас здесь интересует вот это — Авиабилеты. Нажмем сюда. Вот здесь внизу у нас имеется строка состояния, и в данном случае открывается вот этот ***URL***, на который мы перешли.

Нам предлагается заполнить эту форму для того, чтобы заказать авиабилеты. Как это делается, мы с Вами более подробно изучим на последующих уроках, а сейчас мы просто приводим это в качестве примера

использования вот таких гиперссылок. Ну, как мы уже сказали, это — система взаимосвязанных документов, на которые можно переходить с помощью гиперссылок. Она известна как гипертекст. Это второе основное понятие веб-а. А теперь, если Вы в достаточной степени поняли, что представляет собой ***URL*** (адрес веб-страницы) и, соответственно, что такое гипертекст, Вы сможете совершенно спокойно путешествовать по веб. Более подробно мы с Вами познакомимся с этим на следующем уроке. А теперь завершим этот урок, на котором мы запустили три обозревателя Интернет, то есть, три Интернет-браузера: ***Internet Explorer*** (вот он у нас), ***Netscape*** и ***Opera***.

6.3.1.2. Навигация в обозревателе

На прошлом урок мы с Вами начали свое путешествие по веб. Мы узнали, что такое гипертекст и узнали, что такое ***URL*** и, что каждая веб-страница имеет свой собственный ***URL***. Надо сказать, что одним из способов путешествия и блуждания по веб является прямой ввод вот сюда, вот в это поле адреса ***URL*** нужной нам страницы. Предположим, мы вводим сюда следующий ***URL***: ***www.yandex.ru***. ***Yandex.ru***, является одним из самых популярных поисковых систем в России. Он же является и, так называемым, рубрикатором. Там имеется множество рубрик. Сейчас мы с Вами это подробно рассмотрим. Надо сказать, что для ввода адреса в большинстве случаев вот эту первую часть ***http*** вводить не нужно: браузер проставит ее сам. Сейчас мы вводим это в ***Netscape***-е.

Давайте, введем... Переключимся на английский шрифт... "***www***, точка, ***yandex***, точка, ***ru***". И нажмем на ***Enter***. Сейчас ***Netscape*** пытается открыть соответствующую страницу. А мы на примере вот этого введенного адреса подробнее разберем, что представляет собой адрес веб-страницы. Нам это в любом случае пригодится.

В данном случае, как мы с Вами уже заметили ранее, ***URL*** имеет следующую структуру. Вот это начальное ***http***... Что это такое? Это — схема доступа. В данном случае этим нас информируют о том, что мы подключаемся к удаленному компьютеру. А мы знаем, что весь Интернет и весь веб — это удаленные от нас компьютеры. Итак, мы подключаемся к нему с помощью вот этой схемы доступа — ***http***, то есть, с помощью этого протокола.

Дальше идет имя машины и имя домена. Имя машины обычно начинается с трех букв ***www***. Это указывает на работу в паутине ***www***, то есть, в паутине

веб. А имя домена является идентификатором машины удаленного доступа. Что это значит? В данном случае наш домен — это ***yandex.ru***. Через косую черту (вот здесь) могут указываться еще и некоторые имена файлов и тому подобное. К примеру, у нас после точки следует ***ru***. Что это означает? Это означает, что здесь определена Россия. Это — идентификатор страны. Например, Канада указывается через "***ca***", то есть, так называемое, "ка". ***Uk*** — это Великобритания, ***us*** — это США и так далее. Очень часто здесь после последней точки следует, например, ***com***.

Так обозначаются коммерческие организации. Давайте, переключимся вот сюда. Мы видим, что у нас здесь имеется ***msn.com*** (***com*** символизирует коммерческую организацию). Очень часто здесь пишется ***edu***, что определяет организацию системы образования, ***org***, определяющее некоторую организацию и так далее. В данном случае мы здесь имеем ***ru***: мы находимся в России. И мы видим вот такой первый сайт, то есть, первую страницу ***yandex.ru***.

Все элементы, которые мы сейчас видим здесь, представляют собой гиперссылки. Они у нас здесь собраны по рубрикам вот таким образом. Например, Компьютеры, Связь и тому подобное. Компьютеры — это Интернет, Связь — это, собственно, тоже Интернет. Мобильная связь, вероятнее всего, относится к связи. Софт и ***Hardware***, как Вы понимаете, относится к компьютерам. Мы можем щелкнуть по любой из этих гиперссылок.

Выберем, например, Мода и красота (допустим, нас интересует именно это). В этом же Яндекс-е мы перейдем на список ресурсов, который называется Мода и красота. Давайте, попробуем сделать то же самое на других браузерах (обозревателях). Перейдем, например, вот сюда и введем сюда какой-нибудь другой адрес. Точно так же, как и в прошлый раз, мы переключаемся на английский шрифт и вводим ***www*** и, допустим, ***ntv.ru***. Таким образом, мы с Вами попадем на домашнюю страницу телекомпании НТВ. Сейчас она откроется перед нами... Итак, мы оказались на сайте ***ntv.ru***. Этот сайт еще продолжает открываться. Здесь нам сообщают, что сейчас имеет место в эфире, что ожидается в эфире и так далее. Это, так называемая, первая страница этой телекомпании. А сейчас мы попробуем в обозревателе ***Internet Explorer*** войти на сайт Мультимедиа Технологии, то

есть, той фирмы, которая, собственно, и создала для Вас данный курс Интернет 2002. Наш адрес, как Вы уже знаете, ***www***, затем — ***mmt*** (что означает Мультимедиа Технологии). Затем пишем ***dl*** (что означает дистанционное обучение) и ***ru***. Нажмем на ***Enter*** и подождем. Вы видите, что этот метод доступа, то есть, сам ***http***, обозреватель дописал за нас, мы его не вводили. Сейчас мы находимся на головной странице сайта дистанционного обучения фирмы Мультимедиа Технологии. Здесь, если Вы обратите внимание, тоже имеются гиперссылки. Здесь имеются вот такие расскрывающиеся меню.

Если мы просто наводим курсор мыши на эти пункты, то открываются вот такие подменю, которые являются гиперссылками. Мы можем нажать на любую из них, и перед нами откроется страница, посвященная соответствующей тематике. Допустим, что нас интересует изучение какого-либо иностранного языка, например, английского. Для этого нажмем на гиперссылку ***English Platinum*** 2000. Сейчас

Вы видите здесь вот такую, как бы, аннотацию к данному курсу ***English Platinum*** 2000. Иначе говоря, нам удалось с помощью ручного ввода ***URL***-адреса получить некоторую интересующую нас информацию. Если мы переключимся на ***Netscape Navigator***, где уже открылась рубрика Каталог и семья, то мы увидим, что здесь Яндекс нам показывает ресурсы, посвященные моде и красоте. Вот здесь — Красота ***online*** (это журнал о красоте и моде), фитнесс и так далее. Как работать более подробно с рубрикаторами и поисковыми машинами, мы с Вами изучим позже.

А сейчас мы просто путешествуем по веб. Давайте, теперь попробуем вот здесь в браузере ***Opera*** ввести еще что-нибудь, чтобы как следует освоить ввод адреса. Предположим, что нас с Вами интересуют последние голливудские сплетни. Правда, эти сплетни нам придется читать на английском языке, но мы, все же, попробуем это сделать. Итак, мы вводим ***www.hollywood.com*** (коммерческое предприятие). Нажимаем на клавишу ***Enter***. Производится загрузка страницы. Мы сейчас посмотрим, что же там делается в далеком Голливуде. Перед нами открывается первая страница удаленной машины, которая называется Голливуд.

Вы видите фотографии звезд мирового кинематографа. Все это очень интересно. И, если Вы достаточно хорошо владеете английским языком, то Вы сможете узнать отсюда много нового и интересного. А вот эти подсветки тоже являются гиперссылками. Вы можете нажать на любую из них и перейти на следующую страницу, которая связана с данной страницей. Перед нами вот такой красочный голливудский сайт, красочная голливудская страница. Надо сказать, что обычно веб-браузеры запоминают несколько последних

посещенных нами веб-страниц. Щелкнем, например, на какую-нибудь ссылку. Пусть это будет, допустим, вот эта ссылка.

А, вообще, будет лучше, если мы перейдем на какой-нибудь русский сайт и щелкнем там, чтобы нам было более понятно. Щелкнем, например, вот на эту ссылку описания курса ***TeachPro*** 3***D Studio MAX***. Итак, перед нами аннотация курса ***TeachPro*** 3***D Studio MAX***. Допустим, что мы все это уже прочитали, и хотим вернуться обратно на предыдущую страницу.

Мы очень легко можем сделать это. Для этого нам просто нужно на этой Панели навигации щелкнуть по этой кнопке — Назад. Давайте, щелкнем на эту кнопку. Как видите, мы находимся в обозревателе ***Internet Explorer***, который также имеет кнопку — Назад. Последний раз мы здесь изучали аннотацию курса английского языка и мы, собственно, на ней и оказались. Если мы сейчас щелкнем вот на эту кнопку, то есть, на кнопку — Вперед, то мы снова окажемся на странице курса ***TeachPro*** 3***D Studio MAX***. Но дальше этой страницы мы не выходили, поэтому эта кнопка после этого оказывается закрытой (закомментированной).

Точно так же нельзя идти обратно до бесконечности. Попробуем нажать несколько раз на кнопку — Назад. Как Вы помните, мы открыли ***Internet Explorer***, начиная вот с этого адреса, поэтому дальше мы не можем никуда перейти. Теперь перейдем на ***Netscape*** и посмотрим, имеются ли подобные кнопки на Панели навигации ***Netscape***. Мы убеждаемся в том, что и здесь, безусловно, такие две кнопки имеются (Назад и Вперед).

Давайте, попробуем нажать на кнопку — Назад. Как видите, с помощью этой кнопки мы перешли на главный сайт поисковой машины ***yandex.ru***. Если мы еще раз нажмем сюда, то, вероятнее всего, мы вернемся на главную страницу компании ***Netscape***. Да, это именно так, с этой страницы мы начали обзор Интернета в браузере ***Netscape Navigator***. Поэтому кнопка Назад сейчас закрыта. Дальше нам идти некуда.

Но зато активна кнопка Вперед. Если мы нажмем на нее, то снова окажемся на поисковой машине ***yandex.ru***. Вот теперь эта поисковая машина ***yandex.ru*** начинает открываться перед нами. Давайте, проделаем тот же самый эксперимент в обозревателе ***Opera***. Мы видим, что и здесь имеются кнопки Назад и Вперед. Нажмем на кнопку Назад и посмотрим, где мы окажемся. Все правильно: мы оказались на сайте НТВ, в который мы вошли сразу перед сайтом Голливуд. Если мы еще раз нажмем на кнопку Назад, то окажемся на главной странице ***opera.com***. Все верно.

Точно так же, если мы нажмем Вперед, то окажемся на НТВ. Итак, мы с Вами еще раз подробней изучили, что такое ***URL***, ввели несколько адресов, получили определенную информацию и научились пользоваться кнопками Назад и Вперед. На следующем уроке мы начнем изучение остальных кнопок

Панели навигации для каждого обозревателя, а затем изучим все остальные возможности этих браузеров.На этом уроке мы с Вами продолжим изучение наших браузеров. Давайте, сейчас пройдемся по этим трем кнопкам. Это кнопки: Остановить, Обновить и Домой. Что все эти кнопки означают? Очень часто случается, что мы уже сразу в начале загрузки страницы обнаруживаем, что данная страница нам попросту не нужна. Начнем, например, открывать в браузере ***Netscape Navigator*** какую-нибудь страницу.

Ну, например, пусть это будет не ***yandex.ru***, а, допустим, ***rambler.ru***. Надо сказать, что в этом поле адреса мы можем не только вводить наш ***URL*** с нуля, но и редактировать имеющийся адрес. Мы можем сейчас убрать слово ***yandex*** и ввести вместо него, к примеру, слово ***rambler***... Это поможет нам сэкономить время. Итак, начинает открываться страница ***rambler***. Если мы именно в этом момент передумали ее открывать, то мы можем прямо сейчас нажать на кнопку Стоп. Теперь уже в данный момент браузер уже не пытается открыть эту страницу, а просто демонстрирует нам только ту часть страницы, которую он успел к этому времени загрузить.

Это, фактически, только начало этой страницы. Вот оно. Но мы можем и продолжить загрузку этой страницы. Мы с Вами уже делали нечто подобное на первых уроках данного курса. Если Вы помните, мы это делали посредством нажатия вот на эту кнопку тогда, когда у нас оборвалась связь с Интернетом (с провайдером Интернета). Нажимаем на эту кнопку. Фактически, это означает обновление загрузки. И теперь мы видим по этой строке, что загрузка страницы ***rambler.ru*** продолжается.

Сейчас мы ее, практически, уже видим всю. Давайте, попробуем проделать ту же самую операцию и в обозревателе ***Internet Explorer***, и в обозревателе ***Opera***. Допустим, мы сейчас хотим открыть страницу... Какую же страницу нам открыть... Ну, например, вот эту — Компьютерная грамотность, ***TeachPro***, ***Illustrator*** 9.0. Открываем. Вот по этому градуснику мы можем судить о том, что страница уже открывается. Правда, мы сейчас не успели нажать на кнопку Стоп. Давайте, заново попытаемся открыть страницу. Выберем — Информация. Вот теперь у нас начинает открываться данная страница, сейчас тут появится вот этот маленький градусник. Вот и все. Правда, у нас практически открылась вся страница.

Давайте, попробуем еще раз. Выберем теперь ***Online*** Уроки. Теперь мы удачно нажали на кнопку Стоп. Вот эта кнопка — Остановить загрузку. Если мы сейчас нажмем на кнопку — Обновить (вот она), то именно та страница, адрес которой указан вот в этой адресной строке, будет загружена еще раз. Смотрим внимательно. Да, она уже загрузилась полностью со всеми своими имеющимися меню и со всеми сопутствующими картинками, которых в прошлый раз не было, потому что мы нажали на кнопку Остановить загрузку,

в результате чего эти картинки и не были отображены на экране. Короче говоря, мы убедились, что ***Internet Explorer*** имеет те же самые возможности по остановке и обновлению загрузки страниц.

Теперь посмотрим, имеется ли такая возможность в браузере ***Opera***. Давайте, перейдем здесь на какую-нибудь страницу. Допустим, нас интересует телепрограмма. Здесь мы, кстати, видим, что кнопка Обновить тут также имеется. Она же превращается в кнопку Остановить. Давайте, остановим. Вы можете убедиться в том, что новая страница осталась нераскрытой. Здесь уже имеется ***URL***-адрес определенной длины.

И, если мы после этого нажмем на кнопку — Обновить, то загрузка будет продолжена. Вот здесь с помощью этого индикатора мы можем наблюдать процент загрузки нашей страницы. Вот сейчас почти весь документ открылся. А вот теперь здесь показано, что документ у нас открылся на 100 процентов. Иначе говоря, та же самая возможность по остановке и обновлению страницы имеется и в обозревателе ***Opera***.

А теперь мы рассмотрим, для чего предназначена эта кнопка Домой (Перейти на домашнюю страницу). Мы с Вами уже говорили о том, что домашняя страница — это та страница, с открытия которой начинается запуск данного браузера. Если Вы помните, загрузка ***Opera*** начиналась с запуска домашней страницы компании, которая и произвела этот браузер. Нажмем на кнопку Домой. Мы оказались в этом адресе ***MyOpera.com***, где мы и были изначально. Теперь проделаем то же самое и в ***Internet Explorer-е***.

Вот здесь имеется вот такой "домик". Нажмем на него. Сейчас мы окажемся вот на этой странице, с открытия которой и началось наше знакомство с ***Internet Explorer***-ом. Посмотрим, что же у нас имеется в ***Netscape***-е. Здесь у нас такой кнопки нет, но здесь внизу имеется вот такой значок, который называется ***Home***. Если мы нажмем на этот значок, то мы с Вами окажемся на сайте ***Netscape***. Вот, собственно, и сам сайт, с которого и началось наше знакомство с ***Netscape***.

Именно такая похожесть всех обозревателей Интернета и говорит о том, что, в общем-то, они служат для одной и той же цели и имеют, практически, одну и ту же функциональность. Разница — весьма незначительна. Вы это почувствуете тогда, когда мы подробно начнем знакомиться с каждым интерфейсом в отдельности. О курсоре мыши необходимо сказать следующее. Ну, курсор мыши Вы привыкли видеть в разных состояниях.

Во-первых, обычно курсор мыши бывает черного цвета, каким Вы привыкли видеть его. Мы специально установили его именно таким, чтобы его лучше было видно в нашем курсе ***TeachPro***. Во-вторых, курсор может принимать форму ладони, в который он превращается при его наведении на гипертекст. Кроме того, курсор мыши может принимать форму песочных

часов, которую Вы сейчас видите перед собой. Наш курсор мыши превращается в песочные часы, когда производится загрузка какой-нибудь веб-страницы. Сейчас идет загрузка, поэтому мы видим именно песочные часы. Как только загрузка завершится... Кстати, о том, что идет загрузка, свидетельствует вот этот живой значок. Сейчас загрузка будет завершена, и Вы увидите, что курсор мыши с песочными часами превратится в обычную черную стрелку.

Мы видим, что наша страница почти уже загружена, и Вы сейчас увидите привычную черную стрелочку. Подождем немного. Здесь мы можем видеть процесс загрузки нашего документа. После окончательной загрузки документа мы сможем увидеть привычный вид нашего курсора мыши. Здесь появляется вот такое всплывающее окно. Закроем его. Вы убеждаетесь в том, что курсор нашей мыши уже представляет собой черную стрелочку. Тут продолжается некоторая загрузка.

Давайте, здесь нажмем на кнопку Стоп. Итак, Вы видите, что курсор мыши превратился вот в эту черную стрелочку, то есть, загрузка уже завершена, и программа ожидает наших дальнейших действий. Перейдем на ***Opera***. Мы видим, что и здесь курсор нашей мыши имеет вид черной стрелки, а значит, мы уже можем производить определенные действия. Надо отметить, что мы имеем возможность открытия какой-нибудь конкретной гиперссылки в дополнительном окне обозревателя. Что это значит? Установим курсор мыши сюда и попробуем открыть содержимое этой гиперссылки в новом окне данного браузера.

Для этого выполним щелчок правой клавишей мыши, воспользуемся контекстным меню и в выберем нем команду Открыть в новом окне. Вы можете убедиться в том, что этой же команде соответствует левый клик мыши, но при нажатой клавише ***Shift***. Попробуем именно так и сделать. Нажмем на клавишу ***Shift*** и сделаем клик мышью. Сейчас у нас фактически открылась другая страница нашего обозревателя. Вот, видите, здесь имеется вот такое подобие панели задач в ***Windows***-е. Нажимая на каждую кнопку (вот они), мы видим, соответственно, дополнительные окна. Ну, вот это окно у нас в данном случае всплывающее. Оно — отдельное. Давайте, закроем его, в данный момент оно нам не нужно. И мы видим на панели задач открытые две текущие страницы: первая, где мы, собственно,

нажали ***Shift*** и клик, и вторая, которая и открылась. Давайте, попробуем открыть третью и посмотрим, получится ли это у нас. Нажимаем клавишу ***Shift*** и делаем клик вот на этом слове ***Linux***. Вы видите, что в данный момент здесь у нас уже открыты три страницы. Напомню еще раз, что мы находимся в обозревателе, который называется ***Opera***. Это творение норвежских программистов. Надо сказать, что он очень легкий. Возможностей в нем не очень много, и потребляет он незначительные ресурсы Вашего компьютера.

Давайте, попробуем сделать то же самое, но теперь уже в окне ***Netscape***. Посмотрим, как это будет выглядеть вот тут. Нажмем на клавишу ***Shift*** и сделаем клик, например, на этой строке (на этой гиперссылке). Нажимаем. К сожалению, мы видим, что открывается совершенно другое окно. Давайте, закроем его. Это доказывает, что данное сочетание не срабатывает в ***Netscape Navigator***-е. Попробуем на данной ссылке нажать правую клавишу мыши. Мы видим, что в данном случае именно такого сочетания клавиш (***Shift*** и клик) здесь не поставлено в соответствие. Здесь предлагается просто открыть данную ссылку в новом окне. Давайте, сделаем это с помощью мыши. Мы видим, что здесь открылся второй обозреватель ***Netscape Navigator***-а.

Вы можете видеть, что здесь на панели задач ***Windows*** их уже два. Имеется два отдельных вот таких окна. В этом окне сейчас открывается та самая ссылка, на которую мы нажали. Давайте нажмем на Остановить. Нам открывать не нужно, мы просто изучаем принцип. Мы просто закроем вот этот второй ***Netscape Navigator***.

Давайте проведем тот же эксперимент в ***Internet Explorer***-е. Сейчас мы будем умнее и не будем просто так нажимать на ***Shift*** и клик. Делаем клик правой клавишей мыши и видим, что здесь также имеется возможность Открыть в новом окне. К сожалению, и здесь не поставлено никакого сочетания клавиш. Открываем в новом окне и посмотрим, что мы будем иметь здесь. Здесь на самом деле происходит то же самое. Мы имеем два вот таких разных окна ***Internet Explorer***. Они совершенно идентичны (вот они), с тем же набором меню, с панелью навигации и другими разнообразными панелями. Видите, вот они.

На самом деле пользоваться вот таким способом открытия веб-страниц очень удобно, особенно если у Вас не очень хорошая связь с Интернетом. Пока у Вас открывается одна страница, Вы читаете информацию с другой. Таким образом, экономится время. Давайте, закроем этот дополнительный ***Internet Explorer*** и продолжим изучение вот этих основных принципов работы с нашими обозревателями. Ну, как Вы видите, после того, как Вы нашли нужную информацию, у Вас может возникнуть необходимость ее напечатать. Кнопка с таким принтером практически имеется во всех обозревателях. В

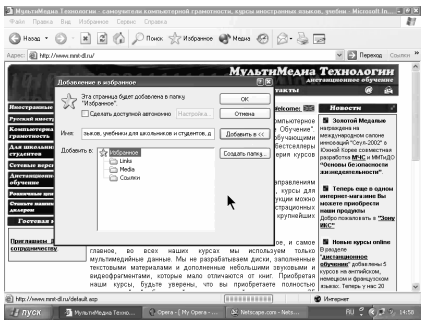
данном случае в ***Internet Explorer***-е она находится здесь. В ***Netscape Navigator***-е она находится вот здесь. И здесь в ***Opera*** также имеется такая кнопка Печать.

Давайте, попробуем на нее нажать. Сейчас происходит печать страницы на принтер, который у нас задан по умолчанию. Без всяких предварительных настроек у нас напечаталась вот эта страница, на которой описывается браузер ***Opera*** 7 ***Beta*** версии для ***Windows***. Попробуем нажать на эту же самую кнопку в обозревателе ***Netscape***. Нажимаем на нее. Здесь у нас появляется окно для того, чтобы мы могли задать дополнительные установки, а не просто производится печать по умолчанию на имеющийся принтер.

Здесь можно выбрать принтер. В данном случае у нас установлен один принтер. Здесь можно установить некоторые определенные свойства, то есть, бумагу и так далее, а затем нажать на кнопку ***OK***. Можно, соответственно, установить, сколько страниц нужно напечатать. Попробуем проделать то же самое вот в этом обозревателе. Нажимаем на кнопку Печать. Здесь никакое окно не появляется и, точно так же, как и в ***Opera***, все выдается на печать.

Давайте, на этом завершим наш урок. Мы с Вами изучили вот эти оставшиеся возможности по навигации и научились производить распечатку данной текущей страницы Интернет, которая загружена на нашем браузере. На следующем уроке мы с Вами начнем более подробное изучение наших браузеров.

6.3.1.3. Любимые страницы. Печать



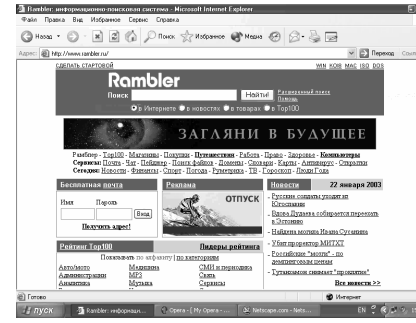
В ***Web***-е есть еще очень много интересных мест, которые стоит посетить. В эти места, очень возможно, Вам захочется еще вернуться. Все создатели хороших браузеров предусмотрели удобный способ запоминать эти места, и нам не нужно записывать все эти адреса на какую-нибудь отдельную бумажку или в отдельный файл, терять их и т. д. Практически в каждом браузере предусмотрена возможность сохранения самых любимых мест. Давайте, попробуем их сохранить и посмотрим, как же они вообще сохраняются? В обозревателе ***Internet Explorer*** эти самые адреса (наши любимые) сохраняются в, так называемом, пункте меню Избранное.

Как это, собственно, делается? Давайте, введем здесь какой-нибудь адрес (новый), который потом мы с Вами захотим сохранить. Или, что проще, мы можем открыть вот таким образом этот список адресов и напрямую попасть в тот адрес, в котором мы с Вами уже были. Ну, предположим, это у нас — ***mmt***

и ***dl***, то есть, сайт нашей компании Мультимедиа Технологии. Сейчас он перед Вами открывается. Мы сейчас увидим его первую страницу. Вот он. И, предположим, мы хотим его сохранить, как наше любимое место, для того, чтобы в последующем уже не набирать этот адрес вручную, не искать его в этом списке, который, кстати, может быть весьма большим, а просто напрямую попасть в него из этого пункта Избранное. Ну, добавляется вот в этот список с помощью команды Добавить в избранное. Давайте, мы это с Вами сделаем.

Эта страница, как нам сообщает вот это диалоговое окно, будет добавлена в папку Избранное. Эта страница у нас может быть доступна автономно, то есть, без подключения к Интернету. То есть, она вся запомниться на наш жесткий диск, в нашу операционную систему. Или, если этот флажок не устанавливать, то эта страница у нас будет доступна только при наличии подключения к Интернету. Имя нашей страницы — Мультимедиа технологии, самоучитель иностранных языков, учебники для школьников и т. д.

Давайте, нажмем вот на эту кнопку — Добавить в. Что это значит? Это значит, что в папке Избранное мы можем создать целую структуру новых папок. Например, давайте, создадим здесь папку, которую назовем, предположим, Саша. Это будет означать, что здесь все Избранное принадлежит Саше. И мы хотим именно для него сохранить вот эту ссылку — Мультимедиа технологии. Выделяем эту папку и нажимаем на кнопку ОК. Посмотрим, что же у нас произошло с меню Избранное? Нажимаем на меню Избранное и видим, что у нас здесь уже есть папка Саша.



При наведении курсора на эту самую папку (на этот самый пункт меню) мы уже увидим, что у нас есть Мультимедиа технологии, самоучители и т. д. Давайте, попробуем перейти на какой-нибудь другой сайт, допустим, на ***Opera.com***. Это тот самый сайт, с которого мы скачали наш третий браузер. И, сейчас мы попробуем по нашей ссылке Избранное (по ссылке Саша мультимедиа) перейти на нашу запомненную страницу, то есть, на наш запомненный адрес, по которому находится наша страница. Давайте, нажмем вот сюда — и мы оказываемся на сайте Мультимедиа технологии. То есть, все, что нам нравиться в Интернете (то, что мы видим в данном окне нашего браузера), мы можем сохранить с помощью этого пункта меню Избранное. Давайте, попробуем сохранить что-нибудь еще. Давайте, именно здесь еще раз наберем ***Rambler***, точка, ***Ru***.

Вот. Появляется страница нашего ***Rambler***-а. Предположим, что мы и ее хотим сохранить в нашей папке Избранное. Для этого выполняем команду Добавить в Избранное. Но в данном случае, мы не создаем новую папку, а добавляем в самый корень, просто в Избранное. Нажимаем на кнопку ОК. И, если сейчас мы откроем Избранное, то увидим, что вот здесь присутствует пункт меню, который называется ***Rambler*** — информационно-поисковая система. Давайте, попробуем перейти вот сюда. Мы перешли на Мультимедиа, сейчас она откроется, а затем перейдем обратно на ***Rambler***. Вот так работает возможность сохранения избранных ***Web***-страниц в обозревателе ***Internet Explorer***. Давайте, посмотрим и освоим то же самое в ***Netscape***. В ***Netscape Navigator***-е есть та же самая возможность, но делается это с помощью меню ***Bookmarks***, то есть, Закладки.

Предположим, мы хотим сохранить вот эту самую страницу у нас в ***Bookmarks***-ах. С помощью вот этого пункта меню выполняем команду ***Bookmarks This Page***, то есть, фактически, Установить закладку. Сейчас эта закладка установлена. Вы можете видеть, что здесь сбоку имеется вот такой пункт, который тоже называется ***Bookmarks***. Если мы на него нажмем, то увидим, что эта самая страница ***Netscape.com*** теперь занесена в список закладок. Вот эта страница ***Netscape.com***.

Попробуем еще раз поработать с этим пунктом меню. Выполним команду ***Manage Bookmarks***, то есть, Управление закладками. Перед нами появилось целое диалоговое окно, с помощью которого мы можем оперировать с закладками в ***Netscape***. Как мы это делаем? Вот здесь имеется список закладок, и, соответственно, вот такие списки папок, которые мы можем открывать вот таким образом, нажимая на такой треугольник, направленный вбок, и, соответственно, закрывать их. Обе эти операции мы делаем с помощью нажатия клавиши мыши. Предположим, что мы хотим создать новую папку.

Нажимаем сюда и именно здесь также набираем Саша. Папка Саша. Нажимаем ОК. Вот здесь у нас теперь есть папка, которая называется Саша. Если мы ухватимся за этот адрес и перетащим его вот таким образом в папку Саша, то здесь рядом с Сашей появится такой треугольник, и в нём будет ссылка на домашнюю страницу компании ***Netscape.com***. В общем-то, примерно понятно. Более подробно это окно Вы можете изучить и самостоятельно. Оно на самом деле очень простое в управлении, и подробно на нём останавливаться не стоит. Это — папки, в которых, соответственно, находятся адреса. В данном случае в папке Саша находится адрес ***Netscape.com***. Давайте, закроем это окно и посмотрим, что же у нас здесь произошло с этими самыми закладками.

Сейчас мы видим, что у нас имеется папка Саша (то есть, все изменения, сделанные в том окне, отобразились у нас и здесь), в котором имеется вот этот

Netscape.com. Нажмем на него, в результате чего откроется наша домашняя страница... Вот поэтому индикатору мы можем судить о том, какой процент страницы осталось открыть. Вот здесь, как видите, у нас открываются дополнительные картинки. В общем, примерно понятно, что происходит. Теперь, давайте, попробуем сделать то же самое в браузере ***Opera***. Что мы тут имеем? Здесь также имеется пункт меню, который называется Закладки. Нажимаем клавишей мыши на этот пункт меню. Здесь в этом пункте мы видим тоже большое количество папок, которые имелись изначально.

Если Вы обратили внимание, такие же папки имелись также и в ***Internet Explorer***-е в папке Избранное. Вот это — наши ссылки: Настройка радио, наша папка Саша, дополнительные ссылки, которые нам могут понадобиться, и информация в корне. В ***Netscape*** имеются вот эти встроенные папки, которые нам даны изначально. Это — поисковые машины (вот они), о которых мы с Вами будем беседовать несколько позже. Вот — Лучшая веб (ну, это, согласно мнению разработчиков и работников компании ***Netscape***). Дальше идет то, что у нас имеется внутри самого ***Netscape*** (это — ***Chat***, Календарь, ***Mail***, Каналы и так далее). И, вот мы с Вами создали папку — Саша.

Собственно, вместо нее мы могли бы создать папку, которая, к примеру, называлась бы Моя любимая музыка. Мы бы помещали в нее ссылки на те музыкальные сайты, которые нам больше всего понравились. Давайте, попробуем сделать что-то в этом роде в ***Opera***. Откроем — Закладки. Выберем, например, строку Управление закладками. Вот здесь, с левой стороны нашего браузера открывается вот такая вкладка, которая называется Закладка. Допустим, мы хотим добавить новую папку. Добавим папку, которую назовем, Моя музыка. Нажмем на кнопку ОК.

В эту самую папку Моя музыка мы сейчас можем добавить какую-нибудь страницу. В данном случае, у нас открыта вот эта страница. Собственно, именно и ее ***Opera*** нам и предлагает добавить в наш список. Но мы ее сейчас добавлять не станем, потому что она нам в таком виде, собственно, не нужна. Давайте, попробуем открыть что-нибудь более интересное. В браузере ***Opera*** также имеется возможность нажать вот на эту стрелочку и увидеть те страницы, которые мы с Вами посещали ранее.

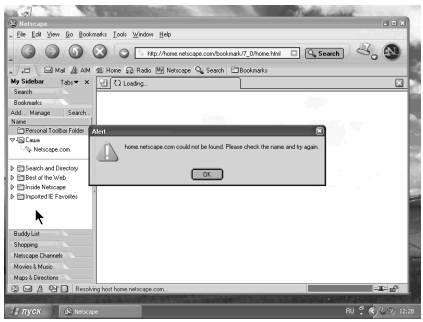
Ну, разумеется, не все, а лишь те, которые мы вводили сюда вручную. Сейчас у нас тут открывается сайт НТВ, и именно его мы постараемся сейчас сохранить в папке Моя музыка. Ну, мы можем уже сохранить этот сайт прямо сейчас, а не дожидаться, пока он откроется полностью. Создаем новую закладку. Пусть это будет официальный сайт телекомпании НТВ. Конечно, эта связь между телекомпанией НТВ и музыкой весьма условная. Мы добавили нашу страницу, вот отсюда.

Давайте, выйдем из этого списка папок, перейдем в наши закладки, найдем папку Моя музыка, который мы только что создали, и сохраним этот экран, то есть, вот этот адрес в папку Моя музыка. Попробуем сделать это. Выполним — Добавить. И — ОК. Посмотрим, что у нас получилось. Нет, к сожалению, добавление мы произвели не туда, куда хотели, а в корневой каталог. Нам, попросту, нужно было войти вот сюда. Давайте, так и сделаем. Войдем в нашу папку, выполним — Добавить и нажмем на кнопку ОК. Вот теперь у нас, наконец-то, все расположилось на свое место. Проверим. Вот наша ссылка. Что же у нас здесь получилось?

Мы теперь имеем ссылки на этот сайт в совершенно неподходящих местах. Как же мы с Вами будем их удалять? Самым простым способом удаления ненужной ссылки является выбор команды Управление закладками. Затем нужно выйти сюда, выделить ненужную закладку и удалить ее, воспользовавшись контекстным меню. Здесь для этого как раз имеется команда, которая называется — Удалить. Если мы теперь перейдём в пункт меню Закладки, то убедимся в том, что здесь у нас уже нет этой закладки.

Она у нас осталась вот здесь (вот она — официальный сайт НТВ). Она нам теперь доступна вот отсюда. Итак, мы с Вами научились сохранять избранное, так называемые, закладки во всех наших трех браузерах. Так что, теперь мы с Вами уже не пропадем: если нам понравится какая-нибудь страница, то мы всегда можем к ней вернуться, мы уже знаем, как ее сохранить. На этом, давайте, завершим наш сегодняшний урок, а на следующем уроке мы начнем беседу об ускорении запуска наших браузеров.

6.3.1.4. Автономный режим работы



На этом уроке мы с Вами поговорим о, так называемых, ***Online***-овых и автономных режиМАХ работы браузера. Надо сказать, что существуют два основных режима работы браузера. Мы сейчас попробуем запустить наш браузер ***Internet Explorer***. Нам сообщается, что та веб-страница, которая у нас загружена в ***Internet Explorer***, как основная, недоступна в автономном режиме. Она доступна только в режиме ***Online***, то есть, при наличии подключения к ***Internet***. Предположим, что мы с Вами хотим выбрать автономный режим работы. Давайте, сделаем это. Для этого нажмем на кнопку Автономно. Посмотрим, что у нас при этом происходит.

Сейчас перед нами открылось окно нашего браузера ***Internet Explorer*** версии 6.0. Нас предупреждают о том, что в данный момент невозможно увидеть эту веб-страницу, она недоступна в автономном режиме. Надо отметить, что при работе в ***Internet***-е те страницы, которые мы с Вами рассматриваем, сохраняются на нашем жестком диске. Если мы сейчас пройдемся по тем страницам, которые мы с Вами уже открывали, и попробуем, к примеру, зайти вот сюда, то у нас откроется сайт фирмы ***Microsoft***.

Мы его посещали ранее, и он у нас сохранился на наш жесткий диск. Поэтому при отсутствии подключения (а Вы видите, что подключения к ***Internet*** у нас сейчас нет), этот сайт у нас спокойно открылся. Автономный режим работы имеет большое количество плюсов. Первый плюс — это его экономичность. В данном случае такая работа с браузером ***Internet Explorer*** не оплачивается Вами. Она происходит очень быстро, и Вы можете вот таким образом, не заботясь о тех минутах, которые Вы проводите в Интернете, внимательно изучить всю нужную Вам страницу. То есть, Вы можете, спокойно и не торопясь, читать информацию, изображенную на этой странице. Давайте, еще раз убедимся в том, что все это происходит достаточно быстро.

Попробуем зайти еще на какой-нибудь сайт, например, на ***rambler.ru***. Вы можете убедиться в том, что смена страниц производится практически мгновенно. Это происходит именно потому, что они считываются не с ***Internet***, а с нашего жесткого диска. Такой режим работы очень удобен для начального освоения вашего браузера, когда Вам приходится много и долго экспериментировать. Надо сказать, что существует огромное количество специальных программ, которые скачивают Интернет-страницы на жесткий диск. Иначе говоря, можно совершенно спокойно просто скачать информацию с Интернета для ее последующего более внимательного изучения.

Такими программами являются ***TeleportPro***, прекрасная качалка фирмы Арсенал и тому подобное. А уже после этого Вы сможете скачанные страницы спокойно с помощью браузера неоднократно просматривать. Давайте, посмотрим, имеют ли такую же возможность и другие браузеры, то есть, ***Netscape Navigator*** и ***Opera***. Закроем ***Internet Explorer*** и попытаемся запустить ***Netscape Navigator***. Сейчас нам сообщается, что страница ***Netscape.com*** не найдена. Это та самая страница, с которой начинает открываться браузер ***Netscape***. Давайте, согласимся с этим (мы поняли, что она не найдена) и посмотрим, может ли этот браузер работать в автономном режиме.

Возможность работы в автономном режиме устанавливается из меню Файл, в котором имеется специальная команда, которая называется ***Work offline*** (работа в автономном неподключенном режиме). Выполним эту команду и попробуем открыть страницу, которую мы с Вами открывали ранее. Пусть это

будет ***yandex.ru***. Вы видите перед собой полностью всю страницу ***yandex.ru***. Она также открылась практически мгновенно в режиме ***offline***, то есть в автономном режиме без подключения к Интернету. Посмотрим внимательно вот сюда. Здесь мы не видим значка с изображением соединенных друг с другом компьютеров, который свидетельствовал бы о том, что связь с Интернетом имеется.

Давайте, откроем еще какую-нибудь страницу, допустим ***rambler.ru***. Она у нас также предварительно была сохранена на нашем жестком диске. То есть, мы зашли сюда, и ***Netscape Navigator*** сам сохранил эти страницы на жестком диске. Ну, что нам теперь осталось сделать? Нам осталось выяснить, имеется ли вышеуказанная возможность у браузера ***Opera***. Закроем ***Netscape*** и откроем браузер ***Opera***. Вот он. Запуск. Здесь нам также сообщается о включении автономного режима и о том, что браузер не может открыть эту страницу. Дело в том, что эта страница на жестком диске не сохранена.

Нам предлагается включить режим ***Online***. Мы отказываемся от этого предложения и сообщаем браузеру, что мы не желаем переключаться в режим ***Online***. Мы делаем вывод, что сейчас мы работаем в автономном режиме. Попытаемся открыть здесь, например, страницу Голливуд. А вот эта страница у нас открылась совершенно спокойно. Теперь она вся — перед Вами, и все это происходит при отсутствии подключения к Интернет.

Точно также мы можем открыть любую другую страницу. Откроем, например, страницу НТВ. Браузер сообщает нам о том, что включен автономный режим, и невозможно открыть данный сайт. Это говорит о том, что с этим сайтом мы с Вами работали достаточно давно, поэтому из жесткого диска она уже стерта. Дело в том каждый браузер (об этом Вы узнаете позже) имеет ограничение, то есть, он может сохранять на Ваш жесткий диск ограниченное количество информации. С данной страницей мы работали достаточно давно, поэтому она уже стерта с жесткого диска. Нам предлагается включить режим ***Online***. Если мы соглашаемся с этим, то нам сообщают, что нет связи с ***Proxy*** сервером. А почему нет связи, да потому что нет подключения. Мы сейчас просто должны подключиться для обеспечения соответствующей связи.

Мы так и сделаем. Подключаемся. Ну, вот, подключение произошло. Мы можем видеть, что подключение у нас уже имеется. Еще раз попробуем обновить вот эту страницу ***Ntv.ru***. К сожалению, мы снова перешли на Голливуд. Давайте, повторно выполним... Перед нами уже в режиме ***Online*** открывается страница ***Ntv.ru***, то есть, страница телекомпании НТВ. На этом мы завершим этот урок. Вот все, что

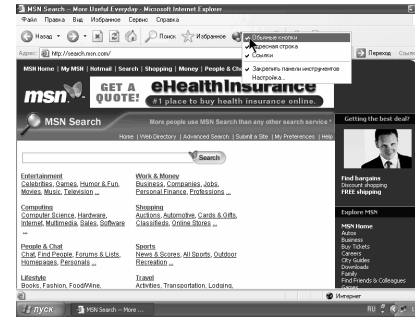
Вам следует знать об автономном режиме работы и о его отличиях от, так называемого, режима ***Online***. Этим уроком мы также заканчиваем

вступительную главу о наших браузерах. Вы уже умеете работать с Интернет, умеете запоминать любимые места, распечатывать их, а также умеете работать в автономном режиме работы. На следующих занятиях мы с Вами начнем изучать главы, посвященные интерфейсам наших браузеров, в которых каждый браузер будет подробно рассматриваться в отдельности.

6.3.2. Интерфейс Internet Explorer 6.0

6.3.2.1. Панели браузера

На этом уроке мы с Вами начинаем изучение интерфейса браузера ***Internet Explorer*** версии 6.0. Для начала, давайте, запустим его. Как Вы помните, он запускается с помощью меню Пуск. Вот здесь имеется вот такой значок с изображением буквы E, с помощью которого запускается браузер ***Internet Explorer***. Сейчас нам сообщается, что та веб-страница, которая установлена в качестве домашней в обозревателе ***Internet Explorer***, недоступна в автономном режиме. Мы с этим соглашаемся и нажимаем на кнопку Подключиться. Именно таким образом и устанавливается связь с нашим Интернетом. Итак, открывается самая первая страница.



Это страница ***msn***, страница фирмы ***Microsoft***. Теперь мы с Вами начнем изучение интерфейса браузера ***Internet Explorer***. Первое, что бросается в глаза, это самая верхняя титульная панель, которую иногда называют Заголовком окна. Что мы видим на этом заголовке? Во-первых, мы видим название страницы, то есть, фактически, ее имя. Страница наша называется Добро пожаловать на ***msn.com***. Дальше идет название нашего браузера — ***Microsoft Internet Explorer*** (Вы это уже знаете). Если бы эта страница была открыта в автономном режиме, то здесь у нас это было бы указано, был бы указан автономный режим. Сейчас у нас этого написано, то есть, страница открыта в режиме ***Online***, то есть, в режиме наличия подключения к Интернету. Еще у нас тут имеются вот такие три кнопочки. Эти кнопочки типичны для большинства приложений ***Windows***, и, для чего они предназначены, Вы, в общем-то, уже знаете.

С помощью этой кнопки минимизируют, то есть, сворачивают браузер ***Internet Explorer***. С помощью средней кнопки устанавливают либо полноэкранный режим работы, либо, так называемый, частично-экранный, то есть, когда обозреватель занимает не весь экран. А вот сейчас устанавливается полноэкранный режим. А с помощью вот этой крайней красной кнопочки

происходит выход из программы ***Internet Explorer***. Если мы сейчас нажмем на нее, то наша программа, как видите, закроется.

Давайте, запустим ее еще раз. Вот, она запущена. Пойдем дальше. Ниже титульной панели вот здесь у нас располагается панель главного меню. Это, фактически, строка с надписями, которые представляют позиции главного меню программ. Это: Файл, Правка, Избранное, Сервис и Справка. Если мы нажмем на один из этих пунктов, то откроется уже вот такое полное меню с определенными командами. Далее. У нас здесь имеется панель инструментов. Вот — эта панель. Фактически, это третья панель, на которой имеются кнопки для быстрого управления системы. Рядом с некоторыми из них имеются надписи, поясняющие работу той или иной кнопки. Некоторые кнопки не имеют таких надписей. Чуть ниже располагается панель адресов и ссылок. Это — четвертая панель.

Сюда вводится ***URL***-адрес нужной страницы, то есть, сетевой адрес того узла Интернета, в котором происходит работа. Здесь еще имеется вот такая команда ссылок, в которой имеются такие основные ссылки нашего браузера. Самую большую часть нашего экрана (нашей программы), фактически, занимает область просмотра документа (вот она). Именно в этой области просмотра открывается текущий документ. Ну, и в самом низу расположена, так называемая, панель статуса (вот она). Это панель, в которой отображается текущее состояние, то есть, статус программы. Здесь имеются различные пиктограммы. В данном случае, здесь имеется сообщение о том, что мы подключены к Интернет.

Вернемся к титульной панели, так сказать, к заголовку окна нашего браузера. Вот она. Если мы выполним вот здесь щелчок мышью, то мы увидим следующие команды: Восстановить, Переместить, Размер, Свернуть, Развернуть и Закрыть наше окно. Эти команды, практически, идентичны вот этим, которые тоже обозначают Свернуть, Восстановить и Закрыть наше окно. Единственное отличие заключается в том, что здесь нет команды Переместить. Ну, а перемещать окно нашего браузера мы сможем после уменьшения его размеров, схватив за заголовок этого окна.

Вот таким образом оно свободно перемещается. Это происходит идентично другим программам ***Windows***, по крайней мере, их большей части, поэтому ничего удивительного в этом нет. Давайте, снова развернем это окно во весь экран и продолжим его изучение. На нашем экране привлекают к себе внимание следующие элементы. Это, во-первых, вот такие маленькие треугольники. Какую роль они играют? Если мы нажмем на них вот таким образом, то перед нами окажется список соответствующих объектов.

Короче говоря, при нажатии на такой конвертик, мы получаем следующие возможности. Мы можем прочитать нашу почту, можем создать новое

сообщение, можем отправить ссылку вот этой открытой страницы по почте, можем отправить всю страницу, а не только ее ссылку (***URL***-адрес), а всю страницу, и можем прочитать новости. Если мы сейчас перейдем по какой-нибудь ссылке, к примеру, вот сюда к ***My msn***, то у нас активизируется кнопка Назад с соответствующим треугольником. Вы можете видеть, что здесь образовался соответствующий треугольник. Если мы теперь нажмем на него, то здесь мы, фактически, получим список страниц, предшествующих текущей странице. То есть, в прошлый раз мы находились на заглавной странице, которая называлась ***Welcome to Msn.com***. Если мы перейдем еще куда-нибудь (перейдем, например, вот сюда), то там уже мы будем иметь список, состоящий из двух адресов. Давайте, убедимся в этом.

Вот, теперь мы, фактически, имеем список, содержащий два адреса. Ну, как пользоваться этими треугольниками, мы уже поняли. Кроме того, имеется возможность увеличения обзор с помощью вот таких двух галочек, напоминающих кавычки. Это означает, что здесь у нас определенный элемент не поместился весь на экране. Но, если мы нажмем вот сюда, то мы уже увидим, как бы, продолжение этого элемента (его содержимое).

Мы увидим вот эти ссылки: ***Windows***, ***Windows Media***, Бесплатная почта ***Hotmail*** и настройка ссылок. Итак, работать с такими элементами, как такой треугольник и такая галочка, мы с Вами уже умеем. А теперь, давайте, завершим текущий урок. На следующем уроке мы с Вами будем работать вот с этой панелью инструментов, то есть, научимся добавлять и удалять определенные элементы этой панели инструментов.

6.3.2.2. Панели инструментов

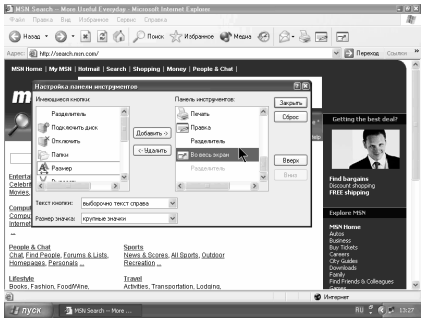
На этом уроке мы с Вами поговорим об изменениях интерфейса ***Microsoft Internet Explorer*** версии 6.0. С чего же начать? Подведем курсор мыши, например, к панели инструментов и вызовем контекстное меню, нажав правую клавишу мыши. Перед нами появилось вот такое контекстное меню с определенным списком. Рядом с некоторыми элементами этого списка стоит вот такая галочка. Что это означает? Это означает, что данный элемент у нас присутствует на экране.

Попробуем отключить какой-нибудь из этих элементов. К примеру, отключим пункт, который называется Обычные кнопки. Щелкнем мышью по этому пункту. Мы видим, что эта панель инструментов просто-напросто исчезла у нас с экрана. Теперь ее уже нет на экране. Почему это произошло? Это произошло по той простой причине, что мы отключили ее из списка вот в этом контекстном меню. Если мы теперь установим эту галочку обратно, то есть, сделаем повторный щелчок мыши, то данная панель инструментов снова появится на экране. Вот она. Точно так же можно поступить и с другими элементами. Попробуем сделать это. Контекстное меню.

Нажимаем на адресную строку. Адресная строка у нас пропала с экрана, сейчас ее на экране нет. Это, опять-таки, произошло по причине ее отключения. Вызовем ее обратно на экран. Адресная строка. И вот она у нас — снова на экране. Вновь воспользуемся контекстным меню. У нас здесь еще имеется пункт Ссылки вот в такой подгруппе. Нажмем на него. Мы убеждаемся в том, что список Ссылки просто-напросто пропал с экрана. Это для нас не является неожиданностью, мы этого и ожидали. Теперь перейдем к следующему, так называемому, подсписку Закрепить панели инструментов. Для чего служит эта команда?

Дело в том, что панели инструментов можно перемещать по своему желанию если мы сейчас снимем эту галочку, то наши панели инструментов станут, как бы, перемещаемыми. Вы можете видеть, что здесь появились вот такие точечки рядом с каждым элементом. Если мы сейчас дернем за эту часть и потянем ее вверх, то панель адреса ссылки переместится вот сюда и установится рядом с этой панелью инструментов, которая в данный момент не вся видна. Она частично находится за пределами нашего экрана, и мы видим только кнопку Назад.

Но, если мы нажмем на эту напоминающую кавычку кнопку вот таким образом, то мы увидим все остальные элементы нашей панели инструментов. А, как пользоваться этой кнопкой, мы с Вами изучили еще на прошлом уроке. Давайте, попробуем восстановить все, как было. То есть, спустим вот этот адрес сюда. Панель инструментов восстановилась на своем месте. Ссылки у нас расположены здесь. Кстати, ссылки мы можем поднять и установить рядом с панелью инструментов. Давайте, попробуем сделать это.



Захватим это мышью и потянем наверх вот таким образом. Если мы потянем эту часть, например, левее, то Вы сможете убедиться в том, что у нас, опять-таки, часть инструментов опять скрывается. На экране снова появляется вот такая кнопочка, напоминающая кавычки. При нажатии на нее мы можем видеть все остальные элементы данной панели инструментов. Ну, восстановим все на свои места. Ссылки должны находиться здесь. Вот так будет очень хорошо. Теперь, воспользовавшись командой Закрепить панели инструментов, закрепим их на своих местах. После этого мы их не сможем перетащить: сколько бы мы не дергали здесь, перетащить их все равно не удастся. И — последняя команда данного контекстного меню это Настройка. На экран вызывается диалоговое

окно, которое называется Настройка панели инструментов. Дело в том, что данную панель инструментов мы можем настраивать по своему желанию. Мы можем добавлять сюда кнопки. Можем также удалять некоторые кнопки и менять их расположение. Давайте, попробуем сделать это. Во-первых, вот здесь мы имеем список имеющихся кнопок.

Могут быть добавлены вот такие кнопки. Первый элемент — это Разделитель. Что значит Разделитель? Вы видите, что некоторая последовательность кнопок отделена от другой последовательности кнопок посредством вот такой черточки. Эта черточка и называется Разделителем. Короче говоря, кнопки у нас, фактически, сгруппированы по темам. Например, эти кнопки относятся к навигации, а это — уже следующая группа кнопок. Это — дополнительная.

Давайте, попробуем вставить какой-нибудь разделитель вот сюда. Выделяем его здесь и нажимаем на кнопку Добавить. Если Вы обратили внимание, здесь добавился такой разделитель. Теперь попробуем добавить еще какую-нибудь кнопку, допустим, Во весь экран. Выделяем ее вот таким образом и нажимаем на кнопку Добавить. Если Вы обратили внимание, кнопка Во весь экран добавилась, практически, мгновенно.

Вот, фактически, наша кнопка. Мы можем и переместить ее с помощью этого списка в любую из этих групп и в любую позицию, что осуществляется с помощью этих кнопок Вверх и Вниз. При последовательном нажатии на кнопку Вверх (обратите внимание вот сюда) эта кнопка меняет свою позицию вот таким образом. Фактически, она путешествует по нашей панели инструментов.

Аналогично при нажатии на кнопку Вниз она будет перемещаться направо. Очень хорошо.Здесь еще имеются два раскрывающихся списка: Текст кнопки и Размер значка. Что значит Текст кнопки? В данном случае выбран пункт Выборочно текст справа. Что это означает? Это означает, что у некоторых кнопок справа имеется поясняющая надпись. Вот — кнопка Назад, и здесь, соответственно, так и написано. А здесь такой надписи нет. Мы можем, во-первых, в обязательном порядке отображать подписи к кнопкам. Вы видите, что здесь сейчас все кнопки имеют свои подписи. Правда, в результате этого часть элементов панели инструментов на экране не поместилась, но при нажатии на такую кнопочку, оставшиеся элементы тоже откроются.

Для этого нужно предварительно закрыть диалоговое окно, которое Вы видите перед собой. Во-вторых, кнопки могут и вовсе не иметь подписей. Вот так. Теперь подписей нет вообще. Кроме того, имеется возможность изменения размеров наших значков с Крупных на Мелкие. Если мы сейчас закроем данное окно (нажмем на кнопку Закрыть), то наши значки поменяются на мелкие. Правда, каждый значок имеет поясняющую надпись, которая

всплывает после наведения на него курсора мыши. Так что становится понятно, какую функцию выполняет та или иная кнопка. Давайте, вызовем Настройку еще раз. Правая клавиша мыши.

Настройка. Восстановим все, как было. Здесь у нас были Крупные значки, здесь — Выборочно текст справа, а кнопки — Во весь экран у нас не было вообще. Выделяем этот элемент и нажимаем на кнопку — Удалить. Как видите, данный элемент удален с экрана. Точно так же удалим и Разделитель, потому что здесь он нам сейчас не нужен, так как после него никаких элементов не следует. Нажимаем на кнопку — Удалить и закрываем данное диалоговое окно. Как видите, мы восстановили внешний вид нашего ***Internet Explorer***-а. Сейчас все так, как и было раньше.

Кроме того, мы можем изменять внешний вид нашего окна с помощью меню Вид. Откроем меню Вид. Мы видим, что здесь всплывает меню, идентичное контекстному меню, которое мы вызывали ранее, и мы можем с ним аналогично работать. Далее идет пункт — Строка состояния, который мы так же можем убирать с экрана. Нажмем сюда. Вы видите, что строка состояния, которая была расположена внизу, исчезла с экрана.

Снова отобразим ее на экране. Вот так. Очень хорошо. Здесь имеются также панели обозревателя: Поиск, Избранное, Медиа, Журнал, Папки и Полезный совет. Эти панели обозревателя вызываются на экран следующим образом. Нажмем, например, на Поиск. В левой части окна ***Internet Explorer***-а открывается вот такая область. Эта кнопка окажется нажатой, и здесь мы уже сможем искать соответствующую веб-страницу.

Поиском мы с Вами займемся на последующих уроках, а сейчас просто попробуем нажать на кнопку Избранное. Мы с Вами это уже делали на прошлых уроках. Здесь появляется такая область, которая посвящена избранным нами веб-страницам. Вы можете тут видеть папку, которую мы с Вами создали на одном из первых уроков, это — папка Саша. Здесь появляются параметры Медиа, вот такой плеер... Все это происходит в левой части нашего окна. Ну, и, соответственно, Журнал. При нажатии на этот Журнал, мы можем видеть список посещенных нами ранее страниц.

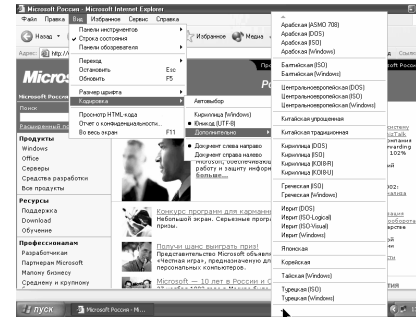
Например, сейчас мы находимся на этой веб-странице, а посещали мы весь такой список. Точно так же мы можем просмотреть, что мы посещали в пятницу, в четверг и в среду. Подробно работу Журнала мы с Вами изучим позднее. Отключим эту кнопку. Как мы уже говорили, то же самое можно осуществлять выбором этих пунктов подменю. Нажмем, например, на кнопку Журнал. Это — то же самое, если бы мы нажали на кнопку Журнал вот с этой панели инструментов. Вы можете убедиться в том, что кнопка Журнал у нас теперь нажата. Вот все, что относилось к изменению интерфейса обозревателя ***Internet Explorer***. Вы сами можете поэкспериментировать и привести внешний

вид Вашего обозревателя к максимально удобному для Вас состоянию. На этом завершим наш урок. А на следующем уроке мы с Вами ознакомимся с некоторыми приемами работы обозревателя ***Internet Explorer***.

6.3.2.3. Некоторые приемы работы

На этом уроке мы с Вами освоим некоторые приемы работы в нашем обозревателе ***Internet Explorer***. С некоторыми из этих приемов работы мы уже ознакомились на прошлых уроках, но ничего страшного не произойдет, если мы слегка повторим их. Давайте, закроем этот элемент Журнал, сейчас мы не будем заниматься им. Начнем осваивать приемы работы с ***Internet Explorer***. Как Вы уже знаете, мы можем расширить область просмотра.

Это мы делаем с помощью меню Вид. Мы можем убрать строку состояния, можем убрать соответствующие панели инструментов и тому подобное. Мы можем также либо вывести на экран панели обозревателя, либо скрыть их. Сейчас, например, у нас нет на экране никаких панелей обозревателя, все они скрыты. Вернем на экран нашу строку состояния и посмотрим, что еще тут можно сделать. С этими пятью кнопочками мы уже познакомились ранее. Давайте, повторим, что представляют собой эти кнопки.



Первая кнопка — Назад осуществляет возврат на ту страницу, которую мы открывали до этой страницы. Как Вы знаете, при нажатии на этот треугольник, мы можем увидеть названия тех страниц, которые были открыты перед этой страницей. Допустим, мы выбираем страницу — Добро пожаловать на ***Msn.com***. Открыли. Вот они. Это — кнопка Вперед. Здесь также имеется вот такой черный треугольник. Если мы нажмем на него, то мы будем иметь адреса тех страниц, на которые мы можем перейти после этой страницы. У нас тут имеется, например, ***Msn Search***. Нажмем сюда. Это, фактически, путешествие по последовательности открытых веб-страниц.

С этими тремя кнопками Вы уже знакомы. Первая кнопка прекратить загрузку страницы — Остановить является весьма полезной в том случае, когда у Вас начинает открываться очень большая "тяжелая" страница (для примера, давайте, начнем открывать вот эту страницу), а затем по какой-то причине Вы передумали ее открывать. Нажмем на эту кнопку. Эта страница у нас открылась не до конца, потому что мы прекратили ее загрузку, поняв ошибочность такого желания.

С помощью этой кнопки мы обновляем загрузку страницы. Нажмем на нее. Сейчас эта страница ***oddity.de*** у нас отроется еще раз. Мы видим, как она, фактически, открывается перед нами. А с помощью этой кнопки мы с Вами возвращаемся на основную страницу нашего браузера, то есть, на ту страницу, адрес которой указан в качестве основной страницы нашего браузера. Нажмем на нее. Мы перешли на нашу основную страницу. Попробуем перейти на какую-нибудь русскую страницу, то есть, на такую, которая у нас указана на русском языке, и поэкспериментируем с кодировкой. Надо сказать, что существует ряд таблиц кодировок знаков. Многие документы имеют специальные указатели или переключатели для установки нужной кодировки. Если перед Вами — текст из странных знаков, то это означает, что у Вас, скорее всего, кодировка установлена неправильно. Давайте, поэкспериментируем с кодировками. Во-первых, кодировку можно менять с помощью меню Вид. Здесь имеется такое понятие, как Кодировка.

Могут быть установлены следующие кодировки: например Юникод, Кириллица ***Windows***, Дополнительно (раскрывается вот такой огромный список кодировок). Мы сейчас попробуем установить, к примеру, вот такую кодировку — к примеру Греческая. Вы можете убедиться в том, что наша страница теперь отображена перед нами в непонятном виде. Мы вообще не понимаем, что здесь написано. Это и означает, что кодировка у нас указана совершенно неправильно. Итак, в случае, если кодировка указана неверно, нам нужно, может быть даже, экспериментируя вот здесь, правильно указать кодировку. Но, если Вы точно знаете, что открытая страница содержит именно русский текст (например, если Вы открыли какую-нибудь страницу русского сайта, а кодировка у Вас указана неверно), то лучше всего поочередно попробовать вот эти кодировки, которые относятся к кодировке Кириллицы:

KOI8-***R***, ***KOI***8-***U***,Кириллица (***ISO***) и Кириллица (***DOS***). Кроме того, еще имеется Кириллица (***Windows***), которая у нас как раз в этом пункте и фигурирует, как одна из использовавшихся в последний раз. Но, так как мы точно знаем, что в данном случае у нас была использована кодировка Юникод (мы это видели, когда открывали пункт Кодировка), то, давайте, выберем ее еще раз и посмотрим, восстановится ли у нас все. Мы убеждаемся в том, что все у нас здесь, действительно, восстановилось, и мы спокойно все можем прочитать.

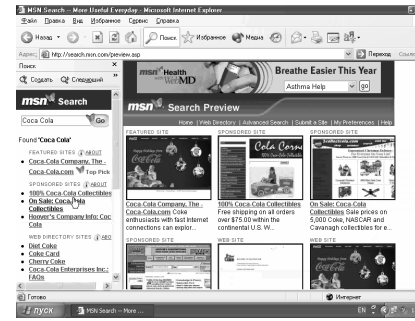
Надо отметить, что иногда текст документов может нам показаться слишком мелким, или, наоборот, слишком крупным. Менять размер нашего шрифта на экране мы можем, опять-таки, с помощью меню Вид и команды, которая называется Размер шрифта (вот эта команда). Здесь имеется список размеров шрифта. Шрифт может быть: Самый крупный, Крупный, Средний, Мелкий и Самый мелкий. Попробуем выбрать Самый мелкий и посмотрим,

что у нас произойдет. Вы убеждаетесь в том, что текст становится, практически, нечитабельным, он стал очень мелким. Для того чтобы пользоваться таким мелким шрифтом, необходимо иметь дисплей с высоким разрешением. Теперь попробуем изменить шрифт на Самый крупный.

Теперь буквы стали огромными. Надо сказать, что эти команды (Размер шрифта и Кодировка) можно также вызывать с панели инструментов. Для этого, воспользовавшись контекстным меню и вызвав диалоговое окно Настройка, мы можем попросту Добавить эти кнопки на экран. Вот кнопка Кодировка, а вот кнопка Размер шрифта. Мы их добавляем вот сюда и закрываем это диалоговое окно. Мы убеждаемся в том, что эти кнопки на экране уже имеются. Нажав на соответствующую стрелочку, мы увидим список кодировок, а здесь, соответственно, — список размеров шрифтов. У нас был установлен Средний размер шрифтов. Давайте, мы так и сделаем. А кодировка, как мы помним, у нас была установлена Юникод.

Лучше всего, все-таки, иметь эти кнопки на панели инструментов. Ну, может быть, кнопка размеров шрифтов Вам и не понадобится, но вот кнопкой Кодировка Вам придется пользоваться достаточно часто. Поэтому, давайте, мы просто уберем отсюда кнопку Размер шрифтов. Удалить. И Закрыть. А кнопка Кодировка мы оставим на экране. На этом завершим наш урок, а на следующем уроке мы займемся поиском, учетом и печатью Интернет-страниц.

6.3.2.4. Кнопки поиск, избранное



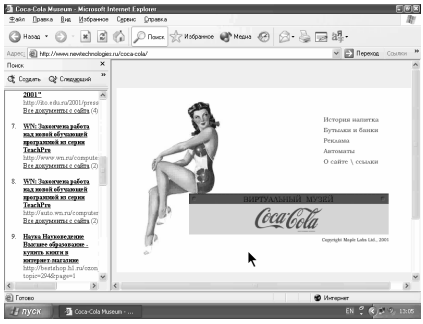
На этом уроке мы с Вами продолжим знакомство с интерфейсом браузера ***Microsoft Internet Explorer*** версии 6.0. Сейчас мы начнем знакомство с этими четырьмя кнопками: Поиск, Избранное, Медиа и Журнал. Начнем с кнопки Поиск. Надо сказать, что в Интернете существуют специальные поисковые машины с поисковой системой, которые помогают быстро найти нужные сведения, разбросанные на серверах ***Internet*** во всем мире. Более подробно о том, как осуществлять эффективный поиск в Интернете, мы с Вами поговорим в последующих главах. Эффективному поиску в Интернете будет посвящена целая тема, а сейчас мы просто предварительно ознакомимся с этим. Надо сказать, что каждый браузер имеет выход, хотя бы, на несколько такие поисковые системы.

Нажмем, например, на кнопку Поиск. В левой части нашего окна открылось вот такое, как бы, подокно, посвященное поиску, и именно здесь мы с Вами

можем искать нужную нам информацию. Предположим, мы хотим найти веб-страницу, которая содержит выражение (или словосочетание) ***Coca Cola***. Давайте, для этого переключимся на латинский шрифт. Вот так. Поиск будет производиться с помощью поисковой машины, которая называется ***Msn Web Search***, о чем здесь, собственно, нас и информируют. Нажмем на кнопку Поиск. В левой части окна появился вот такой список, как бы, гиперссылок, по каждой из которых мы можем перейти на веб-страницу, которая содержит такое словосочетание ***Coca Cola***.

Мы имеет вот такой достаточно большой список страниц. А в правой части нашего экрана, фактически, изображены начальные страницы этих сайтов. То есть, вот здесь, например, имеется ***Coca Cola Company***. Это — сама основная страница фирмы ***Coca Cola***. Здесь Вы можете видеть такой ее набросок, как бы, ***Screenshot***, то есть, изображение экрана (как он выглядит). Вы можете видеть здесь вот такой огромный список страниц. Предположим, что мы хотим осуществить поиск не с помощью поисковой машины ***Msn***, а с помощью какой-либо другой поисковой машины. Здесь при нажатии на эти галочки появляется вот такой раскрывающийся список, который называется Следующий. С помощью этого списка мы можем выбрать поисковую машину, с помощью которой и желаем производить поиск.

Допустим, нас интересует именно поисковая машина ***Yandex***. Выберем ее. Произведен поиск выражения ***Coca Cola*** при помощи поисковой машины ***Yandex***. В данном случае сюда не перешли наши ***Screenshot***-ы соответствующих веб-страниц, но при нажатии на какую-нибудь из этих ссылок вот таким образом, в этой части экрана откроется вся эта страница, в которой данное выражение ***Coca Cola*** используется. Итак, мы сейчас открыли



гиперссылку, которая называется ***Coca Cola*** — клевый напиток. Точно так же мы можем организовать поиск выражения ***Coca Cola*** с помощью любой другой поисковой машины. Попробуем, например, сделать это с помощью поисковой машины

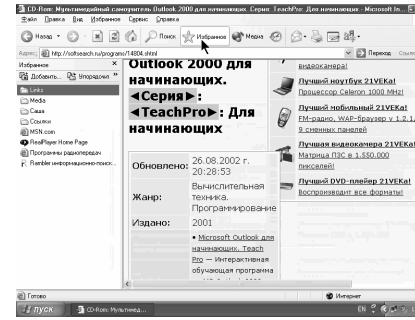
Rambler. Таким образом, мы выбрали другую поисковую машину, и сейчас произойдет поиск с помощью поисковой машины ***Rambler***. Вот здесь это и обозначено. Ну, вот, с помощью ***Rambler***-а найдены вот эти следующие гиперссылки, то есть, переходы на нужные нам веб-страницы. Попробуем, к примеру, перейти по этой гиперссылке — Виртуальный музей Кока Колы. В правой части экрана открывается соответствующая веб-страница. Итак, сейчас

перед нами открывается Виртуальный музей Кока Колы, и здесь мы уже спокойно и полноценно можем работать. Надо сказать, что это количество поисковых машин мы с Вами можем регулировать. Как это можно сделать? Делается это с помощью команды Настроить. Выполнили эту команду. Теперь нам предлагается настроить параметры поиска. То есть, определенными включениями и отключениями вот этих галочек мы регулируем тот список, который у нас открывался при нажатии на эту кнопку.

Правда, было бы лучше, чтобы здесь присутствовали все, но мы можем поменять их местами и сделать поисковую машину ***Rambler*** самой первой. Давайте, именно так и поступим и посмотрим, что еще мы здесь имеем. Надо сказать, что последние десять заданий поиска в данном случае сохранены. Почему? Да потому, что у нас здесь установлена соответствующая галочка. Если мы ее сбросим, то, соответственно, этой команды

Предыдущий поиск мы иметь не будем, и нам придется каждый раз вводить выражение заново. Ну, в том виде, в котором сейчас здесь установлены опции, нас все устраивает. Нажмем на кнопку ОК. Теперь мы видим начальную страницу для поиска. Но, в отличие от прошлого случая, когда у нас здесь было написано Предоставлена поисковой службой ***Msn***, сейчас здесь у нас уже установлено Предоставлена поисковой службой ***Rambler***. Давайте, попробуем еще что-нибудь поискать. Допустим, нам нужно найти Серия ***TeachPro***.

Теперь зададим поиск поисковой машине ***Rambler***. Сейчас у нас на экране — огромное количество ссылок. Отметим, что найдено 143 ссылки на веб-страницы, в которых упоминается сочетание Серия ***TeachPro***. Вот, например, здесь это сочетание явно присутствует. Куда бы мы сейчас не вошли, на экране мы увидим вот такое сочетание Серия ***TeachPro***. Войдем, к примеру, вот сюда. Сейчас здесь перед Вами на экране — реклама самоучителя ***Outlook*** 2000 для начинающих. Вот — эта реклама.



И здесь, как раз, присутствует выражение Серия ***TeachPro***. Иначе говоря, мы с Вами научились (правда, пока не очень уверенно и не очень эффективно) искать веб-страницы по какому-либо признаку, Если мы сейчас нажмем на кнопку Создать, то мы попадем на наш самый первый экран, где мы изначально и задаем категорию поиска. Это пока все, что Вам сейчас стоит знать по поиску ***Microsoft Internet Explorer***. Ну, а с этой кнопкой Избранное мы с Вами уже знакомы. Сейчас мы на нее нажали. Если Вы помните, здесь имеется папка, которую мы с Вами создали, и

ссылка Избранное, которую мы занесли сюда. Это мы делали на одном из наших первых уроках по браузерам.

Давайте, еще раз пройдемся по этим кнопкам. Здесь у нас имеется кнопка Добавить. Нажмем на нее. Эту страницу, которая сейчас у нас открыта в нашем браузере, мы можем также добавить в список Избранное. Причем, мы можем добавить в список Избранное в корень, то есть, ссылка на эту страницу у нас появится вот здесь, а можем добавить и в какую-нибудь из папок. Давайте, все-таки, добавим в папку Саша.

С помощью этого флажка можно сделать следующее. Мы можем сделать эту страницу доступной автономно. Что такое автономный режим работы, мы с Вами уже знаем. При установлении этой галочки эта страница будет записана на жесткий диск, и, не имя подключения к Интернету, мы сможем ее потом открывать. Что у нас произойдет при нажатии на кнопку Настройка? Это — Мастер автономного просмотра избранного. С помощью этого мастера мы можем увидеть то, что нам в данный момент доступно для автономного просмотра, то есть, то, что мы можем просматривать, не имея подключения к Интернет. Нажмем на кнопку — Далее.

Следующая страница. Именно ее мы и хотим определить, как доступную для автономного просмотра. Нам предлагается сделать следующее: Если это избранное содержит ссылки на другие страницы, хотите ли вы сделать их доступными автономно. Что это означает? Это означает следующее. Здесь имеются отдельные гиперссылки, при нажатии на которые мы будем переходить на следующую страницу. Если мы хотим те страницы, которые нам доступны по нашим ссылкам, также сделать доступными в автономном режиме, то все страницы, которые расположены по этим гиперссылкам, тоже будут сохранены на нашем жестком диске. Мы даже можем не посещать эти страницы. Они просто автоматически будут сохранены на диске.

Допустим, мы согласны с этим, и устанавливаем переключатель Да. Здесь нам предлагается загружать страницы на глубину. Что это значит? Это значит, что, если у нас здесь установлено "1", то у нас сохранится вот эта страница и те страницы, которые доступны по этим гиперссылкам. Если мы увеличим это число, то будут сохранены не только страницы, доступные по этим гиперссылкам, но и те страницы, которые доступны по гиперссылкам тех страниц, и на которые мы можем перейти отсюда.

Короче говоря, имеется возможность сохранения всей иерархической ветви на определенную глубину. Ну, здесь, конечное число — это 3. Больше, чем на глубину трех ссылок сохранять не имеет смысла. Ну, давайте, к примеру, установим именно 3 и посмотрим, что при этом произойдет Далее. Теперь нам предлагают синхронизировать эту страницу. Правда, мы с Вами пока еще не говорили про синхронизацию.

Но, если мы сохраняем нашу страницу на нашем жестком диске и регулярно читаем ее, то она у нас может просто устареть. Дело в том, что на самом деле по реальному веб-адресу эта страница может быть уже изменена, а у нас на нашем жестком диске показана еще старая версия. Если мы установим этот флажок, то по команде Синхронизировать, которая у нас будет доступна из меню Сервис, мы сможем периодически обновлять страницу на нашем жестком диске, то есть, получать, тем самым, последнюю информацию.

А еще это можно сделать автоматически, а не с помощью команды Синхронизировать, которую мы будем вызывать сами, когда вспомним об этом. Мы просто будем использовать новое расписание. Но мы сейчас пока не будем устанавливать этот флажок. А просто нажмем на кнопку Далее. Нужен ли пароль для доступа к этому узлу. Надо сказать, что некоторые узлы в Интернете защищены именем и паролем в отличие от других узлов. Для доступа к этому узлу никакого пароля не требуется. Мы, как Вы помните, пароль не вводили. Поэтому мы устанавливаем флажок вот здесь на Нет. И — Готово. Нажмем на кнопку ОК. Сейчас у нас происходит закачка этих страниц, которые мы задали, как Избранные.

Иначе говоря, сейчас вся ветвь пытается закачаться на наш жесткий диск, чтобы быть доступной в автономном режиме просмотра. При нажатии на кнопку Сведения нижняя часть данного окна может открываться и закрываться. Сейчас у нас происходит синхронизация. Но, так как на жестком диске у нас никакой такой информации нет, то, фактически, производится просто закачка информации, а не ее синхронизация (сравнение).

Но в данном случае этот процесс тоже называется синхронизацией. И на Вкладке Результаты мы можем посмотреть, имелись ли при этом ошибки, или все прошло благополучно. А, так как сейчас все это не завершено, то здесь, соответственно, и нет никакой информации. Если Вы обратите внимание, то увидите, что здесь происходит изменение вот этой строки.

Тут показывается, какой конкретно элемент в данный момент скачивается на жесткий диск и откуда. С помощью этих кнопок мы можем и пропустить какой-нибудь элемент, если он нас не интересует. Например, мы можем сейчас пропустить этот элемент. Мы его пропустили, окно у нас, соответственно, закрылось. Значит, ничего у нас на жесткий диск не закачалось. Но сейчас здесь мы видим вот эту новую ссылку ***CD ROM*** мультимедийный самоучитель.

Она у нас попала в список Избранное. Отлично. Значит, какая-то часть элементов с этой веб-страницы на наш жесткий диск, все-таки, сохранилась. Впоследствии при вызове команды. Синхронизировать, фактически, будет произведена докачка той информации, которую мы здесь выберем и нажмем на кнопку Синхронизация. Но сейчас мы не будем этого делать. Мы это

сделаем на уроке, посвященном меню Сервис. На этом мы завершим данный урок. На следующем уроке мы ознакомимся с этими двумя кнопками Медиа и Журнал.

6.3.2.5. Кнопки медиа, журнал

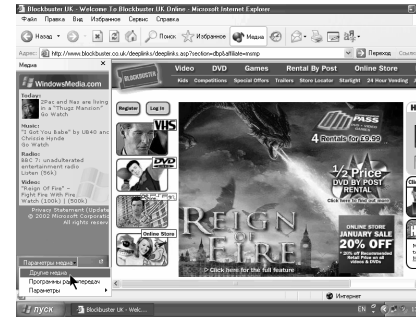
Сейчас мы с Вами перейдем к следующим двум кнопкам. Первая кнопка — это кнопка Медиа. Надо сказать, что в Интернете нам доступна не только информация с веб-страниц, но и множество мультимедийных возможностей. То есть, мы можем прослушивать радио через Интернет, музыку, смотреть видео фильмы. Все это мы можем делать, благодаря встроенной в ***Internet Explorer*** возможности работы с мультимедиа прямо изнутри нашего браузера. После того, как мы нажали на эту кнопку, в левой части окна открылось вот это подокно, которое, фактически, называется Медиа.

Давайте, более подробно ознакомимся с панелью Медиа. В нижней части этого окна у нас имеется вот такая панель, напоминающая панель ***Player***-а ***Windows***. Здесь у нас имеется вот эта кнопка. Это — кнопка Воспроизведения мультимедийного файла. Соответственно, рядом — прекращение воспроизведения мультимедийного файла, Перемотка файла (Назад и Вперед), Отключение звука.

Это у нас — микшер. Мы, фактически, можем поднимать уровень звука с помощью этого рычажка. Вы видите, что мы можем, например, немного уменьшить уровень звука в левой части микшера. Увеличим, например, уровень звука в правой части. С помощью этой кнопки мы можем задать, в отдельном ли окне нам запускать панель мультимедиа или нет. Если мы нажмем, то получим панель мультимедиа вот в таком виде и сможем таскать её вот таким образом по экрану. Если нажмем на кнопку еще раз, то эта панель будет аккуратно встроена в окно ***Internet Explorer***. Сейчас в этой части открыта страница ***WindowsMedia.com***. Нам предлагается сегодня прослушать или просмотреть информацию, которая расположена вот по этим ссылкам.

Предположим, мы хотим просмотреть вот этот видео файл. Давайте, нажмем на эту гиперссылку. Сейчас в отдельном окне у нас открылся медиа плейер, который пытается закачать вот этот видео фрагмент с Интернета. К сожалению, общая скорость подключения у нас достаточно низкая и все, что Вы будете видеть, будет производиться с серьезными задержками и недостаточной скоростью. Вот сейчас у Вас на экране проигрывается видео информация. С помощью вот этого рычажка мы можем переходить чуть вперед и чуть назад. К сожалению, из-за плохой связи с Интернет качество изображения не очень хорошее. Сейчас идет реклама художественного фильма, показываются фрагменты этого фильма.

Давайте, прекратим просмотр. Остановим его с помощью вот этой кнопки. Закроем это окно и посмотрим, что у нас еще имеется. Кстати, вот в этом раскрывающемся списке мы можем выбрать мультимедийный файл, который мы с Вами хотим послушать или посмотреть. Что здесь в Параметрах Медиа еще имеется? Мы можем перейти на Программы радиопередач или на другие медиа страницы. Предположим, нас интересуют другие радиопередачи. Сейчас перед нами открывается программа радиопередач, которые нам доступны. Вот здесь, например — Радиостанции.



Мы можем по любой теме выбрать радиостанцию. Предположим, нас интересует, к примеру, Альтернативный Рок. И — вот перед нами соответствующие радиостанции, которые это и передают. Давайте, еще раз выполним щелчок мыши. Предположим, нас интересует вот это (6 ***Music***). Точно так же мы можем сделать, например, вот здесь. Мы можем открыть следующую, и ее также добавить в список радиостанций. Сейчас наш медиа плэйер настраивается на эту радиостанцию и, фактически, начинает передачу с нее. К сожалению, Вы сейчас не можете услышать это, но звук на самом деле идет. Давайте, закроем это окно и посмотрим, что у нас здесь еще имеется. Откроем этот список и выберем Параметры. Здесь у нас еще имеется следующая возможность.

Мы можем Воспроизводить веб-медиа в панели. Что это значит? Если мы установим этот флажок и выберем, например, тот же видеоролик... Вы видите, что сейчас здесь открывается вот такой экран и именно в нем ***Internet Explorer*** попытается воспроизвести данный фрагмент. Это произошло именно потому, что мы здесь указали Воспроизводить веб-медиа в панели. Вот здесь в углу экрана мы сейчас видим этот рекламный видеоклип. Вот и все, что Вам необходимо знать о мультимедиа сейчас. Более подробно мы это рассмотрим на последующих уроках, а сейчас перейдем к следующей кнопке. Эта кнопка называется Журнал.

При нажатии на эту кнопку у нас в левой части окна появляется список посещенных нами ранее веб-страниц. Если мы нажмем мышью вот сюда, на Сегодня, то список Сегодня закроется, и мы увидим список самого высокого уровня, состоящий из двух пунктов. Предположим, мы хотим узнать, что мы посещали на Прошлой неделе. Вот здесь мы выполняем щелчок мышью и видим вот такой список веб-узлов, который мы с Вами посещали на прошлой неделе. Если мы, опять-таки, щелкнем мышью по любой из этих пунктов, то у

нас откроется более подробный подпункт, где уже непосредственно дан адрес веб-страницы. Если мы нажмем вот сюда, то в правой части окна у нас откроется данная веб-страница. Именно сейчас в правой части окна она у нас и пытается открыться. Если мы закроем этот список На прошлой неделе (список закрывается точно так же, как и открывается, щелчком мыши) и откроем список Сегодня, то, соответственно, увидим в той же последовательности тот список, который мы посетили сегодня. Управлять изображением этих списков мы можем с помощью раскрывающегося списка Вид. Сейчас у нас последовательность изображена По дате.

Мы можем, например, отобразить эту последовательность По узлу. Именно так и поступим? Мы получаем список узлов (вот наши узлы), данные в алфавитном порядке. То есть, начинается, естественно, с цифры (это была наша радиостанция), потом — ***a***, ***b*** и так далее. Можем выбрать и По посещаемости. Вот так. В той последовательности, в которой мы посещали эти узлы. Вот они. То есть, в зависимости от того, сколько раз мы посещали тот или иной узел. По порядку посещения — это, в какой последовательности мы посещали эти узлы, то есть, с чего мы начали. Вот так.

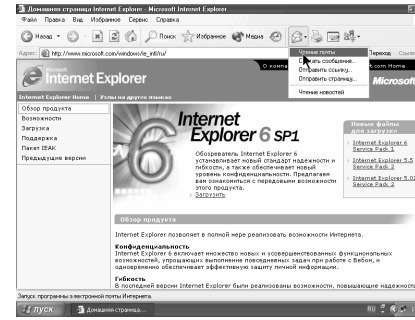
Давайте, все-таки, выберем По дате. Здесь, опять-таки, имеется два пункта: На прошлой неделе и Сегодня. Надо сказать, что элементы Журнала мы можем редактировать. Что тут можно сделать? Нажмем правую клавишу мыши и вызовем контекстное меню.

Данный пункт в Журнале мы можем удалить. Давайте, удалим его. Нас переспрашивают: Вы действительно хотите удалить запись журнала? Мы подтверждаем наше желание. И данный пункт уже в Журнале присутствовать не будет. Мы можем, кстати сказать, удалить весь пункт верхнего уровня. Правая клавиша мыши. Удалить. Согласимся. И теперь пункта — На прошлой недели, у нас уже нет. Мы с Вами его удалили.

Также с помощью контекстного меню можно. Развернуть узел и увидеть адреса уже конкретных страниц вот таким образом — Развернуть. Соответственно, для такого раскрытого узла, если мы вызываем контекстное меню, то эта команда превращается в команду — Свернуть. И, если ее выполняем, то, соответственно, весь узел собирается. Ну это все, что необходимо знать о Журнале ***Microsoft Internet Explorer***. Давайте, завершим этот урок, а на следующем уроке мы с Вами познакомимся с этими тремя кнопками.

6.3.2.6. Кнопки почта, печать, показать в виде HTML

На этом уроке мы продолжим знакомство с интерфейсом обозревателя ***Microsoft Internet Explorer***. Следующая кнопка, с которой мы сейчас познакомимся — вот эта. Она называется Почта. При нажатии на этот черный



треугольник открывается список команд, которые связаны с почтовым клиентом. Первая команда — это Чтение почты. Если мы ее выполним, перед нами откроется наш почтовый клиент. В данном случае это будет ***Outlook Express***. Подробно работе с почтой у нас будет посвящен целый раздел. А сейчас мы просто убедимся в том, что у нас вызывается почтовый клиент ***Outlook Express*** при выполнении команды Чтение почты. Давайте, переключимся на наш ***Outlook Express*** и закроем его. Следующая команда — это Создать сообщение. Прямо из нашего обозревателя ***Microsoft Internet Explorer*** мы можем послать письмо. Создаем сообщение.

Появляется соответствующее окно, которое так и называется Создать сообщение. Здесь мы можем указать Адрес того получателя, которому мы хотим послать сообщение, Тему письма и написать само письмо. Давайте, напишем адрес... Допустим, мы хотим послать это письмо Сергею... ***mail ru***. Темой письма будет просто слово Сообщение. А письмо само будет содержать только одно слово — Привет.

После того, как мы нажмем на кнопку — Отправить, наше сообщение будет отправлено. Ну, давайте сейчас закроем это окно. ***Outlook*** делает попытку отправить сообщение. Продолжим изучение вот этого списка команд. Если мы нашли в Интернете что-нибудь интересное, какую-нибудь интересную страницу, на которой имеется определенная информация, мы можем отправить нашим друзьям либо ссылку на нее, либо саму страницу.

Давайте, выполним команду Отправить ссылку. Опять-таки, появляется окно Отправка. Здесь также можем указать адрес получателя, то есть, того адресата, которому мы хотим отправить данное сообщение. Пусть это также будет ***sergey mail ru***. Но здесь в качестве вложения имеется адрес на ту страницу, на которой мы в данный момент находимся. Это — адрес той страницы, которая сейчас открыта в нашем браузере. И здесь имеется информация о том, что файлы вложены в сообщение и готовы к отправке. При нажатии на кнопку Отправить вот эта информация об адресе уйдет с нашей машины.

Давайте, закроем это окно и выполним следующую команду — Отправить страницу. В отличие от предыдущей команды, в качестве вложения здесь будет дана не просто ссылка, а вся наша страница, которая в данный момент открыта в нашем обозревателе. Сейчас перед нами открылось диалоговое окно

Отправки. Оно похоже на прошлое окно, которое было открыто перед выполнением предыдущей команды, но содержит в себе уже всю страницу, которая в данный момент активна в нашем браузере. К сожалению, здесь просто немного не соответствует кодировка.

Кодировку можно выправить с помощью меню Формат. Нам достаточно здесь ее немного подправить. Давайте, выберем, например, кодировку, которая называется Кириллица (***Windows***). К сожалению, выбранная кодировка в данный момент не помогла: видимо, мы неверно указали кодировку. Сейчас это и не важно. Давайте, здесь наберем адрес получателя... И снова нажмем на кнопку Отправить. На этот раз будет отправлена вся страница целиком. Давайте, закроем это окно, чтобы оно не занимало на экране места. Это все, что было связано с кнопкой Почта. Следующая кнопка — это кнопка Печать. При нажатии на эту кнопку будет распечатана та страница, которая в данный момент активна в Вашем браузере. Принтер должен быть предварительно установлен в Вашей системе.

Следующая кнопка называется Править в блокнот. Дело в том, что гипертекст — это, фактически, компьютерная программа, которая в большинстве случаев бывает написана на языке, который называется ***HTML***. Если мы выполним эту команду — Править в блокнот, перед нами откроется веб-страница, которая активна в данный момент, но на этом самом языке (языке ***HTML***) и вот в таком виде. Если мы хотим сохранить эту страницу на наш компьютер или сделать в ней кое-какие изменения (то есть, изменить определенные команды ***HTML***), то мы имеем эту возможность. Можно это сделать здесь, а затем уже Сохранить ее на жесткий диск. Эта возможность на самом деле очень удобна, если Вы создаете свои собственные веб-страницы и работаете с ними. Обзорно мы с Вами изучим язык ***HTML*** в дальнейшем, а сейчас просто закроем это окно и запомним, что такая возможность в нашем браузере имеется.

Ну, и последняя кнопка, — это кнопка Кодировка. Мы ее сами установили на панель инструментов на одном из прошлых уроков. Вы это помните. В данный момент мы не будем ее рассматривать подробно, так как уже знаем, как пользоваться ею. При открытии списка с помощью этой кнопки открывается список всех имеющихся кодировок, который поддерживает наш браузер. И, если это необходимо, мы можем выбрать любую кодировку из этого списка. А пока мы закроем этот список и на этом завершим наше знакомство с панелью инструментов. На следующем уроке мы с Вами начнем изучение меню браузера ***Microsoft Internet Explorer***.

6.3.2.7. Главное меню. Меню Файл

На этом уроке мы с Вами приступим к изучению главного меню браузера ***Microsoft Internet Explorer***. Вот — наше главное меню. Надо сказать, что

главное меню — это традиционное средство для работы с приложениями. Оно обычно дает полный доступ ко всем возможностям нашего браузера. Главное меню нашего браузера содержит следующие пункты. Первое — это пункт Файл. При выполнении здесь левого клика мыши открывается вот такой набор команд, с помощью которых мы работаем с нашим файлом. Следующий пункт — Правка. Здесь сосредоточены операции правки нашего документа. Далее следует меню Вид. С помощью этого меню мы работаем с интерфейсом нашего браузера: мы указываем шрифты, его внешний вид и так далее.

Избранное. Здесь мы работаем с Избранным. Сервис. С помощью меню Сервис обычно производятся определенные настройки нашего браузера. Ну, и, наконец, меню Справка. Здесь сосредоточены команды по работе со справочной системой. Если Вы обратите внимание, то заметите, что очень многие команды меню имеют, так называемые, подчеркнутые символы. Если, к примеру, мы сейчас нажмем на кнопку ***Alt***, то очень многие команды у нас будут подчеркнуты в определенных своих буквах вот таким образом. Что же это означает? Это означает, что при нажатии сочетания клавиш ***Alt*** и соответствующей буквы данная команда будет выполнена.

Попробуем сделать это. Для начала переключимся на русский шрифт и нажмем на кнопку ***Alt*** и Ф одновременно. Как Вы успели заметить, у нас открылось меню Файл. Открылось именно это меню, потому что мы нажали сочетание горячих клавиш ***Alt***+Ф. Вот здесь подчеркнуто именно Ф. Если бы мы вместо ***Alt***+Ф нажали ***Alt***+П, то открылось бы меню Правка, что и произошло. Аналогично при нажатии ***Alt***+И открывается меню Избранное. Перейдем к первому пункту нашего меню, к меню Файл. Вот — оно. В позиции Файл главного меню сосредоточены основные операции по работе с файлами (мы с Вами об этом уже сказали).

Первая команда, которая здесь имеется, это команда Создать. При наведении курсора мыши на нее у нас открывается подменю, в котором мы и выбираем, что конкретно мы хотим создать. Мы можем создать Новое окно. Давайте, выполним эту команду. Перед нами открывается второе окно обозревателя ***Internet Explorer***. Вы видите, что на Панели задач у нас сейчас активны два окна ***Internet Explorer***. Закроем одно из них и продолжим изучение. Мы можем создать: Сообщение, Контакт, Вызов через Интернет. Ну, команда Сообщение идентична команде, которая вызывается при нажатии этой кнопки. Нажали.

Появилось соответствующее окно, которое так и называется Создать сообщение. Мы его закроем, так как с этим окном мы с Вами уже работали. Далее — Создать контакт.

Что такое Создать контакт? Это, фактически, запись в адресную книгу. Как это делается, Вы узнаете в соответствующих разделах нашего курса. Далее — Вызов через Интернет. С помощью этой команды мы вызываем абонента для проведения телеконференции. Что такое телеконференция, Вы также узнаете позже. Вы научитесь пользоваться с той программой, которая сейчас запустилась. Эта команда называется ***NetMeeting***. С помощью этой программы Вы можете общаться с другими пользователями Интернет.

Мы сейчас закроем это окно, а Вы просто запомните, что с помощью этой команды Создать вызов через Интернет запускается ***NetMeeting***. А теперь перейдем к следующей команде, которая называется Открыть. С помощью этой команды открывается окно поиска нужного файла или адреса. Выполняем эту команду. Здесь нам предлагается ввести адрес документа или папки в Интернете. Мы можем ввести адрес прямо здесь.

Введем, например, ***www.rambler.ru***... Нажмем на кнопку ОК. Перед нами сейчас открывается ***rambler.ru***. А можно сделать и следующее. Выберем Файл / Открыть и нажмем на кнопку Обзор. Теперь уже можно найти файлы, которые сохранены на нашем жестком диске. Как мы с Вами говорили, веб-страницы в основном представляют собой файлы ***HTML*** и они-то, собственно, у нас должны открываться. Давайте, выберем здесь недавние документы. Например, вот тут у нас имеется такая ***HTML***-страничка (***HTML***-файл).

Давайте, попробуем открыть его. Нажмем здесь на кнопку ОК. Перед нами открылся вот такой файл, который мы предварительно сохранили. Мы, все же, вернемся куда-нибудь еще и откроем здесь, к примеру ***Yandex***. С помощью этого диалогового окна мы можем открывать файлы, как из Интернет, так и с нашего жесткого диска. Кроме того, здесь имеется еще одна очень интересная возможность. Мы еще раз выполним команду Открыть и увидим этот список вот с таким треугольником, напоминающим галочку. Если мы щелкнем по этой галочке, то у нас откроется весь список посещенных нами веб-страниц, который открываются при нажатии на эту галочку. Но мы нажмем на кнопку Отмена.

Сейчас мы никуда переходить не будем, а просто продолжим освоение меню Файл. Следующая команда называется Править в блокнот. Эта команда идентична команде, которая выполняется при нажатии этой кнопки. Если мы выполним эту команду, то перед нами откроется та страница, которая активна сейчас, но уже в формате ***HTML***. Закроем пока это окно. Мы сейчас не будем этим заниматься. Следующие команды — это команды Сохранить и Сохранить как. Данная страница у нас открыта с Интернет. Мы можем

сохранить ее на наш жесткий диск. Давайте, попытаемся сделать это. Она у нас будет называться ***Yandex*** и будет располагаться в папке Мои документы. Диалоговое окно, которое Вы сейчас видите перед собой, идентична окну, которое вызывается при сохранении объектов в ***Windows XP***. Мы хотим сохранить в папку Мои документы. Для этого мы выполняем команду

Сохранить. Сейчас производится сохранение веб-страницы. Все необходимые элементы скачиваются на наш жесткий диск. Эта страница у нас будет доступна при автономной работе. Все элементы будут отображены на нашем экране. Чем это удобно? Мы можем, например, открыть эту страницу прямо с нашего Винчестера вот таким образом. ОК. Сейчас наша страница открыта на экране с нашего жесткого диска. И при отключенном подключении к Интернет мы, фактически, сможем ее изучать. Следующие три команды связаны с распечаткой нашей страницы. Первая команда Параметры страницы выводит окно с параметрами текущей страницы. Предположим, мы хотим распечатать нашу страницу на бумаге соответствующего размера. Размер бумаги указывается с помощью этого списка. Допустим, нас интересует бумага не формата А4, а формата ***Letter***. Подача. С какой кассеты, откуда подавать бумагу на наш принтер.

Принтер может затягивать бумагу со своей обычной кассеты, но мы можем подавать ему бумагу на вход и вручную. Допустим, что нас интересует именно ручная подача бумаги. Кроме того, при распечатке можно указать колонтитул. Колонтитул — это определенная информация, которая выводится на каждую страницу при распечатке соответствующего документа и которая располагается в нижней или в верхней части нашего листа. В данном случае здесь указано, что верхний колонтитул должен содержать имя нашей веб-страницы, номер печатаемого листа и общее количество листов.

Иначе говоря, если мы имеем, например, пятистраничный документ, то на первой странице будет распечатано 1 из 5-ти, на второй — 2 из 5-ти и так далее. Аналогично дело обстоит и с нижним колонтитулом. Здесь еще имеется Ориентация бумаги. Ориентация бумаги может быть Книжной (такой, как в данный момент) и Альбомной. Если мы сейчас выберем Альбомную ориентацию бумаги, то, как видите, бумага у нас развернется, и документ будет распечатан не по длине бумаги, а, фактически, по ее ширине. Восстановим Книжную ориентацию бумаги. В этих полях указаны отступы полей на бумаге, то есть, вот эти расстояния: Слева, Справа, Сверху и Снизу, соответственно.

При нажатии на кнопку Принтер появляется диалоговое окно, в котором мы можем выбрать соответствующий принтер, если их имеется несколько. В данном случае мы имеем только один сетевой принтер, который установлен на компьютере ***Apollon***. Это принтер ***LaserJet*** 4***ML***. Выбор у нас, собственно,

небольшой. Нажмем на кнопку Отмена и закроем это диалоговое окно нажатием на кнопку ОК. Рассмотрим следующую команду. Следующая команда — это уже команда Печать. Ну, страницу мы с Вами уже настроили и знаем, как будет выглядеть наш документ при печати.

При выполнении команды Печать появляется диалоговое окно, с помощью которой и происходит управление печатью. То есть, печать производится именно на этот принтер. Количество печатаемых листов указывается здесь. Печатаются либо все страницы, либо например, определенное количество страниц, предположим с первой по третью. Здесь производится Настройка нашего принтера, а тут указывается число копий. Допустим, мы хотим распечатать три копии данного документа. С помощью флажка Разобрать по копиям мы указываем, в какой последовательности необходимо распечатать эти копии.

Если, например, этот флажок сброшен, то сначала будут распечатаны три экземпляра первой страницы, затем — три экземпляра второй страницы и, соответственно, три экземпляра третьей страницы. Если же этот флажок установлен, то будет распечатана последовательно сначала первая копия документа, затем — вторая, а следом — третья копия, соответственно. Нажмем на кнопку Отмена и продолжим изучение меню. Следующая команда — это Предварительный просмотр. С помощью команды Предварительный просмотр мы можем увидеть, как будет выглядеть наш документ после того, как мы его распечатаем. Вот это — лист бумаги. Выглядеть он будет именно вот таким образом. Мы имеем всего две страницы. Как видите, с помощью этих стрелочек мы можем переходить в ту или другую сторону. Мы можем видеть сначала первую страницу, затем — вторую. Здесь в верхнем колонтитуле, как мы с Вами и установили, будут распечатаны номер страницы, общее количество страниц и название соответствующего сайта. Окно Предварительный просмотр идентично стандартному окну ***Windows***-а ***XP***. Закроем этот экран и изучим следующие пункты меню Файл.

Здесь имеется команда Отправить. С помощью этой команды выводится подменю с указанием источника (то есть, куда и что мы отправляем). Мы можем Отправить Страницу по электронной почте и Ссылку по электронной почте. То же самое мы с Вами делали с помощью вот этой кнопки. Мы можем также отправить. Ярлык на рабочий стол. Как это делается? Давайте, попробуем выполнить эту команду. Итак, мы сейчас отправили Ярлык на рабочий стол. Теперь, если мы закроем это окно (закроем и окно ***rambler***-а), то на нашем рабочем столе появится вот такая иконка, которая, фактически, содержит в себе ссылку на эту веб-страницу. Мы, фактически, отправили на рабочий стол ссылку (ярлык). Следующая команда — это Импорт и Экспорт.

Попробуем выполнить эту команду. При выполнении данной команды вызывается Мастер импорта и экспорта. С помощью этого мастера производится импортирование и экспортирование данных из ***Internet Explorer***, например, в другие приложения или в файлы. Мы можем отправлять и принимать Избранное или файлы ***Cookie***. Что представляют собой файлы ***Cookie***, мы с Вами узнаем позже. А сейчас пока мы с Вами незнакомы с этим понятием. Но импортировать и экспортировать эти файлы мы сможем с помощью именно этого мастера. Нажмем на кнопку Далее.

Здесь необходимо указать ту команду, которую мы, собственно, и хотим выполнить. Мы хотим выполнить Импорт Избранного, Экспорт Избранного, Импорт файлов ***Cookie*** и Экспорт файлов ***Cookie***. Допустим, мы сейчас хотим выполнить Экспорт Избранного. Нажмем на кнопку Далее. Здесь мы имеем весь список Избранного, который у нас имеется на нашем компьютере. Предположим, мы хотим экспортировать только папку Саша. Далее. Нам сообщают, что Избранное можно экспортировать в другой веб-обозреватель или в файл. Нам предлагается экспортировать его вот в такой файл, который называется ***Bookmark.HTML***. Допустим, нас это устраивает. Если этот файл не устраивает, то всегда можно нажать на кнопку Обзор и указать другой, куда необходимо экспортировать Избранное. Но в данном случае файл ***Bookmark*** нас вполне устраивает. Нажимаем на кнопку Далее.

Итак, нам сообщается, что мастер Импорта и Экспорта завершил свою работу. Нажимаем на кнопку Готово. Еще раз сообщается о том, что Экспорт прошел успешно. Теперь у нас в папке Мои документы уже имеется файл, который называется ***Bookmarks***. Убедимся в этом. Попробуем открыть этот файл. Мы убеждаемся в том, что такой файл действительно имеется. Открываем его и получаем файл вот в такой странной форме. Вы не удивляйтесь этой форме: это сейчас не столь важно. Важно, что при импорте этого файла в какой-нибудь другой браузер наши Избранные данные скопируются. Перейдем Назад на наш предыдущий экран и продолжим изучение меню Файл. Ну, с Импортом и Экспортом мы уже разобрались.

Следующая команда — это Свойства. При выполнении этой команды выводится окно со свойствами текущей страницы, где указаны Протокол, Тип документа, Тип соединения, Адрес, Размер и так далее. Нажмем на эту кнопку Сертификаты. Здесь нам сообщается, что данный тип документа не имеет сертификатов безопасности. Что такое сертификаты, мы с Вами узнаем позже, а сейчас мы просто имеем сообщение о том, что данный тип документа не имеет сертификатов безопасности. Закроем это окно и окно Свойства. Последняя команда — это команда Закрыть. Выполнение этой команды приводит к закрытию окна ***Internet Explorer*** версии 6.0. Попробуем выполнить эту команду. Вы видите, что наш браузер закрылся. На этом мы завершим этот

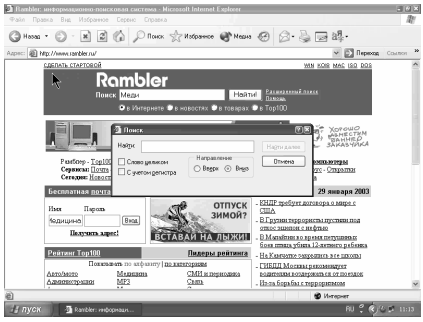
урок, а на следующем уроке мы продолжим изучение главного меню этого браузера.

6.3.2.8. Главное меню. Меню Правка

На этом уроке мы рассмотрим следующий пункт меню, который называется Правка. В первой части этого меню сосредоточены команды, которые служат для операций с Буфером обмена ***Windows***, так называемым, ***Clipboard***-ом. Это команды: Вырезать, Копировать и Вставить. Ну, давайте попробуем поэкспериментировать с этими командами. Перед тем, как что-нибудь скопировать в Буфер обмена, необходимо сначала выделить копируемый элемент. Найдем и выделим здесь любой фрагмент текста.

Допустим, нам нужно выделить это слово Медицина. Выделяем. Делается это нажатием клавиши мыши в начале пункта выделения и последующим протягиванием до конечного пункта выделения. Затем клавиша мыши отпускается. Выделили. А теперь выбираем меню Правка. Вы можете убедиться в том, что команда Копировать здесь уже стала активной, то есть, у нас уже есть, что заносить в буфер обмена ***Windows***. Итак, мы выполняем команду Копировать. После того, как мы с Вами скопировали выбранный текст, мы можем сделать следующее.

Мы можем установить курсор в какой-нибудь области. Обычно курсор можно устанавливать вот в таких полях для заполнения, например, вот сюда. Мы выполнили здесь щелчок мышью — и здесь уже установлен текстовый курсор. Это значит, что сюда мы уже можем вставить определенную информацию, например, из буфера обмена. Еще раз выберем пункт меню Правка. Мы убеждаемся в том, что команда Вставить теперь активна. Если мы выполним эту команду, то слово Медицина будет вставлено в это поле.



Итак, оно вставлено сюда из буфера обмена. Команда Вырезать аналогична команде Копировать. Единственное отличие заключается в том, что при ее выполнении информация исчезает с того места, откуда мы ее, собственно, и вырезаем. Что это значит? Давайте, выделим часть слова Медицина. Выделим, например, Цина и выполним команду Вырезать. Вы можете убедиться в том, что вторая часть слова Медицина попросту исчезла с экрана. Если мы установим курсор в другом месте, например вот здесь в поле Имя, и выполним команду Вставить, то именно этот фрагмент текста и вставится из буфера обмена. Иначе говоря, он и был помещен в Буфер обмена, но при этом исчез с экрана. А, если мы просто

выполним команду Копировать... Нет, давайте, для наглядности выделим этот фрагмент и выполним команду Копировать. На экране слово Меди осталось. Устанавливаем курсор сюда и выполняем команду Вставить. Мы убеждаемся в том, что первая часть слова Медицина успешно вставлена из буфера обмена в это поле Имя.

Пойдем дальше. Меню Правка. Следующая команда — Выделить все. Что она означает? Если мы выполним команду Выделить все, то вся информация, которая находится у нас на нашей веб-странице, будет выделена. Если мы после этого выполним команду Копировать, то вся эта информация будет помещена в буфер обмена (решительно вся информация нашего экрана). Давайте, снимем выделение. Для этого достаточно щелкнуть мышью в любой области экрана вот таким образом. Как видите, выделение снято. Теперь продолжим изучение меню Правка. Последняя команда этого меню — Найти на этой странице. Что это значит? Выполним эту команду. Перед нами появляется диалоговое окно для поиска информации на данной веб-странице. Допустим, нам нужно найти, к примеру, вот это слово Школы. Для этого нам нужно ввести это слово вот сюда.

Теперь попробуем его найти. С помощью этих двух переключателей указывается направление поиска. Первое направление — это Вниз, второе направление — Вверх. Что это значит? Это значит, что, если у нас здесь выбрано направление Вниз, то будет просматриваться информация вот отсюда (с начала страницы) и в этом направлении, фактически, Вниз. Итак, давайте, найдем слово Школы. Это слово у нас найдено. Попробуем повторить поиск. Нам сообщается о том, что поиск в документе завершен. Больше на нашей странице слова Школы не имеется. Согласимся с этим утверждением.

А теперь попробуем спуститься по нашей веб-странице и щелкнуть мышью где-нибудь вот здесь. Попытаемся снова возобновить поиск искомого слова. Вы видите, что слово Школы у нас опять найдено. То есть, поиск, опять-таки, был произведен сверху вниз. Еще раз пробуем выполнить Найти Далее. Слово Школа больше не встречается. Теперь мы попытаемся организовать поиск слова, которое на этой странице повторяется несколько раз. Вероятнее всего, тут несколько раз будет повторяться слово Рамблер. Заодно мы с Вами это проверим.

Введем сюда слово Рамблер... и нажмем Найти Далее. Вот — первое слово Рамблер. Вот оно у нас. Теперь — Дальше. И — найдено второе такое слово вот здесь. Еще раз. Вот — Рамблер в мире. О Рамблере. Кстати, с помощью этого слова Рамблер мы с Вами сейчас почувствуем разницу в поиске Слова целиком и без установления вот этого флажка (то есть, не целиком). Еще раз. Еще один Рамблер. И — еще один. Все. Теперь у нас поиск в документе уже завершен. Давайте, попробуем поменять направление поиска: Вверх. Как

сейчас ***Internet Explorer*** будет производить поиск слова Рамблер на нашей веб-странице? Найти Далее. Вот это слово. Дальше. Фактически, поиск производится снизу вверх: с конца нашей веб-страницы до ее начала. То есть, это слово мы нашли первым. Следующее — здесь. Затем мы, видимо, перейдем сюда. Проверим это.

Вот так. Все правильно. Сюда. Где у нас...? Вот. Еще раз — сюда. И — все: Поиск в документе завершен. Теперь, давайте посмотрим, что означает флажок Слово целиком. Этот флажок означает следующее. Если у нас этот флажок не установлен, то слово Рамблер ищется не полностью, то есть, его падежные разновидности, к примеру, Рамблеру и Рамблера тоже будут найдены. Но, если мы установим этот флажок, то будет разыскиваться только это слово целиком, то есть, при условии, если здесь и здесь, как бы, ничего нет. Итак, если мы установим этот флажок, то слово Рамблеру в этом случае найдено не будет. Проверим это. Вот здесь. Идем снизу вверх. Рамблер... Обсудить Рамблер. И сейчас мы убедимся в том, что в слове О Рамблер***е***, этот фрагмент Рамблер найден не будет. Давайте нажмем Найти Далее.

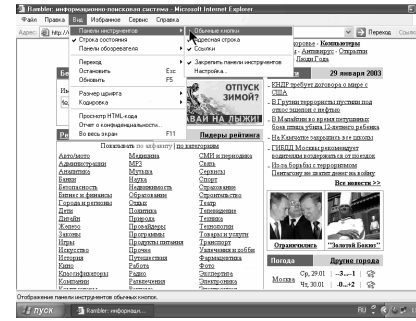
Вы видите, что слово О Рамблер***е*** проигнорировано. Это произошло по той причине, что здесь установлен флажок Слово целиком. Следующий флажок — С учетом регистра. Что означает этот флажок? Дело в том, что слово может быть набрано, как прописными, так и строчными буквами. Если при этом данный флажок не установлен, то ***Internet Explorer*** ищет, в общем-то, совпадение, вне зависимости от того, прописными или строчными буквами набран Рамблер. Давайте, убедимся в этом. Наберем слово Рамблер строчными буквами и возобновим поиск. Найти далее. Как видите, все слова Рамблер отмечаются.

Но, если здесь мы установим флажок — С учетом регистра, то уже ни один из тех вариантов, который был найден ранее, найден не будет. Действительно: нам сообщается, что поиск в документе завершен. Еще раз. Короче говоря, ***Internet Explorer*** ничего не может найти, так как мы ищем полное совпадение с учетом регистра. На экране должна присутствовать надпись Рамблер с маленькой буквы, чтобы мы нашли что-нибудь при установленном этом флажке. Теперь закроем это окно.

С поиском мы с Вами, вроде, уже разобрались. Вот все, что Вам нужно знать о пункте меню Правка. Если Вы обратите внимание, то заметите, что рядом с каждой командой имеется комбинация горячих клавиш. То есть, эти команды можно выполнить и с помощью клавиатуры без участия мыши. Попробуем это сделать. Например, Выделить все — это ***Ctrl+A***, а Найти на этой странице — это ***Ctrl+F***. Постараемся пока не забыть это. Теперь воспользуемся сказанным.

Нажимаем комбинацию клавиш ***Ctrl+A*** и убеждаемся в том, что у нас выделилось все на нашем экране. Снимем выделение и воспользуемся комбинацией клавиш ***Ctrl+F***. Перед нами возникло диалоговое окно Поиск, которое мы до этого вызывали только с помощью мыши. Закроем это окно и на это завершим этот урок. На следующем уроке мы рассмотрим меню Вид.

6.3.2.9. Главное меню. Меню Вид



На этом уроке мы рассмотрим следующий пункт меню, который называется Вид. Итак, мы с Вами уже говорили и демонстрировали, что интерфейс нашего браузера является настраиваемым. И в этом пункте меню Вид сосредоточены команды, которые позволяют менять вид интерфейса. В первую очередь, это опции управления нашими панелями. Сначала идет панель инструментов. Мы с Вами уже знаем, что, если сбрасывать вот эти флажки, то соответствующие элементы будут просто исчезать с нашего экрана. Как видите, мы установили Обычные кнопки обратно, и у нас появилась наша панель инструментов. Мы можем вот таким образом убрать нашу Адресную строку. Вот так.

Мы можем также восстановить ее на нашем экране. Аналогично можем поступить и со Ссылками (вот так). Ну, и Закрепить панель инструментов. Как мы помним, если сбросить этот флажок, то мы сможем свободно передвигать наши панели инструментов. Давайте, сделаем это еще раз. Поменяем, например, Меню и Панель инструментов местами. Вот так. Можем спустить Адрес. Можно и восстановить все на свои места.

Вот так. Еще раз. Но после того, как мы все переставили, нужно выполнить команду Закрепить панели инструментов. Вот так. Хотя, то, что мы сделали сейчас, получилось не очень удачно. Давайте, сбросим этот флажок и вернем Ссылки вот сюда, а затем еще раз выполним команду Закрепить панели инструментов. Ну, Вы помните, что выполняет команда Настройка. Она выводит такое диалоговое окно, с помощью которого можно добавлять и удалять кнопки с панели инструментов. Сейчас мы этого повторять не будем, мы закроем это окно и продолжим изучение меню Вид. С помощью команды Строка состояния, если сбросить этот флажок, можно убрать Строку состояния с экрана. Сделаем это. Как видите, здесь внизу экрана уже нет Строки состояния. Теперь восстановим ее. Отлично. Пойдем дальше. Панель обозревателя. Если мы установим, например, флажок Меню Поиск, то в левой

части окна появится окно поиска, с которым мы уже знакомы. С этим окном мы уже работали при изучении панели инструментов. Еще раз — Вид \ Панель обозревателя, уберем Поиск.

Мы можем вызвать и Журнал, но с ним мы также работали при работе с панелью инструментов. Мы нажимали вот эту кнопку, если Вы помните. Аналогичным образом можно и убрать панель Журнал. Но имеется и такая полезная команда, которая называется Полезный совет. Здесь ***Internet Explorer*** рассказывает нам о новых возможностях данной версии (версии 6.0). Вот здесь нам сообщают: Знаете ли Вы, что можно указать, принимать ли файлы ***Cookies*** для каждой зоны безопасности. Что такое файлы ***Cookies***, мы с Вами пройдем позже в соответствующем разделе. В общем-то, это достаточно полезная информация.

Можно нажать Следующий совет. И снова: Знаете ли Вы, что можно изменить порядок ярлыков на панели Ссылок, перетаскивая их. Вот наша панель Ссылок. Здесь у нас находятся ярлыки, и мы можем их просто перетаскивать. Можно вот таким образом захватить их мышью и перетащить, например, вниз. Вот эта черная указка и показывает, куда конкретно мы перетаскиваем данный ярлык. Отпускаем и видим, что ярлык ***Windows*** оказался перед ярлыком Настройки ссылок. Аналогично можно поступить и с Бесплатной почтой ***Hotmail***. Потянули. И — вот результат. То есть, этот совет оказался полезным: мы воспользовались им. Можно узнать и следующий совет. Здесь нам сообщают, что изменить цвета ссылок на веб-страницах можно, воспользовавшись меню Сервис, выбрать пункт Свойства обозревателя и нажав на кнопку Цвета.

Но этим мы с Вами будем заниматься позже, когда будем изучать меню Сервис. А сейчас, давайте, уберем Полезный совет с экрана, мы уже узнали, что это такое. Итак, мы убрали Полезный совет. Очень хорошо. Следующие три команды относятся к навигации наших веб-страниц. Подменю Переход. В свое время это подменю Переход представляло собой отдельный пункт меню. Так было, например, в ***Internet Explorer***-е версии 4.0. А сейчас команда Переход включена, как подменю Вид. Рассмотрим, что здесь имеется. Это — кнопки Назад и Вперед. Они аналогичны тем кнопкам, которые мы видим на панели инструментов: вот эти — Назад и Вперед.

Если мы сейчас нажмем на кнопку Назад, то вернемся на предыдущий экран нашего ***Internet Explorer***-а, то есть, на предыдущую страницу, которая у нас была открыта. Но сейчас мы нажимать на эту кнопку не будем, потому что мы с Вами уже достаточно хорошо изучили эти кнопки Назад и Вперед. Следующий пункт — это Домашняя страница. Выполнение этой команды аналогично нажатию вот на этот Домик (Домой). При выполнении этой команды производится переход на ту страницу, которая определена, как

основная (домашняя) в нашем обозревателе. Нажмем на эту кнопку и посмотрим, что при этом произойдет. Мы с Вами перешли на страницу ***Msn.com***, которая, как Вы помните, была загружена в качестве основной нашей веб-страницы. Продолжаем изучение меню Вид. Его подменю — Переход.

Здесь имеется список из трех страниц, которые были открыты нами последовательно с момента запуска ***Internet Explorer***. У нас была открыта страница ***rambler***, Вы это помните. Затем мы перешли на домашнюю страницу нашего ***Internet Explorer***-а (это — Добро пожаловать на сайт ***Msn.com***). А перед этим страница не была полностью открыта, поэтому у нас больше нет страницы для отображения. Вот — список из этих трех страниц.

Именно эти страницы у нас были открыты с момента запуска ***Microsoft Internet Explorer***. Далее — команды Остановить и Обновить. Что представляют собой эти команды, Вы также знаете. Они идентичны вот этим командам на панели инструментов: Остановить загрузку (если Вы передумали загружать какую-нибудь страницу) и Обновить (то есть, продолжить или загрузить страницу заново). Размер шрифта и Кодировка.

С этими командами Вы также знакомы. Если Вы помните, при работе с панелью инструментов, мы увеличивали и уменьшали размеры шрифта и меняли кодировку нашего документа. Поэтому мы с Вами сейчас этого делать не будем, мы это уже достаточно подробно изучили. Просмотр ***HTML*** кода. Эта команда практически аналогична вот этой команде, которая называется Править в блокнот. Если мы выполним эту команду, то увидим код нашей веб-страницы на языке ***HTML***, то есть, на языке гипертекста. Закроем сейчас это окно и продолжим дальше изучение меню Вид.

Следующая команда — Отчет о конфиденциальности. Мы сейчас не будем подробно изучать эту команду и не будем рассматривать это диалоговое окно, которое появляется при выполнении данной команды. Дело в том, что все это связано с файлами ***Cookies***, которые мы с Вами будем изучать позднее в соответствующем разделе настройки безопасности. Вот тогда мы и вернемся к этому диалоговому окну Отчет о конфиденциальности.

А сейчас мы это окно просто закроем. Ну, и последняя команда, которую мы видим перед собой, это — команда Во весь экран. Если мы ее выполняем, то меню, заголовок окна и строка состояния исчезают с нашего экрана. На экране остается только панель инструментов (вот она). При этом, мы получаем возможность на полном экране увидеть нашу веб-страницу.

Если мы вот здесь нажмем на эту кнопочку, то все, соответственно, и восстановится. Давайте, еще раз выполним эту команду Вид \ Во весь экран. С помощью этих трех кнопок мы можем управлять окном. Мы можем его Минимизировать, Восстановить в размере (мы это уже делали однажды вот

так: еще раз откроем во весь экран), а можем и просто закрыть это окно вот таким образом. На этом мы завершим изучение меню Вид. Остальные пункты этого меню мы изучим на последующих уроках.

6.3.2.10. Главное меню. Меню Избранное

Продолжим изучение главного меню браузера ***Microsoft Internet Explorer*** версии 6.0. Следующий пункт меню, который мы с Вами еще не изучали, это пункт меню, который называется Избранное. Ну, как Вы уже знаете, наш браузер позволяет организовать списки избранных страниц, которые хранятся в специальной папке или в ряде других папок. Как работать с этим списком, мы с Вами подробно уже обсуждали, когда изучали работу с этой кнопкой Избранное. Давайте, откроем это меню и посмотрим, какие команды в нем нам доступны.

Первая из операций для работы с избранными страницами — это Добавить в Избранное. Если мы выполним эту команду, то перед нами откроется диалоговое окно для подтверждения добавления в список избранных страниц

имени текущей страницы. Текущая страница у нас называется Рамблер — информационно-поисковая система. С этим диалоговым окном мы с Вами уже знакомы и умеем с ним работать. С помощью этого флажка мы устанавливаем доступ к нашей странице. Этот доступ может быть ***Online***, то есть при наличии подключения к Интернету. Вот сейчас у нас это подключение имеется. Вы можете видеть здесь вот эти два компьютера. Мы можем установить доступ к этой странице, автономно, установив этот флажок. Можно и Настроить автономный доступ к данной странице. Сейчас перед Вами — Мастер автономного просмотра избранного. С этим мастером мы уже работали, поэтому нажмем на кнопку Отмена и сбросим этот флажок.

С помощью этой кнопки мы можем скрывать нижнюю часть данного окна или отображать ее. Здесь — "Добавить в" у нас имеется список папок, куда конкретно мы добавим ссылку на информационно-поисковую систему Рамблер. Мы можем добавить ее в папку Избранное или в одну из существующих подпапок: это у нас Ссылки,

Медиа и Саша. Ну, Саша — это та подпапка, которую мы с Вами создали на одном из предыдущих уроков из Ссылки. Мы можем создать и новую папку. Это мы также уже делали. Давайте, попробуем в папке Саша Создать новую папку. Вот так. Назовем ее Моя папка. После того, как мы нажмем на кнопку

ОК, Вы убедитесь в том, что в папке Саша создана новая папка, которая называется Моя папка. Если мы сейчас при выделенной этой папке нажмем на кнопку ОК, то ссылка на нашу страницу будет добавлена именно в эту папку. Попробуем убедиться в этом. ОК. Сейчас, если мы, опять-таки, выполним команду Избранное, то есть, вызовем подменю и перейдем на папку Саша, то мы увидим, что в этой папке имеется подпапка с названием.

Моя папка, а уже в ней — ссылка на информационно-поисковую систему Рамблер. Команда, которая здесь следует за командой Добавить в Избранное, называется Упорядочить Избранное. Перед нами открывается диалоговое окно, которое называется Упорядочить Избранное. С помощью этого диалогового окна мы можем менять порядок следования ярлыков в списке избранных страниц. Рассмотрим, как это делается. Ну, во-первых, в этом диалоговом окне мы можем создать новую папку. Давайте, попробуем сделать это и посмотрим, что у нас получится. Вы видите, что в самом низу у нас создалась новая папка. В данном случае здесь активен курсор, то есть, мы можем изменить название Новая папка на любое другое. Допустим, мы хотим создать папку с названием, к примеру, Музыка.

Теперь выполняем щелчок мыши вне этой рамочки — и эта папка приобретает название Музыка. С помощью следующей кнопки мы можем переименовать выделенный здесь элемент. Таким элементом может быть и папка. Нажмем. Вы видите, что сейчас опять курсор активен. Слово Музыка находится вот в таком черном квадратике. Это означает, что мы можем это слово спокойно изменить. Мы можем набрать новое имя, а можем просто, перемещая текстовый курсор, например, в самое начало, добавить здесь слово Моя. Пишем — Моя и пробел. И, опять-таки, мы можем сейчас нажать либо на ***Enter***, либо выполнить щелчок мышью вне области ввода текста. Ну, в данном случае воспользуемся клавишей ***Enter***. Очень хорошо.

Мы можем переименовывать и сами веб-страницы. Выделим, к примеру, Рамблер и нажмем на кнопку Переименовать. Вы видите, что информационно-поисковая система теперь тоже заключена вот в такой черный квадратик, и мы можем ее Переименовать. Переименуем ее, например, в Моя страница и нажмем на ***Enter***, то есть, на клавишу ввода. Но здесь мы, к сожалению, ошиблись, и у нас образовались две прописные буквы М. Давайте, еще раз нажмем на кнопку Переименовать, перейдем в начало названия и клавишей ***Delete***, удалим, лишнюю букву М. Еще раз нажмем на ***Enter***. Отлично. Здесь внизу мы видим описание данной страницы. Здесь имеется ее адрес, то есть, она находится по адресу ***www.rambler.ru***. С нашей стороны было сделано 9 посещений.

Последнее посещение состоялось 29 января в 12:40. Имеется возможность сделать эту страницу доступной автономно, то есть, без подключения к

Интернет. Если мы сейчас здесь установим этот флажок, у нас появится вот такая кнопка Свойства, при нажатии на которую появятся свойства вот этой Моей страницы. Мы можем, например, сменить значок этой страницы. Можем выбрать не такой значок, который показывает, что страница открывается браузером ***Internet Explorer***, а любой другой. Но сейчас мы с Вами не станем менять этот значок и закроем это окно. Здесь — ***URL***-адрес нашей страницы. Быстрого вызова — Нет, то есть, в автономном режиме она недоступна. Посещений — 9. И так далее.

Если мы сейчас нажмем здесь на кнопку ОК, то это окно у нас в данном случае закроется. Что еще мы тут имеем? У нас еще имеется кнопка, которая называется Переместить. Нажмем на эту кнопку. Появляется диалоговое окно, которое называется Обзор папок. Здесь мы должны выбрать папку, в которую необходимо переместить выделенный файл. У нас выделен файл, который называется Моя страница. Куда же нам его переместить? Попробуем переместить этот файл в папку, которая называется Моя музыка, и нажмем ОК. Как видите, этот файл исчез с экрана, но, если мы тут выполним двойной щелчок мышью, то у нас появится вот эта Моя страница. Она сейчас находится в папке Моя музыка.

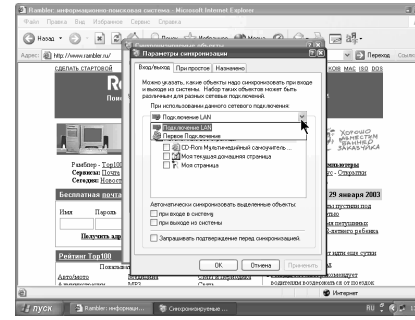
Кроме того, мы можем удалить выделенный элемент. Это может быть, как папка, так и страница. Допустим, мы хотим удалить полностью папку Моя музыка. Нажимаем на эту кнопку. Нас переспрашивают: Действительно ли мы хотим удалить папку Моя музыка и отправить все ее содержимое в корзину. Мы с Вами это подтверждаем, мы именно так и хотим поступить. Все. Папки Моя музыка у нас больше нет. У нас нет также той страницы, которая находилась в папке Моя музыка. Закрываем это диалоговое окно — Упорядочить Избранное. Так как мы установили флажок быть доступной автономно для страницы ***www.rambler.ru***, то теперь эта страница загружена на наш жесткий диск, и она у нас действительно будет доступна автономно впоследствии. Продолжим знакомство с меню Избранное.

Здесь, в нижней части меню у нас, фактически, видны папки, в которые мы заносим наше Избранное, и страницы, которые мы занесли не в папки, а прямо в корень, то есть, в папку Избранное. Еще раз вызовем окно — Добавить в Избранное, чтобы убедиться в сказанном. Вот весь этот список папок и элементы, которые занесены в корень, то есть, в саму папку Избранное, сейчас отображены вот здесь. Вы можете видеть здесь наши 4 папки и те элементы, которые у нас находятся вне папок. У нас имеется доступ к ним всем. Если мы наводим курсор мыши, например, на папку Саша, то у нас открывается ее содержимое. В папке Саша у нас имеются подпапка (вот она), которая называется Моя папка, и два элемента (две веб-странички). На этом мы

завершим наше знакомство с пунктом меню Избранное, а на следующем уроке мы уже будем изучать меню сервис.

6.3.2.11. Главное меню. Меню Сервис

На этом уроке мы поговорим о следующем пункте меню, который называется Сервис. Надо сказать, что в меню Сервис сосредоточены следующие команды. Во-первых, это команды работы с почтовым и новостным клиентом. Большинство из этих команд дублируют команды, которые становятся доступными при нажатии вот этого конвертика. Это: Чтение почты, Создать сообщение, Отправить ссылку и страницу, а также Чтение новостей. Все эти команды у нас доступны и отсюда. Это: Читать почту, Создать сообщение, Отправить ссылку, Отправить страницу и Читать новости. Если мы выберем одну из этих команд (например, мы хотим Читать почту), будет запущен почтовый клиент. В данном случае это ***Outlook Express***, та самая программа, с помощью которой мы можем читать наши письма, получать их, посылать и так далее. А сейчас закроем почтовый клиент, и перейдем к следующим командам.



Здесь также доступна команда — Создать сообщение. Вы уже знаете, как работают с этой командой. Вызовем ее. Появляется вот такое окно Создать сообщение. Здесь мы должны указать Адрес получателя, Адрес получателя Копии, если есть (заполнение этого поля необязательно), Тему нашего письма и, собственно, само письмо. На одном из прошлых уроков мы с Вами это уже делали, поэтому мы сейчас не будем здесь ничего делать, а просто закроем это окно. Никаких изменений мы с Вами в данном сообщении не делали, поэтому ничего нам сохранять и не нужно.

Следующая команда — это Отправить ссылку (отправить ссылку на ту нашу страницу, которая открыта в данный момент). Если мы сейчас выполним эту команду, опять появится знакомое нам диалоговое окно. Здесь мы должны указать Адрес получателя на ссылку нашей страницы. А вот здесь, как вложение, у нас вложен ***URL*** страницы ***www.rambler.ru***.

Вот здесь у нас указано, что это у нас — ***URL***, то есть, ссылка на нашу страницу. Закроем это окно и выполним команду Отправить страницу. Сейчас вся страница, как вложение, присутствует в нашем письме. Все это мы с Вами делали на прошлых уроках. Поэтому закроем это окно и перейдем к изучению остальных команд меню Сервис. Со следующей командой —

Синхронизировать Вы уже знакомы. С помощью команды Синхронизировать синхронизируются те страницы, которые доступны для автономного просмотра. Иначе говоря, если у Вас на экране Ваша страница была сохранена, но по ***URL*** эта страница уже изменилась, то Вам необходимо сравнить эти две страницы и, в случае наличия такого изменения, закачать эту страницу на свой жесткий диск заново, чтобы каждый раз иметь самую последнюю версию этой страницы.

Давайте, например, выделим вот эту страницу и нажмем на кнопку Синхронизация. В данном случае производится сравнение этой нашей страницы, всех ее элементов. Давайте, прервем выполнение синхронизации, потому что синхронизировать она будет достаточно долго, а мы с Вами это уже делали. Останавливаем синхронизацию и закрываем это окно. Еще раз вызовем диалоговое окно Синхронизировать и посмотрим, какие еще здесь имеются возможности.

Здесь была возможность синхронизации, а еще здесь есть кнопка, которая называется Установка. Нажмем на эту кнопку. Появляется диалоговое окно, которое называется Параметры синхронизации. С его помощью мы можем указать, какие объекты необходимо синхронизировать при входе и выходе из системы. Для разных сетевых подключений этот набор может быть совершенно разным. У нас здесь имеются подключение по локальной сети и подключение через Интернет, которое мы с Вами назвали Первое подключение

На каждое из этих подключений мы можем установить соответствующие объекты. Допустим, мы хотим для Первого подключения, то есть, при подключении к Интернету, синхронизировать вот этот объект. В нижней части нашего диалогового окна мы показываем, какие объекты автоматически синхронизировать и когда. То есть, мы хотим синхронизировать, например, При входе в систему (или При выходе из системы).

Если мы установим этот флажок, синхронизация будет происходить при входе в систему. Соответственно, при установлении этого флажка синхронизация будет происходить при выходе из системы. Можно синхронизировать и в обоих случаях: и при входе, и при выходе. Здесь у нас устанавливается флажок Запрашивать ли подтверждение перед синхронизацией. Если мы его установим, то каждый раз, когда будет производиться синхронизация, нам будет выдано диалоговое окно, в котором нас будут спрашивать, синхронизировать или нет.

Здесь мы установили параметры синхронизации при Входе и Выходе из программы ***Microsoft Internet Explorer***. Кроме того, можно производить синхронизацию При простое. Если, к примеру, ваш компьютер бездействует в течение некоторого времени, то и в этом случае можно синхронизировать

Ваши документы. Здесь, опять-таки, мы можем это делать при Сетевом подключении, при подключении к Локальной сети и при подключении к Интернет. Допустим, что этот же элемент мы хотим синхронизировать, так сказать, в Ждущем режиме. Ну, и следующая вкладка — Назначено. Мы можем задать расписание, по которому необходимо проводить синхронизацию наших автономных страниц. Если мы нажмем на кнопку Добавить, то появится Мастер расписания синхронизации, где мы, опять-таки, должны конкретно указать, при каком подключении и что мы будем синхронизировать.

Предположим, ту же страницу мы хотим синхронизировать, допустим, каждый день. Хотя, конечно, мы можем синхронизировать и только по рабочим дням, и каждый сколько-то дней, но мы хотим синхронизировать ежедневно, начиная, к примеру, с 17-ти часов 5-ти минут. Вот так. А начнем мы синхронизацию с сегодняшнего дня, хотя здесь можно выбрать и любой другой день. Ну, давайте, выберем Пятницу. Итак, мы выбрали Пятницу. Теперь нажмем на кнопку Далее.

Имя запланированной синхронизации. Здесь мы можем ввести любое имя. Попробуем ввести следующее: Синхронизация в пять часов... Чем — не имя? Нажмем на кнопку Далее. Теперь мастер расписания синхронизации нам сообщает о том, что мы все сделали успешно. Нам достаточно нажать на кнопку Готово, после чего в этом списке у нас появится вот эта наша задача по синхронизации: Синхронизация в 5 часов. Мы, конечно, можем и изменить эту задачу, если мы ее выделим и нажмем на кнопку Изменить. Мы можем изменить имя расписания,

Объекты синхронизации, само Расписание (когда конкретно мы будем проводить синхронизацию) и Параметры. Но сейчас мы ничего здесь пока не будем менять и нажмем на кнопку Отмена. А с помощью этой кнопки можно удалить данную задачу по синхронизации. Мы сейчас ее удалять не будем. Нажмем на кнопку Применить. Итак, все изменения, сделанные в этом диалоговом окне, у нас сейчас применены. Закроем это диалоговое окно. После нажатия на кнопку Применить все сделанные изменения сохранены, и мы можем спокойно закрыть данное окно с помощью этой кнопки. Закроем и это основное диалоговое окно.

Итак, мы с Вами установили все параметры по синхронизации. Продолжим изучение меню Сервис. Следующая команда называется ***Windows Update***. Что она означает? Если мы выполним команду ***Windows Update***, перед нами откроется соответствующая страница ***Microsoft***, которая посвящена модернизации операционной системы ***Windows***. У нас открылась эта страница. Нам сообщают о том, что поиск последней версии программы ***Windows Update*** может занять несколько минут.

Итак, нас спрашивают, хотим ли мы установить и выполнить ***Windows Update*** версии 4 ***Control***. Здесь, кстати, нам сообщают о том, что компания ***Microsoft*** подтверждает безопасность содержимого. Если мы установим здесь вот этот флажок, то, тем самым, мы с Вами укажем, что мы всегда доверяем содержимому, которое получаем от ***Microsoft Corporation***, и больше это диалоговое окно появляться не будет. Мы установим этот флажок, так как мы уважаем ***Microsoft Corporation*** и доверяем ей. Нажмем на кнопку Да. А сейчас компания ***Microsoft*** сообщает, что мы можем получить последние имеющиеся обновления для операционной системы программы драйверов оборудования нашего компьютера.

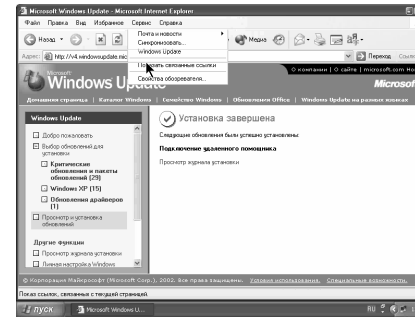
Если мы сейчас нажмем на эту кнопку, то увидим весь набор обновлений, который подходит именно для нашего компьютера. Давайте, именно так и поступим. Сейчас программа ***Windows Update*** выполняет поиск доступных обновлений. Вот здесь в левой части экрана мы видим пункты, по которым нас ведет программа ***Windows Update***. Сначала мы были на первом экране Добро пожаловать, а теперь мы находимся вот здесь. Это — Выбор обновлений для установки. У нас завершено уже 25 процентов выбора обновлений. А теперь уже произведено все 100 процентов выбора обновлений для нашей установки. Итак, у нас имеются следующие обновления: Критические обновления и пакета обновлений — 30, ***Windows***-а ***XP*** — 15, Обновления драйверов — 1 и Просмотр и установка объявлений — 30. Нажмем на эту гиперссылку и прикинем, что из 30-ти имеющихся обновлений мы здесь хотим установить.

Здесь нам сообщают, что в списке выбранных обновлений содержится элемент, который следует устанавливать отдельно от других обновлений. Если мы желаем установить этот элемент, то нужно просто нажать на кнопку Установить. Но сейчас этот элемент здесь имеет размер более 30 мегабайт, поэтому мы пока его устанавливать не будем, так как это займет слишком много времени. Здесь у нас имеется первый пакет обновлений для ***Windows XP***. Будет загружено 1,9 мегабайт. Это, так называемый, ***Service Pack***. Здесь — Пакет исправлений для обозревателя ***Internet Explorer*** 6.0. Сейчас мы его не будем устанавливать таким образом. Мы просто нажмем на кнопку Удалить. Тому, как обновлять обозреватель ***Internet Explorer*** версии 6.0, будет посвящен отдельный урок. Поэтому мы просто нажали на кнопку Удалить.

Обновлять систему безопасности ***Outlook Express*** 6.0 мы сейчас тоже не будем. Это — тоже. Давайте, поступим следующим образом. Мы удалим самые объемные обновления, чтобы не тратить много времени. Вот, к примеру, с размером в 2 мегабайта. Это, конечно, слишком много. Здесь — Подключение удаленного Помощника. Ну, размер тут маленький, и мы вполне можем его установить в дальнейшем. Далее идет Обновление системы

безопасности, которое нам сейчас не нужно. Нажмем Удалить. Ключевое обновление мы пока оставим. А это Ключевое обновление нам тоже не нужно.

Это оставим. Ну, здесь все они достаточно мелкие. А это большой кусок с размеров в 2,8 мегабайта. Закачивать его на наш винчестер мы с Вами не будем. Обновление подключения ***Windows Messenger***-а также великоват, Даже размер 909 килобайт — тоже велик для закачки. Опять свыше 20 мегабайт. Произведем удаление и оставшихся больших по размеру обновлений. Давайте, уберем это тоже. Мы пропустили Обновление системы безопасности — размеров в 1,3 мегабайт (закачиваться будет долго). Его также удалим. Вы видите, что здесь наше число соответственно убывает, в зависимости от того, что конкретно мы здесь удаляем. Ну, здесь остались вот такие мелкие, хотя в сумме они составляют 17 мегабайт, а это не мало. Давайте, удалим и здесь. Здесь — мало. Удалим и эту часть. Вот — 4 мегабайта.... Пакет обновлений с размером 1,9.



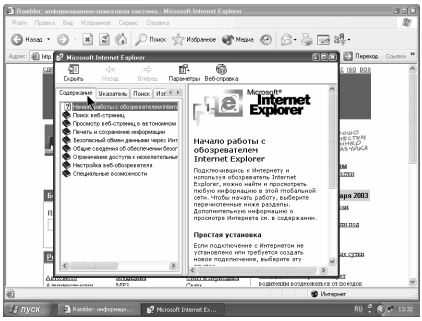
После стольких удалений можно увидеть, что у нас осталось достаточно мало: 3,6. Но и это много. Еще немного удалим... Обновление системы безопасности -Удалить... Удалить, удалить... Еще немного и еще чуть-чуть. По-моему, это — последний. Ну, давайте, установим именно его, удаленного Помощника. Здесь нам предлагают выбрать лицензионное соглашение, то есть, согласиться с ним. Сейчас у нас происходит загрузка обновления, которое устанавливается на наш компьютер, и мы с Вами должны подождать. Здесь индикатор показывает, на каком этапе загрузки мы сейчас находимся. После окончательного завершения загрузки обновления на наш компьютер будет произведена его установка...

Итак, загрузка на наш компьютер завершена. Сейчас производится подготовка к установке... Нам сообщают, что установка обновления на наш компьютер завершена, но необходима перезагрузка нашего компьютера. Нажмем на кнопку Отмена, дабы перезагрузить наш компьютер позднее. Итак, было успешно установлено следующее обновление:

Подключение удаленного Помощника. Это именно то обновление, которое мы с Вами и выбрали. Продолжим изучение меню Сервис. Следующая команда, которая доступна в этом меню, это Показать связанные ссылки. Если мы выбираем эту команду, в левой части нашего окна открываются те ссылки, которые доступны вот отсюда. Те гиперссылки, которые присутствуют на

нашей открытой странице, оказываются доступными именно в этой части. Мы можем перейти по любой из них. Мы продолжаем изучение меню Сервис. Следующий пункт — это Свойство обозревателя. У нас открывается вот такое большое диалоговое окно Свойство обозревателя с многочисленными вкладками, с помощью которых мы можем менять настройки нашего ***Internet Explorer***-а версии 6.0. Сейчас мы этого делать не будем. Настройке нашего браузера посвящена отдельная тема. Мы пока закроем это окно, нажав на кнопку Отмена. Мы с Вами изучили все меню Сервис. Поэтому завершим данный урок. А уже на следующем уроке мы перейдем к изучению меню Справка.

6.3.2.12. Главное меню. Меню Справка



Мы продолжаем изучение главного меню браузера ***Microsoft Internet Explorer*** версии 6.0. Последний пункт меню, который мы с Вами еще не изучили, это пункт меню, который называется Справка. Если мы откроем его, то увидим вот такой набор команд. Давайте, выполним первую команду Вызов справки. Перед нами открывается справочная система нашего браузера ***Microsoft Internet Explorer***. Это — основное окно этой справочной системы. Оно имеет несколько вкладок: Содержание, Указатель, Поиск и Избранное. Вернемся на первую вкладку с названием Содержание. На этой вкладке сгруппированы темы справки по нашему браузеру. Первая тема — это начало работы с обозревателем ***Internet Explorer***. Именно эта тема и открыта у нас вот здесь в правой части окна: Как начать работу с обозревателем ***Internet Explorer***. Если мы потянем вот таким образом за этот скроллинг, то мы увидим всю тему. Справочная система также содержит гиперссылки.

Вот — они, такие синие и подчеркнутые. Если мы подводим сюда курсор мыши, он превращается в такую ладонь, и мы можем перейти по любой из этих гиперссылок. Некоторые пункты нашей справочной системы имеют рядом с собой вот такие знаки "плюс". Что они означают? Если мы щелкнем по такому знаку клавишей мыши, то соответствующий пункт, как бы, откроется, и будет выведено вот такое его пояснение. Если мы еще раз выполним щелчок мышью по этому значку, который уже превратился в "минус", то соответствующий пункт снова закроется. Очень многие элементы обозначены значком, имеющим вид книжки. Что это означает? Это означает, что данный пункт содержит целую тему. Внутри себя она также имеют вот

такие отдельные главы со знаком "вопрос". Страницы со знаком "вопрос" — и есть информация в нашей справочной системе. А вот такие книжечки — это, всего лишь, нечто, вроде подпапок. Если мы щелкаем по ним мышью, они открываются, и становятся доступны их главы. Если мы щелкнем сюда еще раз, то у нас откроется, например, вот эта страничка.

Если мы щелкнем на эту книжечку еще раз, то она, соответственно, закроется. Дальше следует вкладка Указатель. С помощью этой вкладки мы можем вводить ключевое слово для поиска. Предположим, мы с Вами хотим найти ссылки, связанные с синхронизацией. Вводим С, затем — И, и у нас автоматически появляется вот этот пункт Синхронизация веб-страниц. Если мы выполним здесь двойной щелчок мыши, у нас появится диалоговое окно, в котором указаны разделы, которые содержат в себе выражение Синхронизация веб-страниц. Это может быть Предоставление доступа в автономном режиме, Предоставление доступа к избранным страницам и Просмотр веб-страниц без подключения.

Давайте, увеличим место для заголовка. Это делается с помощью технологии ***drag and drop***, то есть, протаскиванием вот таким образом нашего курсора мыши. Допустим, что нас интересует именно это: Предоставление доступа в автономном режиме к текущей веб-странице. Нажмем — Показать. Появляется вот эта информация. Теперь перейдем на следующую вкладку, на вкладку Поиск. С помощью этой вкладки мы можем уже искать слово не таким образом, не в этом списке имеющейся информации, а в той информации, которая у нас имеется на экране.

Например, найдем слово Сервис... Мы ввели ключевое слово для поиска. Теперь нажимаем на Разделы. После этого здесь появляется список всех страниц нашей справки, на которых встречается слово Сервис. Вот — такой огромный список. Мы можем выбрать по желанию, например, Задание программ, используемых обозревателем. Здесь появляется соответствующая тема: Как задать программы, используемые обозревателем ***Internet Explorer***.

Слово Сервис здесь выделено, потому что его мы искали вот тут. А теперь перейдем на самую последнюю вкладку Избранное. Надо сказать, что отдельные места нашей справочной системы нам могут понадобиться впоследствии. Чтобы снова их не искать, мы можем их список задавать. Допустим, что эта справка нам впоследствии понадобится. Здесь указан текущий раздел, то есть, имя вот этой, как бы, справочной страницы. Если мы сейчас нажмем на кнопку Добавить, то этот раздел будет занесен в список доступных разделов.

Если после этого мы здесь перейдем на любой другой пункт, то для возврата нам будет достаточно просто вернуться в Избранное и перейти вот по этому пункту. То есть, нужно будет, как бы, выделить этот пункт и нажать на

кнопку — Показать. Вы видите, что соответствующий пункт у нас сейчас здесь и показан. Ну, мы уже научились работать с этими вкладками, а теперь приступим к рассмотрению вот этой верхней панели инструментов, которая здесь имеется. Что мы тут имеем? Здесь имеется кнопка — Скрыть. Если мы нажмем на эту кнопку, нам будет доступна только правая часть нашего окна Справочная система, то есть, только вот эта часть окна. Нажмем на кнопку — Показать. Вкладки у нас сейчас показаны, то есть, открыты.

Кнопки Назад и Вперед. Эти кнопки идентичны вот этим кнопкам браузера. С их помощью мы совершаем навигацию по нашей справочной системе. Если мы будем нажимать на эти кнопки вот таким образом, то будем, соответственно, переходить с одного экрана справочной системы к другому. Здесь еще имеется кнопка Параметры. Попробуем нажать на нее. Здесь имеется вот такой маленький треугольник, поэтому после нажатия на эту кнопку, открывается, как бы, подменю. Здесь, фактически, дублируются вот эти кнопки: Скрыть вкладки, Назад, Вперед, Домой (с помощью этой кнопки мы переходим на тему Начало работы с обозревателем ***Internet Explorer***). Дальше мы можем — Остановить открытие страницы и Продолжить ее открытие. Эти команды идентичны вот этим командам.

Параметры Интернета. Здесь мы, опять-таки, можем установить определенные свойства обозревателя, так как справочная система ***Microsoft Internet Explorer*** может подгружать отдельные страницы из Интернета. Вы сейчас сами убедитесь в этом. Веб-справка. С ее помощью мы можем потягивать справку из Интернета. Ну, и — Печать и Включить подсвечивание. С помощью Печати мы можем распечатать соответствующий экран.

Давайте, сейчас перейдем на какую-нибудь другую страницу (у нас сейчас не очень хорошая связь с Интернетом) и выполним команду Печать. Здесь нам сообщается, что можно распечатать либо выделенный раздел, либо все разделы текущего заголовка. Ну, если, к примеру, мы выберем распечатку только текущего раздела, то затем мы должны будем здесь нажать на кнопку ОК. Если же мы захотим распечатать все содержимое заголовка, то, соответственно, нужно установить вот этот переключатель.

Мы сейчас ничего печатать не будем, а просто нажмем на кнопку Отмена и продолжим изучение данного списка. Выключить подсвечивание. С помощью этой команды мы можем убрать вот такое подсвечивание результатов. Как видите, мы искали ключевое слово Сервис — и мы здесь его нашли. А результат нашего поиска подсвечен вот таким синим прямоугольником. Если выключить подсвечивание, то, соответственно, при следующем поиске найденные слова не будут подсвечены. Ну, и последняя команда — Веб-справка. С помощью этой команды производится получение информации с Интернет. Выберем эту команду. В данный момент у нас нет подключения к

Интернету, мы с Вами сейчас не подключены к нему, поэтому у нас здесь ничего и не появляется. Закроем окно нашей справочной системы. Мы с Вами уже поняли, как оно работает. Дальше.

Полезный совет. Если мы выполним эту команду, то на нашем экране появится вот полезный совет, идентичный тому, который появлялся при выполнении команды из меню Вид. Но мы закроем это окно, оно нам сейчас не нужно. Пойдем дальше. Следующий пункт — это Для пользователей ***Netscape***. Что это означает? Это означает, что, может быть, Вы привыкли пользоваться обозревателем ***Netscape Navigator***, и для пользователей ***Netscape*** предусмотрены определенные советы.

Тот, кто раньше работал с программой ***Netscape Navigator***, может легко освоить ***Internet Explorer***, и начать работать с этой программой. Здесь дана соответствующая информация и соответствующие советы. Закроем пока это окно и продолжим изучение меню Справка. Техническая поддержка. Если выполнить эту команду, то можно выйти на сайт технической поддержки ***Microsoft***. Сейчас у нас нет подключения к Интернету, поэтому, даже если мы эту команду выполним, все равно никуда не перейдем, а получим запрос на подключение к Интернет. Давайте, выполним команду Подключить, а затем выполним команду

Вызов... Произошло Первое подключение, и теперь у нас открывается страница технической поддержки. Нам предлагается синхронизировать определенные элементы. Вот — это диалоговое окно. Дело в том, что мы только сейчас подключаемся к Интернету, и, фактически, сейчас происходит его первое подключение. Но мы закроем это окно, потому что нам сейчас ничего здесь синхронизировать не нужно.

Открывается страница технической поддержки ***Microsoft*** в России. Это произошло потому, что мы выполнили команду Техническая поддержка в меню Справки. Следующая команда — Отзывы и предложения. С помощью этой команды мы можем передавать по Интернету советы и предложения фирмы ***Microsoft***. Иначе говоря, если мы, например, чем-то здесь недовольны, или хотим им предложить что-либо, то, соответственно, выполняем данную команду Отзывы и предложения. При этом открывается вот эта страница ***Microsoft***, которая называется — Пишите нам.

И, последняя команда меню Справка — это О программе. Выполнение этой команды приводит к появлению диалогового окна с информацией о нашем браузере версии 6.0. Если мы нажмем на кнопку ОК, то это окно, соответственно, закроется. Итак, мы с Вами разобрались со справочной системой. На следующем уроке мы научимся настраивать браузер ***Internet Explorer***.

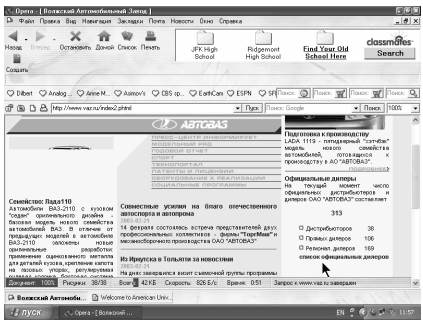
6.4. Путешествие в WEB

6.4.1. Путешествие по миру АВТО

А теперь мы с Вами совершим путешествие по ***Web***. Что здесь нас может заинтересовать? Предположим, мы хотим купить автомобиль. Если мы решили это сделать, то, прежде чем бежать в ближайший магазин, не помешало бы узнать, какие сейчас выпускаются автомобили и, какие автомобили сейчас в моде. Как это можно сделать? Допустим, нас интересуют автомобили фирмы ВАЗ. Как мы знаем из уроков по поиску информации в ***Web***, можно, например, попробовать ввести ***www.vaz.ru***. Иначе говоря, первое, что мы можем сделать для этого, это ввести название фирмы-производителя или компании, которая занимается теми или иными услугами. Итак, ***vaz.ru***. Теперь можно нажать на клавишу ***Enter*** или на кнопку Пуск.

Мы нажмем на клавишу ***Enter***. Сейчас перед нами открывается главная страница АвтоВАЗ. Мы видим здесь эмблему (вот она). Здесь так и написано — АвтоВАЗ. Информация на сайте АвтоВАЗ представлена в двух версиях: в русской и английской. Допустим, нас интересует именно русская версия. Войдем сюда и посмотрим, что здесь представлено. Вот здесь имеется информация о розыгрышах. Нам сообщается, что мы можем купить любую Ладу до 15 февраля и выиграть при этом один из 15-ти автомобилей. Подготавливается к производству новая модель.

Это очень интересно. Если, например, нас это заинтересует, то мы сможем нажать на ссылку Подробнее. Здесь имеется информация об официальных дистрибьюторах. Эта информация может нам пригодиться в том случае, если мы действительно решим купить ту или иную модель автомобиля. Давайте, войдем в список официальных дилеров. Здесь мы видим список дилеров компании АВТОВАЗ (вот он). Он представлен по районам.



По этой ссылке мы можем выйти на дилера любого интересующего нас района. Предположим, что нас интересует представление продукции АВТОВАЗ в республике Татарстан, например вот здесь. Перед нами появляется полная информация о региональном дилере ОАО АвтоВАЗ. Мы можем написать письмо. Как видите, электронный адрес здесь также представляет собой гиперссылку. Если мы сейчас щелкнем по этой гиперссылке, запустится тот почтовый клиент, который установлен по умолчанию, в данном случае, для браузера ***Opera***.

Закроем теперь информацию о дилере и продолжим изучение. Если мы смогли вот таким образом получить информацию об АВТОВАЗ -е, то, вероятнее всего, таким же образом можно получить информацию о производителях каких-нибудь других автомобилей. Давайте, попытаемся таким же образом, набрав здесь ***www***, задать, например, ***ford***. Сначала попробуем набрать здесь ***com***, так как ***Ford*** является коммерческой и, в общем-то, американской компанией. Мы видим перед собой фирменную эмблему компании ***Ford***. Вот она. Открывается первая страница. Спустимся вот сюда. Здесь имеется информация о компании, информация об услугах. Здесь можно задать поиск.

Это может быть любая модель. Кроме того, здесь имеются ссылки на те или иные компании, например, на компанию ***VOLVO***, на компанию ***MAZDA***. Все это нам знакомо. Здесь имеются также ссылки на компании ***JAGUAR***, ***LAND-ROVER***. Иначе говоря, отсюда мы можем перейти на любую из этих компаний. Попробуем точно также поступить с самой известной машиной ***Mersedes-Benz***. Опять-таки, набираем ***www***, затем — ***mersedes*** и, для начала, — ***com***. Попрбуем сначала поискать без ***Benz***. Может, нам и повезет. Нет, к сожалению, нам с Вами не повезло. Это не оказалось адресом компании ***Mersedes Benz***. Мы, в общем-то, оказались совершенно на другом сайте, который называется ***musicplus***. К сожалению, это широкая практика создателей веб-страниц.

Попробуем, все-таки, найти компанию ***Mersedes***. На этот раз мы будем искать эту компанию с помощью поисковой машины ***Google***... Закроем эти всплывающие окна, которые нам в данный момент не нужны. Итак, попробуем осуществить здесь этот поиск. Напишем латинскими буквами... и нажмем на кнопку Поиск в ***Google***. Браузер ***Opera*** спрашивает нас, отправить ли данную форму по сети незашифрованной. Установим здесь флажок Больше не показывать и нажмем на кнопку ОК. Перед нами появилось большое количество ссылок на ***Mersedes***, но, к сожалению, это все не то.

Попробуем теперь еще задать дополнительно ***Benz***. Может быть, так нам больше повезет. Итак, первая же ссылка указывает именно на сайт ***Mercedes-Benz***-а. Компания называется так: ***Mercedes***, тире и ***Benz***. Значит, нам нужно было ввести и остальные буквы. Кроме того, мы допустили ошибку, написав ***S*** вместо ***C***. Если теперь перейти по этой гиперссылке, то можно попасть на официальный сайт компании ***Mercedes-Benz***. Сайт ***Mercedes-Benz*** существует в двух языковых исполнениях: в немецком и английском. Английский язык нам, все-таки, более понятен. Перед нами открывается соответствующее окно. Мы с Вами видим здесь первую машину.

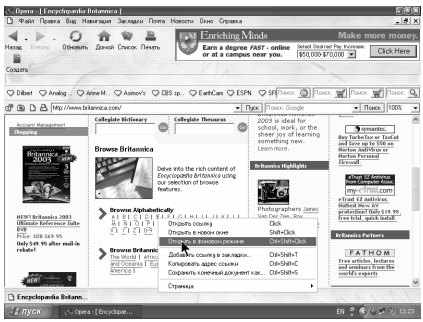
Вот — еще одна машина с видом сзади. Очень красиво выполненный сайт. Здесь присутствует большое количество гиперссылок. Автолюбители могут с удовольствием ознакомиться со всеми этими ссылками. Выберем, например,

эту гиперссылку. Перейдем по ней. Нас еще раз приветливо приветствуют на сайте ***Mercedes-Benz***. Как видите, здесь имеется также большое количество подкатегорий. Рассмотрим, например, вот это (***Coupes***). Здесь тоже имеется гиперссылка. Щелкнем по ней. Вы убеждаетесь в том, что мы можем совершенно спокойно путешествовать по миру автомобилей.

Разумеется, что в Интернете имеются не только страницы ВАЗ-а, ***Ford***-а и ***Mercedes-Benz***-а. Безусловно, все ведущие компании имеют свои веб-страницы. На данном уроке мы с Вами ознакомились только с принципом работы с ними, то есть, каким образом Вы можете просто вызывать их из строки адреса, и, если Вам не удалось что-либо, то найти это с помощью поисковой машины. На этом завершим наш урок, а путешествие по ***Web*** продолжим на следующем уроке.

6.4.2. Энциклопедия и WEB

Мы продолжаем путешествие по ***Web***. Нас здесь может еще заинтересовать получение определенной информации, например, энциклопедии. Самая большая и интересная энциклопедия — это, безусловно, известная всем энциклопедия Британика. Попытаемся обратиться к ней и посмотрим, представлена ли она в Интернете. Поступаем аналогично тому, как мы это делали на предыдущих занятиях: вводим ***www.britanica.com***... Сейчас мы узнаем, представлена ли Британика в Интернете. Как видите, перед нами открывается страница Энциклопедия Британика. Это именно то, что мы хотели. Это и есть главная страница сайта ***britanica.com***. Рассмотрим подробнее эту страницу и посмотрим, какую интересную информацию мы можем получить на этой странице.



Первое, что мы здесь видим, это то, что мы можем посмотреть в алфавитном порядке всю имеющуюся информацию. Оглавление, понятное дело, идет по английскому алфавиту. Попытаемся посмотреть какую-нибудь информацию, начинающуюся, например, с английской буквы ***E***. Говорят, что это самая распространенная буква английского алфавита. Поступим следующим образом. Вызовем контекстное меню данной гиперссылки, нажав на правую клавишу мыши, и попробуем открыть окно в фоновом режиме. Иначе говоря, данная гиперссылка будет открыта в новом окне. Вы видите ее ярлык на панели окон. Она сейчас открывается, а мы продолжаем изучать главную

страницу Британики. Британика, кроме всего прочего, имеет еще и Атлас мира. Посмотрим, что в нем имеется, и какая тут представлена информация. Вероятнее всего, нас заинтересует Азия и Россия, а вернее, Россия. Перейдем уже напрямую по этой гиперссылке. Открывается Атлас мира. Перед нами — вот такая карта. Попробуем, щелкнув мышью, перейти вот таким образом на эту надпись (Страна Россия). Хотя, в общем-то, активна, так сказать, вся территория России для перехода по гиперссылке.

Здесь имеются всплывающие окна, которые мы пока закроем. Посмотрим, что же тут сказано про Россию. Здесь сказано, как официально называется наша страна — Российская федерация. Здесь даже написана транскрипция латинскими буквами. Имеется Флаг. Имеется и карта. Попробуем открыть ее с помощью контекстного меню в фоновом режиме. Пока она будет открываться, мы сможем изучать текущий сайт.

Здесь имеется также и содержание всей статьи, посвященной России. Мы можем получить самую разнообразную информацию: О религии, Народонаселении, Экономике и так далее. Рассмотрим информацию О религии. К сожалению, нам сообщают, что пользование Британикой платное. Правда, мы имеем право на 72 часа свободного доступа. Но, для того, чтобы получить эти 72 часа, нам необходимо зарегистрироваться. Регистрация производится по нажатию на эту ссылку. Пока эта страница открывается, мы рассмотрим другие страницы и посмотрим, что у нас там имеется. Здесь нам разрешается открыть нужные нам статьи. Мы с Вами выбрали букву ***E***. Здесь имеются энциклопедические статьи слов, которые начинаются с этой английской буквы ***E***:

E region, ***e-commerce***, ***e-mail*** и так далее. Допустим, нас интересует какое-то конкретное слово, а не перечисление статей в алфавитном порядке. Наберем что-нибудь типично английское, к примеру, имя Ричард (вот так) и нажмем на кнопку ***Go***. Посмотрим, что найдено по имени Ричард. Надо сказать, что в Британике имя Ричард представлено достаточно широко. Здесь имеются целых два Ричарда-***I***. Далее следует Ричард-***II***, еще один Ричард-***II*** и еще один Ричард-***II***. Три Ричарда-***II***. Здесь имеются и Ричарды-***III***. Теперь уже нумерованные Ричарды исчерпаны, и далее следуют просто Ричарды. Кстати, мы можем нажать на ***Next*** и увидеть следующий список Ричардов. Опять появились всплывающие окна. Закроем их. Причем, можно и бесплатно просматривать статьи по алфавиту. То есть, можно не использовать вот этот сеанс бесплатного доступа в 72 часа. Далее опять следуют Ричарды. Иначе говоря, мы, примерно, поняли, как следует пользоваться этой библиотекой.

Посмотрим, что там у нас открыто браузером в фоновом режиме. Здесь мы, опять-таки, имеем сообщение о том, что необходимо зарегистрироваться для нормального пользования атласом мира. Закроем это окно, потому что форму

для регистрации мы уже открыли вот здесь. Здесь нам предлагается подписаться на пользование Британикой. Можно заплатить за год и заплатить при этом всего 59 долларов. За месяц здесь плата составляет около 10 долларов. Но сейчас мы этого делать не будем. Мы просто просмотрели тот образец, который Вам нужно будет заполнить, если Вы захотите это сделать. Закроем и это окно тоже. Закрыть. Это — алфавитный порядок энциклопедии Британика. Надо сказать, что для изучения энциклопедических статей на английском языке мы, к сожалению, не очень подготовлены. Рассмотрим какие-нибудь российские энциклопедии.

Доступ к российским энциклопедиям легче всего получить с какого-либо портала типа Рамблер или Яндекс. Попробуем ввести ***Rambler*** и посмотреть, какие там имеются энциклопедии. Мы ввели ***Rambler***. Теперь выполним Переход. К сожалению, ничего напоминающего энциклопедию мы здесь на первый взгляд не видим. Попытаемся проделать то же самое в Яндексе. Введем, соответственно, ***Yandex***... Итак, мы имеем сервисы Яндекса, которые называются Энциклопедии. Попробуем перейти по ним и посмотреть, что мы можем найти там. Здесь у нас имеется вот такой список ссылок. Здесь есть ссылки на Большую Советскую Энциклопедию, на Малый энциклопедический словарь Брокгауза и Ефрона и на Толковый словарь Даля. Выберем Большую Советскую Энциклопедию.

Здесь мы можем задать слово для поиска, которое будем искать именно в этой энциклопедии. Здесь имеется вот такой список слов. Допустим, нас интересует кит Касатка. Введем слово Касатка и посмотрим, что же будет найдено в данной Большой Советской Энциклопедии нашим браузером. Итак, были найдены две статьи: Касатка, как птица отряда воробьиных (видимо, такая птица имеется, это деревенская ласточка) и Касатка, как водное млекопитающее семейства дельфинов. Иначе говоря, мы получили нужную нам информацию. Мы можем перейти по любой из этих ссылок (например, сюда) и получить данную энциклопедическую статью уже в отдельном окне. Вы видите, что открылось дополнительное окно, в котором мы получили информацию по слову Каксатка. На примере этих двух энциклопедий мы научились пользоваться ими и искать в них нужную информацию. На этом завершим этот урок, а на следующем уроке мы с Вами продолжим путешествие по миру ***Web***.

6.4.3. Литература в WEB

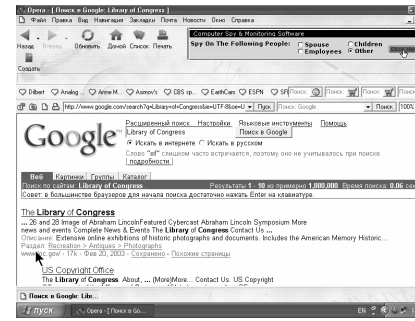
Продолжим изучение. Для чего еще нам может понадобиться Интернет? Для того чтобы, например, покопаться в книгах, ознакомиться с творчеством какого-нибудь поэта из Уругвая или почитать книги писателя Ивлина Во.

Библиотеки очень широко представлены в Интернете. Одной из самых известных библиотек является Библиотека конгресса.

К сожалению, название Библиотека конгресса (***Library of congress***) слишком длинное, поэтому мы не можем предположить, как будет называться в этом случае ***URL***-адрес. Поэтому мы воспользуемся какой-нибудь из поисковых машин, например, поисковой машиной ***Google***. Здесь мы с Вами вполне можем ввести: ***Library of congress***. Убедимся в том, что установлен именно флажок Искать в Интернете, а не Искать в русском Интернете, потому что ***Library of congress*** написано латинскими буквами. Кстати, ***Of*** нам определенно не удалось.

Давайте, исправим: ***of congress***. Производим Поиск в ***Google***. Первая же ссылка, которая появилась здесь, и является Библиотекой конгресса. Если мы посмотрим вниз на этот ***URL***-адрес, то увидим, что он представляет собой ***www.loc.gov***. Можно было бы и предположить, что это будет сокращение, то есть, аббревиатура: ***library — l***, ***of — o***, ***congress — c***. А ***gov*** означает правительственную организацию Соединенных Штатов. Итак, войдем в эту самую библиотеку. Открывается главная страница Библиотеки конгресса Соединенных Штатов. В этой библиотеке мы с Вами можем найти самую разнообразнейшую информацию. Мы можем, например, задать поиск по нашим каталогам, что немаловажно. Можем увидеть индекс этой библиотеки, написать письмо библиотекарям и тому подобное.

Предположим, нас интересует индекс. Вот здесь в алфавитном порядке представлены все книги и статьи, которые имеются в этой библиотеке. Здесь имеется вот такой список. Что конкретно мы можем найти здесь? Например, книги и статьи об арабском мире мы можем найти вот по этой ссылке. Фактически, здесь в основном мы видим список категорий. Перейдем, к примеру, вот сюда на ***Reading room***. Сама ***Reading room*** (читальная комната) тоже разбита на определенные секции. Тут мы также можем углубиться дальше в путешествие по этим ссылкам. Попробуем что-нибудь найти в каталоге. Это есть ***On line*** каталог. С помощью этих двух ссылок мы задаем, как именно нужно искать ту или иную информацию. Это, например, основной поиск, а это — расширенный. Предположим, нас интересует основной поиск, и мы можем назвать автора или предмет, на который он написал то или иное.



Попытаемся, например, найти информацию по ключевому слову Буш (как автора тех или иных источников).

Начнем поиск. Нажимаем на ***Begin search***. Мы видим, что огромное количество Бушей представлено в библиотеке конгресса США. Вы можете убедиться в этом. Сейчас мы видим только 25 имен на странице. На следующую страницу мы можем перейти по нажатию на эту ссылку. Это все пока относится к Альфредам Буш, хотя вот тут имеются и Артуры. Здесь даже есть Барбара Буш. Смотрим дальше. Собственно, если бы мы пожелали искать что-либо за авторством Джорджа Буша, то мы могли бы это задать изначально. А сейчас мы можем просто вот такими переходами к следующей странице добраться до буквы ***J***. Вернемся на ***Search*** и зададим следующий поиск: ***Bush***, пробел, ***J***. Можем еще задать точку. Выполняем ***Begin search***.Итак, большое количество Джорджей Бушей представлено в библиотеке конгресса США...

Вы видите здесь вот такое, можно сказать, обилие. Так как мы не знаем, какой именно нам нужен, попробуем перейти по любому из них, допустим, вот сюда. Оказывается, тот Джордж Буш, по чьей ссылке мы перешли сюда, писал поэмы. Это мы с Вами видим. Мы можем выбрать, в каком именно формате мы хотим получить информацию. В данном случае по умолчанию выбран текстовый формат. Мы должны ввести свой ***E-mail***-адрес, и нам в текстовом формате будет выслан файл о творчестве этого самого Джорджа Буша. Итак, мы с Вами разобрались с Библиотекой конгресса. На этом мы завершим наш урок, а на следующем уроке продолжим изучение этой темы.

Мы продолжаем изучение данной темы. Посмотрим, какие библиотеки представлены в российском Интернете, то есть, в, так называемом, ***Ru***-нете. Перейдем на какой-нибудь портал, например, на ***Yandex.ru***. Мы сейчас переходим на портал ***Yandex.ru***. Посмотрим, представлены ли здесь какие-нибудь библиотеки. Рассмотрим все это внимательней. Нет, вроде, никаких библиотек здесь не представлено. Давайте, убедимся в этом... Зададим поиск По страницам. Найти слово Библиотека (текст вводим на русском языке).

Текст Библиотека найти невозможно. Значит, мы не ошиблись. Введем слово Библиотека для поиска. При этом добавим еще ***Online***, то есть, мы имеем в виду те Библиотеки, которые представлены в Интернете. Посмотрим, что нам найдет ***Yandex***. Неплохо: было найдено 65000 ссылок с 438 серверов. Это указано вот здесь. Здесь имеется большое количество библиотек (например, вот здесь). Это ***Online*** библиотека, имеющая источники на тему Оценки управления стоимостью. Книги, Методические и так далее. Еще — библиотеки, Словари, Медицинская библиотека... Здесь представлено большое количество библиотек. Вызываем контекстное меню и открываем В фоновом режиме. Это мы с Вами уже умеем. Смотрим дальше. ***Online*** библиотека Максима Машкова. Это одна из самых известных библиотек в российском

Интернете. Ее тоже откроем в фоновом режиме, чтобы представить, что она представляет собой. Перейдем на следующую страницу. Здесь у нас, видимо, уже открылась страница. Нет, к сожалению, она не до конца открылась. Мы видим, что данная страница пока еще открывается. Перейдем на библиотеку Машкова. Эта страница у нас уже открыта полностью. Здесь мы можем задать поиск по каталогам этой библиотеки. Но, с другой стороны, каталог библиотеки разбит на категории. Вы можете видеть, что здесь имеется Русская современная проза, Русская и зарубежная поэзия, Самиздат,

Проза и поэзия... Вот такая большая и интересная библиотека. Что нас может заинтересовать в этой библиотеке? Попытаемся с помощью этих категорий найти какой-нибудь литературный источник. Предположим, нас интересует Зарубежная проза. Нажимаем на эту ссылку. Перед нами в алфавитном порядке возникает список писателей, которые эту самую прозу и писали. Допустим, нас интересует Ивлин Во (вот он).

Выполняем щелчок мышью и смотрим, какие произведения Ивлина Во представлены в библиотеке Машкова. Здесь имеются романы: Незабвенная, Черная беда, Мерзкая плоть, Возвращение в Брайдсхед и так далее. Вот такой список романов, рассказов... Здесь также имеется дополнительная ссылка: Ивлин Во представлен на сервере Лавка языков.

Это тоже очень известный сервер, который имеет те или иные литературные источники. Допустим, нас больше всего интересует Незабвенная. Выполняем щелчок мышью. Появляется текст. Теперь можно сидеть перед компьютером и спокойно читать этот текст. В данный момент перед нами открываются все произведения. Итак, мы с Вами разобрались с библиотекой Машкова. Теперь посмотрим, как же поживает та библиотека, которую мы с Вами открыли. Это — список ***Online*** библиотек. Появился рубрикатор Нижнего Новгорода (это как раз и обозначено вот здесь), который имеет в себе ссылки на какое-то определенное количество ***Online*** библиотек. А здесь внизу имеется список ресурсов данной категории. В данный момент здесь имеются в наличие только два ресурса. Это — Медицинская библиотека ***NicoMed***. Это тоже — ***Online*** библиотека, которая содержит более 2000 полнотекстовых книг.

Мы можем потихоньку переходить по ней. Посмотрим, какие еще библиотеки принес нам ***Yandex***. Вы видите здесь вот такое огромное количество библиотек. Литература в Интернете тоже очень хорошо представлена. Вы можете найти здесь произведения на любую тему и на любой вкус. Сейчас перед нами вот здесь открывается эта самая Медицинская библиотека. Нас приветствуют на этом сайте. Мы с Вами не будем подробно рассматривать ее. Итак, Вы уже уяснили принцип пользования электронными библиотеками. Вы уже поняли, что текст может быть прислан нам либо по ***E***-

mail, либо же мы можем просто из самой веб-страницы читать те или иные произведения. На этом завершим данный урок.

6.4.4. Музеи в WEB

Итак, как мы с Вами уже говорили, для сети можно найти множество применений. Интернет нужен и для работы, и для развлечений. Давайте, попробуем сделать следующее. Мы посетим какой-нибудь из музеев мира. Дело в том, что многие музеи размещают репродукции своих коллекций в Интернете. Это делают, практически, все ведущие музеи мира. Мы сейчас попытаемся посетить несколько таких музеев.

Первый музей, который нам приходит в голову, это, конечно же, Парижский Лувр. Этот музей имеет в Интернете свой веб-узел. Попробуем ввести его как-то. Вводим ***www*** и точку. Музей называется Лувр, поэтому вводим ***Louvre.fr***. ***Fr*** означает французский веб-узел. Сейчас перед нами открывается веб-узел одного из лучших музеев мира. Перед нами — главная страница. Здесь имеется следующая информация. Дело в том, что информация на главной странице, если Вы успели заметить, представлена на французском языке.

Ну, французский язык знают не все. В этом смысле был бы предпочтителен английский язык. Информация здесь представлена на английском, испанском и японском языках. Если мы выполним щелчок мыши по соответствующему пункту (вот здесь), то перед нами откроется соответствующая страница. Нам более понятен английский язык, поэтому, давайте, по нему и щелкнем. Мы сделали это. Теперь, как видите, информация левой части нашего окна представлена на английском языке. Что здесь нас может заинтересовать? Мы можем, например, ознакомиться с историей Лувра. Выполним щелчок мышью по этой гиперссылке и посмотрим, что за страница у нас здесь откроется. Открылась страница, посвященная истории Лувра.

Здесь имеются картинки. В правой части нашего экрана мы имеем вот такую информацию. Кроме того, мы можем осуществить виртуальный тур по тем или иным залам Лувра. Если мы выберем вот эту гиперссылку Виртуальный тур, то увидим вот такой список залов. Это, например, Западно-античные времена, это — Египетские и тому подобное. Это, фактически, список залов Лувра. Если мы выполним щелчок мыши по любому из них (вот

таким образом), то будет открыт такой список. Здесь нам сообщается, что для того, чтобы совершить этот самый виртуальный тур,необходимо установить плагин ***Quick Time*** 0.5 для нашего браузера. А уже потом, нажав на соответствующую ссылку здесь, совершить этот тур.

Но мы сейчас не будем этого делать, потому что плагин ***Quick Time*** 0.5 не установлен в нашем браузере, а такая установка отнимет достаточно много времени. Поэтому мы просто ознакомились с тем, что такая возможность имеется. Все музеи периодически проводят те или иные выставки. Если, например, мы нажмем на ссылку ***exhibitions***, то увидим, когда и какая выставка ожидается в этом самом музее. Например, с 17 января по 14 апреля состоится вот такая выставка, информацию о которой Вы можете получить, перейдя по соответствующей гиперссылке. Вот, например, с 26 марта по 23 июня состоится выставка, посвященная Микеланджело.

Что еще здесь имеется? Вот — Леонардо да Винчи. Соответственно, состоится экспозиция. Выполним щелчок мыши. Вот здесь имеется информация о выставке, которая состоится в это время в Лувре. Этот сайт достаточно большой и богатый. В нем имеется разнообразнейшая информация. Музей, соответственно, проводит образовательные мероприятия. Если мы щелкнем по ***Education***, то увидим их. Это различные конференции, обмен опытом и так далее. А это, соответственно, их программа на 2002 и 2003 год. Вы можете связаться с данным музеем, нажав на соответствующую ссылку, которая называется ***Contact as***. По различным пунктам имеются соответствующие возможности. Допустим, Вас интересует вопрос, кто разработал этот веб-сайт, и Вы хотите связаться с создателями этого веб-сайта.

Переходим по ссылке, которая называется Веб-команда. Мы видим список людей, которые создавали этот самый сайт. Это все, что касается Лувра. А что же можно сказать о наших российских музеях. Первое, что приходит в голову, это, конечно же, Государственный Эрмитаж. Безусловно, он также представлен в Интернете. Введем ***www.hermitage.ru*** и нажмем на клавишу Ввод. Мы попадаем на главную страницу веб-узла Государственный Эрмитаж. Информация на этой странице представлена, как на русском языке, так и — английском.

Мы выберем русский язык. Аналогично тому, как это было на странице Лувр, где у нас имелась возможность виртуального визита (виртуального тура), здесь имеется такая же возможность. Вызовем контекстное меню, воспользовавшись правой клавишей мыши, и откроем гиперссылку На новой вкладке. Вы видите, что здесь у нас в фоновом режиме открывается новая вкладка. А мы с Вами пока посетим шедевры коллекции. Здесь мы видим информацию о том, что в государственном музее Эрмитаже собраны около трех миллионов памятников. Тут имеются вот такие категории. Что нас может

здесь заинтересовать? Ну, допустим, Галерея драгоценностей. Галерея драгоценностей имеет следующие рубрики: Золото древних кочевников, Греческое золото, Ювелирное искусство Востока и так далее. Остановим свой выбор на Ювелирном искусстве Востока.

Здесь имеется информация о Ювелирном искусстве Востока, имеются изображения отдельных украшений. Мы можем получить подробное описание каждого из них. Выберем, например, вот это нагрудное украшение в виде воина-орла. Допустим, именно это нас интересует. Выполним щелчок мыши на Подробном описании и посмотрим, что из этого выйдет. Теперь мы видим более удачное изображение. Вы видите эту внушительную картинку. Таким образом, мы можем получить полное представление о том, как именно выглядит этот шедевр.

Посмотрим, что нам предложили на другой вкладке в виде виртуального тура. Здесь, опять-таки, имеется вот такое деление: Первый этаж, Второй этаж, Третий этаж, Вид с крыши и так далее. Правда, я предполагаю, что нам и здесь придется установить какой-нибудь плагин. Хотя, нет. Мы видим полный и подробный план каждого этажа. Соответственно, каждый из этих пунктов, который изображен на этой картинке, имеет гиперссылку. Мы можем нажать на соответствующий пункт на самом рисунке, или же можем нажать на соответствующую строчку внизу. Предположим, нас интересует Зал Юпитера. Зал Юпитера имеет номер 15. Давайте, найдем его на картинке. Где же он у нас? Здесь — 16, а 15 находится где-то в другом месте. Но, это — не беда. Нажимаем на номер 15.

Перед нами появилось изображение зала Юпитера (вот оно). Как видите, это изображение у нас сейчас показывается с разных позиций так, как будто камера перемещается. Таким образом, Вы можете получить полное представление о том, как выглядит Государственный Эрмитаж. На этом мы завершим данный урок. Итак, Вы убедились в том, что в Интернете представлены различные музеи.

6.4.5. Отправление открытки

Что еще радостного и приятного мы можем сделать с помощью Интернет? Может случиться так, что в самый последний момент мы вспомним, что у кого-то из наших друзей наступил день рождения. Что мы можем сделать в этом случае? Мы можем послать ему интерактивную поздравительную открытку. Большую часть подобных открыток можно отправлять бесплатно. Откроем какой-нибудь сайт, с помощью которого можно отправить поздравительную открытку, и посмотрим, что в реальности представляет собой этот сайт.

Самый большой из сайтов, содержащих подобные виртуальные открытки, так и называется ***Virtualcards***. Введем ***www.virtualcard.ru***. Это российская страница. Сейчас перед нами открывается страница этих самых виртуальных поздравительных открыток. И вот тут нам предлагается выбрать, к какому конкретно празднику послать поздравительную открытку. Допустим, мы хотим поздравить человека с днем рождения. Для этого вы выполняем щелчок мыши по соответствующему пункту. Мы с Вами видим примеры таких открыток (их эскизы). Два эскиза уже открылись.

Открываются еще и другие. Предположим, что нам не нравится ни один из приведенных здесь шести вариантов. К сожалению, пока шестой вариант не появился. Вот он и появился с таким цветочком. Мы можем поочередно путешествовать вот по этим гиперссылкам. Допустим, нам понравился вот этот вариант с таким красивым дракончиком. Выполняем здесь щелчок мыши. Сейчас мы увидим этот рисунок полностью. Он у нас потихоньку открывается. Вы видите здесь вот такого дракончика с тортиком. Но здесь мы уже должны заполнить форму, потому что данная открытка содержит не только графику и анимацию, но и свое собственное послание. Кроме того, мы должны задать электронный адрес для того, чтобы открытка была доставлена соответствующему адресату.

В данном случае нам предлагают ввести ***E-mail***. Введем тот самый адрес ***Karapeta@mail.ru***, наше имя Степан Степанов и ***E-mail*** адресата, которому мы посылаем данное послание. Отправим это послание, например, некоему ***motor-head***-у на ***mail.ru***. Имя получателя открытки. Это тоже необходимо ввести. Зовут его Моторная голова. Далее следует Тема письма с открыткой. Понятно, что это должно быть С днем рождения. Так и введем. Здесь нам предлагается выбрать пример дизайна, то есть, как будет выглядеть вся открытка. Мы выбираем дизайн открытки, например, дизайн 6. Здесь мы можем даже выбрать музыку. Если у получателя открытки имеется звуковая карта и динамики, то он может услышать какую-либо музыку. Выберем отсюда что-нибудь оптимистичное. Ну, Свадебный марш пока выбирать рановато... Посмотрим, что тут имеется подходящего для нашего случая. Вот, в данном случае ***Happy birthday*** подходит нам. А теперь — Ваше сообщение в открытке.

Что мы можем сообщить адресату? Введем, например, следующее... С днем рождения тебя... Соответственно, копируем это. С днем рождения тебя...

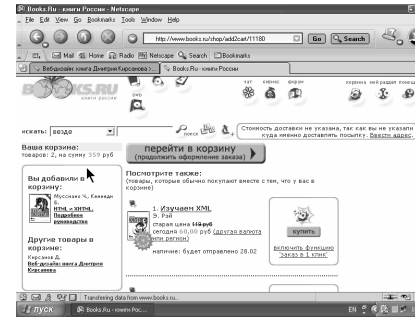
Вводим С днем рождения, милый Моторчик... С днем рождения тебя. По-моему, очень хорошо и мило. Итак, отправляем Немедленно. Но здесь мы можем сделать еще и следующее. Мы можем послать открытку, которую он сможет прочитать в ***Web***. Если же он пользуется каким-нибудь почтовым клиентом, то можно отправить это послание в виде ***HTML***. В этом случае вся эта информация вместе с картинкой будет просто скачана в почтовый клиент, а не будет представлять собой ссылку.

Сейчас мы можем отправить эту поздравительную открытку, как ***Web***. Уведомление о просмотре открытки мы хотим, все-таки, получить, чтобы быть уверенным в том, просмотрел ли именинник нашу открытку. А Дату отправки зададим Немедленную. Но мы сейчас, все же, не будем отправлять эту открытку, а просмотрим ее предварительно... Открытка в реальности будет иметь вот такой симпатичный вид. Вот она с такими душевными стихами. Здесь указан и отправитель тоже. Имеется и вот такая небольшая реклама. Иначе говоря, при получении данной открытки наш именинник увидит ее именно в таком симпатичном виде. Если же нам здесь что-то не понравилось, или нужно исправить что-либо, то мы можем нажать на кнопку Назад и подправить кое-что, то есть, изменить введенную нами информации.

Допустим, мы хотим вместо обращения Милый Моторчик (чтобы он не обиделся от такой фамильярности) написать Милый ***Motorhead***. Еще раз взглянем на то, что у нас получилось. Предварительный просмотр. Итак, изменения уже вступили в силу. Если нас здесь все устраивает, то мы уже можем отправить открытку. Нажимаем на соответствующую кнопку. Открытка отправлена. Нас благодарят за то, что мы воспользовались услугами данного сайта и сообщают об успешной отправке открытки. Если мы хотим отправить какую-нибудь открытку еще кому-нибудь, то можем, соответственно, нажать на эту кнопку. Но мы сейчас этого делать не будем. В общем-то, процедуру отправки открытки мы с Вами освоили.

6.4.6. Покупка книги

Рассмотрим, в каких еще случаях можно использовать Интернет. Например, мы с Вами можем использовать Интернет для покупок. Мы сейчас попробуем отовариться в каком-нибудь магазине. Что мы можем там купить? Здесь имеются: компакт-диски, ***DVD***, билеты на самолет, компьютеры, комплектующие, бакалейные и гастрономические товары, бытовую технику и лекарства. Давайте, попробуем войти на какой-нибудь сайт, где мы можем купить, к примеру, какую-нибудь книгу. Пусть это будет российский сайт, который называется ***Books.ru***. Вводим ***www.books.ru***. В данный момент перед нами открывается главная страница одного из старейшего магазина российского Интернета, которая называется ***Books.ru***.



С помощью этого магазина мы с Вами можем купить книги, компакт-диски, видеокассеты и ***DVD***. Кроме того, этот сайт имеет свой ***Chat***, свой форум и так далее. Мы можем найти любую интересующую нас книгу. Так как наибольшим спросом пользуется компьютерная литература, то здесь имеются вот такие ссылки, по которым мы можем быстро перейти в соответствующий раздел. Это может быть ***Net***, ***Delphy***, ***Linux***, ***My SQL***, ***Oracle*** и ***PERL***. Тут имеется даже форум поклонников Гарри Потера и тому подобное. Вот бестселлеры. Это веб-дизайн, книга Дмитрия Кирсанова. Тут еще есть Компьютерные сети, Последнее слово за мной и так далее. Как же конкретно совершаются в Интернете покупки. Попробуем перейти по какой-нибудь из этих ссылок. Допустим, нас интересует вопрос веб-дизайна. Там уже будет видно, каким образом мы можем купить данную книгу. Здесь нас предупреждают о том, что все цены указаны без учета доставки. Но мы можем узнать стоимость доставки в наш регион. Попытаемся, нажав на правую клавишу мыши, открыть на новой вкладке вот эту ссылку, чтобы узнать условия доставки. Здесь нам сообщают следующее. Оказывается, если мы сейчас купим эту книгу, то она будет отправлена почтой 28 февраля.

Мы можем еще узнать дату доставки. Выполняем правый щелчок мышью. Мы убеждаемся в том, что в данном случае нам не позволяют открыть эту ссылку в новом окне. Почему? Если мы подведем курсор мыши к этой строке, то в статусной строке увидим, что мы имеем дело не с ссылкой, а, всего лишь, с вызовом ***Java script***-а. А ***Java script*** мы не можем вызвать в отдельной странице, а именно, вот здесь. Если мы выполним щелчок мышью вот здесь, откроется дополнительное всплывающее окно, в котором мы можем узнать срок доставки данной книги. Здесь указаны ориентировочные сроки доставки товара.

Например, в Санкт-Петербурге эта книга будет доставлена вот в таком промежутке времени, то есть, либо сегодня, либо третьего марта. В Москве — с 3-го по 8-ое. Это — остальные регионы России. Тут имеются и страны СНГ: США, Германия, Израиль и другие страны. По крайней мере, все конкретно и понятно. А пока закроем это и перейдем сюда. Итак, для того, чтобы производить покупки в Интернет-магазине, желательно предварительно зарегистрироваться. Для чего это делается? Это делается для того, чтобы данный сайт хранил информацию о тех книгах, которые мы заказали, и которые еще должны быть оплачены и тому подобное.

Допустим, я — новый пользователь. Нажмем на эту кнопку и зарегистрируемся. Те поля, которые отмечены красными звездочками, должны быть заполнены в обязательном порядке. Но мы аккуратно заполним все поля. Перейдем на русский шрифт и введем Степан Степанович. Уберем курсор мыши. Введем Степанов. День рождения не будем указывать. Наш ***E-mail***-адрес — ***karapeta@mail.ru***. Вводим пароль и еще один раз — пароль. Регистрируемся. Теперь уже регистрация успешно завершена. Мы можем заполнить почтовый адрес доставки и вернуться обратно в каталог. Мы с Вами сейчас вернемся в каталог, но пока не будем заполнять почтовый адрес. Это мы сделаем как-нибудь в другой раз.

Теперь, допустим, что мы с Вами хотим купить данную книгу по веб-дизайну. Нажмем на кнопку Купить. После того, как мы положили в нашу корзину один товар на сумму 109,79 рублей (это книга "Веб-дизайн" Кирсанова), ***Books.ru*** нам предлагает книги по похожей тематике: может быть, они нас тоже заинтересуют. Это может быть книга "Веб-дизайн" Нильсона, книга по ***HTML***, по ***X-HTML*** и тому подобное. Допустим, нас заинтересовала еще и эта книга "***HTML*** и ***X-HTML***". Она стоит 180 рублей. Сейчас нам сообщают, что данная книга (если мы пожелаем ее) будет отправлена 28 числа. После того, как мы выбрали книгу, посвященную ***HTML*** и его расширенной разновидности, нам предлагают справочники по ***XML***, по программированию на ***PERL*** и тому подобное.

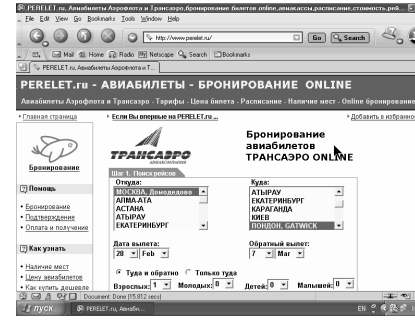
Предположим, что нас больше ничего не интересует. Этих двух книг нам вполне достаточно. Итак, после того, как мы с Вами заказали их, мы должны перейти в корзину и продолжить оформление заказа. Попробуем оформить свой заказ и посмотрим, что при этом получается. Итак, получатель — Степан Степанович Степанов, страна — Россия (регион и область штата нам выбирать не нужно), город — Москва. Введем индекс — 111116. Далее следуют Улица, Квартира и Дом. Вводим: Энергетическая "запятая" 8/2. Укажем контактный телефон: 362-22-22.Какие же дадим примечания? Примечаний никаких не дадим. Сохраняем адрес. Нам сообщают, что у нас не определен регион. Видимо, мы что-то упустили. Верно. Московская (внутри МКАД).

Введем: метро Авиамоторная. Итак, у нас имеется вот такой адрес доставки. Продолжим нашу регистрацию. Нам сейчас сообщают о том, что для нашего адреса доступен только один способ доставки — доставка курьером. Рассмотрим подробнее, что представляет собой доставка курьером. Вызываем контекстное меню и выбираем На новой вкладке. Здесь открывается новая вкладка. Нам сообщается, как именно будет доставлена наша посылочка. Продолжим свою работу. Итак, для нашего региона доступны следующие способы оплаты. Мы можем заплатить курьеру наличными. Это значит, что после того, как курьер доставит наш заказ по указанному адресу, мы,

собственно, ему и оплатим наш заказ. Мы можем и перевести сумму банковским переводом. А можем и оплатить кредитной картой и прочими способами.

Давайте, оплатим банковским переводом. Если Вы собираетесь оплатить покупку от организации... Покупку от организации мы с Вами оформлять не будем, а просто нажмем на кнопку Продолжить. Мы с Вами согласились, что будем оформлять банковским переводом. Итак, у нас все готово к совершению заказа. Нам предлагается проверить правильность указанной информации и подтвердить ее. И в заключении нам необходимо подтвердить свой заказ. Сейчас мы этого делать не станем. После того, как мы подтвердим заказ, останется только лишь перечислить деньги, и нам будут отправлены наши книги.

6.4.7. Бронирование билетов



Итак, на прошлом уроке мы с Вами научились покупать товары посредством Интернета. А теперь попробуем следующее. Мы попробуем забронировать и приобрести авиабилет. Надо сказать несколько слов в защиту этой системы для бронирования и приобретения авиабилетов через ***Web***. Во-первых, с помощью этих систем бронирования можно очень легко выяснить, какие имеются рейсы в интересующем нас направлении и, каковы цены на билеты. После того, как Вы выбрали подходящую авиакомпанию, Вы можете посетить ее веб-узел. Рассмотрим несколько сайтов, с помощью которых можно забронировать авиабилеты. Первое, что приходит в голову, это, так называемая, ***UTS***-2000. Введем ***www***, затем перейдем на Латиницу и введем "точка" ***UTS***-2000 "точка" ***com***. С помощью данного сайта можно забронировать авиабилеты на рейсы аэрофлота.

Это могут быть, как прямые рейсы, так и — со стыковкой на рейсы других авиакомпаний. Эта страница имеет вот такой вид. Здесь нам предлагается в связи с изменением адреса сервера обязательного установить версию вот такого приложения. Сейчас мы ничего устанавливать не будем. Здесь имеются определенные ссылки, с помощью которых мы можем узнать тарифы, правила, счета к оплате и так далее. Здесь имеются: Вопросы, Изменения, Регистрация и прочее. То есть, для начала мы должны получить доступ к этой базе. Нажмем, например, на Регистрацию. Давайте, перейдем вот сюда, потому что

здесь дана ссылка, которая может быть интересна для пассажиров. Нажимаем. Мы можем забронировать билеты компании ТрансАэро. Кроме того, мы можем найти определенные рейсы. Посмотрим, что здесь имеется. Мы можем забронировать авиабилеты ТрансАэро и билеты аэрофлота. Ну, вроде, все (только эти две авиакомпании). Допустим, мы с Вами хотим вылететь из Москвы (это Домодедово). Куда же нам полететь?

Ну, например, в Атырау. Дата вылета. Допустим, мы хотим лететь не сегодня, а в марте. Обратный вылет. Предположим, обратно мы будем лететь в декабре. Детей у нас нет. Итак, нам нужны два билета: туда и обратно. Нажмем на кнопку Наши рейсы и посмотрим, имеется ли у нас возможность осуществить это. Нам сообщают о том, каким образом можно долететь из Москвы в Атырау, да так, чтобы 30-го числа быть на месте. Вот здесь нам предлагается вылететь 29 марта, потому что, видимо, на 30-ое мест нет. Кроме того, мы еще можем вылететь 3-го апреля и 5-го апреля с помощью этих самых ТрансАэро ***Airlines***-ов.

Лететь мы с Вами будем в Боинге в салоне, в котором не курят. А в обратном направлении рейсов не было найдено. Видимо, год — это слишком долгий срок. Вернемся обратно и попробуем то же самое выполнить для рейсов аэрофлота. Для рейсов аэрофлота имеются два сервиса. Мы можем узнать, как именно лететь, и, сколько это будет стоить. Допустим, нас интересует вопрос стоимости билетов Москва — Сидней (туда и обратно). Нас интересует стоимость в долларах. Да, билеты — дороговаты. Лететь первым классом стоит 7000 долларов, бизнес классом — 4000 долларов, а эконом классом — чуть меньше полутора тысяч долларов.

Это весьма сурово. Давайте, попробуем найти, какими рейсами мы можем вылететь из Москвы, к примеру, в тот же Сидней... Здесь имеются агентства, через которые мы можем организовать тот или иной вылет. Допустим, мы хотим найти наши рейсы. Вот рейсы из Москвы в Сидней. Итак, мы можем лететь рейсом аэрофлота, вылететь из Москвы (Шереметьево) и затем прилететь вот сюда (рядом с Токио). После чего мы можем вылететь с Токио и уже прилететь в Сидней. Мы можем также лететь через Бангкок. А обратно мы можем лететь либо через Бангкок, либо, опять-таки, через Токио.

После того, как мы посмотрели всю эту информацию, выбрали для себя тот или иной рейс, необходимо узнать о наличии мест. Имеются ли места на выбранный рейс? Сейчас нам сообщают о том, что эта система бронирования автоматически находит и показывает нужные нам рейсы. Из расписания для указанного направления проверяется наличие мест. Кроме того, прокладывается стыковочный маршрут, если нет прямых рейсов. Как нам проложить стыковочный маршрут для того, чтобы полететь до Сиднея, мы с Вами уже знаем.

Вернемся обратно. Здесь у нас уже все это показано. Причем, свободные места здесь имеются, потому что, как нам сказали только что, эта система перелета показывает только те рейсы, для которых имеются свободные места. А после того, как мы получили всю необходимую информацию, мы можем уже начать бронировать авиабилеты. Для этого необходимо нажать, соответственно, на эту ссылку. Закроем появившееся окно и поработаем вот с этой открытой вкладкой. Теперь уже мы предполагаем, что подходящий рейс у нас выбран. Мы нажимаем на кнопку Продолжить. Нам нужно ввести данные, которые необходимы для бронирования билетов. Здесь у нас маршрут дан полностью (мы с Вами сами выбрали его).

Здесь сообщается, что минимальная стоимость одного билета составляет 172000 рублей (по минимальному допустимому тарифу). Срок оплаты — до 6-ти часов 27-го февраля. Нам предлагают выбрать класс авиабилета. Первый — Эконом, затем — Эконом и — еще один Эконом. Здесь нам уже необходимо ввести данные о пассажирах, дать номера контактных телефонов, ***E-mail*** и тому подобное. Всю информацию, которую мы с Вами должны здесь ввести, мы уже вводили однажды на прошлом уроке при рассмотрении вопроса покупки книг.

Сейчас мы с Вами не будем тратить зря время и не станем бронировать билеты. Процедура, как понимаете, совершенно аналогичная. Необходимо будет ввести сюда нужные данные, затем нажать на кнопку Забронировать места, и места будут забронированы. Нас оповестят об этом и представят счет для оплаты, как, собственно, здесь и написано. Итак, мы сейчас не будем на это тратить время. Вы уяснили для себя основные преимущества покупки авиабилетов и их бронирования через ***Web***. Вы можете выбрать наиболее удобный для Вас маршрут, узнать, стоимость билетов и, во сколько Вам обойдется поездка.

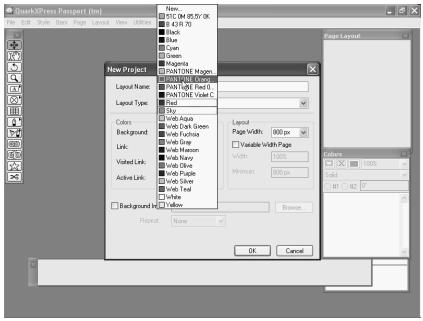
6.5. Отличия от поли графического дизайна на примере публикаций QuarkXPress 6.0

6.5.1. Создание простых Web-проектов

Для создания нового ***Web*** проекта в ***QuarkXPress*** мы пользуемся все той же командой ***File***, ***New***, ***Project***, только на этот раз в поле ***Layout Type*** вместо ***Print*** у нас должно стоять значение ***Web***. Как вы можете видеть, окна создания печатных и ***Web*** проектов отличаются между собой. Поэтому, давайте, рассмотрим с вами подробнее окно создания ***Web*** проекта. В разделе ***Colors*** есть 4 поля. Это ***Background***, ***Link***, ***Visited Link*** и ***Active Link***. Поле ***Background*** показывает, какого цвета фон будет использоваться в нашем ***Web*** проекте. По

умолчанию здесь стоит белый цвет фона. Остальные три поля — ***Link***, ***Visited Link*** и ***Active Link*** — это цвета гиперссылок. Причем имеются в виду именно текстовые гиперссылки. В данном случае установлены стандартные цвета, которые используются для ***Web*** проектов по умолчанию.

В этом варианте стандартная текстовая гиперссылка имеет синий цвет, а гиперссылка, на которой пользователь уже щелкал, т.е. открывал страницу, с которой она связана, имеет пурпурный цвет. И активная ссылка, таковой является ссылка, на которой в данный момент щелкают, имеет красный цвет. Любой из этих цветов можно менять, щелкнув на выпадающем списке. Здесь мы можем увидеть цвета, которые у нас установлены для ***QuarkXPress***. А кроме того мы можем увидеть дополнительные цвета с приставкой ***Web***.



Помните о том, что ***Web*** страницы, в отличие от страниц, предназначенных для печати, имеют ограниченный набор цветов. Поэтому использовать в них те цвета, которые вы определили при работе с печатными публикациями, конечно, не стоит. Проблема заключается в совместимости различных браузеров. Дело в том, что набор цветов, отображающийся одинаково любым из существующих браузеров, состоит из 216 — 216 совместимых с любым браузером цветов. Все остальные цвета, которые мы можем установить, могут в различных браузерах отображаться по-разному. Поэтому, если мы хотим быть уверенными в том, что цвета, использованные нами, в ***Web*** странице все пользователи будут видеть одинаково, мы должны использовать специальные библиотеки цветов. Для того чтобы получить к ним доступ, вместо того чтобы выбирать из данного списка, мы щелкаем на команде ***New*** и здесь прежде чем выбрать цвет, мы выбираем модель ***Web Safe Colors***, т.е. безопасные цвета для ***Web***.

Данная модель содержит набор из 216 цветов, которые поддерживаются абсолютно всеми браузерами. Вы можете обратить внимание на то, что каждому цвету соответствует некий код со знаком решетки перед ним. Данный код на самом деле — это 16-ричное представление комбинации красного, синего и зеленого цветов, т.е. палитры ***RGB***. Иначе говоря, например, сочетание ***FF***, если перевести его в 10тысячную систему будет означать 255. Комбинировать красный, зеленый и синий цвета можно, изменяя каждый из них в пределах 256 градаций. Так вот, если разбить символы в коде цвета на три группы по два символа, то первые два символа обозначают красную

составляющую, следующие два символа — зеленую составляющую и последние два символа — синюю составляющую.

Если же вы теряетесь в такой палитре, то вы можете использовать вместо палитры ***Web Safe Colors*** палитру ***Web Named Colors***. Здесь также цвета имеются, поддерживаемые ***Web***, только в данном случае они имеют конкретные имена, в которых вам будет легче разобраться. К сожалению, палитра ***Web Named Colors*** более бедная, чем палитра ***Web Safe Colors***, но во всяком случае, у цветов здесь есть конкретные названия, такие как, скажем, ***Light Steel Blue*** или ***Magenta***, или, скажем, ***Medium Purple*** и т.д.

Еще раз повторю, что мы используем данные палитры для того, чтобы цвета в наших ***Web*** проектах отображались во всех браузерах одинаково. Если же мы попытаемся использовать, скажем, ***CMYK*** цвета или цвета из любой другой палитры, предназначенной для печати, то само собой разумеется, что в итоге мы потерпим неудачу, т.к. принципы отображения цветов на экране и отображения цветов при печати очень различаются. Давайте, выберем для ссылок на нашей странице какой-нибудь цвет, скажем, вот цвет ***Navy*** — морской, и щелкнем на кнопке ОК.

Таким же образом вы можете изменить цвета остальных элементов для вашей ***Web*** страницы. Но это можно сделать и после создания проекта, так что торопиться здесь некуда. Давайте теперь посмотрим на правую часть окна создания проекта, где находится раздел ***Layout*** — макет. Данный раздел выполняет всего лишь одну функцию — он регулирует ширину вашей страницы. Так как у ***Web*** страницы не существует конечного размера, во всяком случае по высоте, то данный размер, само собой, отсутствует, но ширину ***Web*** страницы можно регулировать. Это имеет значение для таких пользователей, у которых, скажем, мониторы с небольшой диагональю и использующими невысокое разрешение экрана. Ваша страница при неправильной настройке может не очень корректно там отображаться.

Итак, что же у нас есть в поле ***Layout***. Мы можем задать ширину страницы в пикселях. Кстати, привыкайте, при работе с ***Web*** проектами большинство измерений, в отличие от печатных проектов, производится в экранных пикселях, и мы можем задать здесь любую из величин, которые вы видите в данном выпадающем списке — от 600 до 1268 пикселей. Рекомендуется задавать ширину страницы до 800 пикселей, т.к. многие пользователи используют мониторы с разрешением экрана 800x600.

Если ваша страница будет шире, чем данный размер, то она не будет помещаться на таком мониторе. Существует еще компромиссный вариант. Мы можем, скажем, задать ширину нашей страницы в 1024 пикселя, но при этом включить флажок ***Variable Width Page***, т.е. страница переменной ширины. Что это означает?

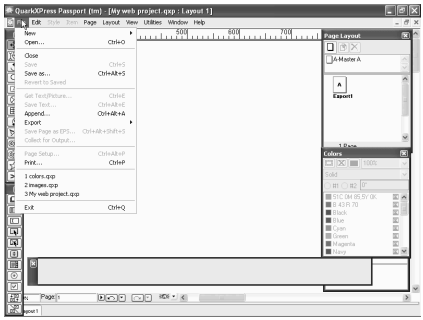
Это означает то, что ширина вашей страницы нефиксирована. Мы задаем желаемое значение ширины для нашей страницы, но при этом говорим, что, во-первых, страница может уменьшаться по ширине и указываем для этого нижний предел. В данном случае у нас стоит значение 800 пикселей, т.е. при просмотре данной страницы с разрешением 800x600 пикселей, браузер просто автоматически сожмет страницу и перегруппирует элементы на ней таким образом, чтобы они помещались вот в данное расширение.

И кроме того, здесь же можно задать ширину страницы относительно окна браузера. По умолчанию стоит 100%. Мы можем, например, задать значение 70 или, скажем, 80%. Мы с вами оставим это значение по умолчанию, т.е. 100%, и теперь, давайте, обратим внимание на последний элемент, находящийся в данном окне, который в данный момент отключен. Это жлажок ***Background Image***.

Не путайте его с пунктом ***Background Color***. Если в ***Background Color*** мы просто задавали цвет фона нашей ***Web*** страницы, то в пункте ***Background Image*** мы можем выбрать изображение, которое будет помещено на фон нашей ***Web*** страницы. Более подробно нюансы наложения фонового изображения мы рассмотрим с вами чуть-чуть позже. А пока мы просто отключим этот пункт.

После того, как параметры, создаваемого ***Web*** проекта нами определены, остается щелкнуть на кнопке ОК и, пожалуйста, новый ***Web*** проект в ***QuarkXPress*** создан.

6.5.2. Палитра Web Tools



После того, как проект создан, следует его в первую очередь сохранить. Для этого мы используем команду ***File***, ***Save*** и выберем для нашего проекта папку ***My Documents***, где создадим новую папку ***My Web Projects***. Самому же файлу дадим название также ***My Web Project***ю Итак, после того, как проект сохранен на диске, давайте присмотримся к интерфейсу. Если вы обратите внимание, то он немного у нас изменился. Например, в стандартной панели инструментов ***QuarkXPress*** у нас прибавилось еще одна палитра, которая называется ***Web tools*** — инструменты ***Web***. В ней находятся такие инструменты, которые вам пригодятся исключительно для создания интерактивных публикаций и публикации ***Web***. Вообще говоря, при работе с

Web публикацией некоторые возможности ***QuarkXPress*** просто отключаются, т.к. они не нужны для работы с ***Web*** макетами.

Это такие команды, как, например, ***Page Setup***, ***Collect for Output*** или ***Save Page as EPS*** стали недоступными. Ну, это и понятно. В общем-то, данные команды вам абсолютно не нужны для работы с ***Web*** публикацией. Давайте, рассмотрим новую появившуюся палитру, пока что кратко, и узнаем, какие же инструменты у нас добавились для работы с ***Web*** страницами, и что они из себя представляют. Самый первый инструмент в палитре — это инструмент, который называется ***Image Map***. Это средство, позволяющее создавать гиперссылки в графических изображениях, и причем не одну гиперссылку, а несколько. Поэтому инструмент так и называется ***Image Map*** — карта изображения, т.е. карта, которая указывает, в каких местах на изображении находятся те или иные гиперссылки. Данный инструмент у нас присутствует в трех видах, т.е. прямоугольный, круглый и, наконец, Безье.

Сразу же за инструментом ***Image Map*** следуют инструмент ***Form Box*** и инструмент ***File Selection***. Они находятся на одной кнопке, и мы можем таким образом переключать их. Эти два инструмента позволяют создавать два типа форм для ***Web*** страниц. ***Form Box*** создает обычные формы, которые заполняются пользователем. В них, например, может заноситься информация о подписке на рассылку или что-либо подобное. Инструмент же ***File Selection*** создает диалоговое окно, предоставляющее пользователю возможность указать путь при загрузке файла.

Третий инструмент в палитре называется ***Text Field***, что переводится как текстовое поле. Данный инструмент обычно представляет собой однострочную версию инструмента ***Form Box***.Если в инструменте ***Form Box*** формы могли быть любого размера, включая в себя любое количество текста, то в поля, и созданные инструментом ***Text Field*** можно вводить текст только одной строкой. Инструменты ***Button*** и ***Image Button***, как можно догадаться по их названию, служат для создания кнопок. ***Button*** предназначен для создания стандартных кнопок, а ***Image Button*** предназначен для создания кнопок в виде графических объектов.

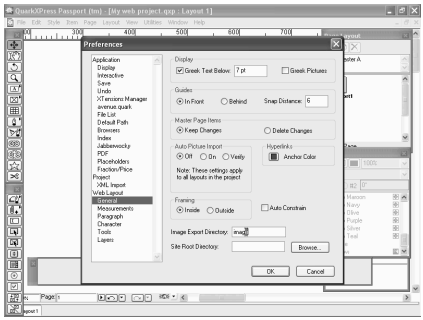
При этом можно использовать любые графические объекты, доступные для ***Web*** макетирования. Следующий инструмент на очереди — это ***Popup Menu***. С его помощью можно создавать раскрывающиеся списки, из которых пользователь может выбирать нужные ему параметры. Инструмент ***List Box*** создает поле, где пользователь может выбрать список элементов, например, для заполнения базы данных, расположенных в Интернет. Включать или отключать различные параметры на ***Web*** странице можно также с помощью, так называемых ***Radio Button*** и ***Check Box***. Эти два элемента предназначены для каких-либо настроек на ***Web*** страницах или выбора параметров. Различие

состоит в том, что ***Check Box*** позволяет включить или отключить параметр. А ***Radio Button*** используется только в списке из нескольких вариантов одного и того же параметра, причем выбран может быть в таком списке только один из пунктов.

Ну и, наконец, последние два инструмента в палитре ***Web Tools*** — это ***Roll Over Linking*** и ***Roll Over Unlinking***. Данные инструменты позволяют устанавливать и разрывать связь для двух позиционных, так называемых, ролловеров, т.е. элементов, которые появляются при наведении указателя мыши на какой-либо элемент. Палитра ***Web Tools*** так же, как и любая другая палитра может быть убрана с экрана. Для этого мы используем команду ***Window***, ***Tools***, ***Hide Web Tools***.

Соответственно для вызова ее на экран мы снова воспользуемся все той же командой ***Window***, ***Tools***, но на этот раз команда у нас будет называться ***Show Web Tools***. Так же как и любые другие палитры, ее можно сворачивать двойным щелчком или перемещать по экрану. Тем не менее, я, конечно, не рекомендую ее убирать с экрана, т.к. палитра ***Web Tools*** содержит большинство нужных инструментов для создания несложных ***Web*** страниц и, скорее всего, вам часто придется ею пользоваться.

6.5.3. Диалоговое окно Preferences. Вкладка Export Выбор браузера. Палитры, ориентированные на Web



Само собой, что отличие ***Web*** проектов от проектов печатных, не ограничивается только палитрой ***Web Tools***. Например, изменились некоторые пункты в диалоговом окне ***Preferences***, в котором мы делаем все настройки, касающиеся ***QuarkXPress***. В частности, вместо раздела ***Print Layout*** здесь у нас появился раздел ***Web Layout***, который содержит, в общем-то, все те же подразделы, но при этом в каждом из них есть небольшие или большие изменения. Давайте, рассмотрим их. Например, во вкладке ***General*** появилось текстовое поле ***Image Export Directory***. Оно предназначено для того, чтобы указать программе ***QuarkXPress***, куда будет выполняться экспортирование изображения для ***Web*** документа. Программе, которая используется для размещения вашей страницы в Интернет, требуются сведения о том, где именно расположены все необходимые изображения. Поэтому этот параметр довольно-таки важен. Если

указать здесь некий путь, то это все равно, что воспользоваться средством ***Collect For Output*** при печати документа. В текстовом поле ***Site Root Directory***, т.е. корневой каталог сайта, вы можете указать расположение начальной страницы вашего ***Web*** сайта. Например, щелкнув на кнопке ***Browse***, мы зайдем в нашу папку ***My Web Projects*** и укажем ее, как корневую папку нашего сайта. Тогда все экспортируемые изображения будут храниться в папке ***My Web Projects*** в подпапке ***Image***.

Также во вкладке ***General*** отсутствуют такие параметры, как ***Auto Page Insertion***, если помните, она использовалась в печатных публикациях для автоматического добавления страниц при размещении текста. А также у нас отсутствует параметр ***Hypper Link Color*** — это цвет гиперссылки. Естественно его здесь нет, потому что он теперь указывается еще при создании нашего ***Web*** макета. А также его можно указать в диалоговом окне ***Layout Properties***, которое мы с вами еще рассмотрим. Вкладка ***Measurements*** — измерение. Здесь отсутствует параметр ***Item Coordinates*** — координаты элементов, так как на ***Web*** страницах не существуют разворотов, в отличие от печатных публикаций.

И, наконец, вкладка ***Trapping*** в диалоговом окне ***Preferences*** для ***Web*** проектов отсутствует вовсе. Но это, наверное, понятно — для экранных публикаций, использование трепинга абсолютно не нужно. Изменения коснулись также диалогового окна ***Modify***. В данный момент оно недоступно, но, давайте, создадим графический блок, импортируем в него какое-нибудь изображение командой ***Get Picture***, ну скажем, вот это, и теперь вызовем для данного изображения диалоговое окно ***Modify***. Здесь, в общем-то, вам все более менее будет знакомо, кроме закладки ***Export***. В закладке ***Export*** мы указываем, куда будет экспортироваться графика в нашем проекте при окончательном конвертировании файла, созданного в ***QuarkXPress***, в обычную ***Web*** страницу.

Вы, наверное, и сами понимаете, что большинство элементов, создаваемых инструментами ***QuarkXPress***, абсолютно несовместимы в ***Web***, поэтому единственный выход при конвертации проекта уже в окончательный вид, т.е.в вид ***Web*** страницы большинство таких элементов экспортировать в графический формат, который будет совместим с ***Web***. Так вот, в закладке ***Export***, мы можем для каждого элемента по отдельности, указать, как его экспортировать, куда его экспортировать.

Ну, вот здесь вы, например, видите, папку ***Image***, т.е. ту самую папку, которую мы задали еще в диалоговом окне ***Preferences***. Можно выбрать также качество картинки. Это в том случае, конечно, если мы используем конвертацию в формат ***JPEG***, т.к. он у нас, как вы знаете, поддерживает различные степени сжатия, ну и от которых, само собой, зависит и качество

картинки как таковой. Кроме того, у формата ***JPEG*** есть еще дополнительный режим, который мы с вами не рассматривали при разговоре о печатных публикациях, т.к. там он не применяется. Это флажок ***Progressive***.

Данный режим вы можете увидеть на очень многих ***Web*** страницах, и это очень легко заметить. Если на какой-либо ***Web*** странице вы видите, что картинка подгружается сразу же, но при этом очень низкого качества, а потом, как бы поэтапно она становится все более и более четкой, то скорее всего это именно изображение в формате ***JPEG Progressive***.Ну и, наконец, еще один нюанс, на который хотелось бы обратить ваше внимание, на кнопку с изображением земного шара в нижней части экрана, рядом с кнопками отмены и повтора операции.

С помощью этой кнопки мы можем предварительно просмотреть, что же у нас получается в любом стандартном браузере, который установлен на нашем компьютере. Причем можно просматривать результат нашей работы сразу в нескольких браузерах. Для этого рядом с кнопкой предварительного просмотра у нас есть кнопка со стрелкой, где мы можем выбрать доступные нам браузеры. Ну, в данном случае это у нас только ***Internet Explorer***, который, как вы знаете, является стандартным модулем операционной системы ***Windows***. Для того чтобы включить предварительный просмотр, мы просто щелкаем на данной кнопке и наша публикация экспортируется в ***Web*** страницу, которая и открывается в браузере ***Internet Explorer***.

Для того чтобы ваша страница корректно отображалась у максимального числа пользователей, желательно, конечно, для предварительного просмотра использовать различные типы браузеров, чтобы исправить ошибку некорректного изображения страницы в каком-либо из них. На самом деле список браузеров для данной кнопки не задается автоматически программой ***QuaekXPress***. Для того чтобы добавить сюда любой другой браузер, который установлен у вас в системе, вам нужно воспользоваться диалоговым окном ***Preferences*** и разделом ***Application***. Здесь есть подраздел ***Browsers*** и, например, для своей системы я могу, кроме ***Internet Explorer***, добавить, скажем, браузер ***Opera***, который также у меня установлен. Для этого я воспользуюсь кнопкой ***Add*** и теперь от меня требуется найти и папку с программой опера и ее главный файл для запуска.

Вот у меня папка ***Opera*** 7, стандартно она садится само собой разумеется на диск С в ***Program Files***, и вот здесь у меня файл запуска программы, который называется ***Opera.exe***. Щелкаем на кнопку ***Open***, и, пожалуйста, браузер добавлен. Теперь закрываем диалоговое окно ***Preferences*** щелчком на кнопке ОК, и давайте заглянем в список браузеров в нижней части окна. У нас здесь появился пункт ***Opera.exe***. Галочка, которая стоит рядом с ***Internet Explorer***, обозначает, что данный браузер всегда будет использоваться по умолчанию,

т.е. просто при щелчке на кнопке с земным шаром страница будет всегда открываться в браузере ***Internet Explorer***. Если же мы хотим воспользоваться ***Opera***, то тогда мы открываем список и щелкаем уже здесь на ее названии. Пожалуйста. Теперь мы просматриваем нашу будущую ***Web*** публикацию через браузер ***Opera***.

Браузер по умолчанию также можно изменить, если вы чаще хотите делать предварительный просмотр в браузере ***Opera***, а не в браузере ***Internet Explorer***, то для этого все в том же диалоговом окне ***Preferences***, в разделе ***Browsers*** щелкаем рядом с ***Opera.exe*** в поле ***Default***, и галочка теперь стоит рядом с браузером ***Opera***.

Благодаря этому, по умолчанию теперь на щелчке предварительного просмотра наша публикация будет открываться в браузере ***Opera***. Ну и, наконец, последний штрих, касающийся отличий ***Web*** макетов от макетов печатных — это наличие дополнительных палитр. Таких палитр три. Это палитра ***Hyperlinks***: она содержит список всех гиперссылок на внешней ***Web*** странице или другие страницы текущей публикации. Что такое гиперссылка, мы уже с вами говорили и подробнее о них мы будем говорить дальше.

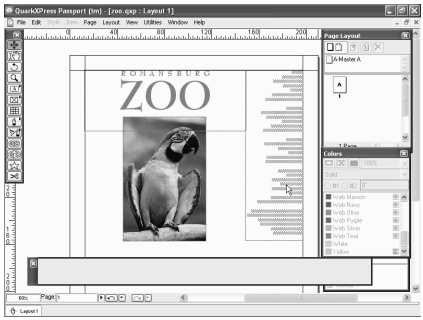
Вторая палитра — это палитра ***Placeholders***. В ней содержится информация о файлах ***XML***. Она содержит записи, определенные для конкретного ***XML*** файла. И, наконец, палитра ***Sequences***, которая содержит в себе список объектов, отображаемых на ***Web*** странице, или связанных с ней. Последовательность, заданная в этой палитре, указывает порядок размещения объектов. В данный момент она у нас пуста, т.к. объект у нас на странице всего один.

Итак, как вы можете видеть, при работе с ***Web*** проектами ***QuarkXPress*** у нас кое-что изменилось по сравнению с печатными проектами. Но основные инструменты, которые служат для размещения графики, текста остались теми же, изменились всего лишь правила, которым мы должны следовать. Естественно, как мы уже говорили, ***Web*** страница намного более ограничена в возможностях по сравнению с обычной печатной публикацией. Об этом всегда следует помнить.

6.5.4. Преобразование печатных макетов в Web формат

В шестой версии ***QuarkXPress*** при работе с ***Web*** макетами появилась новая возможность. Она заключается в том, что вместо того, чтобы создавать макет с нуля, вы можете просто перевести ***Web*** формат в уже существующий печатный макет. Насколько эта возможность является полезной и пригодится она вам или нет, трудно сказать. Сложность заключается в том, что, как вы уже знаете, печатные и электронные публикации сильно между собой отличаются.

Поэтому при конвертации очень много возникает конфликтов, и вам просто-напросто приходиться добрую половину макета все равно переделывать.



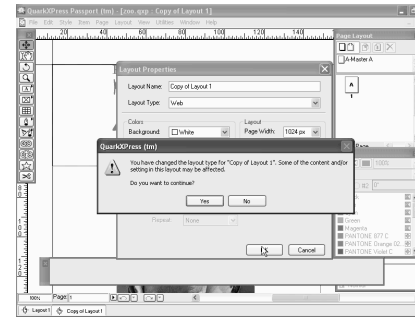
А иногда, это даже не половина, а больше, и фактически получается, что проще макет создать с нуля для публикации в ***Web***, чем конвертировать его из печатной публикации. Но тем не менее у нас такая возможность есть и мы рассмотрим, как она работает, рассмотрим ее плюсы и минусы. Для конвертации мы выбрали простенький, если не сказать, примитивный макет, так как если мы возьмем сложный макет, то в нем будет очень много конфликтов, мы не ставим сейчас целью показать весь процесс переделки макета печатного в макет формата ***Web***, мы хотим просто рассмотреть ключевые моменты этого процесса. В первую очередь, если вы решили конвертировать печатный макет в ***Web*** формат, следует создать копию этого печатного макета, т.к. наверное, в своем печатном виде он вам также нужен и вы, возможно, будете его еще печатать в типографии.

Для того чтобы создать копию макета, мы можем воспользоваться как командой ***Save As***, т.е. просто сохранить текущий файл под другим именем и потом с ним уже экспериментировать. Старая версия макета у нас останется. Или можно сделать еще проще. Как вы помните, в проекте ***QuarkXPress*** могут находиться несколько макетов. Поэтому мы можем просто продублировать текущий макет, сделать его дубликат. Это возможно с помощью команды ***Layout***, ***Duplicate*** — макет, дублировать, и здесь щелкнуть на кнопке ОК. Итак, мы получили с вами копию нашего макета. Она находится в том же проекте, поэтому это даже, в общем-то, удобнее, чем сохранять его в отдельном файле. Т.е. мы можем переключаться между оригиналом и копией.

Следующим шагом для нас будет как таковая конвертация нашего макета из печатного формата в формат для ***Web***. Для этого мы снова войдем в меню ***Layout*** и вызовем диалоговое окно ***Layout Properties***. Окно ***Layout Properties*** представляет собой практически копию окна создания нового проекта. Как мы уже говорили, после создания проекта, можно изменять его параметры. И вот эти параметры мы можем изменять именно с помощью данного окна. Итак, нам нужно перевести печатный макет в макет в ***Web*** формате, и для этого нам достаточно просто в поле ***Layout TYple*** вместо ***Print*** выбрать ***Web***.

Появляются знакомые вам разделы, которые мы видели при создании нового ***Web*** макета. Они один к одному совпадают с тем окном, и, в общем-то,

сейчас нам здесь менять больше нечего. Оставим все настройки, как они есть, и просто щелкнем на кнопке ОК. При конвертации макета в ***Web*** формат, ***QuarkXPress*** всегда спрашивает вашего подтверждения, т.к. это почти необратимая операция. На самом деле, конечно, можно затем конвертировать обратно ***Web*** макет в печатный, но при этом вы не получите в конечном результате то, что у вас было в самом начале. Что-то потеряется, какие-то настройки исчезнут, в любом случае, этого делать не рекомендуется.



В нашем случае мы уверены, что мы собираемся конвертировать наш макет, поэтому нажимаем на кнопку ***Yes***. И буквально через пару секунд мы с вами получаем наш макет в формате ***Web***. Давайте, посмотрим внимательно, что осталось прежним и что здесь теперь изменилось. Помните об одном — это очень важный момент, за которым всегда надо следить. ***C***овсем не обязательно, что макет будет выглядеть в браузере пользователя так же, как он выглядит в ***QuarkXPress***, несмотря на то, что он уже переведен в ***Web*** формат. Вот, обратите внимание на два текстовых блока, которые используются в макете. Они, в общем-то, перишли сюда без каких-либо особых изменений, но единственный штрих — это вот такая иконка фотоаппарата, которая появилась у каждого из них в углу. Что это означает?

Это означает то, что каждый из этих текстовых блоков имеет некое форматирование, несовместимое с публикацией в ***Web***. Что касается, например, нашего заголовка, здесь, в общем-то, все понятно. Для использования в ***Web*** страницах рекомендованы исключительно системные шрифты, такие как ***Arial***, ***Times New Roman***, ***Curier*** и т.д. Само собой разумеется, что данный заголовок у большинства пользователей, по крайней мере тех, у которых отсутствует шрифт ***BernhardMod BT***, отображаться корректно не будет.

Что же делает в таком случае ***QuarkXPress***? При создании макета, видя такой текстовый блок, ***QuarkXPress*** присваивает ему атрибут ***Export To Graphics***. Это означает, что при окончательном экспортировании нашего ***Web*** макета уже в настоящую в ***Web*** страницу, т.е. формат ***HTML***, данный текстовый блок будет перенесен не в виде текста, а в виде изображения. Т.е. будет переконвертирован ***bitmap***

Хорошо ли это, конечно же, нет. ***Web*** страницы создаются в расчете на всеобщий доступ. Они расположены на неких серверах в Интернет и, конечно,

скорость доступа к этим серверам не безграничная. Поэтому, чем меньше занимает место материал на ***Web*** странице, имеется в виду место в КБ и мегабайтах, само собой разумеется, тем быстрее будет загружаться данная страница. Если же мы, даже в таком простейшем макете умудрились получить три ***bitmap***-а, то представьте себе, если мы будем использовать ту же практику при каких-либо сложных проектах. Поэтому здесь, скорее всего, следует поступить иначе.

Во-первых, снять для этого блока атрибут экспорта в графику. Для этого мы вызываем диалоговое окно ***Modify*** и в нижней части раздела ***Text*** отключаем флажок ***Convert to Graphic on Export***, т.е. конвертировать в графику при экспорте. Это первый шаг. Второй шаг — постараемся перевести данный заголовок в какой-либо из системных шрифтов и обработать его так, чтобы он выглядел максимально похоже.

Например, возьмем тот же ***Times New Roman***, вот так, в общем-то, можно считать такой заголовок приемлемым. И второй несовместимый с ***Web*** нюанс данного заголовка — это его цвет. Обратите внимание, здесь используется для него цвет ***PANTONE***, который естественно в браузере понимать не будут. Мы должны изменить цвет этого заголовка на ***Web*** совместимый. В нашей палитре есть базовый набор ***Web*** цветов. Если он нас не устроит, мы можем добавить сюда цвета с помощью диалогового окна ***Colors***.

Это делается также, как и в обычных печатных публикациях. Единственный момент, это то, о чем мы уже говорили, цвета надо добавлять из палитр ***Web Named Colors*** или ***Web Safe Colors***. В данном случае нам добавлять цвет не нужно, мы просто выберем цвет ***Web Silver***. Он максимально соответствует тому цвету, который у нас использовался до этого, или можно даже использовать просто ***Web Gray***. Он подходит даже больше.

Итак, если мы разобрались с заголовком, то теперь можно обратить внимание и на небольшую колонку текста в правой части страницы. Этой колонке ***QuarkXPress*** при конвертации также присвоил атрибут конвертации в графику, но здесь для этого объективных причин я не вижу. Т.к. у нас в этой колонке используется шрифт ***Arial***, который является системным, и цвет у него стандартный — черный цвет.

Здесь может быть еще один маленький нюанс. Если в тексте используется какое-нибудь нестандартное форматирование, скажем, кроме атрибутов ***Bold***, ***Underline*** или ***Italic*** присвоены еще какие-либо не совсем стандартные для ***Web*** атрибуты текста, в таком случае ***QuarkXPress*** у нас реагирует соответствующе, или, например, может быть задана окантовка для самого текстового блока. Ни того, ни другого здесь нет, поэтому мы просто сейчас в окне ***Modify*** отключим флажок ***Convert to Graphic on Export***, вот таким образом. Как видите, значок фотоаппарата исчез. И последний элемент на

нашей странице — это фотография. Но, в общем-то, фотография у нас экспортировалась без проблем, поэтому я думаю, что ее дополнительно обрабатывать в данном случае не надо. По крайней мере у нас сейчас промежуточный момент, когда мы можем в предварительном просмотре посмотреть, что же у нас получается из данной страницы на самом деле в браузере.

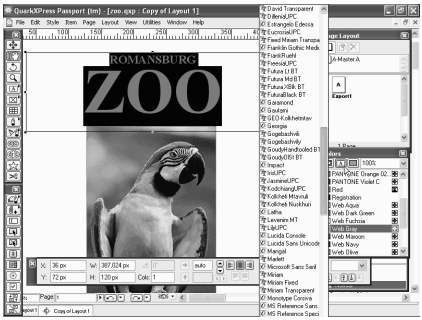
Давайте щелкнем на кнопке предварительного просмотра. И вот, такой приблизительно у нас получается ***Web*** макет при опубликован в ***Web*** и прссмотре через браузер. Единственный нюанс, на который хотелось бы здесь обратить внимание, — это то, что заголовок, как видите, расширился, т.е. увеличилось расстояние между строками, по сравнению с исходником. Почему? Потому что в исходнике у нас средствами ***QuarkXPress*** насильно уменьшен интерлиньяж между этими двумя строками, а при конвертации в формат ***HTML*** этот интерлиньяж был задан по стандарту, поэтому у нас получился вот такой эффект. Для того чтобы этого избежать, можно, например, эти два слова, эти две строки расположить в разных текстовых блоках, тогда на них не будет влиять интерлиньяж и расстояние между ними не будет изменяться при конвертации.

Итак, исходя из того, что мы с вами узнали на данный момент, получается, что мы можем конвертировать обычные публикации ***QuarkXPress*** в ***Web*** формат, но это связано со многими сложностями и нюансами. В частности, в переделке цветов, перенесенных из печатного макета в цвета, соответствующие стандарту ***Web***, удаление форматирования, не поддерживаемого на ***Web*** страницах, смена шрифтов, используемых в макете для печати на стандартные шрифты, которые будут поддерживаться любым браузером с любого пользовательского компьютера.

Единственное, о чем можно не беспокоиться, во всяком случае начинающим пользователям, — это об изображениях. При экспортировании ***Web*** макета в формата ***HTML***, ***QuarkXPress*** преобразовывает графические изображения, как импортированные в ***QuarkXPress***, так и созданные в самом ***QuarkXPress***, в соответствующие форматы, используемые в ***Web***. Поэтому вам об этом беспокоиться не стоит

6.5.5. Стилизованный текст

Итак, мы узнали с вами, что нужно делать для того, чтобы наш ***Web*** макет отображался корректно при открытии его в браузере, что нужно изменять, какие стандарты поддерживаются в ***Web***-е и, какие не поддерживаются. А теперь давайте рассмотрим вопрос обратный. Что, если мы имеем элемент, который не можем изменять под стандарты ***Web***?



Не можем, потому что, например, этот элемент является логотипом фирмы, который не подлежит изменению, его стилистика уникальна, и менять в нем ничего нельзя. Например, мы вернем сейчас нашу надпись в то состояние, в котором она была сразу после конвертации нашего печатного макета в ***Web*** макет. Там использовался шрифт ***BernardMod BT***. Ну и, кроме того, мы еще переделали цвет нашей надписи. Можно представить такую ситуацию: если это логотип, то логотип должен сохранять свой цвет, т.к. цвет тоже является фирменным признаком логотипа и его менять ни в коем случае нельзя. В самом крайнем случае должен быть подобран цвет, максимально соответствующий оригинальному. На самом деле в ***QuarkXPress*** для этого есть достаточно возможностей. Программа предлагает нам два компромисса. Первый компромисс — это конвертировать элементы, которые вы не хотите изменять при экспорте в ***HTML*** в графику.

Помните, мы в предыдущем занятии мы отключали этот параметр. Сейчас мы сделаем наоборот. Мы этот параметр включим для нашего заголовка. Для этого вызовем окно ***Modify*** для этого текстового блока и в разделе ***Text*** в нижней части экрана включим параметр ***Convert to Graphic on Export*** — конвертировать в графику при экспорте. После щелчка на кнопке OK, мы можем видеть значок фотоаппарата, появившийся в углу текстового блока. Что происходит при этом? Мы можем взять любые элементы на странице, которые, возможно, в своем оригинальном виде не будут корректно изображены при просмотре в браузере, и указать для них вот этот атрибут.

Благодаря нему, при экспорте в ***HTML*** нашего макета все элементы с этим атрибутом будут преобразованы в графику, которая уже, само собой разумеется, будет просматриваться корректно любым браузером. Давайте сейчас убедимся в этом, щелкнув на кнопке предварительного просмотра. Как мы можем видеть, вид нашей надписи сохранился таким, каким он должен быть, единственный нюанс — вы можете обратить внимание, что надпись у нас получилась чуть грубоватая. Но это неизбежно, т.к. мы конвертировали векторный шрифт в растровое изображение.

Естественно растровое изображение выглядит не настолько идеально. С этим поделать, к сожалению, ничего нельзя. Второй вариант решения данной проблемы — это использование каскадных таблиц стилей. Что такое каскадные таблицы стилей? Прежде чем это выяснить, давайте уясним для

себя вот что — любой браузер оперирует не названиями шрифтов, а шрифтовой классификацией. Т.е. для любого браузера на странице существует всего лишь 5 видов шрифтов.

Это шрифт с засечками, шрифт без засечек, моноширинный шрифт, декоративный шрифт, и шрифт для заголовка. Поэтому мы, само собой разумеется не можем предугадать, какие шрифты будут стоять у пользователя. И даже когда мы используем, скажем, в заголовке или в тексте стандартный системный шрифт, такой как ***Times New Roman*** или ***Arial***, то даже в этом случае нет гарантии, что пользователь увидеть именно ***Times New Roman***, потому что у него браузер может быть перенастроен так, что вместо ***Times New Roman*** он будет отображать, например, шрифт ***Georgia***. Что же такое каскадные таблицы стилей? С помощью каскадных таблиц стилей мы можем задать приоритеты подмены шрифтов. Что это означает? Например, мы используем в нашей публикации шрифт ***BernardMod BT***.

Давайте, зайдем в пункт меню ***Edit*** и найдем в нем команду ***CSS Font Families*** — семейство шрифтов ***CSS***. Аббревиатура ***CSS*** — это и есть каскадные таблицы стилей — ***Cascade Style Sheets***. После щелчка мышью мы можем увидеть на экране вот такое окошко. Давайте, разберемся, что же здесь есть. У нас здесь есть список шрифтов, причем некоторые из них активны, некоторые нет. И если мы внимательно присмотримся, все эти шрифты — это системные шрифты ***Windows XP***. Шрифты ***Arial***, ***Courier***, ***Courier New***, ***Georgia***, ***Helvetica***, ***Times***, ***Times New Roman***.

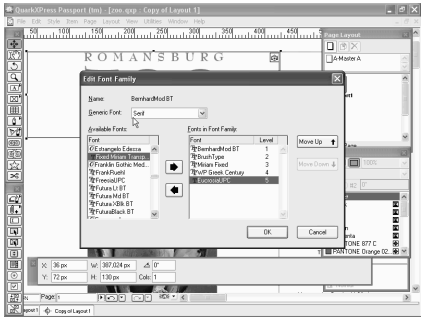
В данном окне мы можем настроить следующий процесс. Если у меня на ***Web*** странице упоминается шрифт ***Arial***, и он отсутствует у пользователя, то я могу представить еще дополнительный список шрифтов, который следует по приоритету, и из этого списка шрифтов будет подставлен любой шрифт, присутствующий у пользователя в компьютере.

Например, давайте посмотрим, какие настройки у нас для шрифта ***Arial***. В нижней части экрана мы видим следующую надпись. Первая — ***Arial***, вторая — ***Helvetica Generic Font Sans-serif***. Или, например, для шрифта ***Times New Roman***: первая — ***Times New Roman***, вторая — ***Times Generic Font***: ***Serif***. Что это означает? Это означает, что любая надпись, которая сделана шрифтом ***Times New Roman*** в нашем проекте будет отражаться им в том случае, если шрифт ***Times New Roman*** установлен в компьютере пользователя и он установлен, как шрифт с засечками по умолчанию. Если такого шрифта нет, то следующий за ним по приоритету идет так называемый ***Times Generic Font***, это стандартный шрифт ***Times*** с засечками, который установлен в системе. Какой это шрифт будет, неизвестно.

Мы можем добавить от себя дополнительные шрифты. Например, возьмем тот шрифт, который мы используем в нашем заголовке. Щелкаем на кнопке

New, находим в списке шрифтов наш шрифт ***BernardMod BT***, и для того чтобы добавить его в список, щелкаем вот на этой стрелке, указывающей вправо. На этом можно, в общем-то, и закончить.

Единственное, что остается выбрать для этого шрифта, — это какого типа системный шрифт будет использоваться при его отсутствии в компьютере пользователя: стандартный шрифт с засечками ***Serif***, без засечек ***Sans-serif***, ***Cursive*** — заголовочный, ***Fantasy*** — декоративный или ***Monospace*** — моноширийный. Но на самом деле этим все не ограничивается. Мы можем добавить в данный список любое количество шрифтов. Но при этом, конечно, на самом деле следует добавлять шрифты, которые похожи на данный шрифт.



Сейчас, конечно, мы не будем копаться в списке шрифтов и искать, какие из этих шрифтов похожи и какие нет, мы просто добавим их наугад вот таким образом. Главное здесь — понять принцип действия данного процесса. Итак, в списке у нас 5 шрифтов. Изначально в документе у нас используется шрифт ***BernardMod BT***. Если он отсутствует на компьютере пользователя, то следующим после него по приоритету следует шрифт ***Brush Type***, и система будет искать этот шрифт. Если он у пользователя есть, то она его подставит. Если шрифта ***Brush Type*** нет, то соответственно будет искаться следующий шрифт по приоритету. И т.д., пока не будет найден шрифт из списка, или не закончится список шрифтов и ни один шрифт не будет найден.

Тогда в самом конце будет подставлен стандартный системный шрифт с засечками, который у нас вот здесь указан. На самом деле сам принцип действия каскадных таблиц стилей и семейств шрифтов, конечно, очень удобен, но правда состоит в том, что у большинства пользователей, если, конечно, они не занимаются дизайном полиграфии, установлен стандартный системный набор шрифтов.

И поэтому, подставляй шрифты или не подставляй, скорее всего, у пользователя их все равно не будет. Поэтому более действенным способом, хоть и не очень-то удобным в смысле получающихся больших размеров ***Web*** страницы, является перевод нестандартных надписей просто в графические изображения. Но перебарщивать с этим, конечно, не стоит. Т.е., если вы переведете все ваши текстовые элементы на странице в графику, то размер вашей страницы достигнет таких высот, что страница будет загружаться у среднего пользователя, который выходит в Интернет по модему через

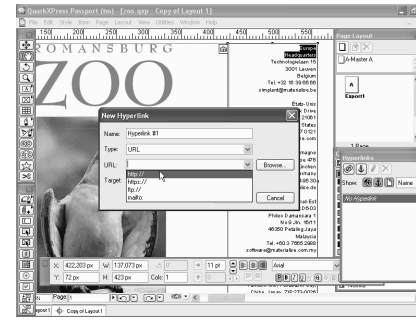
телефонную линию, в течение 15-20 минут. Конечно же, это не очень хорошо. Т.е. в большинстве случае пользователь просто не дождется конца загрузки страницы и переключиться на что-нибудь другое.

6.5.6. Создание гиперссылок

Один из важнейших элементов, по принципу которого работают ***Web***, — это гиперссылки. В общем-то, благодаря ним, мы можем перемещаться по Интернет между различными ***Web*** страницами, географические расположенными абсолютно на противоположных концах земного шара. В ***QuarkXPress*** для создания гиперссылки не нужно прилагать особенно много усилий. Гиперссылки могут быть как текстовыми, так и графическими. Но мы с вами пока рассмотрим текстовый вариант. Итак, мы выделяем тот текст, который у нас будет гиперссылкой. В данном случае не важно, какой. Вот выделить две эти строчки.

Затем используем палитру ***Hyperlinks*** — она у нас в данный момент не отображена. Давайте откроем ее. Напомню, это делается командой ***Window***, ***Show Hyperlinks***. В этой палитре щелкаем на кнопке ***New Hyperlink*** — создание новой гиперссылки. При щелчке на кнопке появляется окошко, где есть 4 поля. В первом поле ***Name*** можно ввести произвольное имя, оно ни на что не влияет. Оно предназначено только для того, чтобы затем в палитре ***Hyperlinks*** отслеживать те ссылки, которые у нас расположены на странице. Давайте, назовем эту ссылку ***Hyperlink*** #1.

В следующем поле ***Type*** нам предлагается выбрать, какой вид гиперссылки мы хотим создать, на что она будет ссылаться. Она будет ссылаться на ***URL***, т.е. адрес некой страницы. Она будет ссылаться на страницу, находящуюся в том же текущем проекте, в котором мы с вами работаем, или она будет ссылаться на ***Anchor***. Что такое ***Anchor***, мы поговорим с вами позже.



Пока что, давайте, оставим значение ***URL*** и в следующем поле, которое так и называется ***URL***, мы должны ввести абсолютный или относительный адрес расположения страницы, на которую будет вести эта ссылка. Дело в том, что если ссылка ведет на страницу того же проекта, т.е. вашего же сайта, то, скорее всего, вам следует использовать относительную ссылку, которая не использует полный Интернет адрес, т.е. начиная с ***http*** и прочее и прочее, адрес сайта, название страницы и т.д. Она использует только имя ***HTML*** файла, который обычно, если страницы

находятся на одном и том же сайте, расположены в одном и том же каталоге или в каталогах, вложенных в корневой каталог вашего сайта.

Кроме того, ссылка может вести не только на ***Web*** страницу, но и на ***FTP*** сайт. Вот у нас здесь есть даже подсказки. Т.е. если мы собираемся сделать ссылку на ***FTP*** сайт, то мы щелкаем на ***FTP*** и потом вписываем дальше адрес сайта. Или ссылка может быть предназначена для того, чтобы при щелчке на ней у пользователя открывалась его почтовая программа, открывалось автоматически новое письмо, в котором будет уже вписан почтовый адрес, который вы введете после слова ***mailto***.Т.е. фактически это создание ссылок на подобие тех, которые обычно расположены на стартовых страницах сайтов в нижней части: это контактная почта ***Web*** мастера или создателя сайта для того, чтобы пользователь, заходящий на страницу мог бы с ним связаться. И, наконец, последнее поле — это поле ***Target***.

Здесь также фиксированные значения, которые можно ввести и, в общем-то, его можно даже оставить пустым. Если же мы выберем одно из 4 значений, которое у нас здесь есть, т.е. ***blank***, ***self***, ***parent*** или ***t***ор, то соответственно, в случае бланк страница, на которую будет зачислен переход по ссылке, откроется в новом окне браузера. Если будет указано значение ***self***, то в случае, если на вашем сайте используются фреймы, то страница будет открыта в текущем фрейме.

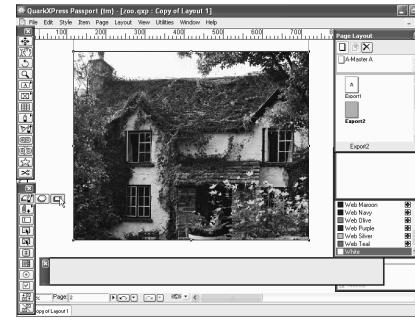
Если фреймы не используются, то страница будет открыта в текущем окне браузера. Значение ***parents*** также касается только тех сайтов, где используются фреймы. При значении ***parents*** открывающаяся страница будет открыта в родительском фрейме по отношению к тому фрейму, где расположена ссылка. И, наконец, значение ***t***ор, которое выполняет практически ту же функцию, что и ***self***, но в отличие от нее есть на странице фрейма или нет, она открывает страницу по ссылке в том же окне, и при этом все фреймы убирает, если они есть. Итак, давайте оставим это поле тоже пустым, так как в общем-то необязательно указывать значение ***Target*** и введем какое-нибудь ***URL***

Ну, на данном этапе у нас нет страницы дополнительной, на которую мы могли бы создать ссылку. Давайте, договоримся так, что в дальнейшем мы создадим страницу с именем, скажем, ***animals.htm*** и щелкнем на кнопке ОК. Итак, что же мы с вами получили. В палитре ***Hyperlinks*** у нас появилась новая запись с названием ***Hyperlink*** #1, т.е. это то имя, которое мы ввели при создании гиперссылки, а текст, который был у нас выделен, превратился теперь в гиперссылку и выглядит, как стандартная гиперссылка по умолчанию.

Давайте, просмотрим нашу страницу в браузере. Как видите, ссылка у нас появилась и здесь и при подведении курсора к ней, он у нас превращается в знакомую всем пользователем Интернета ручку с пальчиком. Теперь при щелчке на этой ссылке, нас ***Internet Explorer*** пытается перебросить на

страницу, которая в ней указана. Но этой страницы пока что не существует, поэтому мы получаем сообщение, что эта страница не может быть отображена. Гиперссылки — это основа любой страницы, поэтому относитесь к ним внимательно. Наличие большого количества гиперссылок, при этом ссылающихся на различные страницы и перекрещивающихся, очень легко запутаться. Для этого используете палитру ***Hyperlinks***, чтобы всегда видеть список гиперссылок, используемых в документе.

6.5.7. Создание карты изображения



Гиперссылки на вашей ***Web*** странице могут быть не только текстовыми, но и графическими. Т.е. мы можем назначить гиперссылку для любого изображения на нашей странице. Для того чтобы это сделать, мы поступаем практически точно так же, как и в случае с текстом. Только на этот раз мы выделяем не текст, а изображение на странице, и после этого щелкаем на кнопке ***New Hyperlink*** в палитре ***Hyperlinks***. Весь остальной процесс, в общем-то, здесь повторяется, так что мы не будем его рассматривать.

Только, давайте вспомним, что в поле ***Name*** вводим название гиперссылки, которая появится в палитре ***Hyperlinks***. Тип гиперссылки может быть ***URL***, ***Page*** или ***Anchor***, т.е. адрес в Интернете, локальная страница или анкер, сам адрес, по которому будет указывать ссылка и наконец, где будет отображаться страница, которая называется данной ссылкой — в данном же окне, в другом окне и т.д. Но изображения нам позволяют использовать сразу несколько гиперссылок.

Это называется созданием карты изображения. Что же это такое, давайте, мы сейчас с вами рассмотрим. Для этого мы в нашем макете создадим новую страницу, используя палитру ***Page Layout***, мы выполним команду ***Insert Pages*** и получим новую страницу. По умолчанию она будет называться ***Export*** 2. Перейдем на нее и чтобы долго не мучаться создадим здесь графический блок, в который импортируем изображение.

Теперь увеличим размер блока таким образом, чтобы он совпадал с размерами изображения, и расположим изображение приблизительно по центру страницы. Теперь можно еще задать заголовок для страницы. Назовем ее, например, ***My House***. Вот таким образом. Мы зададим для заголовка шрифт ***Times New Roman***, отцентруем его приблизительно по изображению, и вот

наша, в общем-то, такая примитивная страница готова. Что здесь можно теперь сделать? Как я уже упоминал, мы будет работать с вами с картой изображения. Что такое карта изображения? Как мы только что говорили, мы можем использовать любое изображение на ***Web*** странице, будь то импортированная фотография или любое изображение, созданное средствами ***Color Draw***.

Использовать мы любые изображения можем для гиперссылок. Ккак это делается, мы только что с вами рассмотрели. Но, кроме того, мы можем создавать на одном изображении сразу несколько гиперссылок. Для этого мы будем использовать карту изображения. Карта изображения позволяет нам обозначить на изображении некие области, каждая из которых будет привязана к определенной гиперссылке. Для этого в ***QuarkXPress*** у нас есть инструмент ***Image Map*** — карта изображения. Причем здесь у нас имеется три типа инструмента — это прямоугольник, овал, и инструмент Безье.

Создается карта изображения очень просто. Мы фактически рисуем на изображении области, которые у нас будут работать, как гиперссылки. И как вы уже догадались, области эти могут быть прямоугольными, овальными или, в общем-то, любой формы, если мы будем использовать для этого инструмент Безье. Работают все эти три инструмента по такому же принципу, как и собратья, которые отвечают для создания обычных графических элементов. Ну, мы с вами для ускорения работы используем обычный прямоугольник, а теперь, давайте, посмотрим на само наше изображение. Когда мы создаем карту изображения, то области, которые у нас будут служить гиперссылками, должны быть привязаны логически к каким-нибудь объектам на изображении. У нас здесь имеется изображение дома, и мы можем использовать окна этого дома, как объекты для гиперссылок.

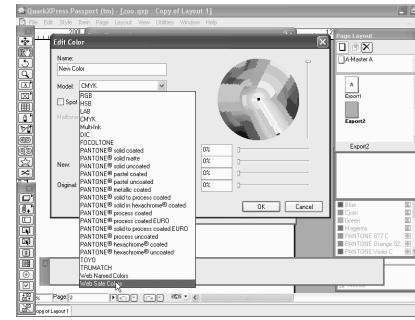
Для этого мы поверх каждого окна нарисуем область, которая у нас будет служить гиперссылкой. Приблизительно это будет выглядеть вот так. Это одна область. Это вторая. Эта третья. Ну и, наконец, мы можем четвертую область создать еще и на входе в дом. Вот, таким образом. Как видите, на изображении у нас появились прямоугольники. они полупрозрачные и имеют красный контур. Каждый из этих прямоугольников мы можем использовать как гиперссылку. Назначить гиперссылку теперь очень просто. Выделяем нужную область и в палитре ***Hyperlinks*** в гипперссылке создаем новую гиперссылку. В дальнейшем мы рассмотрим с вами на практике, каким образом можно применять карты изображения для построения вашего сайта.

Обратите внимание также на то, что в верхнем правом углу изображения появился значок с прямоугольником. Он указывает на то, что в данном изображении используется карта ссылок. Если вам понадобится уже после создания областей на изображении удалить какую-либо из них, это сделать

очень просто. Его достаточно выделить щелчком мыши и нажать на кнопку ***Delete*** на клавиатуре. Если же вы хотите удалить сразу все области на изображении, то для этого можно использовать правый щелчок мыши на изображении и команду ***Delete All Hot Areas*** — удалить все области. Вот таким образом. Кстати, если дизайн вашей страницы покажется вам слишком пресным на белом фоне, то, как вы помните, вы можете использовать цвет для вашего фона

6.5.8. Установка фона страницы. Использование изображения в качестве фона страницы

Если мы задали не тот фоновый цвет изначально при создании проекта, то теперь мы можем менять фоновый цвет для каждой страницы по отдельности. Например, для данной страницы мы можем изменить фоновый цвет с белого на какой-либо другой с помощью команды ***Page***, ***Page Properties***. Данное диалоговое окно, как вы видите, очень похоже на диалоговое окно ***Layout Properties*** — свойств макета. Но в данном случае оно касается только текущей страницы. Не забываете об этом, т.к. изменения внесенные здесь отразятся только на данной странице. Все остальные страницы макета останутся такими, какие они и есть. Давайте рассмотрим, что здесь у нас есть. В общем-то, в данном случае нас интересует фоновый цвет страницы. Он находится в разделе ***Colors*** и здесь у нас имеется выпадающий список ***Background***. В данный момент здесь стоит значение ***White***. Вместо него мы можем выбрать любой из доступных нам цветов. Но, как мы уже говорили, желательно, чтобы это был цвет, совместимый с ***Web***. Мы можем выбрать его из выпадающего списка.



Можно также создать новый цвет, использовать палитру ***Web Named Colors*** или ***Web Safe Colors***, например, и выбрать отсюда тот цвет, который нас устраивает. Например, вот этот. И вот так теперь выглядит наша страница. Теперь, если мы, скажем, перейдем на вторую страницу макета, то вы увидите, что здесь остался фоновый цвет белый. На самом деле можно изменить фоновый цвет сразу для всех страниц макета. Вы помните о шаблонных страницах в печатных публикациях ***QuarkXPress***. Так вот, в ***Web*** макетах тоже есть свои шаблонные страны, на которых вы можете размещать какие-то

элементы, которые должны повторяться на всех страницах макета. Так вот, перейдем на шаблонную страницу, она у нас в данном случае присутствует в палитре ***Page Layout*** под названием ***A-Master-A***, и зададим для нее в свойствах любой другой фоновый цвет вместо белого, например, вот такой.

Как видите, все страницы вашего макета будут иметь такой фоновый цвет, за исключением тех, на которых вы уже фоновый цвет меняли вручную. Как видите, вот на эту страницу, на которой мы фоновый цвет с вами поменяли, изменения на шаблонной странице не действует. Ну, и, наконец. для ***Web*** страницы, вместо цвета фона мы можем задать фоновое изображение. Это означает, что вместо, скажем, сплошной голубой подложки мы можем использовать для нашей ***Web*** страницы некое изображение, которое будет служить нам фоном. Как это делается? Давайте, обратимся опять же к диалоговому окну ***Page Properties*** — свойства страницы, и посмотрим на нижнюю часть этого окна, где у нас есть флажок ***Background Image*** — фоновый рисунок.

В данный момент этот флажок отключен. Мы включим его щелчком мыши, и, как видите, активизировалось у нас два поля. В первом поле мы должны указать файл, содержащий изображение, которое будет использоваться, как фоновое. Воспользуемся кнопкой ***Browse*** — просмотр и найдем наше изображение. Такие изображения у нас хранятся в папке ***Textures*** — текстуры. Фоновое изображение в большинстве случаев — это специализированное изображение, которое называется текстурой.

Текстура — это некое небольшое изображение, содержащее некий орнамент или что-либо подобное и причем, позволяющее использовать себя в качестве фона. Что это значить, мы сейчас с вами рассмотрим. Возьмем, например, изображение под названием ***brick***007, щелкнем на кнопке ***Open***. Здесь у нас в строке автоматически прописался путь к этому изображению, и теперь, давайте, обратим внимание на второе поле, это поле ***Repeat*** — повтор.

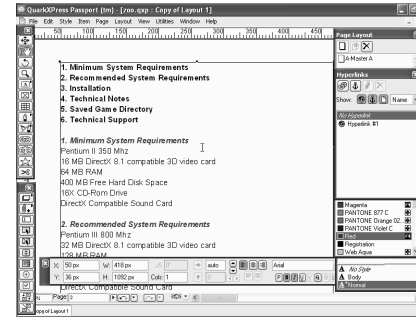
В данный момент стоит значение ***None***. Это означает, что если мы оставим все так, как сейчас есть, то у нас просто появится указанное нами изображение в единственном экземпляре посередине страницы, абсолютного смысла для нас не имеет, так как мы хотим создать фон. Конечно, можно взять большое изображение с высоким разрешением, которое у нас займем целиком всю страницу. Но здесь есть другой минус, который заключается в том, что такое изображение будет занимать очень много места. Если, конечно, у пользователя быстрый Интернет, то это для него большого значения не имеет. А вот для таких пользователей, которые выходят в Интернет с помощью модемов, у которых скорость не выше, чем 56 км/ сек, ваша страница будет грузиться долгие и долгие минуты. Поэтому для фоновых изображений применяются вот такие маленькие картинки, которые можно просто размножить по всей

странице. И вот для этого у нас есть здесь пункт ***Repeat***. В данном пункте у нас есть три варианта — ***Vertical***, ***Horizontal*** и ***Tile***.

Значение ***Vertical*** и ***Horizontal***, как вы уже наверное догадались, позволяют размножить ваше изображение по странице или вертикально, сверху вниз, или горизонтально, слева направо. Но, в случае нашего изображения нам нужно чтобы оно было размножено по принципу, скажем, декоративной плитки, которые украшают, например, стены дома или пол. Самым простейшим примером может быть кафельная плитка у вас в ванной. Итак, выбираем значение ***Tile*** и щелкаем на кнопке ОК. Получаем фоновый рисунок, созданный из большого количества одного и того же изображения, размноженного по всей нашей странице.

Обратите внимание на то, что когда включен фоновый рисунок, то уже цвет фона никакого значения не имеет, т.к. он в любом случае уже не виден. Фоновое изображение для страницы можно в любой момент поменять, воспользовавшись опять же диалоговым окном ***Page Properties*** и выбрав какое-либо другое изображение. Например, вот это, имитирующее черный мрамор. Если же вы хотите отказаться от фонового изображения вовсе и вернуть обычный фоновый цвет, то опять же через окно ***Page Properties*** мы просто отключаем флажок ***Background Image*** и щелкаем на кнопке ОК. Фоновое изображение исчезает и к нам возвращается наш фоновый цвет, который мы установили до этого.

6.5.9. Анкеры



Как мы уже говорили, гиперссылки бывают трех типов. Это локальные гиперссылки, которые указывают на рисунки, находящиеся на нашем же сайте. ***URL*** — это гиперссылки, которые указывают на сайт или страницу в Интернете и, наконец, анкеры, о которых мы с вами сейчас поговорим. Что представляют собой анкеры. Начнем от противного и рассмотрим документ, который у нас расположен на экране. Что он из себя представляет? Он представляет собой некий текст, который разбит на параграфы с подзаголовками, и оглавление данного текста, в котором перечисляются все его параграфы.

Что требуется получить? Требуется получить ***Web*** страницу, в которой данное оглавление будет представлять собой гиперссылки. При щелчке на

каждой из них, должен производиться переход к соответствующему параграфу. С помощью обычной гиперссылки, которая ссылается на локальные страницы, это сделать невозможно, т.к. ссылаться мы будем не на другую страницу, а просто на какое-то конкретное место на текущей же странице. И вот для этого, мы можем использовать анкеры. Что такое анкеры? Анкеры — это некие места на ***Web*** странице, каждое из которых имеет свое уникальное имя и на которое можно производить переход по гиперссылке.

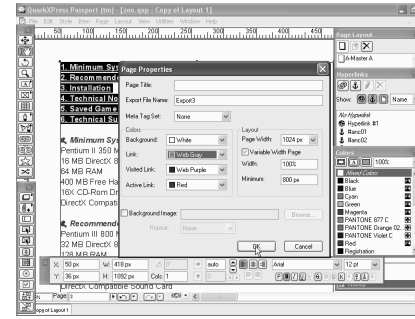
В нашем случае мы должны расставить анкеры на каждом из подзаголовков нашего текста, для того чтобы потом на них ссылаться с помощью гиперссылок в начале страницы. Как это делается? Для начала, конечно же, выделяем подзаголовок, а затем создаем анкеры. Его создание производится через палитру ***Hyperlinks*** опять же. Только на этот раз вместо кнопки ***New Hyperlink***, мы используем кнопку ***New Anchor***, что в переводе с английского означает якорь. Соответствующее изображение вы видите на кнопке. Итак, щелкаем на кнопке ***New Anchor*** и в появившемся окне вводим для него название. По умолчанию ***QuarkXPress*** предлагает название анкера по тексту, который у нас выделен.

Вместо него мы можем просто задать, например, такое название ***anc***01. Следующий анкер мы обозначим соответственно ***anc***02, ***anc***03 и т.д. Щелкаем на кнопку ОК, анкер создан. Он появился у нас в палитре ***Hyperlinks***, и мы можем его в дальнейшем редактировать. Кроме того, рядом с текстом, как видите, появился вот такой значок в виде стрелки. С помощью этого значка мы можем определить, в каких местах на странице у нас расставлены анкеры. Давайте, расставим анкеры для остальных подзаголовков на нашей странице, соответственно ***anc***02, ***anc***03, ***anc***04, 05 и, наконец, 06. На самом деле это только половина процесса. После того, как анкеры расставлены, ими нужно как-то воспользоваться. И вот теперь мы будем создавать гиперссылки, которые будут указывать на данные анкеры.

Для гиперссылок мы используем текст вот этого маленького оглавления. Соответственно каждый пункт оглавления будет соответствовать параграфу с установленным в нем анкером. Итак, выделяем первый пункт оглавления и создаем теперь уже обычную гиперссылку. Единственное отличие от тех видов ссылок, которые мы с вами создавали раньше, это то, что в поле ***TYpe*** мы вместо ***URL*** или ***Page***, указываем ***Anchor***, т.е. ссылка у нас будет указывать на анкер. И теперь обратите внимание на то, что следующий пункт поменял название.

Он теперь называется ***Anchor***, и если мы щелкнем на выпадающем списке, то мы увидим список всех анкеров, которые у нас установлены на данной странице. Соответственно для этой ссылки у нас будет первый анкер, анкер под названием ***anc*** 01. Соответственно расставим гиперссылки для всех

остальных пунктов оглавления. Кроме того, обратите внимание, что при создании гиперссылки она автоматически приобретает синий цвет и подчеркивание. Как вы помните, мы при создании нашего проекта указываем стандартные цвета для ссылок.



Тем не менее, если вас не устраивают конкретно для данной страницы стандартные цвета, то вы можете их изменить. Итак, гиперссылки созданы. Теперь теоретически при щелчке на каждой из этих гиперссылок будет происходить переход к соответствующему параграфу в тексте. Давайте, как мы уже сказали, изменим цвет наших гиперссылок и вообще их стилистику. Для этого мы вызовем уже знакомое нам диалоговое окно ***Page Properties*** и зададим для ссылок вместо синего цвета какой-либо другой. Например, серый. Щелкнем на кнопке ОК. И вот так примерно будет выглядеть наша ***Web*** страница.

Давайте теперь посмотрим, будет ли она работать и, как она будет вообще выглядеть уже в браузере. Для этого, напомню, мы используем кнопку в нижней части экрана, щелкаем на ней, и наша страница открывается в браузере ***Internet Explorer***. В предварительном просмотре у нас все ссылки пометились, как уже посещенные, поэтому они приобрели пурпурный цвет. Это, в общем-то, для нас в данный момент неважно. Давайте проверим, как они вообще работают. Щелкнем на любой из ссылок, например, на 5 пункте, и убедимся в том, что ссылка работает. Теперь для возврата назад мы можем прокрутить страницу, и вернуться обратно к оглавлению.

На самом деле, подобного рода документы всегда в конце каждого параграфа содержат ссылку, для того чтобы вернуться обратно в начало страницы. Давайте вернемся к редактированию страницы и создадим здесь новый анкер. Создадим его для самой первой строчки нашего оглавления. Она уже содержит гиперссылку, но одно другому не мешает, т.е. один и тот текст может служить и гиперссылкой и анкером. Итак, создаем новый анкер. Называем его ***top***, а теперь создаем новую ссылку в конце первого параграфа. Мы наберем текст, который будет приблизительно звучать следующим образом — ***Top of Page***. Создадим на его основе гиперссылку, которая будет указывать на анкер вверху страницы, который мы с вами только что создали. Щелкаем на ОК, гиперссылка создана.

Теперь мы просто скопируем ее с помощью команды Copy и ***Paste***. Видимо у нас слишком короткий текстовый блок. Давайте мы его растянем. Итак, мы

теперь скопируем нашу ссылку в конец каждого из параграфов. Вот таким образом. Это делается для удобства пользователя, чтобы тот, особенно если документ длинный, при переходе, скажем, в конец страницы, не прокручивал затем весь документ обратно наверх, чтобы вернуться к оглавлению, а мог сделать это одним щелчком. Итак, мы расставили с вами ссылки и теперь давайте снова включим предварительный просмотр страницы в браузере. Итак, щелкаем на пункте в оглавлении, переходим к соответствующему параграфу, а теперь для того чтобы вернуться наверх, вместо прокручивания, используем ссылку ***Top of Page*** и возвращаемся обратно.

Кстати, в данном случае смысл отображения посещенных или не посещенных ссылок теряется. Поэтому можно избавиться от этого неприятного пурпурного цвета и заставить ссылки не менять цвет вне зависимости от того, посещал их пользователь или нет. Для этого, давайте, снова вернемся обратно к редактированию нашей страницы и опять же войдем в свойство страницы, где, как вы помните, есть пункты цвета ссылки, и есть пункт цвета посещенных ссылок. Пункт цвета посещенных ссылок у нас пурпурный, а мы можем вместо этого задать цвета для обычной и посещенной ссылки абсолютно одинаковые. Вот таким образом. Теперь вне зависимости от того, посещались ссылки или не посещались, они цвет свой менять не будут. И документ будет выглядеть именно так, как мы хотели.

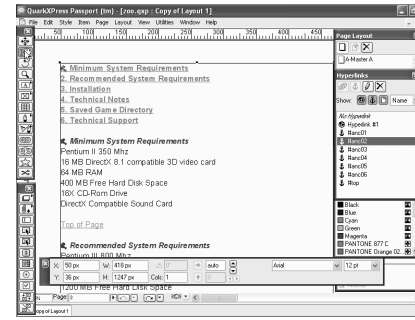
Итак, мы выяснили с вами, что такое анкеры, и каким образом их можно использовать, осталось уточнить еще один нюанс. На самом деле можно создавать гиперссылки на анкеры не только на текущей странице, но можно, например, сослаться на какую-либо другую страницу и при этом сразу же указать анкер, который находится на ней. Это делается очень просто. В конце текста ссылки после названия файла, который содержит ***Web*** страницу, указывается название анкера, но при этом между ними устанавливается знак диеза. Обратите внимание, что в палитре ***Hyperlinks*** все анкеры у нас начинаются со знака диеза. Это и есть признак того, что данный элемент является анкером. Т.е., например, ссылка на страницу ***Page HTML*** будет звучать так — ***Page HTML*** диез и название анкера, который на ней находится.

6.5.10. Редактирование гиперссылок. Создание мета-дескрипторов

Любые гиперссылки и анкеры, которые вы создаете в вашей ***Web*** публикации, подлежат редактированию. Для того чтобы отредактировать ту или иную гиперссылку, мы можем воспользоваться, во-первых, меню ***Hyperlinks***, выбрав в ней нужную гиперссылку и щелкнув на кнопке с карандашом, после чего появляется окошко редактирования гиперссылки. А следующий вариант — это выделить текст гиперссылки в документе и затем,

вызвав правой кнопкой контекстное меню, выбрать команду ***Hyperlink***, ***Edit***. Ну, и, наконец, третий вариант — это опять же выделить текст гиперссылки и выполнить команду ***Style***, ***Hyperlink***, ***Edit***.

Кроме того, любые гиперссылки подлежат удалению. Например, мы можем удалить гиперссылку прямо из палитры ***Hyperlinks***. Для этого всего лишь нужно выделить нужную и щелкнуть на кнопке с крестиком, которая называется ***Delete*** — удалить. ***QuarkXPress*** предупреждает, что гиперссылка будет удалена, мы даем подтверждение, щелкнув на кнопке OK. Для чего может понадобиться редактирование гиперссылки? Например, файл, на который ссылается гиперссылка, изменил название. Чтобы оно после этого корректно работала, нужно ее отредактировать следующим образом. Входим в режим редактирования и здесь меняем имя файла, на который будет указывать наша гиперссылка. Вот и весь процесс ее редактирования.



Есть еще некоторые нюансы, которые вам следует знать. Например, рассмотрим случай с гиперссылками, которые указывают не на файлы или ресурсы в сети, а на анкеры на той же странице. Если удалить анкер, на который ссылается существующая гиперссылка, например, вот таким образом, то, обратите внимание, название этой гиперссылки в списке ***Hyperlinks*** выделено более жирным шрифтом.

Это означает, что анкера, на которые ссылается данная гиперссылка, нет в документе, и она не будет соответственно работать. Еще один нюанс, на который хотелось бы обратить ваше внимание, — это название гиперссылок в палитре ***Hyperlinks***. Обратите внимание на то, что, например, если мы зайдем в режим редактирования для гиперссылки ***Hyperlink***#1, то увидим, что мы для нее задали имя ***Hyperlink***#1 и поэтому она именно так отображается в палитре ***Hyperlinks***. Для остальных же ссылок, которые у нас есть в палитре ***Hyperlinks*** мы не задавали имен. Тем не менее, они отображаются в палитре. Каким образом это происходит?

Если мы не задаем имени для гиперссылки, то ей автоматически присваивается имя того элемента, на который она ссылается. В данном случае у нас происходит ссылка на анкер и соответственно гиперссылка у нас отображается в виде анкера, на который она ссылается. Если же это будет, скажем, адрес некой ***Web*** страницы, то если для гиперссылки не задано конкретного имени, то отображаться она будет именно, как адрес. Мы можем

в этом убедиться, если используем выпадающий список в палитре ***Hyperlinks***, в котором есть всего лишь два параметра ***Name*** и ***Link***. Что это означает? Это означает, отображать гиперссылки по их наименованиям, которые мы им задали. В данный момент включен этот режим. Или отображать гиперссылки по адресам, на которые они ссылаются. Если мы включим данный режим, то у нас преобразится только первая гиперссылка, у которой есть свое собственное имя. Если мы включим режим ***Link***, то теперь вместо ее имени, будет отображаться название файла, на который она ссылается.

Иногда это бывает удобнее при работе, т.к. вы можете забыть какой именно гиперссылке вы дали это имя. Но если вы будете видеть, на какой именной файл ссылается данная гиперссылка, то вам будет легче разобраться, где она находится. Как мы уже говорили, гиперссылки могут ссылаться не только адреса в Интернете или на анкеры, но есть еще третий вариант — они могут ссылаться на странице в этой же публикации. Каким образом это происходит? Давайте, создадим новую гиперссылку, но прежде, давайте внимательно рассмотрим палитру ***Page Layout***, как она выглядит в режиме ***Web*** макета.

Больших отличий здесь, как видите, нет, единственное, что страницы данного макета именуются по-другому. По умолчанию всем страницам задается название ***Export*** — ***Export***1, ***Export***2, ***Export***3 и т.д. При желании и, в общем-то, даже не то что при желании, а очень рекомендуется эти названия изменять. Почему? Потому что эти названия фактически названия файлов, которые образуются при экспорте вашего ***Web*** макета в ***HTML*** формат. Если вы создаете ссылку на страницу, которая находится в вашем же макете, то вы можете задать в этой ссылке просто название данной страницы. Например, сейчас мы здесь создадим ссылку, которая будет ссылаться на страницу ***Export***3. Т.е. вот на эту страницу. Давайте сделаем это.

В общем-то, сейчас нам не важно, где создавать ссылку. Выделим любой текст на данной странице и создадим новую гиперссылку. Называться она у нас будет, допустим, ***link*** #2, тип гиперссылки у нас ***Page***, т.е. она ссылается на страницу нашего же макета. А в следующем разделе, в разделе ***Page***, нам нужно выбрать ту страницу, на которую она будет ссылаться. У нас в макете всего три страницы, поэтому здесь у нас в выпадающем списке вот эти три страницы отображаются. Как мы уже сказали, мы будем делать ссылку на страницу с названием Экспорт3.

Ну и, наконец, поле ***Target***, которое мы с вами обсуждали до этого, мы оставим пустым. Т.е. ссылку будет создаваться по умолчанию. Щелкаем на кнопку ОК. Теперь проверим нашу страницу в браузере. Появится ли там эта ссылка. Да, вот она у нас. При щелчке на ней, конечно, работать она пока не будет, так как при предварительном просмотре в браузере экспортируется только текущая страница. Соответственно та страница, на которую ссылается

наша гиперссылка, пока не существует. Но при окончательном экспорте все это будет работать нормально. Главное же сейчас для нас, что ссылка создана и она работает. Теперь давайте посмотрим на палитру ***Hyperlinks***. Здесь у нас появилась новая ссылка. В самом конце вы можете ее видеть, она называется ***Export***3.

Обратите внимание, что у этой ссылки значок отличается от всех остальных. Как вы помните, у нас три типа гиперссылок. Это гиперссылки, ссылающиеся на ресурсы Интернет, гиперссылки, ссылающиеся на анкеры, и гиперссылки, ссылающиеся на страницы в нашем же макете. Соответственно у гиперссылок в Интернет — значок в виде земного шара, у гиперссылок, ссылающихся на анкеры, — значок в виде якоря, и, наконец, у гиперссылок, которые ссылаются на страницы в нашем же макете, — значок в виде вот такого листка бумаги. Так как в палитре ***Hyperlinks*** у нас включен режим отображения ***Link***, то гиперссылка отображается не под своим названием, а в виде адреса, на который она ссылается. Т.е. в данном случае это страница ***Export***3.

Если же мы переключимся сейчас на режим ***Name***, то соответственно это гиперссылка у нас будет называться ***link*** #2. Здесь есть маленький нюанс, который помогает вам избавиться от головной боли. Представьте ситуацию, когда вы создали несколько гиперссылок, которые ссылаются, скажем, на страницу ***Export***3, как мы сейчас с вами сделали. И вдруг в один прекрасный момент вы решили переименовать данную страницу. Что происходит? Неужели придется во всех гиперссылках, которые ссылаются на эту страницу, их редактировать и менять название страницы, на которую они ссылаются.

Давайте посмотрим. Мы переименуем нашу страницу с названием ***Export***3 в страницу с названием, ну скажем, ***Text***. Давайте, теперь посмотрим, как выглядит наша гиперссылка, которая ссылалась на эту страницу. Мы включим режим ее редактирования и увидим, что страница, на которую она ссылается, — это страница ***Text***. Т.е. что произошло? При переименовании любой страницы в вашем макете, если в нем есть какие-либо гиперссылки, которые на нее ссылаются, они автоматически изменяются программой ***QuarkXPress***. Т.е. вам не нужно беспокоиться о том, что от изменения названия страницы, какие-либо гиперссылки перестанут работать. Все это делается автоматически. Ну и, наконец, хотелось бы поговорить о таком понятии при создании ***Web*** страниц,

как мета дескрипторы. В любой, абсолютно в любой ***Web*** странице существует область, в которой можно создавать некую служебную информацию.

Например, ключевые слова, которые будут использовать Интернет поисковики при поиске в ***Web***, или информацию о создателе ***Web*** сайта, ***Web*** страницы и т.д. Для того чтобы добавить мета дескрипторы в ваш ***Web*** проект, нужно использовать диалоговое окно ***Meta Tags***, которое открывается через меню ***Edit***. В данный момент здесь задан список мета дескрипторов по умолчанию, которые мы можем редактировать, щелкнув на кнопку ***Edit***. Как видите, здесь в данный момент всего два мета дескриптора.

Мы рассмотрим их. На самом деле существует намного большее количество различных мета дескрипторов, которые могут вам помочь при создании ***Web*** страницы. Но, тем не менее, это не входит в наш курс и при желании узнать о мета дескрипторах побольше вы можете изучить документацию по языку ***HTML***. Давайте, начнем с мета дескриптора ***Name — Keywords***. Для того чтобы его отредактировать, выбираем его в списке и щелкаем на кнопке ***Edit***. И здесь мы можем добавить ключевые слова, которые будут считывать поисковые системы в Интернет.

Если ключевые слова, которые вы здесь зададите, совпадут каким-либо образом с запросом, заданным пользователем в поисковой системе, то ссылка на вашу страницу, будет ему выдана. Ключевыми словами могут быть тематика вашей страницы, название кампании, темы статей и другая информацию, которая наиболее полно характеризует содержимое вашей ***Web*** страницы. Все их нужно вводить вот в это поле ***Content***. Второй мета дескриптор, который у нас используется по умолчанию, — это так же мета дескриптор типа ***Name***, но другого подвида — ***Description***.

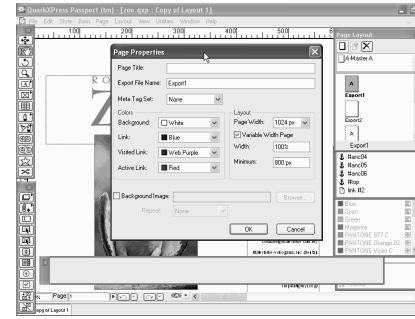
В этом мета дескрипторе вы можете просто задать краткое описание вашей страницы. Если поисковая система обнаружит, что ключевые слова соответствуют запросу пользователя, то, выведя гиперссылку на вашу страницу, ниже гиперссылки он может вывести текст, который вы введете для мета дескриптора ***Name — Description***. Текст должен быть не очень длинным и кратко описывать суть страницы, суть информации, которая на ней расположена. Например, вот такого рода информация. На самом деле, как я уже говорил, существует огромное количество мета дескрипторов.

Они разделяются на две большие группы. Это мета дескрипторы ***Name*** и мета дескрипторы ***http-equiv***. В каждой группе находятся мета дескрипторы, описывающие информацию определенного рода. В мета дескрипторе ***Name*** в основном описывают контактные сведения, сведения об авторских правах и авторе. А вот дескриптор ***http-equiv*** позволяет браузеру правильно отобразить текст вашей страницы, подобрать кодировку, правильно отобразить адреса электронной почты на странице и т.д. Т.е. грубо говоря, мета дескрипторы

Name содержат чисто текстовую информацию, предназначенную для пользователя. А мета дескрипторы ***http-equiv*** содержат служебную информацию, которая нужна только браузеру.

Например, если вы создаете ***Web*** страницу не на английском языке, то нужно задать мета дескриптор ***http-equiv*** со значением ***charset*** и указать номер кодовой страницы для языка, используемого в вашей ***Web*** странице. Для русского языка, например, это будет кодовая страница ***Windows*** 1251. Более подробно о мета дескрипторах, как я уже сказал, вы можете узнать, почитав документацию по языку ***HTML***. Там это все очень подробно описано.

6.5.11. Экспортирование Web-макета в HTML



Конечной целью создания нашей ***Web*** публикации является, само собой разумеется, размещение ее в Интернет. Поэтому, когда готова хотя бы промежуточная версия вашей ***Web*** страницы, вы захотите протестировать ее на ***Web*** сервере. Для этого в первую очередь нужно экспортировать вашу ***Web*** публикацию в обычные ***HTML*** файлы, которые понимают Интернет браузеры. Делается это с помощью команды ***File***, ***Export***, ***HTML***. Но, прежде чем начать экспортирование, нужно уделить внимание некоторым нюансам. Страницам вашей ***Web*** публикации нужно дать правильные имена. Что это означает? Если ваша ***Web*** публикация состоит из нескольких страниц, то в любом случае одна из них должна быть стартовой. Стартовая страница обычно имеет стандартное имя. Это имя индекс ***HTML***, т.е. название файла, в котором будет содержаться ваша стартовая страница. В нашей случае данная страница, т.е. первая наша страница является стартовой, и поэтому нам желательно переименовать ее индекс.

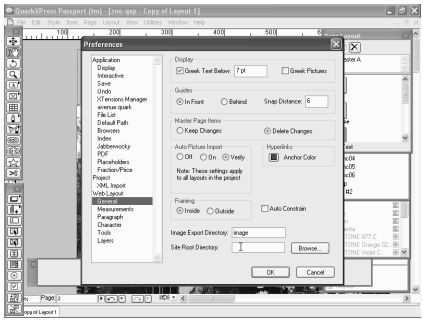
Мы уже видели, как происходит переименование страниц, но на самом деле это можно делать не только через палитру ***Page Layout***, но и с помощью команды ***Page***, ***Page Properties***. Дело в том, что в диалоговом окне ***Page Properties*** — свойства страницы мы можем задать, во-первых, само название страницы, в данный момент это у нас ***Export*** 1, как мы уже сказали, мы назовем нашу страницу ***index***. Во-вторых, как вы уже помните, в данном диалоговом окне можно настроить стандартные цвета ссылок, ширину

страницы, фоновый рисунок, ну и, наконец, в самом вверху диалогового окна у нас есть поле ***Page Title*** — заголовок страницы.

Не путаете его, пожалуйста, с названием страницы. Т.е. фактически с названием файла, в который она будет экспортирована. ***Page Title*** — это всего лишь заголовок страницы, который отображается в браузере в самом вверху, когда страница в нем открыта. Кроме того, данный заголовок страницы будет отображаться в поисковиках при обращении к ней.

Давайте и мы зададим заголовок для нашей страницы, в общем-то, это делать не обязательно, т.е. и без заголовка все будет работать, но более грамотным все-таки будет задание для каждой страницы вашей публикации своего заголовка. Вот так. Если сейчас мы просмотрим нашу страницу в браузере, то увидим, что в заголовке браузера появился тот самый заголовок страницы, который мы ввели в диалоговом окне ***Page Properties***. Остальным страницам также желательно задать экспортируемые имена. Для третьей страницы мы с вами это уже сделали. Давайте, это сделаем и для второй страницы.

Опять же входим в диалоговое окно ***Page Properties***, а в поле ***Export File Name*** пишем, к примеру, ***house***, в поле ***Page Title*** — заголовок страницы, пишем ***my house***. ОК. В общем-то, сколько бы страниц у нас не было, желательно их всех таким образом обработать. На самом деле задавать экспортируемые имена и заголовки страниц лучше всего с самого начала. Так вы избежите путаницы, т.к. в данный момент мы работаем с публикацией, в которой всего три страницы. Если же в ней будет 33 страницы и они будут иметь названия, подобные ***export*** 1,2, 3,5,10 и т.д., то вы в них очень легко запутаетесь. Если же каждой странице вы будете задавать осмысленное название, то вы будете очень легко ориентироваться в своей публикации.



Кстати, еще раз напомню, что диалоговое окно ***Page Properties*** нужно вызывать для каждой страницы. Т.е. изменения в нем влияют только на текущую страницу. Следующим шагом может быть изменение установок в диалоговом окне ***Preferences***. Здесь у нас есть раздел ***Web Layout***. Мы уже с вами говорили о нем. В данный момент хотелось бы обговорить единственный нюанс — это название папки для изображения. Когда будет происходить экспорт вашей публикации, все файлы, которые будут образованы при экспорте, будут помещены в определенную папку. Но, кроме ***HTML*** файлов, ваша публикация будет содержать и графические файлы. Т.е. все изображения,

которые у вас размещены на страницах, будут также экспортированы. Поэтому можно заранее указать название папки, в которой будут сохранены все эти изображения. По умолчанию это название папки ***image***.

Ниже у нас расположено еще одно поле, которое называется ***Site Root Directory***. Это папка экспорта по умолчанию. Т.е. фактически, когда мы будем производить экспорт, нам нужно будет указать папку, в которую будут экспортироваться наши файлы. Если вы делаете несколько разных ***Web c***айтов, то, в общем-то, здесь вводить какую-то папку большого смысла нет. А если же вы делаете один большой проект, то каждый раз при экспорте очередной страницы, указывать нужную папку не очень рационально. Поэтому можно заранее создать папку для экспорта и указать ее в диалоговом окне ***Preferences*** в поле ***Site Root Directory***. Тогда каждый раз при экспорте файлов у вас сразу будет открываться эта папка. Мы создали заранее папку, которую назвали ***My Web Projects***. Вот, давайте, ее и укажем, как папку, для экспорта ***HTML*** файлов по умолчанию.

Итак, мы можем приступить к самому процессу экспорта. Как я уже сказал, он выполняется с помощью команды ***File***, ***Export***, ***HTML***. Появляется диалоговое окно ***Export***, ***HTML***. Как я уже говорил, мы сразу оказываемся в папке ***My Web Project***, которую мы указали в диалоговом окне ***Preferences***. При желании мы можем указать любую другую папку, но каждый раз при новом экспорте по умолчанию будет диалоговое окно ***Export HTML*** открываться именно в этой папке.

Давайте рассмотрим диалоговое окно подробнее. Здесь у нас есть несколько параметров. Например, поле ***Export As***, в котором мы можем указать: экспортировать ли нашу ***Web*** публикацию в формате обычный ***HTML*** или в формате ***XHTML*** 1.1. Далее вам может понадобиться экспортировать не все страницы публикации, а только некоторые. Если вы экспортируете все страницы, то в поле ***Pages*** мы оставляем значение ***All*** — все, а если же мы хотим экспортировать конкретные страницы, например, 1 и 3, то пишем вот так. Далее флажок ***Launch Browser***. Это всего лишь означает, что при выполнении операции экспорта сразу же ваша страница будет открыта в браузере по умолчанию, для того чтобы вы смогли просмотреть результат.

И, наконец, флажок ***External CSS File*** означает, что каскадные таблицы стилей, которые вы создали в своей публикации, будут сохраняться не отдельно в каждой из страниц публикации, а будут сохранены в специальном внешнем ***CSS*** файле, который будет храниться в той же папке, где хранятся изображения. После того, как все требуемые параметры установлены, можно нажать на кнопку ***Export***. Итак, экспортирование закончено. Первая страница нашей публикации открыта в браузере. Мы можем убедиться в том, что она экспортировалась удачно, без потерь. Посмотрим, работает ли гиперссылка.

Она работает. Посмотрим, работают ли гиперссылки на странице, на которую мы перешли. Они также работают.

Ну и теперь, давайте, закроем браузер и посмотрим, что же представляет собой наша экспортированная публикация физически, т.е. из каких файлов она состоит. Как мы помним, мы экспортировали нашу публикацию в папку ***My Web Project***. Давайте войдем в нее и увидим здесь два файла. Это файлы ***index.htm*** и ***Text.htm***. Как вы помните, при экспортировании мы указали, что мы экспортируем только две страницы, 1 и 3. Соответственно, 1 страница — это у нас ***index.htm***, мы задали для нее именно это экспортируемое имя, и третья страница — это у нас ***Text.htm***. В папке ***image*** находится изображения, которые используются на этих страницах. Одно из них, это то самое изображение попугая, которое у нас находится на стартовой странице.

Второе — это, если вы помните, надпись, которая у нас импортируется в виде графики. Вот этот заголовок. И, наконец, третье изображение — это вспомогательное изображение, которое позволяет ***QuarkXPress*** располагать элементы именно так, как нужно на экране. Некоторые пользователи удивятся тому, что у ***HTML*** файлов незнакомый значок. Многие привыкли к тому, что ***HTML*** файлы обозначаются значком ***Internet Explorer***, стандартного браузера операционной системы ***Windows***. Дело в том, что в данной системы браузером по умолчанию является также известный и распространенный браузер, который называется ***Oper***.

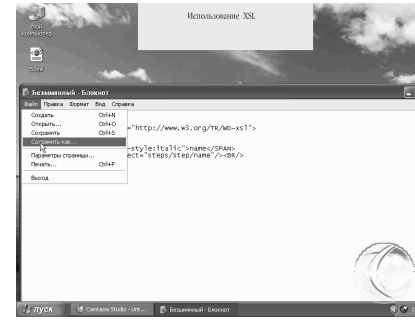
Поэтому, ***HTML*** страницы имеют именно его значок. Но суть их от этого абсолютно не меняется. Итак, в папке, которую вы видите сейчас на экране, находятся файлы, фактические готовые для размещения в Интернет на ***Web*** сервере. Все, что здесь вы видите, достаточно для того, чтобы вашу страницу мог просматривать любой пользователь через Интернет. Размещение страницы в Интернет мы рассмотрим с вами позже. А пока, давайте, вернемся в ***QuarkXPress*** и рассмотрим дополнительные возможности этой программы при создании ***Web*** публикации.

6.6. XML

Здравствуйте, уважаемые дамы и господа! Сегодня мы пройдем краткий курс языка ***Web*** программирования ***XML***. Что такое ***XML — Extensible Markup Language***. Этот язык возник не на пустом месте, а является и упрощенной частью языка ***HTML***. Зачем же было новые языки, если есть ***HTML***. Все просто. Первая идея, это отделить данные от оформления. В настоящий момент задача в ***HTML*** частично решается с помощью каскадных таблиц стилей ***css***. Второе, это расширяемость языка своими тегами. В ***HTML*** такая возможность исключена. В ***XML***, возможна. Давайте, попробуем создать ***XML*** документ. Для этого нам нужно открыть текстовый файл.

Вставить в него наш код, сохранить в формате ***XML***, закрыть текстовый файл, чтобы впоследствии открыть его в браузере. Мы видим четкую древовидную структуру, которую можно просматривать. Мои поздравления! Это наш первый ***XML*** документ. У нас есть возможность использовать каскадные таблицы стилей ***css*** для настройки отображения ***XML*** документа. Давайте включим заголовок ссылку на ***css***. Для этого обратимся к уже созданному нами ранее документу с помощью блокнота. И отредактируем наш файл. Вставим наш код, в котором имеется ссылка на файл ***css***. Сохраним все изменения. И теперь создадим сам стиль.

Для того чтобы создать стиль, нам понадобится еще один файл. Откроем Блокнот. Вставим наш код. Как вы видите, здесь описывается отображение тега ***step***. В этом и есть суть разделения оформления и содержания. Сохраним наш новый файл, как каскадную таблицу стиля. Закроем его. А теперь попробуем запустить программу и посмотреть, как будет выглядеть наш документ. На основе стилей он выглядит совсем по-другому. Это не единственный способ. Есть еще ***XSL***, но об этом мы поговорим позже. Подстановка повторяющихся данных. ***XML*** имеет много новых возможностей. Например, определение типа ***define*** C++, что позволяет использовать повторяющиеся данные на основе макро подстановки. Давайте создадим ***XML*** документ с макро подстановкой. Как мы уже знаем, для того чтобы создать документ, нам нужен блокнот. Вставим наш код.



Сохраним все в формате ***XML***, закроем файл и откроем его в браузере. Как мы можете видеть, у нас все та же четкая древовидная структура, которую можно просматривать. Подстановка из внешнего файла. Конечно же, на предыдущем шаге мы получили хорошие возможности, но лучше, хранить подобные повторяющиеся данные в отдельном файле. Что очень удобно при необходимости смены этих данных. Оформляем ссылку на внешний файл. Для этого, как всегда используем Блокнот. Вставляем наш код и сохраняем все в ***XML*** формате. Документ ссылка создан. Теперь создадим сам файл. Вновь используем Блокнот. Внутри, пока что, будет всего одна строчка.

Сохраняем все в формате ***XML***. У нас уже создан сам файл. Теперь обратимся к головному нашему файлу и посмотрим, что же у нас получилось. Как вы можете видеть, эффект тот же. Но на сей раз, наш документ выполнен

более грамотно по сравнению с предыдущим. Использование ***XSL***. ***Extensible Style sheet Language*** — расширяемый язык таблиц стилей. По принципу работает аналогично каскадным таблицам стилей, объясняя, как отображать ***XML*** документ. Но возможности ***XSL*** намного шире, включая сортировку и фильтрацию. Мы с вами посмотрим простой пример с ***XML*** документом. Создаем ссылку на ***XSL***. Открываем предыдущий нами созданный документ с помощью блокнота и редактируем его вписывая ссылку на ***XSL*** документ.

Сохраняем все. Создаем сам файл ***XSL***. Опять же с помощью Блокнота. Вписываем в него содержание, сохраняем все, но уже в формате ***XSL***. Запускаем браузер и смотрим, работает, как не странно. Атрибуты тега. Атрибут — это пара и назначение, который связан с данным элементом. Атрибуты не отображаются в документе и являются альтернативным способом представления информации. Они не отображаются, если, конечно, вы не сами не хотите. Использовать атрибуты можно только ***XSL*** или в средствах работы с ***XML*** документом. Давайте, попробуем создать ***XML*** документ с атрибутами. Пишем код. Здесь вы можете видеть, находится ссылка на ***css*** документ, который мы сейчас создадим. Создаем наш старый добрый ***css*** документ, который мы уже ранее использовали. И смотрим в браузере наш головной ***XML*** документ.

Как мы можем видеть, атрибуты не отображаются. ***CDATA*** и отображение запрещенных символов. Внутри символьных данных запрещено помещать скобки тегов и знаки амперсантов, хотя есть специальные знаки. Если текста, который использует эти знаки много, то лучше использовать раздел ***CDATA***. Подобным примером могут быть листинги ***HTML*** документа. Посмотрим пример. Создадим файл, впишем в него код. Сохраним все, как обычно, в формате ***XML***. И откроем наш головной файл в браузере. Вот и наш результат, отображения в браузере. Эта древовидная структура, которую можно просмотреть. Использовать теги ***HTML*** в ***XML*** нужно, например, для того, чтобы была возможность использовать ссылки на другие документы. Что мы сейчас и попробуем сделать.

Откроем Блокнот, вставим наш код, сохраним файл, как обычно, в формате ***XML***, откроем его в браузере и здесь мы можем видеть ссылку на сайт. Нажимаем на нее, переходим по этой ссылке в вышеуказанный сайт. В данном случае это сайт ***firststeps.ru***, который, как вы понимаете, не существует. Одним из практических способов использования ***XML*** является использование ***XML*** внутри ***HTML*** кода. Этот процесс называется связыванием. Включить ***XML*** в внутрь ***HTML*** кода очень просто. Для этого используются теги, ***XML*** и идентификатор. Этот идентификатор в последствии пригодится для отображения документа. А дальше, просто нужно связать данные в ***XML*** с

HTML документом. В нашем примере создается таблица, которая отображает данные. Вы можете видеть код ***HTML***, который мы подключаем в ***XML*** код.

Заканчивается наш документ также кодом ***HTML***. В виду того что мы ***XML*** код подключаем к ***HTML***, наш документ мы должны сохранить в формате ***HTML***. Откроем наш документ в браузере, посмотрим, что у нас вышло. Как вы можете видеть, верхняя строчка была создана кодом ***HTML***, остальные три строчки, которые нам хорошо знакомы, созданы кодом ***XML***. Мы с вами имеем возможность использовать и внешний ***XML*** документ. Для этого достаточно оформить на него ссылку. Как вы можете видеть, эта наша ссылка на ***XML*** файл. Сохраняем данный файл в формате ***HTML***. Теперь мы должны создать ***XML*** файл, на который у нас уже оформлена ссылка. Сохраняем его в формате ***XML***.

В данном случае оба наших файла должны находится в одном и том же каталоге. Давайте откроем наш ***HTML*** документ в браузере. А вот и наш результат, который является таким же, как и предыдущий. Использование внешнего ***XML*** файла позволяет нам удобно организовать перемещение по записям таблиц. Т.е. практически смоделировать работу с базой данных. Управлять объектом данных ***XML*** можно используя ***JavaScript***. Посмотрим код. Сохраним все в формате ***HTML*** и загрузим наш файл в браузере. Вот таким будет результат. Мы сможем перемещаться по записям щелкая на вот этих кнопках. На этом краткий курс по изучению языка ***Web*** программирования ***XML*** закончен. Благодарю вас за внимани

6.7. Общие сведения о программировании на Java

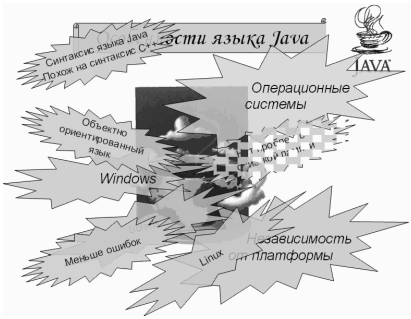
6.7.1. Введение

Рассмотрим подробно язык программирования ***Java***, который является, несомненно, одним из лучших языков программирования, доступных серьезным программистам. И одним из самых существенных преимуществ языка ***Java***, это независимость от платформы, на которой выполняются его программы. Один и тот же код можно запускать под управлением совершенно разных операционных систем, таких как ***Windows***, ***Solaris***, ***Linux*** и т.д. И это, особенно необходимо, когда программы загружаются через Интернет, для последующего выполнения на компьютерах пользователя в совершенно различных операционных системах, которые на них установлены.

Другое преимущество языка ***Java*** заключается в том, что он очень похож на синтаксис языка ***C***++. И программисты, которые работают в основном на этом языке и привыкли к этому синтаксису, довольно легко переходят на язык ***Java*** и в нем довольно быстро адоптируются. Надо сказать, что язык ***Java***, это

полностью ориентированный язык, даже в большей степени, чем сам ***C***++. Причем надо сказать, что разрабатывать программы, не содержащие ошибок, на языке ***Java*** гораздо легче, чем на языке ***C***++, поскольку разработчики снабдили язык ***Java*** средствами, позволяющими исключить достаточно распространенные ошибки.

Например, в языке ***Java*** исключена возможность явного выделения и освобождения памяти. Память в нем освобождается автоматически с помощью сборки мусора, и программист гарантирован от ошибок, связанных с неправильным использованием памяти. Также введены истинные массивы и запрещена арифметика указателей, вследствие чего в принципе невозможно стереть данные из памяти из-за ее неправильного использования. Исключена также возможность, перепутать оператора присваивания с оператором сравнения. Также исключено множественное наследование, которое заменено новым понятием интерфейса.



Рассмотрим основные характеристики языка ***Java***. Первая из них, это то, что язык ***Java*** является достаточно простым языком. По существу синтаксис языка ***Java*** представляет собой очищенный вариант языка ***C***++, в котором нет заголовочных файлов, арифметики указателей, самых указателей, структур, объединений и т.д. По сравнению с обычными, визуальными средами программирования, которые на данный момент используются достаточно широко, например, ***Visual Basic***, язык ***Java*** может показаться сложным. И при работе на языке ***Java*** приходится гораздо больше программировать и писать коды. В отличие от визуальных средств программирования, которые позволяют почти автоматически создавать всю инфраструктуру приложения, весь его внешний вид. А на языке же ***Java*** необходимо программировать довольно много чего вручную.

Правда, надо сказать, что все это, может быть, окупается еще и тем, что в результате получаются достаточно короткие программы. Поскольку для языка ***Java***, как для языка, который в основном предназначен для работы в Интернете, краткость, это очень существенное достоинство, поскольку коды написанные на этом языке, все время приходиться пересылать по Интернету, по сетям. Да и вообще ***Java*** программы можно выполнять на компьютерах совершенно разной мощности. Поскольку размер его основного интерфейса и средств поддержки, даже вместе с его стандартными библиотеками, занимает

всего лишь пару сотен КБ. Но, конечно, с поддержкой графики этот размер немножко больше, но все равно достаточно мал для современного развития программирования.

Особенностью языка программирования ***Java*** к тому же является и то, что он, так же, как и подавляющее большинство современных языков программирования является объектно-ориентированным. И надо отметить, что объектно-ориентированные особенности языка ***Java*** и языка ***C***, довольно похожи друг на друга. Надо отметить еще такую особенность языка ***Java***, как-то, что он является распределенным языком. Т.е. он предоставляет мощные и удобные средства для работы в сетях, обладая большой библиотекой для передачи данных на основе таких протоколов, как, например, ***TCP IP***, ***HTTP***, ***FTP*** и т.д., которые широко используются в Интернете. И надо отметить, что на языке ***Java*** даже самые трудные задачи, например, такие как открытие и закрытие сетевых соединений, решаются очень легко. Механизм, состоящий из так называемых ***servlet***, делает работу на сервере чрезвычайно эффективной, и это поддерживается многими популярными веб-сервирами.

Одной из самых ярких особенностей языка ***Java***, наверное, является то, что язык ***Java*** очень надежный язык. Поскольку он предназначен для создания программ, которые должны надежно работать в любых ситуациях. И поэтому создатели языка ***Java*** уделили большое внимание раннему обнаружению возможных ошибок, также динамической проверки во время выполнения программы. И надо сказать, что компилятор языка ***Java*** выявляет такие ошибки, которые в других языках программирования, компилятор пропускает, и они обнаруживаются только на этапе выполнения. И, конечно, такая достаточно тяжелая ситуация, как повреждение памяти из-за неверного указателя, в принципе исключена в этом языке.

Надо еще отметить такую особенность языка, как то, что он является достаточно безопасным при работе в сетях. Конечно, надо сказать, что абсолютно точно этого гарантировать нельзя, и в принципе в механизме безопасности языка ***Java*** были определенные проблемы. Но надо сказать, что группа разработчиков языка ***Java*** заявила о своей полной нетерпимости к любым ошибкам в системе защиты.

И немедленно исправляет любые возникающие в этой сфере вопросы. И надо сказать, что, учитывая все это, с языком ***Java*** можно совершенно безопасно работать в сетях. Ну, например, в Интернете. Можно скачивать программы, написанные на ***Java***, выполнять их на своем компьютере и при этом, никаких нежелательных последствий проявляться не будут. В отличие от такого альтернативного механизма, как технология ***ActiveX***, которая для безопасности использует только цифровые подписи. Но, конечно же, этого совершенно недостаточно.

Любой пользователь программного обеспечения ***Microsoft*** и этих элементов ***ActiveX***, может подтвердить, что такие программы достаточно часто открываются аварийно, давая тем самым опасность повреждения данных. А система же безопасности языка ***Java*** намного надежнее. Поскольку контролирует приложение с момента запуска и не позволяет ему причинять ущерб компьютеру и имеющимся на нем данным. Надо еще отметить такую особенность, как то, что язык ***Java*** не зависит от архитектуры. Т.е. в результате работы компилятора, получается объектный файл, формат которого не зависит от архитектуры компьютера и может выполняться на любых процессорах под управлением системы выполнения программы языка ***Java***.

Надо также сказать отметить, что язык ***Java*** является машинонезависимым. Т.е. в спецификации языка ***Java*** нет аспектов, которые зависят от системы реализации на конкретных системах компьютеров. Например, размер основных типов данных и арифметические операции над ними, точно определены. Если это тип ***integer***, то на языке ***Java***, это всего 32 — разрядное целое число.

В отличие, например, от языков ***C*** и ***C++***, где разрядность целого числа может принимать различные значения, как 16, так и 32. Надо также отметить, что язык ***Java*** по своей природе является интерпретируемым, поскольку интерпретатор языка ***Java*** может пересылаться на любую машину и выполнять его код непосредственно на ней. Правда, надо сказать, что сам компилятор языка ***Java***, входящий в стандартный набор ***Java SDK***, работает довольно медленно. Правда, скорость при компиляции, это далеко не самый важный фактор при работе с программами.

Надо также отметить, что язык ***Java*** является высокопроизводительным. Надо сказать, что обычно интерпретируемые байт коды, в которые превращаются, после компиляции, программа на языке ***Java***, имеет более чем достаточную производительность. Но бывают ситуации, когда требуется что-то сделать еще быстрее. В этом случае надо сказать, что эти коды можно налету, во время выполнения, транслировать в машинные коды для конкретного процессора, на котором выполняется некая программа или приложение. Лучшую интерактивность и выполнение программы, обеспечивает многопоточность языка ***Java***. Причем надо сказать, что это сделать на языке ***Java*** достаточно просто. И потоки в

языке ***Java*** могут использовать преимущество многопроцессорных систем. Но, конечно же, если операционная система позволяет это сделать.

Ну и надо сказать, что язык ***Java*** является динамичным языком и легко адаптируется к постоянно изменяющейся среде. В библиотеки можно свободно добавлять новые методы и объекты, не причиняя никакого вреда основам. Ну и к тому же, в среде языка ***Java*** легко можно получить информацию о ходе выполнения программ.

Рассмотрим такой аспект использования языка ***Java***, как его работа в Интернете. Причем идея эта, достаточно простая. Это просто-напросто пользователи загружают байт кода языка ***Java***, который получается в результате компиляции из Интернета, и выполняют их на своих машинах. И в результате получается, что на веб-странице появляется некая часть, которая отведена под изображение программы ***Java***. Программы ***Java***, которые работают под управлением веб-браузеров, называются аплетами (***Applet***). Например, в нашем случае, у нас загружена программа ***Internet Explorer***, внутри которой находится аплет ***Java***.

Вот этот прямоугольник. Загрузка этого аплета напоминает просто-напросто внедрение изображения на веб-страницу. А текст и другие части этой веб-страницы просто-напросто обтекают этот аплет. При этом надо сказать, что основное отличие простого изображения от веб-аплета, является то, что этот аплет является живым. Т.е. он реагирует на нашу мышь. На наши действия. Например, если мы нажмем на кнопку мыши, и, не отпуская, попробуем подвигать ее, то можно увидеть, как в результате этого изменяется изображение, как оно двигается.

В нашем случае реализовано отображение внешнего вида молекулы в ее трехмерном варианте. Этот пример поставляется с стандартной постановкой ***Java SDK***, который можно загрузить из сервера компании ***SUN***. Такого рода манипуляции, конечно же, невозможно реализовать на статических веб-страницах, однако аплеты, делают это возможным.

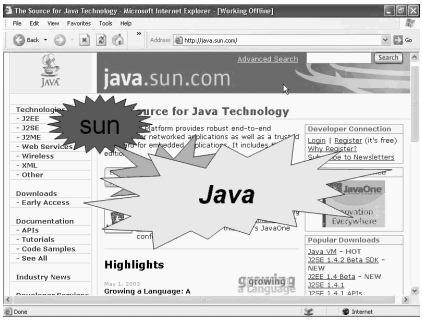
6.7.2. Среда программирования

*6.7.2.1. Установка языка **Java***

*Я*зык ***Java*** был разработан инженерами компании ***Sun***. Ну и, поскольку большинство сотрудников этой компании имели опыт работы с операционной системой ***Unix***, то в основу этого языка был положен язык ***C***++. Сначала этот язык назвали ***Oak***, дуб. Возможно потому, что под окнами офиса этой компании рос большой дуб. Но в последствии, поскольку это имя уже использовалось для другого языка программирования, его изменили и назвали ***Java***, и лицензия на исходный код языка ***Java*** принадлежит компании ***Sun***,

которая настаивает на неизменности, как самого языка, так и структуру и его основных библиотек.

Хотя, к сожалению, в реальности все обстоит немножко сложнее и ответвления от этого единообразия все-таки есть. Разные версии браузеров, таких популярных, как ***Netscape*** и ***Internet Explorer*** поддерживают разные версии языка ***Java***. Ну и наиболее полную и современную версию пакета ***Java***, пакет ***Java***2 ***Standard Edition*** можно загрузить с сайта компании ***Sun***. Вот с этого — ***JAVA.SAN.com***, на котором находится пакет ***Java***2 ***Standard Edition***, который предназначен для работы в операционных системах ***Solaris***, ***Linux*** и, конечно же, ***Windows***. При этом можно увидеть, какие наиболее популярные версии этого языка ***Java*** загружают чаще всего пользователи. Как мы видим, на данный момент разработана ***Beta*** версия 1.4.2 и основная версия 1.4.1 языка ***Java***.



Если перейти по этой ссылке, то можно попасть на следующую страницу, с которой уже можно загрузить конкретно необходимый нам пакет. Здесь у нас есть несколько возможностей. Если мы хотим загрузить минимальную версию этого пакета, которая позволит только проигрывать уже скомпилированные программы, а также запускать веб аплеты, то нам достаточно вот этой версии ***JRE***. Ну, и, конечно, если нам достаточно только английской версии этой программы, то можем воспользоваться вот этой загрузкой.

А если же нам нужны другие языки, то, конечно же, нам понадобится следующая версия, вот эта, которая поддерживает также и другие языки, а не только английский. Если же нам нужна полная версия вместе с компилятором, примерами и т.д., то лучше загрузить вариант ***Java SDK***. Но, конечно, при этом надо учесть, что различные версии этого пакета имеют так сказать различный вес, различный объем.

Минимальная версия, которая нужна для ***Windows***, для поддержки работоспособности аплетов языка ***Java*** и программ, занимает около 8 MB, а полная версия вот эта ***Java SDK*** вместе с примерами и вместе с компилятором, занимает около 37 MB, что надо учесть при скачивании пакета из Интернета. После скачивания этого пакета, мы получаем exe файл, который можно запустить и который уже автоматически установит этот пакет на наш компьютер. Правда, надо еще учесть, что после этого, если мы хотим

использовать полный пакет с компилятором и т.д., нам понадобится провести еще несколько дополнительных операций.

Если мы работаем в системе ***Windows*** 98, нам понадобится задавать путь к каталогу наших выполняемых файлов пакета ***Java***, в список путей, по которому операционная система может найти эти файлы. Это нам придется сделать в файле ***Auto axis bat***. Там нам понадобится добавить в путь ***Path***, путь к этим программам. А также понадобится добавить путь к классам. Ну, это мы сейчас рассматривать не будем, поскольку приблизительно аналогичное нам понадобится сделать в системе ***Windows XP***, которая установлена на нашем компьютере. Правда, сделать это в системе ***XP***, так же, как и в системе ***Windows*** 2000, надо немножко в другом месте, не в файле ***Auto axis bat***, а в том месте, где находится описание нашей системы.

Для этого нам надо вызвать меню ***Start*** в нашем компьютере. Щелкнем на нем, далее подвести курсор к пункту ***My Computer***, щелкнуть на правую кнопку мыши и выбрать его свойства — ***Properties***. Щелкнем. Появляются вот эти стандартные свойства нашей системы. А если же мы перейдем на вкладку ***Advanced*** (дополнительно), а дальше щелкнем на кнопке ***Environment Variable*** (переменные среды), то появляется вот это окно, в котором у нас и находятся все переменные и их значения. И нас интересуют две из них. Это ***Path***, а также ***CLASSPATH***, которые мы и должны изменить. Для того чтобы их изменить, надо щелкнуть на кнопке ***Edit***. Появляется стандартное значение этой переменной и в нем нам нужно добавить путь к нашим файлам. Для этого надо перейти в конец этого списка. Далее обязательно точка с запятой, и путь к нашим файлам.

В нашем случае это диск ***D*** и папка ***j2sdk*** с номером версии, и далее папка ***bin***. Это именно та папка, в которую при инсталляции на наш компьютер и устанавливается инсталляционный пакет программы ***Java***. А теперь щелкнем на кнопку ОК, и это изменение у нас зафиксировалось в соответствующей переменной. Теперь же нам нужно задать еще и путь к нашим классам, для того чтобы правильно работал наш компилятор.

Щелкнем теперь на кнопку ***Edit***. Перейдем в конец нашей строки и далее добавим опять точку с запятой и просто-напросто точку, которая означает, что наш компилятор в случае компиляции нашей программы и ее выполнения, будет искать классы в текущей папке, что и означает введенная точка. Щелкнем теперь на кнопку ОК.

Опять ОК. И закроем наше окно с системными свойствами. Щелкнем опять на кнопку ОК. После этого, в принципе надо перезагрузить наш компьютер, для того чтобы эти изменения вошли в силу. И мы можем начать работу с этим пакетом.

6.7.2.2. Компиляция и выполнение простой программы

После того, как мы установили на нашем компьютере весь пакет ***Java***, также установили пути к его исполняемым файлам и к классам, все эти переменные окружения, теперь посмотрим, что же находится в этом пакетном каталоге. Для этого подойдем к пункту меню ***Start*** на нашем компьютере, щелкнем на нем. Далее Мой компьютер. Далее выберем локальный диск ***D***, поскольку именно на нем инсталлирован наш пакет. Щелкнем на нем дважды. Вот появились папки и файлы, находящиеся на этом диске. И здесь мы можем видеть папку, в которую инсталлирован наш пакет ***Java***. Щелкнем на нем дважды.

И рассмотрим подробнее, какие здесь есть у нас папки. В папке ***bin*** у нас находятся все исполняемые файлы, компилятор, программы запуска и т.д. В папке ***include*** находятся файлы с машинно-зависимыми кодами, в папке ***lib*** находятся, конечно же, библиотечные файлы. В папке ***jre*** находятся файлы среды выполнения программ на языке ***Java***. А в сжатом файле ***src*** находятся исходные тексты всех открытых библиотек. Тут же у нас находятся папка ***demo***, в которой сконцентрированы всевозможные примеры, которые стандартно поставляются с пакетом ***Java***. В каждом из этих пакетов находится довольно много стандартных примеров. Вот их можно просмотреть, запустить и увидеть. Закроем теперь это окно. И посмотрим, какие у нас есть возможности по работе с этим пакетом.

Если раньше вы программировали на каком-нибудь из стандартных языков программирования, обладающем интегрированной средой разработки, тогда наверняка, сталкивались со встроенным текстовым редактором, меню для компиляции, запуска программы, а также с отладчиком. В отличие от них, стандартный пакет ***Java SDK*** не имеет никаких средств для визуальной разработки программ. Все программы выполняются путем их набора в окне оболочки. Правда, надо сказать, что полномасштабная среда разработки программ, в которой, наряду с процессом облегчения нашей работы, поскольку мы можем визуально видеть все, что у нас получается и достаточно легко, так сказать, рисовать внешний вид интерфейса нашей программы, но, тем не менее, программы при этом получаются достаточно большими. А для языка ***Java*** это существенно.

Интегрирование среды разработки становится все более неудобным для создания именно простых программ, поскольку работают медленнее, требуют более мощные компьютеры и т.д. Такие среды достаточно удобны, если мы хотим создать проект, состоящий из многих файлов, может быть, даже распределенные, между разными разработчиками.

Для простых же программ, для которых в основном и предназначен язык ***Java***, удобнее пользоваться в принципе нечто средним между работой в режиме командной строки и интегрированной средой разработки. А именно, каким-либо простым текстовым редактором, который интегрирован с пакета ***Java***. Т.е. в принципе у нас есть один из нескольких возможных вариантов при работе с этой программой. А именно, воспользоваться какой-либо средой разработки. Например, мы можем воспользоваться таким широко распространенной средой ***Java***, как ***Microsoft Visual Java***++, которая дает нам возможность создавать большие пакеты на этом языке.

Мы можем воспользоваться пакетом, который предоставляет компания ***Borland Java Builder***, с помощью которой опять-таки можно создавать достаточно большие пакеты. А можем воспользоваться и программой, которую поставляет сама компания ***Sun***, а именно, ***Sun ONE Studio***. Эту программу опять-таки можно скачать с сайта компании ***Sun***. Или же можно просто-напросто воспользоваться командной строкой, режимом командной строки. И для этого воспользоваться какой-либо простой программой, предоставляющей нам эту возможность, например, ***Norton*** или ***Far***. Вот, перед нами некая маленькая программа, написанная на языке ***Java***, которую мы сейчас можем скомпилировать и запустить именно в режиме командной строки. Первое, что можно сделать, это нажать на клавишу ***F***3, и посмотреть, что же у нас находится в этой программе.

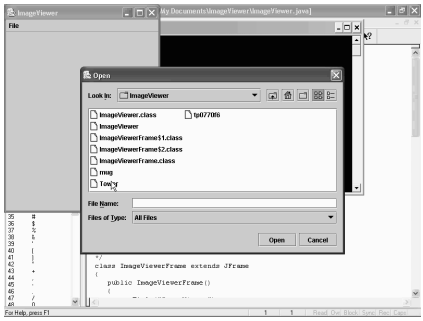
Вот в принципе весь текст нашей программы, который состоит всего лишь из нескольких строчек. Сейчас останавливаться подробно на этом содержимом не будем, щелкнем на клавишу ***Esc*** и, для того чтобы попробовать ее выполнить, первое, что нам надо сделать, это ее скомпилировать. И для этого, в режиме командной строки нам надо набрать следующие строчки. ***Java***, далее буква С, ***javac*** и затем имя нашей программы ***Hello***.***java***. ***Hello*** точка и ***java***. Далее щелкнем на клавише ***Enter***. Мы можем видеть, что, наряду с файлом ***Hello***.***java***, в котором у нас находился текст нашей программы, появился новый файл ***Hello***.***class***, который и является исполняемым модулем нашей программы, полученным после компиляции ***Hello***.***java***.

Сейчас попробуем его запустить в режиме выполнения. Для этого нам нужно набрать ***java*** и далее имя нашей программы ***Hello***. Причем, надо отметить, что нам надо набрать именно ***Hello***. Регистр букв имеет существенное значение, он должен быть написан именно так, как мы его

используем внутри нашей программы. И без расширения ***class***. Только имя ***Hello***, первая буква H с большой буквы. Щелкнем на клавише ***Enter***. Ну, эта программа выполнилась, но пока ничего не видно, поскольку у нас панели навигации закрывают весь наш обзор. Для того чтобы посмотреть, что у нас получилось, щелкнем на клавишах ***Ctrl***+***F***1. Этим мы скрываем левую панель и открываем вид на нашу командную строку, и то, что там выполняется. И можем видеть, что у нас выполнилась программа ***java Hello***, в результате чего на нашем экране консоли появилась надпись ***Hello World***, что и было целью нашей программы.

Но, в принципе это достаточно простая программа, которая, конечно же, больше ничего и не делает, т.е. только выводит на наш экран вот эту надпись ***Hello World***. Т.е. вот таким путем, путем выполнения так сказать двух строчек. Первая, эта компиляция нашего текстового файла с программы, мы получаем исполняемый модуль, и его запуском, мы уже получаем выполнение и его результаты на нашем экране. И все это можно выполнять в режиме командной строки. Причем не только, конечно же, пользуясь такой оболочкой, как ***Norton*** или ***Far***, но и просто, запуская программу с командной строки, прямо из оболочки ***Windows***. Хотя, конечно же, это менее удобно.

*6.7.2.3. Текстовый редактор **TextPad***



Конечно, работать с программами при помощи командной строки можно, но в принципе это достаточно мало удобно. А с другой стороны, при использовании интегрированных средств разработки, хотя они и предоставляют программистам много удобств, однако там есть и достаточно много неудобств. В частности, для достаточно простых программ, которые состоят всего из одного файла, не разбиты на проект, состоящий из нескольких файлов, то среда разработки ***c*** долгой загрузкой, всякими свистками, звонками и посторонними окнами, может оказаться чрезмерно избыточной.

Поэтому, наверное, для достаточно маленьких и простых программ, золотой серединой может оказаться использование обычных текстовых редакторов, которые в принципе дополнительно оснащены возможностью запуска компилятора и интерпретатора языка ***Java***. И одним из, именно таких текстовых редакторов, является текстовый редактор ***TextPad***, который в принципе соответствует всем стандартам в системе ***Windows*** и работа в нем напоминает работу в обычном редакторе ***Notepad***. Этот редактор можно найти

на странице ***www.textpad.com***, вот на этой странице, с таким адресом. И при этом, этот текстовый редактор относится к условно свободно распространенным программам. Установка его достаточно проста. Требования достаточно неприхотливы. Она устанавливается практически на всех компьютерах при любой версии ***Windows***, от ***Windows*** 95 до ***Windows*** XP.

И, если мы его запустим, то внешний вид этого текстового редактора, выглядит вот таким образом. Попробуем теперь загрузить ту же самую программу, которую мы только что запускали с помощью командной строки. Для этого щелкнем на кнопку Открыть. Вот появляется стандартное окно открытия файлов, в котором, как мы видим, в строке выбора типов файлов выбран тип ***Java***. Хотя, если мы щелкнем на треугольнике, то можем видеть, что в этом же текстовом редакторе можно редактировать и файлы ***C***++, ***HTML*** файлы, и т.д.

Но, сейчас нас интересуют файлы ***Java***. Откроем сейчас директорию 1, в которой и находится наш пример ***Hello***, выберем этот файл, и щелкнем на кнопке ***Open***. Вот этот файл загрузился в нашем текстовом редакторе. Мы можем увидеть его внешний вид и содержание. Он состоит всего лишь из 5 строчек. И попробуем теперь его запустить и выполнить. И первое, что нам надо сделать, это его скомпилировать. Для этого выберем пункт меню ***Tools*** и далее пункт ***Compile Java*** (скомпилировать ***Java*** программу). Щелкнем на этой команде.

Теперь можно видеть, что в левой части вот этого окна, у нас наряду с окном ***Hello.java***, в котором мы и находимся, появилось еще одно окно ***Command Results***. Если мы на него перейдем, то мы можем видеть, что наша программа была успешно скомпилирована. Попробуем теперь ее запустить, выполнить ее. Для этого, опять развернем пункт меню ***Tools*** и на этот раз выберем пункт ***Run Java Application***, т.е. выполнить программу ***Java***. Щелкнем на этой команде.

И можем видеть ту надпись ***Hello World***, которую в принципе мы и должны были получить. Далее нам предлагают нажать на какую-либо клавишу, чтобы выйти из этого консольного окна. В результате мы опять попадаем на наше окно текстового редактора. Попробуем теперь выполнить и запустить другое, более сложное приложение. Для этого опять щелкнем на кнопке Открыть. Мы попадаем в ту же самую директорию, в которой мы были в прошлый раз. Перейдем на Мои документы затем в каталог ***ImageViewer***, щелкнем на нем дважды, и выберем вот этот файл ***ImageViewer***. Это простое приложение, которое позволяет просматривать разного рода графические файлы.

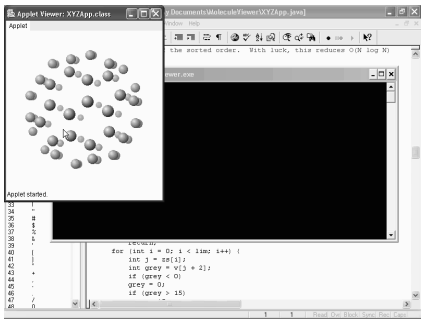
Щелкнем на кнопке ***Open***, и мы можем видеть текст этой программы. В принципе, он уже гораздо более длинный, чем текст, который мы видели в прошлый раз, текст программы ***Hello***. Ну, попробуем ее опять-таки

скомпилировать и запустить на выполнение. Для этого вновь развернем меню ***Tools***, далее выберем команду ***Compile Java*** (скомпилировать), можем переключиться и проверить, что компиляция прошла успешно. А теперь попробуем ее выполнить. Для этого опять меню ***Tools*** и ***Run Java Application*** (запустить программу ***Java***). Вот, можно видеть окно, которое появилось перед нами.

В принципе оно пока пустое, но у нас здесь есть пункт меню ***File***. Если мы на нем щелкнем, то появляется возможность открывать новые файлы. Щелкнем на команде ***Open***. Появляется вот такое окно, в котором мы можем видеть все файлы, находящиеся в нашей текущей папке, в папке ***ImageViewer***. Выберем теперь один из этих двух пунктов, в которых находятся файлы с картинками. Если мы щелкнем на каком-либо из других файлов, то просто-напросто ничего открываться не будет, поскольку эта программа открывает только файлы с картинками.

Выберем, например, вот этот файл и щелкнем на кнопке ***Open***. И вот это изображение загружается в окне нашего приложения. А если мы еще раз развернем меню ***File*** и активизируем команду ***Open***, выберем вот этот файл ***Tower***, щелкнем на кнопке ***Open***, то можно видеть другую картинку. Т.е. таким образом, в этом окне мы можем открывать картинки. Закроем это окно. Для этого в меню ***File*** выберем пункт ***Exit*** и нажмем на любую клавишу.

6.7.2.4. Апплеты



Можно видеть, что вот эти два приложения, которые мы только что рассматривали, это программы, написанные на языке ***Java*** и самостоятельно выполняющиеся в своей среде. Но с другой стороны, особенностью языка ***Java*** является то, что с его помощью можно выполнять апплеты под управлением веб-браузеров. Рассмотрим какой-либо пример из этой сферы. Рассмотрим ту самую молекулу, которую мы уже видели в наших предыдущих уроках. Для этого опять щелкнем на кнопке Открыть. Перейдем в Мои документы, далее вот эта папка ***MoleculeViewer***. И, как мы видим, здесь у нас находится два файла, которые представляют ***Java*** программы. Это, ***XYZApp*** и ***Matrix3D***.

Загрузим оба файла. Щелкнем сначала на вот этом. Это в принципе стандартный пример из пакета ***Java*** и можно видеть, что он достаточно большой и объемный. Наряду с этим файлом, здесь же находится, если мы

щелкнем на кнопку ***Open***, другой файл, в котором у нас находятся все вспомогательные функции для рисования трехмерного изображения, всякого рода преобразования от трехмерного изображения к двухмерному, к ее проекциям. Попробуем теперь скомпилировать этот апплет и запустить его. Развернем меню ***Tools***. Далее ***Compile Java***. Можно увидеть, что компиляция прошла совершенно успешно.

А теперь попробуем посмотреть этот файл, как он выполняется. Для этого подойдем к пункту меню ***Tools*** и здесь у нас есть опять-таки две возможности ***Run Java Application*** и ***Run Java Applet***. Правда, если мы его запустим, как ***Run Java Application***, то мы получим просто-напросто ошибку, поскольку это программой, как таковой, не является. Это аплет, который встраивается в ***HTML*** страницу, для показа в веб-браузере. Поэтому закроем это окно, далее опять ***Tools*** и выберем команду ***Run Java Applet***.

Далее нас спрашивают, из какого ***HTML*** файла мы хотим посмотреть этот аплет. Здесь у нас есть три возможности. Это ***example***1, ***example***2 и ***example***3, в каждой из которых используется этот аплет. Ну, возьмем ***example***2, щелкнем на нем, щелкнем теперь на кнопке ***Yes***, и можно видеть этот апплет в этой веб-странице. Причем, надо сказать, что, так как мы запустили нашу программу, выполняется именно только апплет, без показа содержимого нашей веб-страницы, которая вот эту среду показа апплета просто-напросто игнорирует.

Сейчас у нас есть возможность двигать изображение. Оно трехмерное. Вот таким образом все это будет выглядеть. Закроем теперь это окно, щелкнем на этом крестике. Если же мы теперь развернем меню ***Tools*** и вновь выберем команду ***Run Java Applet***, опять появляется тот же самый пример, и на этот раз согласимся с показом ***example***1. Щелкнем на кнопке ***Yes***, и на этот раз мы можем видеть, что наш апплет выглядит несколько по-другому. Поскольку в примере ***example***1 этот же апплет используется с несколько другими параметрами. В результате чего рисунок выглядит по-другому. Но опять-таки свойства у него те же самые. Мы можем его крутить таким образом. Закроем это окно. Щелкнем на крестике.

Попробуем теперь загрузить эту ***HTML*** страницу с примером ***example***1 и посмотреть, как же все это выглядит там. Для этого, опять-таки щелкнем на кнопке Открыть. На этот раз тип файлов выберем не файлы ***Java***, а ***HTML*** файлы. Редактор ***TextPad*** позволяет это сделать, и откроем первый же пример ***example***1. Щелкнем на кнопке ***Open*** и можем видеть текст нашего текстового редактора, со всеми его тегами и параметрами. Здесь же можем видеть вставку нашего апплета. Вот это место. Именно вот здесь мы вставляем наш апплет и выводим его, как изображение на ***HTML*** странице. Но и в принципе, те, кто знаком с ***HTML*** страницами, может увидеть ее все достаточно стандартные возможности.

Попробуем теперь посмотреть, как же все это выглядит в виде ***HTML*** страницы. Для этого щелкнем на кнопке ***View in Web Browser***. Вот запускается программа ***Internet Explorer***, внутри которой мы можем видеть всякие надписи, ***MoleculeViewer***, ***example***, ***source*** и т.д. И среди них изображение нашей молекулы, нашего апплета. Вот этот квадрат и является апплетом, который представляет собой ***Java*** программу. И в отличие от всех остальных обычных стандартных рисунков на окне, он является трехмерным и может двигаться. Вот, если мы воспользуемся мышью, нажмем на нее, то можно видеть эту молекулу с разных сторон. Закроем это окно. Щелкнем на кнопке Закрыть.

Конечно же, тот же самый ***HTML*** файл можно запустить не только из редактора ***TextPad***, но и обычными стандартными средствами ***Windows***. Для этого, например, подведем курсор к панели задач нашего компьютера, запустим ***Internet Explorer***, далее пункт меню ***File***, ***Open***, ***Browse***, далее перейдем в папку ***MoleculeViewer*** и откроем какой-либо из примеров. Ну, откроем первый ***example***1. Щелкнем на нем дважды. Далее ОК.

И вот появляется то же самое окно, которое мы только что видели. С теми же самыми надписями и с тем же самым апплетом. В принципе этого и нужно было ожидать. Этот апплет выполняет все то же самое. Мы можем схватить его мышью и крутить нашу молекулу. Т.е. апплет, это абсолютно стандартная для веб-страниц часть, которая может на нем отображаться в виде встроенного окна и выполнять различного рода функции.

В частности в нашем случае, это отображать вот такую трехмерную молекулу. Закроем теперь программу ***Internet Explorer***, щелкнем на крестике, вернемся в наше стандартное окно редактирования. Ну, надо, конечно же, сказать, что все то же самое, в том числе и просмотр апплетов, можно, конечно же, организовать и из командной строки. Вот мы активизировали стандартную командную строку и теперь выполним следующие действия.

Во-первых, скроем левую панель, для того чтобы у нас здесь отображалось все то, что мы будем выполнять. Для этого щелкнем на клавишах ***Ctrl***+***F***1, и теперь напишем такие строчки. Во-первых, ***Javac***, для того чтобы мы могли скомпилировать. Именно, ***Javac*** программа компиляции. Далее ***XYZApp***.***Java***. Хотя, в принципе процесс компиляции можно и пропустить, поскольку все это здесь уже сделано. А соответствующий файл с расширением ***Class***, уже можно видеть на экране. Перекомпилируем еще раз, щелкнем сейчас на клавише ***Enter***.

Как мы видим, компиляция прошла успешно, никаких, так сказать, предупредительных сообщений не появилось, и теперь выполним эту программу. Т.е. покажем наш апплет на экране. Конечно, мы можем просто-напросто запустить какой-либо из этих ***HTML*** файлов, щелкнув на нем

дважды. Но мы можем и посмотреть апплет, который входит в эти примеры. Но для этого в стандартном пакете ***Java***, есть специальный просмотрщик апплетов, ***Appletviewer***. Так и напишем. ***Appletviewer*** и далее ***XYZApp***.

Щелкнем на клавише ***Enter***. Конечно же, еще раз надо напомнить, что большие и маленькие буквы надо набирать точно так же, как это набрано в соответствующем файле. Расширение ***Class*** или ***Java***, здесь уже не нужно. Щелкнем на клавише ***Enter***.

Можно видеть, что в результате всего этого, мы все-таки допустили ошибку, поскольку нам нужно было запускать не файл ***XYZApp***, а именно ***example***.***HTML***, внутри которого находится наш апплет. Что нам в принципе и указали, при помощи вот этого текста нашей ошибки. В случае необходимости мы можем просмотреть этот текст целиком, для этого надо нажать на клавиши ***Ctrl***+***F***2, чтобы правая панель тоже исчезла. Вот теперь текст нашей ошибки отображается целиком. Опять нажмем на клавиши ***Ctrl***+***F***2, чтобы появилась наша панель, и напишем команду ***Appletviewer*** еще раз.

Теперь, например, укажем файл ***example***1.***HTML***. Причем, надо сказать, что программа ***Appletviewer*** устроена таким образом, что она реагирует только на апплеты, которые находятся на этом ***HTML*** файле, пропуская все остальные строчки изображения, картинки и т.д. Щелкнем сейчас на клавише ***Enter***, и мы можем увидеть наш апплет, запущенный при помощи командной строки. Но в принципе он выглядит, конечно же, точно также. Щелкнем на крестике. Закроем это окно. И закроем командную строку. Для этого щелкнем на клавише ***F***10. Щелкнем на ***Yes***. И мы вернулись в наше стандартное окно редактирования.

Надо отметить еще одно обстоятельство, а именно, как же наш текстовый редактор ***TextPad*** запускает ***Java*** программы на компиляцию и выполнение. Для этого мы можем зайти в пункт меню ***Configure***, далее ***Preferences*** (опции) и далее у нас здесь есть пункт ***Tools***, в котором сконцентрированы все те три пункта, которые у нас появляются в пункте меню ***Tools***. Первый из них, это ***Compile Java***, который, как мы видим, состоит из команды ***javac***.***exe***, той же самой стандартной команды, которую мы выполняли с командной строки. И дальше, в качестве параметра, ему передается имя нашего файла и текущая папка. А у пункта ***Run Java Application***, программой является ***java***.***exe***, а в качестве параметра, опять-таки имя и папка.

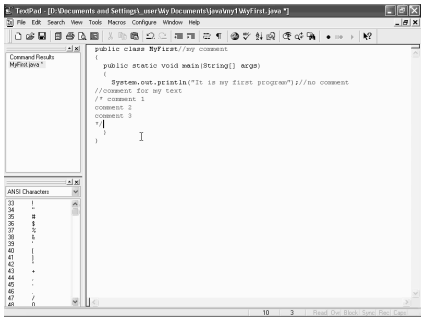
А у третьего пункта ***Run Java Applet***, программой является ***Appletviewer***, та же самая программа, которую мы только что запускали. В качестве параметра, опять передаются имя и директория. Щелкнем теперь на кнопке ***Cancel***. Здесь мы ничего менять не будет. Это мы просто-напросто открывали для ознакомления. И мы вновь вернулись в стандартное окно редактирования.

6.7.3. Основные структуры языка

6.7.3.1. Простая программа на языке Java

Рассмотрим поближе самую простую программу, которую можно себе представить на языке ***Java***. Это программа, которая просто вводит на консольное окно простейшее сообщение. Надо сказать, что исходный файлы на языке ***Java***, это простой обычный текстовый файл, который содержит в себе одно или несколько описаний классов.

Причем надо сказать, что транслятор ***Java*** всегда предполагает, что этот текстовый файл имеет расширение ***java***. Т.е. после имени файла стоит точка и далее ***java***. В результате работы транслятора компилятора, получается код, который для каждого из классов, записывается в отдельный файл, имя которого совпадает в точности с названием класса, затем идет точка и слово ***class***. Т.е. у него расширением является слово ***class***. Причем надо отметить, что в языке ***Java*** так же, как и в языке ***C***, об этом всегда стоит помнить, большие и маленькие буквы, надо всегда различать. Как написано имя класса, так и должно быть написано имя соответствующего файла.



И первое, что теперь нам нужно делать, это написать первую строчку, в которой мы объявим наш класс. Поэтому напишем таким образом ***public***, далее ***class***, и далее имя, которое мы хотим придать нашему классу. Ну, пусть это будет, например, ***MyFirst***. Рассмотрим теперь подробнее, что мы написали. Первое, что мы написали, это слово ***public***, которое является модификатором доступа. Этот модификатор управляет доступом к данному коду из других частей программы. В данном случае слово ***public*** означает, что наш класс доступен из любых других частей.

Таких модификаторов доступа всего 3. Это слово ***public***, которое мы только что использовали, также ***private*** и ***protected***. Назначение каждого из них мы рассмотрим позже. Следующее слово, это слово ***class***, которое напоминает о том, что все объекты в языке ***Java*** находятся внутри классов. Подробный разговор о классах мы продолжим позже, а сейчас просто будем считать, что это некие контейнеры, внутри которых и заключена вся логика программы, и все ее операторы, которые определяют работу нашего приложения. Т.е. классы, это некие элементарные законченные строительные блоки, из которых и состоят наши приложения и наши апплеты, написанные на языке ***Java***. Следующим словом, после слова ***class***, является слово ***MyFirst***. Надо отметить,

что слово ***public*** и слово ***class***, являются так сказать уже внутренними словами ***Java***, т.е. словами, которые входят в базовый словарь языка ***Java***. А слово ***MyFirst***, это уже слово, которое так сказать мы придумали, в качестве названия для слова ***class***.

Вместо ***MyFirst***, могло быть любое слово, которое мы можем себе представить, и которое состоит из букв и цифр. Например, вместо ***MyFirst***, могло быть ***FirstSample*** или просто ***First*** или ***My*** и т.д. Правило образования для этих имен достаточно просты и не очень строги. Единственное, первая буква должна быть именно буквой, а остальная часть может быть любой комбинацией из букв и цифр. Единственное, в качестве такого имени, нельзя использовать зарезервированные слова языка ***Java***. Например, нельзя использовать вот это слово ***public*** или слово ***class***. После этой первой строки, нам надо написать наш дальнейший код, который заключим внутри фигурных скобок. Вот мы написали одну фигурную скобку, скобка, которая открывает текст нашей программы и вторая фигурная скобка, между которыми и будет находиться весь наш текст.

В языке ***Java***, так же, как и в языках ***C*** и ***C***++, фигурная скобка используется для выделения частей программы, ее блоков. Причем, надо сказать, что поскольку язык ***Java***, так же, как и большинство других языков программирования, игнорирует пробелы, и поэтому всегда надо стремиться, чтобы открывающая и закрывающая фигурные скобки, находились одна под другой, для лучшей визуализации и читабельности нашей программы. Теперь же напишем следующую строчку. Причем, опять-таки для лучшей визуализации, немножко отступим от края текста, для того чтобы все, что находилось внутри вот этих фигурных скобок, выделялось небольшим отступом. Поэтому введем лишние пробелы и далее напишем такую строчку. Опять ***public***, далее ***static***, ***void***, далее ***main***, ***String***, ***args***, закроем скобки.

Рассмотрим теперь подробнее, что мы написали в этой строчке и здесь надо отметить первое из слов, после ***public***, которое мы уже рассматривали, это слово ***static***, с помощью которого можно объявлять и методы и переменные класса, используемые для работы с классом в целом. Методы, которые объявлены со словом ***Static***, могут непосредственно работать только с локальными и статическими переменными. Ну, подробнее о слове ***Static***, так же, как и о ***public*** и так же, как о слове ***void***, мы будем разговаривать попозже. А сейчас просто учтем это.

Следующее слово, это слово ***void***, которое означает, что наш метод никакого значения не возвращает. Ну а если бы наш метод возвращал какое-либо значение при выполнении, например, целое, тогда бы мы написали здесь ***integer*** или какое-либо другое слово, которое соответствует типу возвращаемого значения. Ну, а слово ***void*** означает, что возвращаемого

значения, как мы сказали, вообще нет. Ну и самое главное слово, которое есть в этой строчке, это слово ***main***. В принципе метод с этим именем и является самым главным, поскольку ***Java***-интерпретаторы при работе начинают свою работу именно с вызова этого метода. И поэтому метод с именем ***main*** обязательно должен присутствовать в нашем классе. Правда, для компилятора это ограничением не является. Мы может скомпилировать нашу программу даже и в том случае, если метода с именем ***main*** не существует. Но при запуске уже этой откомпилированной программы для выполнения, мы столкнемся с ошибкой. Мы получим сообщение о том, что нет метода с именем ***main***.

После слова ***main*** у нас внутри круглых скобок написано ***String*** и ***args***. Что это означает? Это означает те параметры, которые мы хотим передать методу ***main***. Все эти параметры пишутся внутри круглых скобок и разделены между собой точкой с запятой. Ну, сейчас у нас точки с запятой нет, поскольку у нас всего один параметр. Это параметр с именем ***args*** и класса ***String***. Причем вот эти квадратные скобки, открывающие и закрывающие, которые находятся после слова ***String***, означают, что это у нас массив, а не единичный элемент этого класса.

А теперь же перейдем на следующую строчку и напишем то, что должен выполнять этот метод, метод ***main***. Опять таки все это нужно написать внутри фигурных скобок. Напишем открывающую фигурную скобку и далее закрывающую. Опять-таки их лучше расположить одну под другой, для того чтобы мы в процессе написания нашей программы, точно знали, какая закрывающая фигурная скобка соответствует открывающей. А теперь же напишем следующую строчку, для написания которой опять чуть-чуть отступим от уровня вот этих фигурных скобок и напишем такую строчку ***System***, далее точка, ***out***, точка, ***println***, откроем скобку и внутри вот этой обычной скобки, напишем какой-либо текст, который мы хотим, чтобы наша программа выводила нам на окно. Например, введем следующий текст. ***It is my first program***. Далее закроем кавычки, закроем скобку, точка с запятой.

*6.7.3.2. Простая программа на языке **Java**. Комментарии*

Рассмотрим теперь опять подробнее ту строчку, которую мы только что написали. В данном случае мы используем стандартный объект ***System.out*** и его метод ***println***, который предназначен для вывода сообщений на окно. Заметим, что для вывода метода этого объекта, между самим объектом и его методом стоит точка, которая и является разделителем между объектом и его методом. А далее, внутри круглых скобок, написан тот текст, который мы хотим вывести, причем он у нас помещен внутри двойных кавычек.

Также надо сказать, что после вывода текста, метод ***println*** выполняет переход на следующую строку. Отметим еще, что в языке ***Java***, методы, так же, как и функции в любом другом языке программирования, могут иметь один или несколько параметром. Либо их не иметь вовсе. Причем, надо сказать, что в языке ***Java***, даже если метод не имеет ни одного параметра, тем не менее, скобки писать все равно надо. В этом случае, это просто-напросто будут пустые скобки, открывающая и закрывающая.

Например, в нашем случае, если бы мы не хотели ничего выводить на экран, а просто хотели бы провести перевод строки, чтобы у нас каретка вывода переместилась бы на строчку вниз, тогда мы могли бы использовать метод ***println*** со скобками, но без, так сказать, внутренней части. Вот таким образом. Вот, если это выделить и удалить целиком, вот таким образом тоже можно использовать этот метод, который ничего не будет выводить на наше окно, а будет просто проводить перевод строки. Но, сейчас возвратим обратно текст, который у нас тут был. Для этого, щелкнем на кнопке отмены последнего действия.

Вот появился этот текст, и отметим, что вместо метода ***println***, мы могли использовать и метод ***print***, без этих двух букв ***ln***. Но, тогда в этой конкретной программе, ничего от этого не изменится. Но в принципе, разница только в том, что после вывода этой строки, перехода на следующую строку не будет. И следующий текст выведется непосредственно после вот этой строчки. Т.е., например, следующая буква следующего текста выведется после буквы ***m***, сразу же за ней, а не со следующей строки. Ну, опять-таки вернемся к тому исходному варианту, который у нас был, щелкнем на кнопке отмены последнего действия. Теперь у нас наша программа уже полностью готова и попробуем ее записать в какой-либо файл, а затем выполнить. Для этого щелкнем на кнопке сохранения, ***Save***.

Вот появляется стандартное окно сохранения файлов, в котором у нас требуют, чтобы мы задали имя для нашего документа и его тип. Ну, как мы сказали имя документа должно совпадать с именем вот этого класса ***MyFirst***, поэтому тут так и напишем ***MyFirst***. Причем, большие и маленькие буквы здесь должны быть точно в таком же соотношении, как и в названии класса. Ну и, конечно же, тип нашего файла будет не ***HTML***, а файл ***Java***, с расширением ***java***, вот этот.

Далее создадим новый каталог, который назовем каталогом ***java***, перейдем в эту папку. И внутри нее создадим еще один каталог, который назовем ***my***1. Т.е. это у нас просто будет первый пример. Щелкнем дважды. И теперь уже, наконец, можем нажать кнопку ***Save***, чтобы внутри вот этого каталога ***my***1, появился файл ***MyFirst*** с расширением ***java***. Вот мы уже сохранили эту программу, и, как мы видим, после того, как мы этому текстовому редактору

дали понять, что это у нас файл ***Java***, вот эта картинка с нашим текстом, приобрела несколько более живописный вид. Соответствующие фигурные скобки, ключевые слова и т.д. выделены соответствующим цветом. Конечно, в принципе при работе без этого тоже можно обойтись, но это дает дополнительные подсказки для визуализации нашего текста нашей программы.

Теперь же попробуем эту программу скомпилировать и выполнить. Для этого нам нужно зайти в пункт меню ***Tools***, далее выбрать пункт ***Compile Java***. Компиляция прошла успешно, что подтверждается появившейся надписью ***Tool completed successfully***, операция завершена успешно. Перейдем обратно на наш файл ***Java*** и попробуем его выполнить. Для этого опять меню ***Tools*** и на этот раз ***Run Java Application***, запустить программу ***Java*** на выполнение.

Ну, вот появился результат действия нашей программы с надписью ***It is my first program***. Конечно, результат не особенно впечатляет, появилась всего одна строчка, это все, что нам удалось получить в результате действия нашего кода. Закроем данную программу щелкнув на крестик, и вернемся в наш текстовый редактор. Ну, конечно же, в принципе, от нескольких таких строчек, большего ожидать было и невозможно. Но зато мы смогли написать нашу первую программу самостоятельно и заставили ее что-то сделать.

Теперь, после того, как мы успешно скомпилировали и выполнили вот эту простейшую программу, рассмотрим основы синтаксиса языка ***Java***. И одной из этих основ, является то, что язык ***Java***, так же, как и большинство других языков программирования, довольно свободно обращается с пробелами, кроме табуляции и сменами перевода строки. Т.е., например, всю вот эту программу можно было написать вообще в одну строчку или в две строчки и т.д., вставлять между отдельными словами любое количество пробелов, переводов строки, знаков табуляции и т.д.

Главное, чтобы между двумя соответствующими словами был хотя бы один пробел или один перевод строки, или же один знак табуляции. Все остальное, все вот это выравнивание, это уже для удобства чтения программы, а сам язык ***Java***, всего вот такого выравнивания, красивого выстраивания текста и т.д., не требует. Правда, для удобства и читабельности нашей программы, лучше, конечно же, выравнивать текст относительно друг друга, соответствующим образом. Еще одним элементом, который очень часто применяется во всех программах, а не только в языке ***Java***, это комментарии, которые в ***Java***, так же, как и в большинстве других языков программирования просто-напросто игнорируются при выполнении программы.

Т.е. соответствующий компилятор, просто пропускает эти строчки. Таким образом, в программу можно добавлять столько комментариев, сколько мы считаем нужным. Совершенно при этом, не опасаясь, что от этого наша

программа увеличится в объеме после компиляции. В языке ***Java*** есть несколько способов, ввода в текст нашей программы комментариев. Чаще всего для комментариев используются две косые черты и далее текст комментария. Например, введем после вот этой строчки, которая у нас выводит на окно, вот эту надпись, какой-либо комментарий. Для этого введем здесь две косые черты и далее введем сюда какой-либо текст. Например, напишем таким образом ***no comment***. Точно также, комментарии можно написать не только, к этой строчке, но и к любой другой.

Например, вот к первой строчке. Введем две косые черты и далее ***my comment***. Конечно же, комментарии можно писать не только после какого-либо оператора, но и можно сделать так, чтобы комментарий находились в самом начале нашей строчки. Например, щелкнем на клавише ***Enter***, освободим строчку, далее введем две косые черты, что означает начало комментария, и далее напишем ***comment for my text***. Если же нам нужно ввести комментарии, которые будут состоять из нескольких строчек, конечно, можно начинать каждую из них с двух вот таких косых черточек, но можно поступить и по-другому.

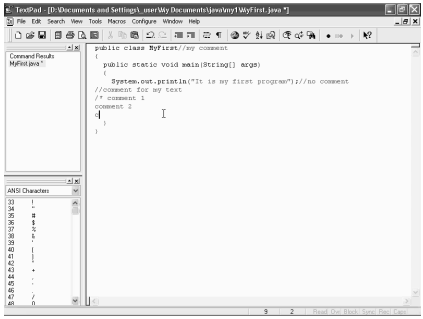
А именно, ввести в косую черту и звездочку, далее ввести любой текст. Например, напишем ***comment*** 1, после этого у нас будет вторая строчка комментария, третья строчка. И, если мы считаем, что на этом у нас комментарии должны заканчиваться, то введем звездочку и косую черту, после чего восстанавливается ввод обычного текста. Т.е. все то, что находится между косой чертой и звездочкой, это начало комментария и, наоборот, звездочка и косая черта, все это является комментарием. И опять-таки компилятор все это будет пропускать.

Если мы теперь запустим нашу программу, скомпилируем и выполним ее, то можно увидеть, что вот эта программа с нашими комментариями, хотя сейчас текст программы стал уже гораздо длиннее, будет выполнять абсолютно так же, как и прежде. Попробуем это сделать. Развернем меню ***Tools***, далее ***Compile Java***, компилировать. Компиляция прошла успешно. И далее попробуем ее выполнить. Для этого в меню ***Tools*** выберем пункт ***Run Java Application***. Мы можем видеть, что результат абсолютно тот же самый. Надписи и ***It is my first program*** появились вот в этой первой строчке нашего окна. Закроем программу. Щелкнем на крестике и вернемся в наш обычный режим.

6.7.3.3. Типы данных. Переменные. Целые числа

Основным элементом для хранения информации в языке ***Java***, является переменная. Причем, надо отметить, что в языке ***Java*** каждая переменная обязательно имеет тип. При объявлении переменной, сначала указывается ее

тип, а затем ее имя. Например, можно объявить переменную таким образом: ***int***, т.е. ***integer***, целое число и точка с запятой. Вот теперь мы объявили переменную ***i*** и она у нас будет целым числом. Надо сосредоточить внимание на том, что после оператора ***int i***, мы поставили точку с запятой, поскольку именной точка с запятой и является разделителем между операторами. Имя переменной может быть практически любое. Единственное ограничение, это то, что первая буква переменной обязательно должна быть именно буквой, а следующие знаки в нашей переменной могут быть буквами и цифрами.

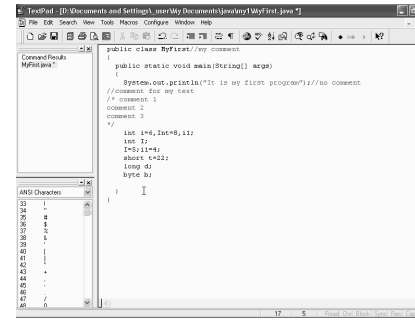


Причем надо сказать, что термины, буквы и цифры в языке ***Java***, имеют несколько более широкое толкование, поскольку буквами считаются не просто символы от ***A*** до ***Z***, и большие и маленькие, а также знак подчеркивания, но и любой символ в кодировке уникода. Например, можно использовать и немецкие буквы и греческие. Например, такие, как альфа, бета, пи и т.д. Правда, есть ограничения. Символы плюс, минус, знак деления, пробел т.д., нельзя использовать в именах переменных. И еще надо отметить, что регистр клавиатуры имеет существенное значение. Т.е. вот эта переменная ***i***, эта одна переменная. А если мы напишем ***I*** с большой буквы, это уж будет совершенно другая переменная.

Например, если мы напишем после этого определения еще ***int*** затем ***I*** большое, то мы определили две целочисленные переменные, ***i*** маленькая и ***I*** большая. Правда, надо отметить одно обстоятельство, что в качестве имен переменной нельзя использовать зарезервированные слова языка ***Java***. Например, вот такие, как ***public***, ***Class***, ***Static***, ***void***, ***String*** и т.д. Правда, есть одно исключение. В качестве переменной мы можем использовать какой-либо из зарезервированных слов, если какую-либо из букв этого слова, мы напишем с другим регистром. Например, в качестве целочисленной переменной, здесь мы можем написать идентификатор ***Int***, если мы первую букву ***I*** возьмем вот такую большую.

Т.е. у нас теперь определены в этой строчке две переменные. Переменная ***i*** целого типа, и переменная ***Int*** тоже целого типа. Надо еще отметить то, как мы написали эту строчку. Т.е. мы одним определили ***int***, объявили две целочисленные переменные ***i*** и ***Int***, написав между ними запятую. Т.е. таким образом, мы можем одним определением задавать несколько переменных. После того, как мы объявили переменную, ее, конечно же, нужно

инициализировать или во время объявления, или чуть позже. Но, например, переменной ***I*** можем присвоить какое-либо значение. Вот этой большой " ***I*** " присвоим, например, значение 5. Для этого, конечно же, сначала нужно написать имя этой переменной, затем знак равенства и далее выражение на языке ***Java***, которое задает требуемое значение. В данном конкретном случае, мы просто-напросто написали цифру 5.



А инициализацию переменной можно проводить и во время ее задания. Например, вот в этой строчке мы можем задавать, что переменной ***i*** сразу же задается, например, число 6. А переменной ***Int***, мы с самого начала присваиваем значение 8. Далее, через запятую зададим еще одну переменную. Например, ***i***1, никакого значения присваивать тут не будем. А присвоим чуть позже, а вот после строчки ***i*** равно 5, напишем ***i***1 равно, допустим, 4.

Язык ***Java*** является строго типизированным. Это значит, что тип каждой переменной обязательно должен быть объявлен. Мы не можем использовать какую-либо переменную, если она не определена, к какому типу она относится. Типов переменных в языке ***Java*** восемь. Из них 4 — целых типа, 2 — числа с плавающей точкой, еще один тип, — символьный тип ***char***, используемый для представления символов в формате уникод.

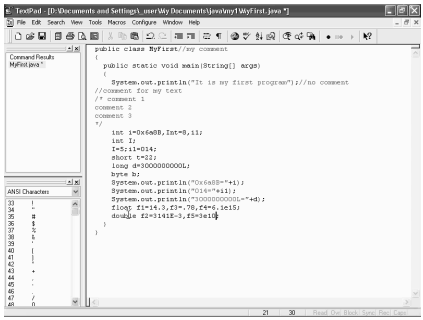
И последний, это булевский тип ***Boolean***, применяется для логических величин, которые имеют два значения ***True*** или ***False***. Рассмотрим их более подробно. Первый из них, это тип ***int***, который мы уже использовали в нашей программе. В ее памяти, требующей для хранения этой переменной, 4 байта и он определяет целое число в диапазоне от (-2) до (+2) миллиардов. Следующий тип, это тип ***short***, который тоже является целочисленным типом. Единственное отличие, что для его хранения требуется всего 2 байтов и соответственно, диапазон его намного меньше от (-32) до +32 тысяч.

Зададим, например, в качестве применения этого типа, например, переменную ***t***, которой присвоим значение 22 точка с запятой. Следующий тип, который у нас есть целочисленный, это тип ***long***, который самый большой в представлении целых чисел и требует для себя 8 байт памяти. Но и в связи с этим, соответственно, в этих переменных можно хранить достаточно большие целые числа и достаточно большие отрицательные целые числа, конечно же, тоже. Возьмем в качестве переменной ***long***, например, переменную ***d***. Определять ее пока не будем. И последний целочисленный тип, который

можно использовать, это тип Байт. Пусть это будет у нас переменная ***b***. Байт, как видно из самого своего названия, занимает в памяти один байт ровно. И его диапазон меняется от (-128) до 127.

В языке ***Java***, диапазоны целочисленных типов, не зависят от машины, на которой выполняется программа. Это облегчает работу программистов при переносе программного обеспечения с одной платформы на другую, и даже из одной операционной системы в другую. В отличие, например, от языка ***C***, где переменные типы ***int***, ***long*** и т.д., могут иметь в своем представлении различное количество байтов для своего хранения в памяти, что можем привести к совершенно различным результатам, при выполнении одной и той же программы на различных машинах, на различных платфор MAX.

6.7.3.4. 16-ричные, 8-ричные и действительные числа



Целые числа, которые мы до этого использовали, это были обычные целые числа в десятичном представлении. Но кроме этого, в языке ***Java*** можно использовать числа и в 16-рычном представлении и в 8-рычном представлении. А также можно использовать и длинные целые числа. Для записи этих чисел в языке ***Java*** имеются специальные возможности. Например, присвоим переменной ***i***1 вот в этом месте не число 4, а какое-либо число в 8-рычном представлении. Например, напишем вот таким образом, ноль 10. Это будет означать, что мы ***i***1 присвоили не число 10, а на самом деле число 8, поскольку 10 в 8-рычном представлении, это есть 8. А признаком того, что это число является числом в 8-рычном представлении, является цифра 0, которая у нас стоит перед самим числом 10. Правда, такая запись чисел, в принципе, достаточно сило противоречит нашим представлениям. Поэтому числа в 8-рычном представлении, в принципе используются достаточно редко.

И надо сказать при этом, что вот, например, вот такая запись, 019, будет просто-напросто ошибкой и при компиляции, компьютер выдаст нам ошибку. Потому что цифра 0, которая стоит впереди, указывает калькулятору, что это 8-рычная, а цифра 9, которая здесь используется, явно выпадает из 8-рычного представления числа. Поэтому, вот, например, 014, это правомерная запись, а 019, конечно же, уже нет. Для записи же числа в 16-рычном представлении, впереди надо приписывать суффикс 0X. Например, переменной ***i*** присвоим

какое-либо число в 16-рычном представлении. Для этого напишем таким образом: 0, затем X и берем 6, далее пусть будет, например, 8 и, например, Б, поскольку для 16-рычного представления числа, кроме цифр от 0 до 9, используются еще буквы: A, B, C, ***D***, ***E*** и ***F***, как большие, так и маленькие.

В принципе числа в таком представлении достаточно удобны для использования в компьютере. Например, для задания каких-либо адресов памяти. Кроме этого, мы можем использовать и большие числа. Для этого, например, для этой переменной ***D***, напишем таким образом. ***D***, затем знак равенства. Далее напишем большое число, которое состоит из большого количества цифр.

Например, таким образом. В конце напишем букву ***L***, что дает знать компьютеру, что это большое число, а не синтаксическая ошибка. Попробуем целые числа, которые мы только что задавали, вывести на окно нашей консоли. Для этого, опять-таки используем вот эту конструкцию ***System.out.println***. Для этого проще всего ее скопировать, выделим. Правая кнопка мыши, Copy.

Вставим ее вот в этом место. ***Paste***. Выровняем. И далее напишем таким образом. Сначала выделим вот это число ***i***. Для этого напишем в кавычках вот это число, 0X, далее 6а, 8б, точно так же, как и здесь было написано. Равняется. Закрыть кавычки. Плюс. И далее само число ***i***. Закрыть скобку. Точка с запятой. Скопируем еще раз. На этот раз воспользуемся опять правой кнопкой мыши. Далее ***Paste***, поскольку в буфере обмена у нас уже есть эта строчка. Далее опять кавычки. На этот раз возьмем 014. Мы видим эту переменную ***i***1, равняется, кавычки и напишем переменную ***i***1. Закрыть скобку. Точка с запятой. Опять скопируем вывод, ***Paste***, и выведем вот это большое число ***D***. Для этого опять кавычки, напишем вот это число 3. Затем буква ***L***.Равняется. Закрыть кавычки. Далее плюс. И число ***D***. Закрыть скобку, точка с запятой. Теперь же скомпилируем эту программу и посмотрим, что у нас получилось. Для этого ***Tools***, далее ***Compile Java***, скомпилировать.

Компиляция у нас прошла успешно. Переключимся на нашу программу и запустим ее на выполнение. Для этого ***Tools*** и ***Run Java Application***. И вот результат выполнения нашей программы, мы можем видеть на нашем экране. Вот это 16-рычная переменная, 6а 8б, это на самом деле число 27275. А 8-рычное число ***i*** с индексом, которое означает минус впереди и далее 14, это на самом деле число 12. А вот это большое число с тройкой впереди и с буквой ***L*** в конце, это вот такое число. Т.е. все те присваивания к целым числам, которые мы проводили, все они успешно были выполнены. Закроем это консольное окно и вернемся в наш обычный текстовый редактор.

Кроме целых чисел, где в языке ***Java*** существует возможность использования и чисел с плавающей точкой, чисел, имеющих дробную часть.

Для представления такого типа чисел, в языке ***Java*** есть две возможности. Это ***float***, она требует 4 байта для своего представления. Введем переменную такого типа, Пусть эта переменная будет у нас, например, ***f***1. И переменная типа ***double***. Пусть это будет ***F***2. Для переменной типа ***double*** в памяти требуется 8 байт. Это так называемое число с двойной точностью и в нем у нас есть 15 значимых десятичных цифр, в отличие от класса ***float***, у которого приблизительно 6-7 значимых десятичных цифр.

Числа с плавающей запятой типа ***float*** и ***double*** можно записывать в различных представлениях. Например, переменной ***F***1 присвоим какое-либо значение. Например, просто 14 целых и 3 десятых. Вот такая форма записи. Кроме этого, введем, например, переменную ***F***3, которой присвоим значение 0,78. Как видим, число 0 впереди мы пропустили, это вполне допустимо. Кроме таких прямых представлений чисел, можно числа представлять и в такой форме. Например, введем переменную ***F***4, которой присвоим значение, например, 6,1e в 15 степени. Для разнообразия зададим теперь переменную типа ***double***. Например, ***f***2 присвоим значение 3141e в — 3 степени. Это приблизительно число пи, как можно заметить.

Зададим еще переменную ***F***5, которой присвоим значение, например, 3e в 10 степени. Вот таким образом можно задавать числа с плавающей точкой. Числа, имеющие дробную часть. Числам, типа ***float***, можно приписывать справа букву ***F***, для явного задания его типа. А если буква ***F*** нет, тогда числа по умолчанию считаются типа ***double***. Но им можно и явно приписывать букву ***D***. Надо еще отметить, что в языке ***Java*** есть три специальных числа с плавающей точкой, которые означают собой положительную бесконечность, отрицательную бесконечность и не число, которые используются для разного рода переполнений и ошибок. Например, при делении положительного числа на 0, получается положительная бесконечность, при делении 0 на 0, или извлечения квадратного корня, получается неопределенность, т.е. число ***num***, не число.

Попробуем теперь эти числа тоже вывести на наше окно приложения. Для этого опять скопируем вот эту строчку, в которой у нас происходит вывод, используем ***Paste*** и далее таким образом. Выведем ***f***1. Далее в качестве разделителя используем 2 пробела, затем закрыть кавычки, выведем число ***f***3. Затем в качестве разделителя опять 2 пробела. Затем ***f***4. Закрыть скобку, точка с запятой. Теперь выведем числа из второй строчки. Опять правая кнопка мыши и далее ***Paste***. Выведем число ***f***2, затем в кавычках два пробела и затем число ***f***5. Закрыть скобку. Точка с запятой. Попробуем теперь все это скомпилировать и выполнить.

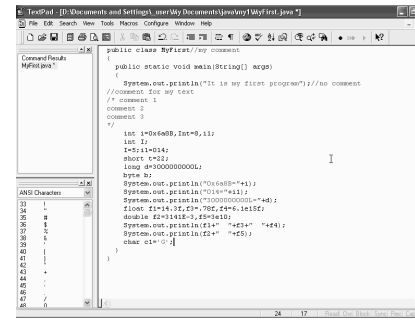
Для этого опять пункт меню ***Tools***, далее ***Compile Java***. В результате компиляции, мы получили три ошибки. Т.е. все три присваивания, и ***f***1, и ***f***3, и

f4, там, где мы задавали тип переменной ***float***, у нас оказались с ошибками. В чем тут дело? Дело в том, что, поскольку мы после чисел написали букву ***F***, то, например, вот это число 14, 3 компьютер по умолчанию взял, как ***double*** с двойной точностью. И присваивание его числа к переменной, которая является числом с одинарной точностью, произошло соответственно с ошибкой. То же самое, относительно переменных ***F***3 и ***F***4, во всех трех случаях. Что же нам теперь делать? Как выйти из этой ситуации.

Для этого перейдем в наш текст и далее в конце каждого числа, припишем букву ***f***, что будет означать, что мы так сказать, принудительно задаем его спецификацию, как число ***float***, числу с одинарной точностью. Теперь опять попробуем скомпилировать то же самое, ***Tools***, далее ***Compile Java***, на этот раз, как видно, все прошло вполне успешно. А теперь запустим. ***Run Java Application***.

И можно увидеть, что в результате у нас получилось. Вот 14,3; 0,78, так же, как здесь есть. А вот число 6,1, которое у нас здесь было, задалось в несколько странной форме, 6,0 и далее 999 и 8. Ну, приблизительно, конечно, это 6,1. Это уже ошибка округления данного числа. А число с двойной точностью, 3,141, которое получилось из вот этого числа 31,41e — 3, и вот это 3e10 у нас конвертировалось в 3,0 с "e" в 10 степени. Закроем теперь окно и вернемся в наше приложение, будем дальше писать нашу программу.

6.7.3.5. Символьный и булевый типы данных



Еще один тип переменной, который мы можем использовать на языке ***Java***, это символьный тип. Определим какую-нибудь переменную этого типа. Для этого мы должны использовать слово ***char***, символьный тип. Пусть это у нас будет, например, переменная ***c***1, которую зададим, как какую-либо букву. Например, пусть это будет буква ***J***. В принципе, символьный тип переменной, это некоторая буква в формате уникода, который была специально разработан для обработки практически всех символом, всех письменных языков, которые только существуют в мире. Он является двухбайтовым кодом, и в нем допускается 65536 символов.

Этот формат намного богаче стандартного набора ***asi***2, который представляет собой однобайтовый код, состоящий из 128 символов. Или его расширения, которое состоит из 256 символов, которое на самом деле является

подмножеством уникода, и эти символы представляют собой просто-напросто первые 256 его символов. Символьную переменную можно задавать не только в виде отдельной буквы, но и можно ее задавать просто числом. Хотя, конечно, оно числом не является. Тогда, просто-напросто берется соответствующая буква из набора уникода, которая соответствует этому порядковому номеру. Например, возьмем переменную c2 и присвоим ей значение 66.

Также возьмем переменную c3, которой сопоставим значение 66 в 16-рычном представлении. Для этого напишем, таким образом, 0X и 66. Вот эти два, конечно же, должны быть две различные буквы c2 и c3. Кроме этого, то же самое можно задавать и таким образом, c4 переменная, равняется, далее кавычки, внутри которых напишем таким образом. Косая черта ***U***, что означает управляющий символ, который показывает, что дальше будет идти 16-рынчная величина, состоящая из 4 цифр. 0066, закрываем кавычки. Так вот, c4 и c3, должна быть одна и та же буква. Попробуем теперь вывести все это на печать. Для этого ***Enter***, далее правая кнопка мыши и ***Paste***. Далее выведем все эти переменные на экран. Для этого напишем таким образом, c1+c2+c3+c4. Закрыть скобку, точка с запятой.

Если теперь мы попробуем скомпилировать и запустить эту программу, то получим следующее. Скомпилируем сначала. Щелкнем на пункте ***Compile Java***. Компиляция прошла успешно. Теперь ее запустим. ***Run Java Application***. И если посмотреть теперь на консольное окно, то получим несколько неожиданный ответ. Там, где у нас должны были получиться 4 буквы, мы получили 341. В чем дело?

Дело в том, что, так как мы написали вывод на печать, c1+c2+c3+c4, он просто-напросто сложил вот эти 4 значения, просто, как целые числа. Но и в результате получили некое целое число 341. Закроем это консольное окно. Попробуем сделать по-другому. Для того, чтобы компьютер понял, что мы хотим вывести именно в виде буквы c1+c2+c3+c4, введем вначале просто-напросто вывод пустого знака. Далее плюс. Теперь попробуем опять скомпилировать. ***Compile Java***. Все прошло достаточно успешно, и теперь запустим наше приложение. ***Ran Java Application***. И вот в результате как раз мы и получили 4 буквы. ***J***, ***B***, ***F*** и ***F***. Как видим, последние две буквы совпадают. Вот эти два представления в виде 66 и в 16-рычном виде. И в таком виде они выдают нам одну и ту же букву ***F***. Закроем это окно, и мы опять вернулись в наш обычный текстовый редактор.

Кроме префикса, косая черта ***U***, которая означает символ в формате уникода, есть еще несколько управляющих символов. Это такие, как возврат на одну позицию, которая выдается, как косая черта и маленькая буква "***b***". Вот таким образом. Можно задавать символ табуляции. Это будет косая черта

и буква "***t***", символ, при помощи которой можно перейти на следующую строку, это косая черта и буква "***n***". Символ, при помощи которого можно задавать возврат клетки, это косая черта и буква "***r***". Также можем задавать вывод двойных и обычных кавычек. Для этого попробуем ввести косую черту и далее вот такие кавычки, тогда будет присваиваться просто-напросто кавычка. И если возьмем простую, одинарную кавычку, тогда будет присваиваться переменной ***c***1 значение одной кавычки, для, например, вывода на печать. Кроме этого, если мы зададим две косые черты подряд, то это будет означать символ одна обратная косая черта.

Попробуем применить эти наши управляющие символы для вывода в этой строке. Например, выведем знак кавычки. Для этого косая черта и кавычка, как мы помним. Далее пускай выводиться одна косая черта, это у нас переменная ***c***1. Также возьмем перевод строки. Для этого, как мы помним, надо в кавычках написать косую черту и букву ***n***, которая у нас является спецификатором перехода на другую строку. И далее, в конце, еще знак возврата на одну позицию назад. Это у нас будет косая черта и ***b***. Далее плюс и введем еще одну какую-нибудь букву. Пускай выведем букву ***o***. Закрыть кавычки. Мы здесь забыли ввести управляющий символ, косая черта.

Попробуем теперь это скомпилировать и посмотрим, что у нас получилось. ***Tools***, далее ***Compile Java***, скомпилируем. Все прошло успешно. Теперь запустим наше приложение ***Run Java application***. И вот можно посмотреть, что у нас получилось. В начале мы ввели знак кавычек. Вот при помощи этого символа. Затем у нас косая черта из переменной с1. Затем буква Б, это у нас переменная с2 дальше у нас перевод строки.

Вот этот управляющий символ косая черта и ***n***, в результате чего мы перешли на новую строчку. Затем у нас С3 и С4, как мы помним, они оба должны были вводить букву ***F***. Одна буква ***F*** есть, а второй буквы ***F*** нет, поскольку вместо нее у нас произошел возврат на одну позицию назад и затем мы на эту букву ***F*** сверху, так сказать написали цифру 0. В результате одна буква ***F*** у нас затерлась. Т.е. все получили все так, как мы и задавали. Закроем теперь это окно приложения.

Рассмотрим последний тип переменной, это переменная типа ***Boolean***. Пусть эта переменная у нас будет переменная ВВ и она может принимать два значения, ***True*** или ***False***. Пусть ВВ принимает значение ***True***, а зададим другую переменную, В1, которой присвоим значение ***False***. Точка с запятой. Надо отметить, что в отличие от языка ***C***, тип переменной ***Boolean***, никакого отношения к целым числам не имеет. Попробуем теперь опять вывести на экран. Для этого правая кнопка мыши, ***Paste*** и далее выведем эти переменные В и В1. Для того чтобы у нас система вывела абсолютно правильно эти булевские переменные, введем вначале опять-таки кавычки, пустой знак, для

того чтобы компьютер понял, что нам нужны символьные представления этих переменных.

Теперь попробуем скомпилировать. ***Compile Java***. Компиляция прошла успешно, и теперь запустим нашу программу. Вот, что получилось в результате всего этого. Вот компьютер нам в последней строчке вывел эти значения, ***True*** и ***False***. Закроем окно нашего приложения и выйдем в наш обычный режим. В языке ***Java***, кроме переменных, можно использовать и контакты, для обозначения которых используется ключевое слово ***final***. Например, вот эти две переменные B и B1 представим в виде констант. Для этого впереди, перед словом ***Boolean*** напишем слово ***Final***.

Что будет означать, что эти два идентификатора B и B1 являются константами, т.е. присвоить им значения можно только один раз. Если же нам нужно употреблять константу, которой будут доступны нескольким методам внутри одного класса, тогда они называются константами класса и тогда перед словом ***Final*** нужно написать еще слово ***Static***. И тогда константы такого класса будут задаваться вне метода ***mine***. И в этом случае ее можно будет использовать в других методах того же класса. Например, зададим его вот в этом месте.

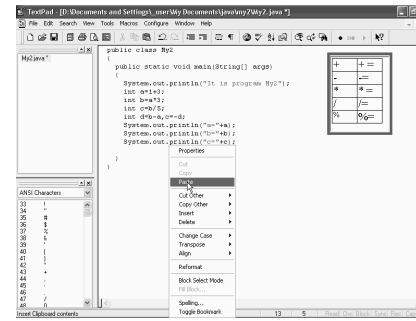
Щелкнем здесь на ***Enter*** и далее напишем, таким образом, ***public***, далее ***static***, ***final***, поскольку это константа и далее переменную, какого типа мы хотим задавать. Например, пусть эта будет ***double***. Число с плавающей точкой двойной точностью Возьмем эту переменную, как пи и введем для него численное значение. Например, такое, 3,14. Это, конечно, не точно, но все-таки это в каком-то приближении число пи.

6.7.3.6. Операторы. Арифметические действия

Начнем нашу работу с минимальной программы, написанной на языке ***Java***, которая просто-напросто пока выводит на консольное окно вот такой текст:' ***it is program My***2'. Эту программу назовем ***My***2, и сохраним ее на жестком диске. Для этого щелкнем на вот этой кнопке сохранения. Откроем новую папку, которую назовем ***My***2. Щелкнем на ней дважды, чтобы туда зайти и выберем имя нашего файла. Как мы помним, оно должно быть точно такое же, как и название нашего класса, причем с учетом регистра.

Т.е. первая буква М, в нашем случае должна быть большая. Наберем ***My***2 и щелкнем на кнопку Сохранить. Тип файла ***Java*** у нас есть, поэтому все остальное у нас остальное нормально. Теперь же рассмотрим вопрос, как в языке ***Java*** используется операторы. Для выполнения обычных арифметических операций сложения, вычитания, умножения и деления, в языке ***Java*** используется стандартные знаки, плюс, минус, знак умножения — звездочка, деление — косая черта, также проценты, это остаток от деления

целых чисел. Напишем, например, маленький пример, при помощи которого используем вот эти несколько арифметических операций.



Например, создадим переменную ***a*** класса ***integer***, присвоим ей значение, например, 1+3, далее возьмем опять целое число, пусть это будет ***b***, которому присвоим значение ***a*** умножить на 3, точка с запятой. Далее возьмем целое число ***c***, которому мы присвоим значение ***b*** деленное на 5, например. Точка с запятой. Затем возьмем целое число ***d***, которому присвоим значение ***b-a***. Ну и тут же на этой же строчке возьмем еще одну переменную ***c***, которой присвоим значение минус ***d***. Теперь же попробуем все эти числа вывести на наше консольное окно. Для этого воспользуемся вот этим оператором ***println***. Выделим его. Далее правая кнопка мыши, Copy. И сюда вставим. Правая кнопка, ***Paste***.

Теперь напишем таким образом, кавычки, это у нас сначала будет переменная A, далее плюс и выведем ее значение ***a***. Закроем скобку. Точка с запятой. Правая кнопка. ***Paste***. Выведем на этот раз переменную B. Далее то же самое с переменной ***c***. И затем то же самое с переменной ***d***. И последняя переменная, это переменная ***e***. Тут мы допустили ошибку, у нас ***c*** повторяется два раза. Конечно же, следующую переменную нужно было обозначить, следующей буквой ***e***. Вставим сюда и выведем на экран значение переменной ***e***. Закрыть скобку. Точка с запятой.

Введем еще одну переменную. Пусть эта будет ***f***, которой присвоим значение, например, ***b*** и остаток по модулю, т.е. вот этот значок проценты и далее 5. Остаток по модулю при делении на 5. Выведем значение этой переменной на экран. Правая кнопка мыши, ***Paste***, далее ***f***, равняется и его значение. Теперь же скомпилируем эту программу. Для этого, как всегда нам нужен ***Tools***. Далее ***Compile Java***. Компиляция прошла успешно, и поэтому теперь мы можем его запустить на выполнение.

Для этого ***Tools*** и ***Run Java Application***. Вот мы видим результат выполнения нашей программы. Первой строчкой является та строка, которая у нас была здесь: '***it is program My***2'. А далее у нас идут значения наших переменных. Переменная ***a*** равна 4, что в принципе и понятно, ***b*** — это у нас будет 4 умноженное на 3, будет 12, а ***c*** у нас будет 2, поскольку получается, как деление ***b*** на 5, но поскольку целиком 12 на 5 не делится, то это у нас целочисленное деление.

Далее у нас значение ***d***, которое равняется ***b-a***, 12- 4=8, все правильно и затем переменной ***e*** мы присвоим значение ***d*** с обратным знаком. Получается (-8). И последнее значение ***f***, это целочисленное деление числа ***b***, которое равно 12, на 5, что остается в остатке число 2. Закроем это окно, щелкнем на этом крестике. Наряду с обычными арифметическими операциями, широко используется и арифметические операции сокращенного вида, которая позволяет выполнять дополнительные операции. Например, вместо операции ***a***, равняется ***a***+4, вот вместо этой операции можно написать операцию при помощи вот этого оператора плюс равно. Т.е. вместо него, можно написать такое выражение, что несколько сокращает внешний вид этой строчки. ***a***+ =4. Что по результату абсолютно одно и тоже.

Такие сокращения операторы обычно получаются приписыванием пред символом равно, соответствующих арифметических символов — плюс, минус, умножения и т.д. Выполним еще несколько операций. Увеличим, ***b***, например, в 3 раза. Для этого пишем ***b***, далее знак умножения равно и 3. переменную ***c*** тоже увеличим. Плюс — равно. Увеличим на значение ***a***+***b***. И далее напишем значение ***c***, как остаток по модулю от деления, например, на 5.

Теперь же выведем все эти значения ***a***, ***b*** и ***c*** на экран. Для этого нам вполне нужно скопировать вот эти 4 строчки. Выделим. Далее правая кнопка мыши и Copy. Вставим сюда. Правая кнопка и ***Paste***. Теперь, единственное нам, осталось выполнить еще одну операцию с переменной ***d***, с ней мы ничего не делали. Поэтому, например, для ***d*** применим вот этот последний неиспользуемый нами оператор: минус равно, т.е. увеличим значение ***d*** на какое-либо выражение. Например, уменьшим на 3 единицы.

В принципе мы можем выровнять все эти строчки, чтобы все они были одинаковыми. И после этого, попробуем скомпилировать эту программу. Для этого, пункт меню ***Tools***, далее ***Compile Java***, все прошло успешно. Переключимся на ***Java*** и запустим теперь наше приложение ***Run Java Application***. Вот мы можем видеть результат выполнения нашей программы. Число "***a***" у нас увеличилось на 4. Если оно раньше было 4, еще 4. то получилось значение 8. "***b***", же у нас увеличилось в 3 раза. Вместо 12 стало 36. "***c***", после выполнения вот этих двух операции, увеличение на ***a***+***b*** и далее целочисленное деление на 5, стало единицей, как остаток от деления на 5, а "***d***" мы уменьшили на 3 и вместо 8 стало 5. Т.е. все операции опять у нас прошли нормально.

Закроем теперь это приложение. Щелкнем на этом крестике и вернемся в наш обычный текстовый редактор. Программисты, которые часто сталкиваются с разного рода программами, конечно же, знают, что одним из самых распространенных действий над числовыми переменными, являются увеличение и уменьшение их значения на единицу. И для этих действий, так

же, как и в языке C***i*** и ***C***++, в языке ***Java*** есть специальные операторы. Например, если мы напишем, таким образом, ***a***++, то после выполнения этой операции, значение ***a*** увеличится на единицу. А если напишем ***b***– –, то ***b*** уменьшится на единицу.

Эти операторы ++ и – – можно использовать не только отдельно, для увеличения и уменьшения на единицу, но и внутри выражения. Например, таким образом. Создадим переменную типа ***integer k***1, пусть его значение будет 5. Запятая. Создадим еще переменную ***k***2, которой присвоим тоже значение 5 и далее напишем такие выражения: ***int s***1, которое будет равняться 2 умноженное на ++ ***k***1.

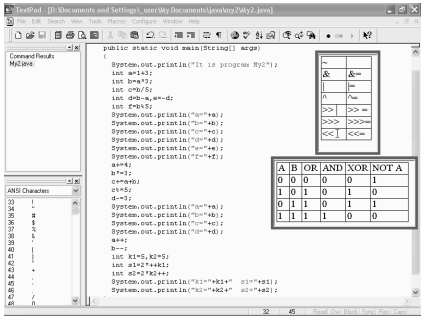
Теперь напишем, практически, такое же выражение: ***int s***2, равняется 2 умножить, но напишем теперь не ++ ***k***1, а ***k***2 и затем ++, т.е. переставим плюс и переменную местами. Посмотрим теперь, что у нас получилось. Для этого выведем это значение на экран, скопируем вот эту строчку. Правая кнопка мыши, Copy. Далее вставим сюда. Правая кнопка мыши, ***Paste***. Выведем значение ***k***1. Для этого напишем, таким образом, в кавычках ***k***1 равняется, затем само значение ***k***1. Теперь выведем значение ***s***1. Равняется. ***s***1, закрыть скобку, точка с запятой.

Теперь опять вставим этот текст. ***Paste***. Кавычки. На этот раз нам нужно значение ***k***2. Равняется. Кавычки. Само ***k***2. Теперь выведем ***s***2. Кавычки и плюс ***s***2. Теперь попробуем скомпилировать эту программу и выполнить ее. Для этого ***Tools***, ***Compile Java***. Все прошло успешно, и теперь запустим саму программу: ***Run Java Application***. И вот можно видеть, что у нас получилось: ***k***1 и ***k***2 у нас стало по 6 их значение. Мы присвоили им 5 и далее при выполнении вот этих выражений, каждая из них увеличилась на единицу, в связи с тем, что около них не написали ++. А вот значение ***s***1 и ***s***2 у нас разное. ***s***1 — 12, а ***s***2 — 10. Дело в том, что если ++ у нас написан спереди, то сначала происходит увеличение ***k***1 на единицу и затем вот это полученное выражение 6, уже используется для дальнейших вычислений. В результате чего и получается 6x2=12.

А при вычислении ***s***2, тут дело происходит немножко по-другому. Для вычисления вот этого выражения ***s***2, берется старое значение ***k***2, 5 и 2 раза 5 получается 10, и лишь после этого ***k***2 увеличивается на единицу. Закроем теперь это окно. Щелкнем на вот этом крестике.

6.7.3.7. Битовые операции. Математические функции

Если мы работает с любыми целочисленными типами, то мы можем применять и оператор, который работает непосредственно с битами, из которых и состоят эти числа. И для этих операции в языке ***Java*** существует несколько операторов. Вот они. Это оператор не, который меняет



A	B	OR	AND	XOR	NOT A
0	0	0	0	0	1
1	0	1	0	1	0
0	1	1	0	1	1
1	1	1	1	0	0

соответствующий бит на противоположный. Оператор "и". Оператор "или". Оператор, исключающий "или", а также несколько операторов, которые выполняют сдвиг вправо и влево. Вот этот оператор выполняет сдвиг вправо, в битовом представлении, вот этот оператор выполняет сдвиг в битовом представлении влево, а оператор, состоящий из трех таких значков, заполняют старшие разряды нулями в битовом сдвиге. Ну и в правом столбце здесь же приведены те же самые операторы в виде использования, как сокращенное выражение. Если мы хотим их использовать аналогично с тем, как мы использовали плюс, минус и т.д., ранее. В нижней таблице приведены действия этих операторов, если у нас есть один бит, А и В, и их значения, вот такие, как внутри таблицы, то в случае действий операторов ***or***, ***and***, ***xor*** и ***not A*** принимают вот такие значения, как видно из таблицы. В языке ***Java***, наряду с обычными операторами, которые работают с целыми числами, действительными и т.д., есть и операторы, которые работают с булевыми выражениями и булевыми переменными. Это такие операторы, как ***and***, ***or***, ***xor*** и т.д. Вот этот оператор, оператор "и ", следующий оператор, оператор "или".

Далее оператор ***xor***, исключающий "или", и следующие два оператора, это операторы ***or*** и ***and*** с быстрыми вычислениями. Т.е. при этих двух операторах, выражение до конца не вычисляются, если ее значение получается уже при выполнении первого из операндов.

Следующей, это оператор отрицания ***not***, но далее три оператора, которые повторяют, оператор "и", "или" и исключающий "или", но вместе с присваиванием. А также есть оператор проверки на равенство, равны ли два выражения друг другу, и оператор проверки, если два выражения не равны друг другу.

А также есть оператор, который заменяет оператор ***if-then-else***. Ниже, в этой таблице приведены результаты работы операторов ***or***, ***and***, ***xor***, и ***not A***, "и" , "или", и, исключающий "или", при различном значений первого и второго операндов А и В, для булевых выражений. Как и во всех языках программирования, в языке ***Java*** существует тоже своя иерархия операторов. Т.е. то, в каком порядке они выполняются в выражениях.

Конечно же, самый большой приоритет имеют круглые и квадратные скобки, а далее, вниз по убывающей. Операторы, находящиеся на одном уровне, выполняются слева направо, за исключением тех случаев, когда они

имеют правую ассоциативность. Это относится к вот этим операторам, они выполняются справа налево. Также, справа налево выполняется оператор с присваиванием. Во всех остальных же случаях, выполняются операции подряд, слева направо, если операторы одного уровня. А если случаются операторы разного уровня, то сначала выполняется тот оператор, который стоит выше по приоритету.

Например, сначала, конечно же, выполняется умножение и деление, затем, во второй очереди, сложение и вычитание. Надо сказать, что, если при написании какого-либо выражения у нас есть подозрение, неопределенность, в каком порядке выполняются те или иные операции, то лучше всегда воспользоваться скобками, которые, конечно же, имеют наивысший приоритет. И в этом случае, мы точно можем быть уверены, что операторы будут выполняться именно в той последовательности, в которых мы их и задумали. И это может предотвратить некое количество ошибок.

Наряду с обычными операторами, в языке ***Java*** реализована возможность использования набора математических функций, часто оказывающихся необходимыми при решении разного рода задач. Для этого в языке ***Java*** есть официальный класс ***math***, в который и входят все эти функции. Например, чтобы извлечь квадратный корень, для этого у нас есть специальная функция ***sqrt***. Зададим переменную с двойной точностью, ***double***. Пусть эта переменная ***x***, которая будет равна 2.

И затем зададим еще одну переменную с двойной точностью. Пусть это у нас будет ***y***, которая будет равняться корень из ***x***. Для этого напишем таким образом: ***math***, тот класс, в котором у нас сосредоточены все математические функции и далее ***sqrt***. И далее в скобках ***x***. Вот теперь у нас к ***y*** будет присвоено значение корня из ***x***, т.е. корня из 2. Выведем это число на экран. Для этого скопируем эту строчку. Выделим. Правая кнопка мыши. Сору. Встанем сюда. Правая кнопка мыши и ***Paste***. И теперь напишем таким образом. Переменная ***y*** равняется, и само значение ***y***. Закрыть скобку, точка с запятой.

Теперь скомпилируем и выполним эту программу, для этого ***Tools***, далее ***Compile Java***, теперь опять ***Tools*** и ***Run Java application***, вот можем видеть, чему равняется корень из 2, 1,4 и т.д. Все цифры по порядку. Закроем это окно. Кроме корня квадратного, в классе ***math***, еще много всяких математических функции. Это такая функция, как ***pow***, которая реализует возведение числа в степень, в корень мы уже видели.

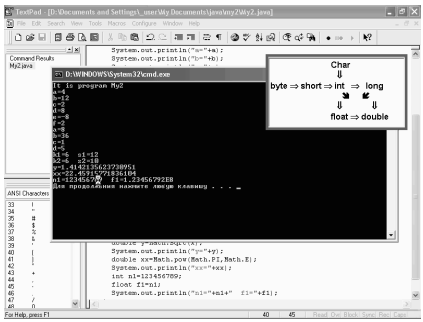
Далее синус, косинус, тангенс, котангенс, это все обычные тригономические функции. Далее экспонент и логарифм, и две математические константы, это ***Pi*** и ***e***. Например, попробуем возвести какое-нибудь число, например, ***Pi*** в какую-либо степень при помощи возведения

pow. Для этого воспользуемся следующим способом. Напишем, таким образом, ***Double*** число с двойной точностью, пусть это будет у нас ***xx***, равняется, и далее напишем, таким образом, ***math***, точка, ***pow***.

Далее возведем число ***Pi***, ***math Pi***, ну, например, в степень "e", ***math*** "e". Закрыть скобку. Точка с запятой. Теперь попробуем это все вывести на окно. Для этого правая кнопка мыши и ***Paste***. Теперь напишем, что это у нас переменная ***x***. И выведем ее значение. Закрыть скобку, точка с запятой. Если же мы сейчас скомпилируем эту программу, и запустим, ***Tools***, ***Run Java application***, то мы можем удовлетворить наше любопытство и узнать, чему равняется число "пи" в степени "e". Это 22 целых и т.д. Закроем теперь это окно и вернемся в наш обычный текстовый редактор.

6.7.3.8. Операторы. Преобразования типа

Очень часто возникает необходимость, преобразовывать один числовой тип в другой. Тут на диаграмме показаны разрешенные преобразования, при которых компилятор не выдает ошибки, и проводит их без всяких вопросов. Это преобразование из байта в ***short***, из ***short*** в ***integer***, и из ***integer*** в ***long***, а также из ***char*** в ***integer***, происходит без всяких потерь, так же, как преобразование из числового типа ***float*** в ***double***, а также из ***integer*** в ***double***. А вот преобразование из ***long*** в ***double***, из ***integer*** в ***float***, а также из ***long*** в ***float***, может происходить с потерей точности. Например, создадим целочисленную переменную типа ***int n***1, которое у нас будет равняться 123456789.



При преобразовании его к типу ***float***, пусть это будет ***f***1, равняется ***n***1, как раз и будет происходить потеря точности. Посмотрим, как это происходит. Для этого скопируем вот этот вывод на консоль. Правая кнопка мыши, Copy, далее вставим сюда, правая кнопка мыши, ***Paste***.Выведем ***n***1 и ***f***1 на печать. Для этого напишем, таким образом, ***n***1 равняется, ***n***1. И далее выведем ***f***1. Закрыть скобку. Точка с запятой. Попробуем это скомпилировать. Для этого ***Tools***, далее ***Compile Java***, далее ***Tools*** и ***Run Java application***. И можно видеть результат, который мы получили: ***n***1, это точное значение, как мы задавали 123456789. А ***f***1 у нас даже, если учесть, что у нас 10 в восьмой степени, тем не менее, как мы видим, 567, а затем, вместо 89 у нас 92, произошла некоторая потеря точности. Закроем теперь это консольное приложение.

И учтем еще, что при выполнении какого-либо бинарного оператора (сложения, вычитания, умножения и т.д.) перед выполнением операции оба оператора, которые стоят по разные стороны от знаков сложения, вычитания и т.д., преобразовываются в числа, которые имеют одинаковый тип. И происходит это по следующему законам. Если, хотя бы один из них, или тип ***double*** двойной точностью, то второй тоже преобразовывается к такому же типу. Если же этого нет, и один из операторов имеет тип ***float***, действительный, тогда второй же преобразовывается к этому же типу. Если же оба числа целые, и один из оперантов имеет тип ***long***, тогда второй тоже преобразовывается к типу длинного целого, к типу ***long***.

Ну и, в сущности, если оба операнда не являются не действительными, и не длинными целыми, тогда оба операнда преобразуются к типу ***integer*** (***int***). Очень часто, при написании программ, нам требует и обратная операция. Т.е. преобразование, например, от действительного числа к целому. Такие преобразования на языке ***Java*** возможны, однако, конечно же, может происходить потеря точности. Эти преобразования называются примитивного типа и синтаксически это должно выглядеть, как пара скобок, внутри которых указывается желательный тип, а затем само имя переменной.

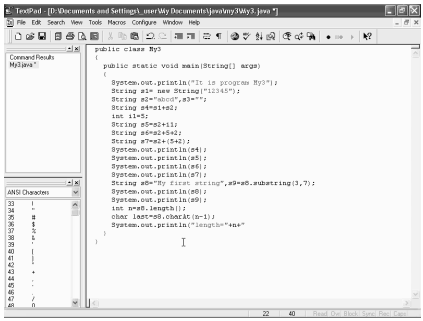
Например, если мы хотим произвести обратное преобразование вот этой переменной ***f***1, которая у нас является типа ***float*** к целому, мы можем написать таким образом: ***int*** и далее пусть это будет ***n***2, равняется. Далее в скобках мы должны указать непосредственно, какой тип преобразования нам нужен. Нам нужен тип ***Int***. Преобразование к типу ***Int***, целому и далее саму переменную. Это переменная ***f***1. Выведем теперь это число тоже. Для этого воспользуемся опять буфером обмена.

Щелкнем на правую кнопку мыши, ***Paste*** и далее напишем, таким образом, ***n***2, равняется, и само значение ***n***2. Закрыть скобку. Точка с запятой. Скомпилируем: ***Tools***, далее ***Compile Java*** и теперь выполним: ***Tools*** и ***Run Java application***. Вот можно увидеть, что у нас при этом получилось. Значение ***n***2, это 1234567, а далее у нас уже 2 цифры отличные. Вместо 89 — 92, что в принципе соответствует вот этому действительному числу ***f***1. Закроем это окно.Напишем еще один пример такого рода. Например, возьмем действительное число типа ***double***. Пусть это будет, например, ***z***1, которое будет равняться 2 целое 7 десятых, точка с запятой.

Далее возьмем преобразование типа к типу ***integer***. Пусть это будет ***i***1, которое будет равняться преобразованию типа ***integer*** и далее эта наша переменная ***z***1. Точка с запятой. И в этом случае у нас к переменной ***i*** и ***i***1, у нас будет присвоено значение 2, которое получается из 2,7 отбрасыванием дробной части. Ну, конечно же, нас больше бы устроило преобразование типа, при котором мы округляем число ***z***1 до ближайшего целого. Это можно

сделать немножко по-другому. Для этого введем еще одну целую переменную. Пусть это будет ***ii***2, равняется. Далее преобразование к типу целого ***int***, и далее воспользуемся функцией ***math***.Далее ***Round***, округление, которое как раз и даст нам округление ***z***1 до ближайшего целого. Вот в этом случае мы уже получим число 3. Проверим это, выведем все это на окно консоли. Для этого щелкнем на правую кнопку мыши и ***Paste***. Далее ***ii***2 равняется, +***ii***.2, далее ++ и внутри в кавычках напишем ***ii***1. Кавычки и значение ***ii***1.Закрыть скобку. Точка с запятой. После ***ii***1 лучше еще написать знак равенства и далее скомпилируем эту программу. ***Tools***, ***Compile java***, далее ***Tools***, ***Run Java Application***, запуск программы. И можно увидеть, что как мы и ожидали у ***ii***2 значение 3, которое получится округлением из 102, 7, а у переменной ***ii***1 получило значение 2, которое получается отбрасыванием дробной части. Закроем консольное окно и вернемся в наше обычное окно редактирования.

6.7.3.9. Строки: создание, слияние, выделение подстроки, длина



Рассмотрим, какие возможности есть у языка ***Java*** для работы со строками. Для этого начнем работу с простейшей подготовки на языке ***Java***, которая пока реализована только выводом одной строки на консольное окно "***It is program My***3".Сохраним его на жестком диске. Для этого щелкнем на кнопку сохранения, создадим новую папку под названием ***my***3. Откроем ее. Для этого щелкнем на ней дважды и введем имя для нашего файла. Как мы помним, большие и маленькие буквы имеют значения, поэтому напишем первую букву М большую, ***My***3, расширением ***Java*** и щелкнем на кнопку Сохранить.

Для работы со строками стандартная библиотека языка ***Java*** содержит специальный встроенный класс ***String***. Причем надо заметить, что любая строка, которую мы заключим в кавычки, сразу же представляет собой отдельный экземпляр класса ***String***. В языке ***Java***, встроенного типа для строк, такого, как, например, целого числа ***integer***, для действительного ***real*** и т.д., нет. Т.е. все, что касается строк, нужно представлять себе в виде экземпляра какого-то класса. Создадим несколько переменных типа ***String***. Для этого щелкнем здесь на кнопку ***Enter*** и напишем таким образом. ***String***, создадим экземпляр этого класса. Пусть, например, это будет ***s***1, знак равенства, и сразу присвоим ему какое-либо значение. Для этого, так же, как и при создании любого другого класса, напишем ***new String***.

В скобках укажем кавычки, внутри котрых напишем то, что мы хотим ему присвоить. Например, числа от 1 до 5. Закроем кавычки, скобку и точка с запятой. В принципе путь, который мы выбрали для создания новой строки ***s***1, вполне допустим, но на самом деле, в языке ***Java*** встроена поддержка более простой возможности. Для этого мы можем взять тип ***String***, далее возьмем переменную ***s***2. И после знака равенства, мы можем написать прямо то, что мы хотим ему присвоить внутри кавычек. Например, буквы ***abcd***. Закроем кавычки, точка с запятой.

И эта строка инициализации, которая находится выше, и вот эта, которая находится ниже, на самом деле абсолютно равнозначны, обе создают новую строку. Создавать можно не только строку, в которой будет какое-либо значение, но и пустую строку. Для этого напишем здесь запятую, далее ***s***3, знак равенства, и напишем пустую строку, в которой нет ни одного символа. Кроме этого, есть возможность объединения двух строк и присвоение третьей. Для этого напишем таким образом. ***String***, переменная ***s***3, которая будет равняться объединению строки ***s***1 и ***s***2. Это будет ***s***4.

Для этого напишем ***s***1+***s***2, достаточно простой синтаксис этой операции. Надо отметить, что к строке можно прибавлять не только строку объединения двух строчек, но и можно прибавлять, например, переменные других типов, например, целого типа. И в этом случае, компьютер его преобразует к его символьному представлению. К примеру, возьмем целое число ***i***1.

Введем знак равенства. Присвоим ему какое-либо значение. Например, 5, точка с запятой. И теперь возьмем класс ***String***, ***s***5, знак равенства. И пусть он является объединением строки ***s***2, прибавить ***i***1, точка с запятой. То же самое можно сделать и, используя вместо переменной, просто обыкновенные числа. Напишем таким образом. ***String***, ***s***6, знак равенства. Далее опять ***s***2, плюс 5. Вот эти два выражения на самом деле абсолютно идентичны. И в том и в другом случае, мы получим одно и то же. Теперь к 5 добавим еще какое-либо число. Например, 2, точка с запятой.

Правда, к выражению такого типа надо быть осторожным. Потому что, если мы ниже напишем такую строку, как ***String s***7, знак равенства, ***s***2, прибавить, в скобках 5+2, то результат выполнения вот этого выражения будет уже совсем другим. Будет не точка ***s***2, т.е. ***abcd***, 5, 2, а строчка ***abcd***, и 7, как результат суммы 5+2. Посмотрим, как все это будет выглядеть. Для этого распечатаем значение этих переменных, которые у нас есть. Для этого воспользуемся функцией ***println*** и скопируем ее в буфер обмена. Сору, далее правая кнопка мыши, ***Paste***.

Выведем сначала ***s***4, закроем скобку, точка с запятой. Правая кнопка мыши, ***Paste***. Далее ***s***5, закроем скобку, точка с запятой. Правая кнопка мыши ***Paste***, ***s***6. И выведем последнее значение ***s***7. Закроем скобку, точка с запятой. Теперь

попробуем все это скомпилировать. Для этого ***Tools***, ***Compile Java***, далее ***Run Java Application***, запустим на выполнение. И вот можно видеть, что у нас получилось. ***S***4 у нас равняется ***s***1+***s***2, т.е. 12345, а затем ***abcd***, что и выведено в этой строчке. Причем можно видеть, что между цифрами и буквами ***abcd***, нет никакого пробела, они идут подряд. Затем, у нас есть сумма ***s***2 и числа 5, что будет обозначать ***abcd*** и далее 5. В следующей строке к 5 еще добавлено 2, что отражается на нашем экране, а затем, в последней строке, сначала выполняется операция 5+2, получается 7, и далее выводится все это на экран.

Закроем теперь консольное окно и вернемся в режим редактирования. Из любой строки можно выделить и подстроку. Для этого, в классе ***String*** существует специальный метод ***substring***. Посмотрим, как это делается. Для этого возьмем ***String***, переменную ***s***8, которая будет равняться следующей строчке. Кавычки и далее напишем ***My First String***. Закроем кавычки, запятая. И далее возьмем переменную ***s***9, которая будет равняться какой-либо части из строки ***s***8. Для этого возьмем таким образом.

S8, далее точка и метод ***substring***. В скобках возьмем числа 3 и 7, закроем скобку, точка с запятой. Что означают числа 3 и 7? Тройка означает номер позиции в строке ***s***8, начиная с которой мы и хотим взять букву из этой строки. Причем, надо сказать, что нумерация начинается с 0, т.е. буква M это нулевой символ, ***Y***- первый, пробел будет второй, а буква ***F*** будет третим символом. Число 7 означает номер символа, до которого мы и хотим взять символы из этой строки. Если попробовать вывести это все на консоль, для этого опять правая кнопка мыши, ***Paste***, далее выведем сначала ***s***8, чтобы у нас было с чем сравнивать, а затем опять правая кнопка мыши, ***Paste*** и выведем на экран ***s***9. Закроем скобку, точка с запятой.

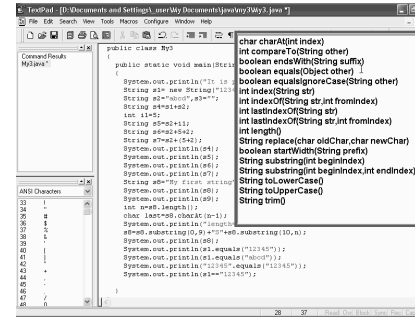
Теперь попробуем все это выполнить. Для этого ***Tools***, ***Compile Java***, далее ***Run Java Application***, и можно видеть результат выполнение этой программы. Сначала мы выводим ***My first string*** на экран, а затем символы с 3 по 7. Т.е. получается ***f***, ***i***, ***r***, и ***s***. Как мы и сказали, буква ***f*** у нас стоит на третьей позиции, а буква ***t***, в слове ***first***, стоит на 7 позиции. Поэтому, копируются и используются все буквы до буквы ***t***, до 7 позиции. И получается то, что мы видим на экране. Закроем это окно. Отметим теперь, что для определения длины строки в языке ***Java*** существует специальный метод ***length***, длина. Попробуем определить длину строки.

Для этого возьмем переменную типа ***integer***. Пусть длина строки будет ***n***, знак равенства, определим длину строки ***s***8, строки ***My first string***. Напишем ***s***8, точка, ее метод ***length***, затем скобки, точка с запятой. Вот теперь, в переменной ***n*** мы поместили длину строки. Для того чтобы выделить отдельный символ, в языке ***Java*** существует тоже специальный метод ***charAt***. Если мы возьмем таким образом, ***char***, далее ***last***, последний символ, знак

равенства. Выделим последний символ в этой строке ***s***8. Для этого ***s***8, точка, ***charAt***. В скобках укажем номер позиции этого символа. Поскольку мы хотим последний, для этого мы можем воспользоваться вот этой функцией ***length***, показать номер ***n***. Напишем ***n***, поскольку, как мы помним, нумерация в строке у нас начинается с 0, поэтому последним символом будет ни ***n***, а ***n***-1. Закроем скобку, точка с запятой. Попробуем все это вывести на экран. Для этого, опять воспользуемся правой кнопкой мыши, далее ***Paste***. Выведем таким образом, ***length***, длина, знак равенства, плюс. Напишем значение ***n***, плюс и далее выведем последнюю букву ***lastchar***, знак равенства, закроем кавычки, плюс и вот этот символ ***last***. Закроем скобки, точка с запятой. Посмотрим, что у нас получилось.

Для этого скомпилируем. ***Tools***, ***Compile Java***. Далее ***Run Java Application***. Теперь надо запустить программу.И, как можно увидеть, мы получили то, что ожидали. Длина нашей строки ***My first string*** равна 15 буквам, включая пробелы и последний знак, буква ***g***, что в принципе так и есть. Закроем окно и вернемся обратно в наш текстовый редактор.

6.7.3.10. Строки. Изменение, сравнение, работа с ними



После того, как мы выяснили возможность выбора отдельного элемента из нашей строки, возникает вопрос, а нельзя ли этот элемент изменить? Оказывается в языке ***Java*** это сделать нельзя. В принципе это противоречит тем навыкам, которые вырабатываются при работе с другими языками программирования. Например, с таким, как ***C***, где любая строка- это объединение отдельных ее символов. И поэтому, замена любой из ее букв не вызывает никаких и затруднений. В языке же ***Java***, это происходит совсем по-другому. Например, что бы поменять какую-либо букву в строке, можно поступить следующим образом.

Разбить ее на несколько частей, заменив ту часть, которую мы хотим заменить на какую-либо другую, затем все это сложить и присвоить к той же самой переменной. Конечно, это достаточно сложный путь, но, тем не менее, это так. Это связано с тем, что такие неизменяемые строки имеют большие преимущества, компилятор может делать строки совместно используемыми, и при этом происходит работа со строками, выигрыш во времени работы с ними, и в эффективности. Попробуем в строчке ***My first string*** заменить маленькую

букву ***s***, на ***S*** большую. Для этого можно написать таким образом, напишем переменную ***s***8, далее знак равенства, опять ***s***8 и ее подстроку ***substring***, которая начинается с нулевого символа и кончается девятым символом. 9-й символ является пробелом в данном случае.

Закроем скобку. Плюс. Теперь напишем в кавычках вместо этой маленькой буквы ***s***, которая здесь стоит, большую букву ***S***. Затем плюс, и опять ***s***8 с подстрокой ***substring***, начиная от следующей 10-й позиции и уже до конца, до ***n***. ***N***, как мы помним, равняется длине этой строки. Вот это место, где мы определили ***n***, как длину строки ***s***8. Точка с запятой. Теперь попробуем все это вывести на наш экран. Для этого опять правая кнопка мыши, ***Paste***, как мы помним, метод ***println*** у нас находится в памяти все время, и далее выведем ***s***8 на экран. Закроем скобку, точка с запятой.

А теперь скомпилируем, ***Tools*** и ***Compile Java***, далее ***Tools***, ***Run Java Application***, как мы видим, мы получили ***My first String*** уже с большой буквой ***S***, ***String***, т.е. в языке ***Java*** мы поменяли одну букву в строковой переменной. Закроем это окно. Для того чтобы проверить, совпадают ли две строки, можно использовать метод ***equals***. Можно написать таким образом. Как мы помним, строка ***s***1 равняется 1, 2, 3, 4, 5, поэтому выведем на наше консольное окно уравнение, в котором указано, равняется ли оно какой-либо строке. Для этого, правая кнопка мыши, ***Paste***.

Вот теперь мы получили ***println***, и далее напишем таким образом. ***S***1, точка. Далее метод ***equals***, скобка. И строка для сравнения. Напишем ту же самую строчку 1, 2, 3, 4, 5. Закроем кавычки, скобки, и точка с запятой. Теперь опять правая кнопка мыши, ***Paste***. И на этот раз напишем несколько по-другому. Сравним ***s***1 ***equals*** с какой-либо другой строкой. Например, ***A***, ***B***, ***C***, ***D***. Закроем кавычки, скобки. Точка с запятой. А теперь, посмотрим, что у нас получилось. Для этого ***Tools***, ***Compile Java***, далее ***Tools***, ***Run Java Application***, и как мы видим, выполнение первого из этих операторов, вызывает результат ***True***, поскольку ***s***1 у нас как раз и равняется 1, 2, 3, 4, 5.

А выполнение следующей за ним строки вызывает ответ ***False***. Поскольку ***s***1 не равняется ***A***, ***B***, ***C***, ***D***.Закроем теперь наше консольное окно и отметим, что-то же самое можно написать несколько по-другому. Щелкнем на правую кнопку мыши, ***Paste***, мы получили ***println***. Затем в кавычках напишем опять какую-либо строку. Например, ту же самую 1, 2, 3, 4, 5, кавычки, точка. Используем эту строку как раз в качестве переменной строкового класса. Далее ***equals***, откроем скобки, и опять 1, 2, 3, 4, 5, закроем кавычки, скобки, точка с запятой. Т.е. вот в этой строчке 1, 2, 3, 4, 5 мы сравниваем опять-таки с 1, 2, 3, 4, 5. Результат, конечно же, должен получаться такой. Просто в этой строке строку мы использовали в качестве переменной, в качестве экземпляра класса ***String***.

Теперь же отметим еще одно обстоятельство, использование вот такого значка двойного равенства для сравнения двух строчек, на самом деле не правомочно. Поскольку, написать такое можно, это не воспримется в качестве синтаксической ошибки, но результат может быть совершенно непредсказуемым. Потому что при сравнении, компьютер сравнивает на самом деле не две строки, а две ссылки. Приведем пример такого порядка. Как мы помним, результат сравнения ***s***1 и 1, 2, 3, 4, 5 у нас был ***True***, поскольку ***s***1 равняется этой строке. Напишем это немножко другим образом. Для этого правая кнопка мыши, ***Paste***, ***s***1 и на этот раз воспользуемся знаком сравнения равенства. И далее в кавычках 1, 2, 3, 4, 5.

Закроем скобку, точка с запятой. Так вот, результат выполнения вот этого оператора, будет ***False***, поскольку ***s***1 и как раз равен этой строке, но на самом деле, как ссылка, это два совершенно различных участка памяти. Посмотрим, так ли это. Для этого ***Tools***, ***Compile Java***. Теперь ***Tools***, ***Run Java Application***, и как мы видим, результат выполнения нашего последнего оператора будет ***False***. ***S***1 и 1, 2, 3, 4, 5, это совершенно различные вещи, хотя, как строчки они одинаковые. Но и предыдущий оператор выдал нам результат ***True***. Результат сравнения 1, 2, 3, 4, 5 самим с собой, как с оператором, выдал нам результат ***True***. Закроем данное окно.

Из всего этого можно сделать следующий вывод, пользоваться этим оператором для сравнения двух строчек совершенно нежелательно. Поскольку мы можем получить совершенно непредсказуемый результат. Если это один и тот же ресурс, мы будем получать равенство, а если — разные, то будем получать значение ***False***. Хотя, как строчки, это может быть абсолютно одинаковые выражения. Рассмотрим теперь основные методы, которые есть у класса ***String*** в языке ***Java***.

Они выведены вот в эту сводную таблицу, которую мы можем видеть на экране. Первый из них это метод ***charAt***, с которым мы уже сталкивались и который возвращает символ, расположенный в указанной позиции, в позиции индекс. Следующий метод, метод ***compareTo***, позволяет сравнивать две строки и возвращает отрицательное значение, если строка предшествует строке ***other*** в лексикографическом порядке, и положительное значение, если эта строка ***other*** предшествует ей. А также выдает значение 0, если две строки совпадают.

Этот метод очень полезен, если мы хотим расположить строчки в алфавитном порядке. Следующий метод, метод ***endsWith***, позволяет определить оканчивается ли данная строка под строкой ***suffix***. Если это верно, то возвращается значение ***True***, а в противном случае, значение ***False***. А метод ***equals***, с которым мы уже знакомы, сравнивает две строки. Если они равны друг другу, то выдается значение ***True***, а если они не равны, не совпадают, то

выдается ответ ***False***. Следующий метод, метод ***equalsIgnoreCase*** сравнивает две строки без учета регистра, без учета больших и маленьких букв.

Если они с этими условиями не учета регистра равны, то тогда выдается ***True***, а в противном случае — ***False***. Метод ***index***, так же, как и метод индекс ***Of***, помогает найти подстроку ***str*** в исходной строке. И возвращает начало первой подстроки, совпадающей с этой строкой, начиная с позиции 0 в первом случае, или, начиная с позиции ***fromIndex*** во втором случае.

Следующие два метода очень похожи на то, что мы только что обсуждали, они возвращают начало последней подстроки, равной этой строке ***str***. Начиная с позиции 0 или с позиции ***fromIndex*** во втором варианте. Метод ***length*** возвращает длину нашей строки. А метод ***replace*** возвращает в новую строку, которая получается заменой всех символом ***oldChar*** на символы ***newChar***. Метод ***startWidth*** позволяет определить, начинается ли наша строка подстрокой ***prefix***.

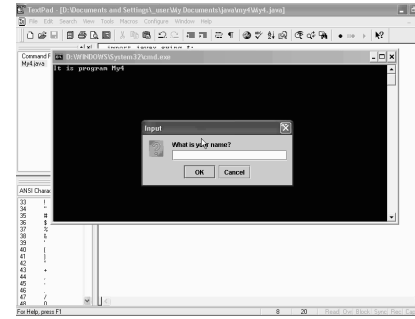
Если это верно, возвращается ***True***, в противном случае, возвращается в ***False***. Следующие два метода, которые называются одинаково, это методы ***substring***. У них различное количество передаваемых параметров. Возвращают в новую строку, которая состоит из всех символов, начиная с ***beginIndex***, и заканчивая концом строки в первом случае, или же заканчивая позицией ***endIndex***, не включая при этом именно эту позицию во втором случае.

Следующие два метода ***toLowercase*** и ***toUpperCase***, превращают нашу строку в новую строку, преобразовывая при этом прописные буквы в строчные в первом варианте. И наоборот, все строчные буквы в прописные во втором варианте. Последний, выписанный здесь метод ***trim***, который возвращает новую строку, из которой исключены все предшествующие и последующие пробелы.

6.7.3.11. Ввод данных с диалоговым окном и операции с ними

Рассмотрим, какие средства есть в языке ***Java***, для возможности вызова диалогового окна, для ввода каких-либо данных с клавиатуры и для их печати. Одним из этих выводом мы уже не раз пользовались. Это метод ***println***, при помощи которого можно выводить текст на консольное окно. Начнем работу с простейшего шаблона приложения, в котором создается консольное окно, и выводится вот такая строчка ***It is program My***4. Сохраним его. Для этого щелкнем на кнопку ***Save***, создадим для него новую папку с названием ***My***4. И зададим имя для нашего файла. Как мы помним, он должен в точности повторять название нашего класса, включая, маленькие и большие буквы. Поэтому назовем так, ***My***4.

Щелкнем на кнопку сохранения. И после этого мы вернулись на наш текстовый редактор. Попробуем теперь сделать так, чтобы мы могли ввести в некое диалоговое окно какие-либо данные. Для этого мы можем воспользоваться следующим методом ***JOptionPane***. Затем точка и вызовим его метод ***showInputDialog***. Далее скобка. Теперь нам нужно написать ту строчку, которая будет выводится на это диалоговое окно. Введем самый стандартный вопрос ***What is your name***, далее поставим знак вопроса, закроем кавычки и скобку. Точка с запятой. Вот мы ввели тот оператор, при помощи которого можно вызвать диалоговое окно. Ну, если мы в таком виде запустим наше приложение на компиляцию выполнения, то произойдет ошибка.



Дело в том, что для того чтобы мы могли использовать вот этот оператор, нам нужно использовать дополнительный пакет ***javax.swing***. Для того чтобы мы могли это сделать, нам нужно в самом начале нашего приложения, написать следующую строчку. Освободим строку и далее напишем ключевое слово ***import***. И далее пакет ***javax.swing***. Именно в этом пакете и определен класс ***JOptionPane***, у которого мы и хотим позаимствовать метод вызова диалогового окна. Теперь в конце нашей программы, напишем еще строчку ***System***. ***Exit***, и в скобках зададим код выхода нашей программы. Зададим 0, поскольку у нас теоретически все должно закончится нормально.

Теперь же попробуем скомпилировать. Для этого щелкнем на пункте меню ***Tools***, далее ***Compile Java***, теперь ***Tools***, ***Run Java Application***, и можно видеть, что у нас получилось. Наряду с этим консольным окном, которое у нас все время появлялось, появилось окошко диалогового типа, которое в принципе независимо от консольного окна. Его можно передвигать в разные стороны, а также можно, конечно же, ввести какой-либо текст.

Например, вписать имя ***John*** и ОК. И, как мы видим, наша программа после этого прекратила свою работу. Правда, на консольное окно больше ничего выведено не было. Закроем теперь наше приложение и попробуем вывести на окно результат ввода. Для этого нам достаточно использовать вот этот метод ***println***, а также задать какую-либо переменную, внутрь которой поместим данные из нашего диалогового окна. Зададим теперь какую-либо переменную строкового типа. Для этого наберем ***String***. Пусть эта переменная у нас будет ***name***, поскольку мы вводим имя. Далее знак равенства и после этого вот этот оператор ***showInputDialog***.

Щелкнем на кнопку ***Delete***, чтобы объединить вот эти две строки. Вот теперь после выполнения нашего диалогового окна поместим в переменную ***name*** наш введенный текст. Теперь же попробуем его вывести на консольное окно. Для этого воспользуемся оператором ***println***. Выделим вот этот текст. Правая кнопка мыши, Copy. Теперь встанем сюда, правая кнопка, ***Paste***. И теперь выведем ***name***. Напишем таким образом. ***Hello***. Далее +***name***. Закроем скобку, точка с запятой. Теперь попробуем опять скомпилировать. Для этого ***Tools***, ***Compile Java***, ***Tools***, ***Run Java Application***. Вот опять появилось консольное окно вместе с диалоговым. Напишем наше имя ***John***, и ОК. Как мы видим, появилась надпись и на нашем консольном окне.

Закроем данное приложение. Теперь попробуем при помощи нашей программы выполнить что-либо более существенное. Например, ввести два числа и затем их перемножить. Для этого опять воспользуемся вот этим оператором, который вызывает диалоговое окно. Проще всего его скопировать. Правая кнопка мыши, Copy. Теперь вставим сюда. Правая кнопка, ***Paste***. Вместо ***name***, пусть это будет ***i***1. Изменим текст. Напишем ***Number one***. Опять скопируем эту строку.

Щелкнем правой кнопкой мыши, Copy. Далее вставим сюда. Правая кнопка, ***Paste***. Пусть это будет теперь два и ***Number two***. Теперь надо преобразовать строковую переменную ***i***1, ***i***2 в их числовые представления. Поэтому напишем таким образом ***int i***1 знак равенства, воспользуемся специальным оператором, который предназначен для этого и который называется ***Integer.parseInt***. Далее точка. И его метод ***parseInt***, который и производит такое преобразование. Далее наша строковая переменная ***i***1, закроем скобку, точка с запятой. Теперь скопируем эту строчку еще раз. Правая кнопка мыши, Copy. Встанем сюда. Правая кнопка, ***Paste***.

Переделаем единицы на двойки. Далее попробуем все это вывести на наше консольное окно. Для этого опять воспользуемся методом ***println***. Выделим. Правая кнопка, Copy. Встанем сюда. Правая кнопка, ***Paste***. Теперь напишем то, что мы хотели вычислить. Для этого напишем таким образом. Сначала выведем первое число ***i***1, затем знак умножения. Второе число ***i***2, знак равенства и их произведение. ***ni***1 умножить на ***ni***2, закроем скобку, точка с запятой. Попробуем теперь скомпилировать наше приложение и посмотреть, что у нас получилось. Для этого развернем меню ***Tools***, далее ***Compile Java***.

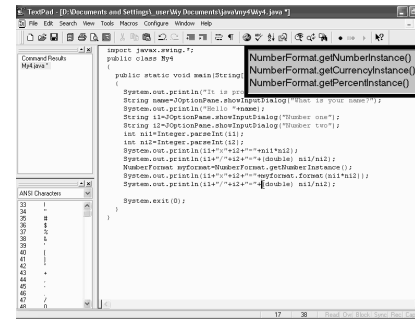
Опять ***Tools*** и ***Run Java Application***. Вот появилось наше консольное окно вместе с окном диалога. Введем какое-либо имя. Пусть будет то же самое, ОК. После того, как здесь написалось приветствие, появилось опять диалоговое окно, которое у нас спрашивает первое число. Возьмем любые числа и ***OK***. Появилось опять диалоговое окно. Теперь нам нужно ввести второе число, ОК. И можно видеть результат выполнения этого произведения. Первое число,

умноженное на второе равно вот этому числу. Как мы видим, какие-то простейшие действия нам удалось выполнить при помощи нашей программы. Закроем теперь это приложение и вернемся в наш обычный режим редактирования.

6.7.3.12. Форматирование данных для отображения на консольном окне

Попробуем теперь в качестве результата вывести не только произведение этих двух чисел, но и их деление друг на друга. Для этого напишем таким образом. Скопировать эту строчку, для того чтобы не писать ее во второй раз. Выделим, правая кнопка мыши, Copy. Далее встанем сюда. Правая кнопка мыши, ***Paste***. И теперь, вместо знака умножения, введем знак деления. А также введем знак деления между двумя переменными ***i***1 и ***i***2. Но, если мы выполним нашу программу в таком виде, то мы получим целочисленное деление первого числа на второе.

Для того чтобы у нас произошло деление с получением точного результата, нам нужно написать впереди преобразование типа к ***Double***. Для этого в скобках напишем ***Double***. Закроем скобку и попробуем скомпилировать. Развернем меню ***Tools***, ***Compile Java***. Опять ***Tools*** и ***Run Java Application***. От нас требуют опять ввести какое-либо имя. Пусть, например, будет имя Билл, OK. Теперь надо ввести какое-либо число. Напишем любое число, OK. Второе число. Пусть будет 38, OK.



И можно видеть результат выполнения этих двух операции. В результате произведения мы получили одно число, а в результате деления — другое. Надо сказать, что при этом, в результате деления, мы получаем все цифры нашего числа, что иногда бывает абсолютно не нужным. Например, если мы хотим вывести только определенное количество цифр, после запятой. Для этого нужно воспользоваться не просто выводом на консольное окно, а некими свойствами форматирования. Закроем теперь наше консольное окно и попробуем реализовать форматированный вывод. Для того чтобы управлять форматом вывода данных, в языке ***Java*** есть три специальных метода, которые предоставляют стандартные средства форматирования. Для чисел, это первый метод ***getNumberInstance***, который находится в классе ***NumberFormat***. Следующий метод, метод ***getCurrencyInstance***, позволяющий форматировать

денежные суммы. И последний метод ***getPercentInstance***, который служит для форматирования процентов.

Попробуем применить это на практике. Для этого напишем следующую строку. Первое, что нам нужно сделать, это получить объект типа ***NumberFormat***. Поэтому напишем таким образом, ***NumberFormat***. Пусть этот объект называется ***myformat***, знак равенства. А теперь применим первый из этих методов ***getNumberInstance***, поскольку мы хотим форматировать числа. ***NumberFormat***, точка. Теперь ***getNumberInstance***. Далее скобки и точка с запятой. И теперь это средство форматирования нам нужно применить для вывода наших чисел. Для этого скопируем вот эти две строки. Выделим. Теперь правая кнопка мыши, Copy. Далее вставим сюда. Правая кнопка мыши, ***Paste***. И теперь переделаем эти выводы. Напишем таким образом. ***Myformat***, точка. И воспользуемся его методом ***format***. Далее скобки. В конце нам нужно добавить еще одну скобку. Вот теперь форматированный вывод первой из этих чисел, нами обеспечен. Теперь то же самое сделаем во втором случае, при выводе делении одного числа на другое.

Для этого выделим вот этот кусок текста. Чтобы не писать еще раз, скопируем, правая кнопка, Copy, и вставим, правая кнопка и ***Paste***. Нам, конечно же, нужно ввести еще одну закрывающуюся скобку. А теперь попробуем скомпилировать и запустить нашу программу. Но, прежде чем мы запустим нашу программу на компиляцию, нам нужно написать еще одну строчку импорта. Для того чтобы воспользоваться классом ***NumberFormat*** и его методами, нам нужно импортировать пакет ***java.text***. Поэтому щелкнем здесь на ***Enter*** и напишем слово ***import***. Далее ***java***, точка, ***text***, и точка с запятой. Вот теперь попробуем скомпилировать нашу программу. Для этого щелкнем на кнопку ***Tools***. Далее ***Compile Java***.

Опять ***Tools*** и запустим нашу программу ***Run Java Application***. Введем опять какое-либо имя. Например, пусть будет Роберт, ОК. Введем первое число. Пусть это будет 1, 2, 3, 4, 5, 6, ОК. Теперь второе число. Например, возьмем 7, ОК. И теперь можно видеть, что у нас получилось. В результате форматированного вывода, как мы видим, большое число было разделено по три цифры, и в качестве разделителя взята запятая. А в качестве разделителя между дробной и основной частью, взята точка. Само число тоже разбито по три цифры. Кроме этого, для вывода действительных чисел, взяты три цифры после запятой. Закроем теперь наше приложение. Щелкнем на вот этом крестике, и мы опять попали в наш текстовый редактор. Если же мы хотим изменить эти установки по умолчанию, для того чтобы изменить количество отображаемых цифр после запятой, мы можем воспользоваться специальным методом ***setMaximumFractionDigits***.

Воспользуемся этим методом для нашего формата. Теперь ***setMaximumFractionDigits***. Далее скобка, и теперь нам нужно указать количество цифр, которые мы хотим взять. Возьмем всего одну цифру после запятой. Закроем скобки, точка с запятой. Кроме этого, мы можем указать и ***MinimumFractionDigits***, если мы хотим, чтобы количество цифр после запятой было не меньше, чем какое-то другое число. В том случае, если цифр меньше, они будут заполняться нулями.

А также мы можем указать общее количество цифр для отображения перед десятичной точкой в форматируемом объекте. Для этого надо воспользоваться методом ***setMaximumIntegerDigits***. Но с таким методом нужно пользоваться с осторожностью, поскольку, если число превысит это значение, и значащих цифр намного больше, то оно усечется и мы получим совершенно неправильный ответ. Попробуем сейчас запустить наше приложение, и посмотреть, что у нас получилось. Для этого ***Tools***, далее ***Compile Java***. Теперь ***Tools*** и ***Run Java Application***. Введем какое-либо имя. Например, пусть будет ***Stiv***, ОК. Введем какое-либо число, ОК. Введем второе число, ОК.

И можно видеть, что на этот раз у нас после запятой только одна цифра для вывода действительного числа, как мы и задумали. Все остальное осталось точно таким же. Единственное, надо сказать что, как мы видим, последняя цифра, которая после точки, цифра 8, получается округлением нашего числа. Поскольку на самом деле у нас одна часть начинается с 7, затем 9, и т.д. И округляя, получим цифру 8. Закроем теперь наше приложение. Щелкнем на вот этом крестике и вернемся в наш обычный текстовый редактор.

6.7.4. Основные структуры языка (продолжение)

6.7.4.1. Оператор if. Синтаксис

Рассмотрим теперь, как в языке ***Java***, реализованы операторы, которые управляют ходом выполнения программы. Для этих целей, так же, как и в любом другом языке программирования, в языке ***Java*** есть условные операторы, операторы цикла, оператор переключения ***switch***, а также оператор так сказать непосредственного перехода. Рассмотрим, как они действуют. Для этого рассмотрим опять-таки стандартный шаблон приложения, при помощи которого наша программа может выводить простейший текст на наше консольное окно. Вот этот шаблон. Пусть он будет класса ***My***5, сохраним его на жестком диске. Для этого щелкнем на кнопке сохранения.

Создадим новую папку ***my***5. Щелкнем на кнопке Открыть. И зададим имя для нашего приложения. Как мы помним, это имя должно в точности повторять имя нашего класса ***My***5. Щелкнем на кнопке Сохранить. И теперь рассмотрим вопросы видимости переменных внутри блоков. Для того чтобы

разобраться в этом вопросе, определим какую-либо переменную. Пусть это будет переменная целого типа ***integer***, ***i*** равняется, допустим, 5. Точка с запятой. Теперь откроем какой-либо новый блок при помощи фигурных скобок и внутри него определим другую целую переменную ***int***, например, ***k***. Далее определим еще одну переменную ***s***, которая будет равняться ***i***+***k***.

А за пределами этого блока, определим опять-таки целую переменную, которая будет называться, например, ***g*** и будет равняться ***i***+2. Но, конечно же, предварительно, нам нужно было бы определить какое-либо значение ***k***. Пусть оно равняется 4. И в таком виде наша программа синтаксически правильно записана. Т.е. внутри вот этого блока от скобки до скобки, мы можем пользоваться всеми теми переменными, которые уже определены внутри этого блока. Вот, в частности, переменной ***k*** для выполнения неких функций в следующей строке.

А также всеми теми переменными, которые определены вне этого блока, ранее. Ну, в частности можно пользоваться переменной ***i***. И в этой строчке мы написали, что ***s*** будет равняться сумме ***i***+***k***. ***k*** определен внутри этого блока, а ***i*** вне этого блока, раньше. А вот в этой строчке, мы определили новую переменную ***g***, которая определилась на основе переменной ***i***, которая определена ранее в том же блоке и затем число 2. Переменную ***k*** здесь мы использовать не могли. Это было бы ошибкой.

Если мы здесь напишем вот таким образом, ***i***+2+, например, ***k***, то компилятор нам в этом случае выдаст ошибку. Если мы развернем меню ***Tools*** и выберем команду ***Compile Java***, то, как мы видим, компилятор это не принял. На символе ***k*** он показал ошибку, что в принципе и ожидалось. Но вторая ошибка является на самом деле отголоском первой. Просто-напросто переменная ***k*** у нас не определена.

Перейдем опять на нашу программу. Сотрем здесь переменную ***k***, которую мы использовали неправомочно и покажем еще одну ошибку, которую можно допустить. Ну, если мы определим наряду с переменной ***k*** внутри этого блока опять переменную ***i***. Напишем ***int I***, равняется, например, 7. Вот такая запись тоже является ошибкой. Т.е. переопределять переменную внутри блока, если она уже определена ранее, нельзя, в отличие от других языков программирования.

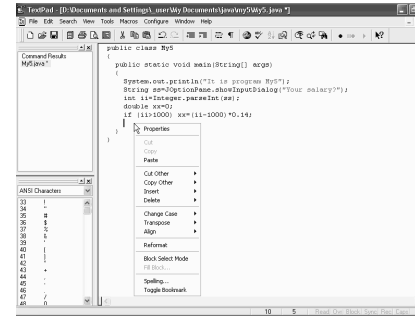
Например, в ***C*** это возможно. И при выполнении такой последовательности действий в языке ***C***, у нас переменной ***s*** присвоилось бы значение 4+7, а не 4+5. А в языке ***Java***, это просто-напросто ошибка. Если мы развернем меню ***Tools*** и выберем команду ***Compile Java***, то, как мы видим, компилятор опять выдал ошибку. Вот это определение переменной ***i***, ему явно не понравилось. Если мы чуть-чуть сдвинемся, то, как мы видим, здесь написано, что

переменная ***i*** уже определена ранее. Переключимся обратно на нашу программу и сотрем вот это неправильное выражение.

Рассмотрим теперь такой популярный оператор, как условный оператор ***if***. Сотрем сначала вот эти строчки, которые мы только что написали и напишем другую строчку. Введем переменную типа ***String***, пусть это будет ***ss***, далее равняется, и далее попробуем ввести некую информационную строку с диалогового окна. Для этого, как мы помним, надо воспользоваться методом ***showInputDialog*** из класса ***JOptionPane***. Теперь откроем скобку и теперь надо указать тот текст, который будет появляться в этом диалоговом окне. Откроем кавычки и далее напишем, например, ваша зарплата, ***Your salary***? Закроем кавычки, закроем скобку. Точка с запятой.

Теперь же переведем эту строковую переменную в число. Для этого воспользуемся переменной типа ***int***, которая пусть называется ***ii***. Теперь нам надо преобразовать переменную ***ss*** к типу ***int***, к типу ***ii***. И для этого воспользуемся следующей функцией ***integer***, точка и далее класс ***parseInt***. Это стандартная функция преобразования от строки к числу, и далее укажем переменную ***ss***. Закроем скобку. Точка с запятой. Теперь же рассчитаем, какой нужно заплатить налог с этой суммы. Предположим, что условия у нас на налоги такое, что если наша зарплата меньше 1000 рублей, то она налогами вообще не облагается, а если больше 1000, то с каждого рубля, который больше этой 1000, надо заплатить 14%.

В этом случае, как раз нам и понадобиться использовать условный оператор ***if***. Но сначала определим какую-либо переменную, в которой будем сохранять сумму налога. Пусть она будет у нас типа ***double***, действительное число, поскольку нам понадобиться оперировать с дробными частями. Пусть эта переменная называется ***xx*** и равна, для начала, 0. И теперь напишем оператор ***if***. Далее в скобках нам нужно написать условие. Условие у нас будет такое. Если ***ii*** больше, чем 1000, тогда нам надо высчитывать налог. И в этом случае ***xx*** будет равняться ***ii***-1000 умножить на коэффициент 0,14. 14 процентов, это как раз и есть 0,14.



Дальше попробуем все это вывести на консольное окно. Для этого скопируем вот этот оператор ***println***, щелкнем правой кнопкой мыши, Copy. Далее вставим сюда. Щелкнем правой кнопкой мыши и ***Paste***. И выведем переменную ***xx***. Напишем таким образом "***You will pay***" и далее ***xx***. Точка с запятой. А теперь нам еще нужно, для того чтобы мы

могли использовать вот это диалоговое окно ввода, произвести импорт соответствующего пакета, который напишем в начале нашей программы. Напишем ***Import***, далее ***javax***, поскольку именно в этом пакете и находится возможность диалогового окна, далее точка, ***swing***, и точка с запятой.

Теперь попробуем выполнить наше приложение. Для этого развернем меню ***Tools*** и выберем команду ***Compile Java*** (компилировать программу). Вновь развернем меню ***Tools*** и выберем команду ***Run Java Application***. Появилось вот это диалоговое окно, в которое введем, например, сумму 3000. Щелкнем на кнопке ОК. И как мы видим, с суммы 3000 надо заплатить 280 рублей. Теперь закроем это консольное окно. В принципе надо учесть, что мы забыли написать ***System.exit*** в нашей программе, что необходимо при использовании диалогового окна. И в результате при выходе из диалогового режима, на конец программы сразу же не попали. Сейчас закроем консольное окно. В конце нашей программы напишем ***System.exit***. Далее код выхода. В принципе можно воспользоваться, конечно же, нулем, если у нас все завершилось нормально. По умолчанию обычно берется код выхода 0, в случае нормального завершения работы, и ненулевое значение, будет означать код ошибки.

*6.7.4.2. Оператор **if**. Составной оператор. Вложение операторов **if***

Теперь предположим, что текст, который мы вводим на консольное окно, хотим сделать разным, в зависимости от того, сколько мы платим. Платим ли какой-то налог или вообще ничего не платим. Тогда, нам уже надо воспользоваться не просто оператором, а составным оператором, который нужно поместить внутри фигурных скобок. Напишем теперь таким образом. Откроем фигурные скобки. И теперь скопируем оператор ***println***. Щелкнем правой кнопкой мыши, Copy. Вставим сюда. Вновь щелкнем правой кнопкой мыши, ***Paste***.

И теперь напишем таким образом ***You mast pay***. Далее кавычки, и сумма ***xx***. Закрыть скобку. Точка с запятой. Теперь же, если у нас сумма меньше 1000, тогда напишем оператор ***else***, и далее следующую строку, вот в этом случае, который, конечно же, будет с несколько другим текстом. Текст будет такой ***You mast pay nothing***! Ну и, конечно же, ***xx*** нам уже писать не нужно. Добавим слово ***not***. Запустим наше приложение и посмотрим, что у нас получилось. Развернем меню ***Tools***, далее ***Compile Java***, далее в меню ***Tools*** выберем команду ***Run Java Application***. Введем опять какое-либо число.

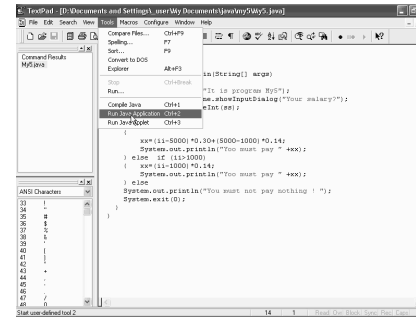
Например, 4000, ОК. И как мы видим, в этом случае нам нужно заплатить 420 рублей. Правда, здесь есть еще довольно большой остаток, который определяется ошибками округления. Сейчас закроем наше приложение. Попробуем его запустить и ввести на этот раз число, которое будет меньше

1000. Для этого развернем меню ***Tools***, компилировать нам уже не обязательно и поэтому прямо выберем команду ***Run Java Application***.

Введем число, например, 500, ОК. И как мы видим, нам ничего платить не надо. Закроем приложение. Чуть-чуть еще более усложним нашу задачу, и пускай с суммы, которая больше, например, 5000 рублей нам нужно платить по другой ставке налога. Например, 30% от всех денег, которые больше этих 5000. Поэтому нам опять надо воспользоваться оператором ***if*** и на этот раз их нам придется так сказать вкладывать друг в друга. Использовать повторяющиеся операторы ***if***. Поэтому сначала выполним проверку на то, будет ли наше число ***xx*** больше 5000, и в этом случае нам нужно будет выполнить какие-либо действия, которые поместим в фигурных скобках.

Если же это число меньше или равно 5000, напишем ***else***, далее пусть выполняется все то, что у нас было ранее. Т.е., если оно больше 1000, но понятно, что меньше 5000, выполняется вот этот блок, а в случае, если это число меньше 1000, то просто-напросто будет выводиться одна, вот эта надпись. Теперь напишем формулу вычисления вот в этом случае. Она у нас будет выглядеть следующим образом. ***xx*** = сумма ***ii***-5000, из которой надо заплатить 30%, 0,30. Плюс еще та сумма, которая платится с денег, которые меньше 5000. Она будет высчитываться по следующей формуле. 5000-1000, т.е. это минимальная сумма, которая освобождается от налога, умножить на предыдущую вставку 0,14, Точка с запятой.

Теперь нам эту сумму надо вывести на консольное окно. Для этого просто-напросто скопируем вот эту строчку. Выделим ее. Щелкнем правой кнопкой мыши, Copy. Далее встанем сюда. Правая кнопка мыши и ***Paste***. Скомпилируем нашу программу, используя команду ***Compile Java***. Теперь же запустим эту программу, используя соответствующую команду. Вот опять появилось это вопросительное окно, в которое введем числа, маленькие мы уже вводили, и теперь введем большое число. Например, 7000, которое явно больше 5000. Щелкнем на кнопке ОК и посмотрим, сколько нам надо заплатить. Оказывается, от 7000, нам надо платить 1160 рублей. Закроем наше консольное окно и вернемся в наш текстовый редактор. Попробуем теперь сделать так, чтобы все эти вычисления проводились не один, а несколько раз. Например, до тех пор, пока в качестве зарплаты мы не введем число 0. Для этого нам надо воспользоваться циклом ***do while***.



Введем здесь новую строку и далее напишем оператор ***do***. Теперь откроем фигурные скобки, внутри которых поместим все вычисления нашей программы. Далее будет закрывающая фигурная скобка. И затем то условие, при котором наш цикл будет выполняться.

Здесь напишем таким образом. ***while***, скобки и условия. Далее напишем ***ii*** больше 0, закроем скобку. Точка с запятой. А теперь, если мы в таком виде запустим нашу программу, при компиляции, компилятор выдаст нам ошибку. Дело в том, что у нас переменная ***ii*** является локальной, внутри вот этого блока ***do*** и соответственно вне этого блока мы ее использовать не можем. Поэтому нам просто-напросто надо вынести определение переменной ***ii*** за вот эти скобки. Поэтому определение ***integer*** здесь удалим. И снаружи нашего цикла проведем ее инициализацию, ***int ii***, точка с запятой. Теперь еще в начале наших вычислений, сразу после диалогового запроса, распечатаем саму нашу зарплату. Поэтому скопируем оператор ***println***, Copy. Вставим сюда, ***Paste***. Затем напишем "***Your salary***", плюс и переменная ***ss***. Закроем скобку. Точка с запятой. Теперь скомпилируем наше приложение.

Для этого развернем меню ***Tools***, ***Compile Java***. Запустим программу, активизировав команду ***Run Java Application*** и, как мы видим, возник тот же стандартный вопрос, который мы уже ни раз видели. Введем какую-либо зарплату. Например, 444. Щелкнем на кнопке OK. Как мы видим, в этом случае мы платить ничего не должны. А если наша зарплата составляет 4444, тогда нам надо платить 482 рубля с копейками.

В случае же, если зарплата еще больше 44444, тогда надо заплатить более 12000. А если же мы сейчас введем 0, щелкнем на кнопке OK, то, как мы видим, при зарплате 0, платить ничего не надо и мы вышли из нашей программы. Закроем теперь наше консольное приложение. И мы вернулись в наш текстовый редактор.

*6.7.4.3. Операторы цикла **do***

Рассмотрим теперь, как в языке ***Java*** реализована возможность, выполнять повторяющиеся операторы. Для этого в нем есть специальный оператор, оператор цикла. Но, начнем мы работу с этого простейшего шаблона, который выводит стандартную надпись на наше консольное окно, и сохраним его на нашем жестком диске. Для этого щелкнем на кнопку сохранения, откроем новую папку. Открыть. Переименуем.

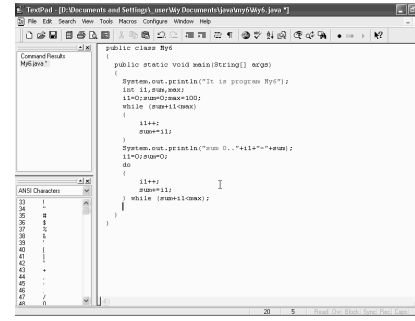
Как мы помним, файл должен называться так же, как и наш класс и щелкнем на кнопку Сохранить. В языке ***Java*** существует два варианта повторяющихся циклов, которые можно использовать в том случае, если мы точно не знаем, сколько повторений должно быть выполнено. Так называемые неопределенные циклы, в ходе выполнения которых, мы все время проверяем

какое-либо условие и лишь пока выполняется это условие, и будут выполняться операторы цикла.

Попробуем, например, выполнить какое-либо простейшее действие. Например, сложить первую последовательность натуральных чисел, до тех пор, пока эта сумма не превысить число 100. Для этого введем несколько переменных целого типа (***integer***) ***i*1, эта переменная, в которой мы будем держать последовательность натуральных чисел. Далее переменная ***sum***, в которой мы будет аккумулировать результат. Также переменная ***max***, до которой мы и хотим сложить наши натуральные числа. Теперь же организуем цикл. До этого проведем инициализацию наших переменных. Пусть ***i*1 равняется 0 с самого начала. Далее сумма тоже равняется 0.

А число ***max*** пусть равняется, например 100, точка с запятой. Теперь оператор ***File***, который является оператором цикла. Условием у нас будет, чтобы следующая сумма, а именно ***sum*+*i*1 должна быть меньше числа ***max***. Пока это условие будет выполняться, мы будем выполнять наши действия. Введем составной оператор.

Для этого напишем фигурные скобки, внутри которой напишем теперь следующее. Число ***i*1 увеличим на единицу. Для этого воспользуемся оператором ++, далее нашу сумму увеличим на ***i*1. Воспользуемся такой записью оператора. И теперь выведем на печать наш полученный результат. Для этого воспользуемся оператором ***Println***, скопируем его из вот этого места, правая кнопка мыши. Copy. Вставим сюда. Правая кнопка мыши и ***Paste***.



Теперь кавычки, ***sum***. Далее напишем, что мы проводим сумму от 0 до какого-то числа. В нашем случае, следующим числом, которое мы складываем, это будет ***i*1. И эта сумма должна равняться знак равенства в кавычках и далее плюс теперь результат суммы. Закрыть скобку. Точка с запятой. И теперь скомпилируем нашу программу и выполним для этого ***Tools***. Далее ***Compile Java***. Теперь ***Tools***, ***Run Java Application***. Можно увидеть результат выполнения нашей программы. Мы, оказывается, сложили числа от 0 до 13, и в результате получили сумму 91. Если же прибавить следующее число, оно будет 14, то, конечно же, вы уже получите число больше 100. Закроем теперь это окно и вернемся в наш редактор.

Ту же самую операцию можно проделать и с помощью другого варианта оператора цикла. Напишем, например, таким образом. Заново определим ***i***1 и ***sum***, ***i***1 равно 0, далее ***sum*** равно 0, и дальше напишем опять оператор цикла. Но на этот раз его напишем при помощи операторов ***do*** и ***while***. Далее выпишем, что мы хотим сделать. Сделать мы хотим то же самое, ***i***1 увеличим на единицу. Далее сумму увеличим на величину ***i***1. Точка с запятой. Теперь надо закрыть фигурные скобки. И теперь надо написать: ***while***. И внутри скобок условие, в течение которого и будет выполняться наш оператор цикла. Ну, опять напишем условие то же самое: ***sum*** + ***i***1 будет меньше ***max***. Закрыть скобки. Точка с запятой.

И далее выведем на печать то же самое выражение. Для этого скопируем эту строчку. Выделим. Правая кнопка мыши. Copy. Теперь встанем сюда. Правая кнопка мыши. ***Paste***. И попробуем скомпилировать и запустить наше приложение. Для этого пункт меню ***Tools***. Далее ***Compile Java***. Далее ***Tools***, ***Run Java Application***, и, как и нужно было ожидать, в случае первого цикла и в случае второго цикла, мы получили абсолютно идентичный результат. Закроем консольное окно.

Отметим то существенное отличие, которое есть между этими двумя операторами цикла. Как мы видим, в первом операторе цикла у нас условие стоит в самом начале цикла, а во втором случае, условие стоит в конце цикла. Из этого следует такой вывод, что во втором случае, этот цикл выполнится хотя бы один раз в любом случае, поскольку проверить это условие можно будет лишь при прохождении всего целого цикла хотя бы один раз. А в первом случае, мы сразу же проверяем условие и поэтому в случае, если оно неверное это условие, все эти операторы, которые находятся внутри фигурных скобок, будут пропускаться и будут выполнять следующий оператор. Т.е. различие между двумя операторами цикла проявляется в том случае, если у нас с самого начала условие неверно. Продемонстрируем это. Скопируем все эти строчки. Выделим. Правая кнопка мыши ***Copy***.

Теперь ***Enter***. Вставим. Правая кнопка мыши. ***Paste***. И единственное, что сделаем, изменим, назначение переменной ***max*** со 100 на (-100) и посмотрим, что в этом случае произойдет. Конечно же, в этом случае, условие неверно с самого начала. Посмотрим, что тогда мы получим. Для этого ***Tools***. Далее ***Compile Java***. Теперь запустим наше приложение, ***Tools***, ***Run Java Application***. И вот можно видеть, что, если в первом случае, когда мы складывали все суммы до 100, у нас результат получился абсолютно идентичный, и в первом и во втором случае.

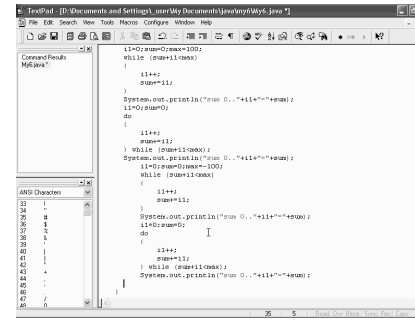
То, во втором случае, когда мы в качестве максимального числа взяли (-100) сразу же неверное условие, результат у нас получился при выполнении цикла так сказать первого вида и второго, два различных результата. Поскольку, в

первом случае цикл не выполнился ни разу, и сумма и переменная ***i***1 равняется 0, а во втором случае, несмотря на то, что условие у нас с самого начала неверно, один раз цикл тем не менее выполнился и поэтому результат получился и для переменной ***i***1 и для суммы 1. Закроем теперь это консольное приложение и вернемся в наш текстовый редактор.

6.7.4.4. Операторы цикла for

Еще один очень распространенный оператор цикла, это оператор ***for***, который широко применяется для выполнения таких повторений, количество которых можно проконтролировать счетчиком или какой-либо другой переменной, которая обновляется во время каждого цикла. Возьмем, например, и выполним тот же самый пример, который мы только что делали. А именно, сложить все натуральные числа от 0 до некоторого значения, до тех пор, пока эта сумма не будет превышать 100. Для этого проведем опять сначала инициализацию, переменной ***sum*** присвоим значение 0.

Переменная ***max***, пусть опять равна 100, и далее напишем оператор цикла ***for***. Теперь скобка, внутри которой нам нужно написать условие нашего оператора ***for***, который содержит в себе несколько действий. Первое среди них задает инициализацию, потом идет само условие цикла, и дальше условие обновление счетчика. После первого выражения напишем переменную ***i***1, которой присвоим значение 0. Затем второе выражение, в качестве которого опять-таки вот это 100+***i***1 меньше ***max***. И затем третье выражение, в качестве которого возьмем приращение счетчика ***i***1 на единицу. Закроем скобку.



Далее выполним некий оператор, в качестве которого возьмем просто-напросто приращение ***sum*** на ***i***1. Поэтому напишем ***sum*** += ***i***1. Конечно, если бы у нас была необходимость использование нескольких операторов, тогда надо было бы воспользоваться составным оператором. Т.е. поместить все эти операторы внутри фигурных скобок. Ну, поскольку сейчас здесь оператор всего лишь один, то фигурные скобки нам не нужны. Далее распечатаем, воспользуемся вот этой уже готовой строкой. Выделим ее. Правая кнопка мыши. Copy. Далее встанем сюда. Правая кнопка и ***Paste***.

Теперь запустим наше приложение и посмотрим, что у нас получилось. Выберем пункт ***Tools***. Далее ***Compile Java***. Теперь опять ***Tools***, ***Run Java Application***. Как мы видим, наряду с теми 4 строчками, которые вычисляли эти суммы и которые мы уже видели, появилась еще одна дополнительная строка.

Она вычисляет такую же сумму и получается тот же самый результат 91. Правда, счетчик у нас остановился не на 13, а на 14. Это уже связано с конкретной реализацией цикла ***for***. Просто-напросто превращение ***i***1 у нас произошло, а вот это превращение при переменной ***i***1 равной 14, еще не прибавилось. И именно на этом моменте и прекратилось выполнение оператора ***for*** Закроем эту консоль.

Теперь выполним этот цикл немножко по-другому. А именно, более стандартным образом: ***for***, далее определим переменную. Переменную счетчик цикла можно определить прямо внутри цикла ***for***. Для этого напишем ***int***. Ну, пусть переменная у нас будет ***i***2. Она пусть равняется от 0 и до тех пор, пока эта переменная ***i***2 меньше, например, 10. Точка с запятой.

И к ***i***2 применим приращение на единицу. Закрыть скобку. Теперь посчитаем сумму: ***sum*** +=, и ***i***2. Точка с запятой. Конечно же, предварительно нам нужно было бы задать в этой сумме значение 0, точка с запятой. И вы выведем теперь это значение ***sum*** на печать. Из-за этого опять воспользуемся вот этой строкой, которая у нас уже находится в буфере обмена. Для этого правая кнопка мыши и ***Paste***. Теперь ее немножко переделаем, ***i***1 у нас вообще нет в цикле. Не используется. А вот переменную ***i***2, здесь мы использовать, как раз не имеем право, поскольку она определена внутри этого цикла. Ее можно использовать только внутри вот этого блока ***for***. Но мы знаем в принципе, что она меняется от 0 до 10, поэтому напишем вот так: 0 и 10.

Все это можно убрать. Но, конечно, если у нас сумма от 0 до 10, тогда в этом случае тут лучше взять не знак меньше, а знак меньше или равно, чтобы число 10 тоже вошло у нас в эту же сумму. Теперь возьмем ***Tools***. Далее ***Compile Java***. Опять ***Tools***, ***Run Java Application*** и, как можно видеть, сумма чисел от 0 до 10, является числом 55. Что при желании можно проверить хотя бы вручную. Но это так. Закроем теперь это окно.

В цикле ***for*** можно организовать не только вот такой возрастающий цикл, но и убывающий, при котором, например, вот эта переменная ***i***2 будет убывать. Например, сложим все числа в убывающем порядке от 20 до 10. Для этого сначала опять проинициализируем переменную ***sum***. Присвоим ей значение 0. Далее оператор ***for***. Скобка. Определим опять переменную внутри цикла ***int*** и далее ***i***2.

Начальное значение пусть будет 20, точка с запятой. А далее условие цикла, ***i***2. Должно быть больше или равно 10. Точка с запятой. И далее переменную ***i***2 уменьшим на единицу. Напишем два знака минус. Закрыть скобки. А теперь к сумме надо прибавить вот эту переменную ***i***2. Напишем это таким образом. Плюс равно и далее ***i***2. Точка с запятой. Теперь все это выведем на печать. Для этого правая кнопка мыши. Далее ***Paste***. Теперь опять переделаем вот эту строчку вывода. Как мы помним, мы складываем от 20 до 10. И попробуем

выполнить это приложение. Для этого сначала ***Tools***, ***Compile Java***. Далее опять ***Tools***, ***Run Java Application***. Как мы видим, сумма чисел от 20 до 10 равняется 165. Закроем это окно.

В операторах цикла можно, конечно же, использовать не только целые числа в качестве счетчиков, но в принципе можно использовать и действительные числа. Но, правда, обращаться с ними надо с осторожностью, помня об ошибках округления. Например, если мы выполним цикл с действительными числами, предварительно определим значение ***sum*** равным 0, далее оператор ***for***, теперь определим переменную действительного типа ***double***. Пусть это будет ***d***1. Начальное значение пусть у нее будет равно 0. Точка с запятой. Затем условие. Пусть условие будет ***d***1 меньше 10. Запятая. И далее приращение к ***d*** за один цикл пусть будет равно 0,1. Закрыть скобку. И просто-напросто подсчитаем количество итераций.

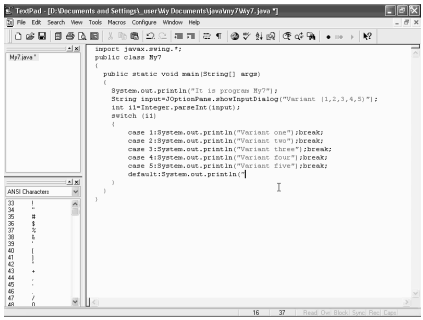
Для этого напишем таким образом: ***sum***+ равно 1. Конечно же, на предыдущей строке нам не хватает точки с запятой. Теперь скопируем вот этот ***println***. Для этого выделим его. Правая кнопка мыши. Copy. ***Enter***. Теперь правая кнопка мыши. ***Paste***. И выведем вот этот результат выполнения нашего цикла. В принципе, если посмотреть на сам цикл, то можно увидеть, что он должен выполняться примерно 100 раз от значения ***d***1 равно 0, до значения ***d***1 равно 9,9. Что, как раз составит ровно 100 циклов.

Посмотрим, так ли это. Выведем просто ***sum***. Закрыть скобку. Точка с запятой. Далее скомпилируем ***Tools***, ***Compile Java***. Далее ***Tools***, ***Run Java Application***. И, как мы видим, наш компьютер выполнил этот цикл не 100 раз, а 101. Вот этот один цикл прибавился за счет ошибки округления. Дело в том, что 0,1 в реальном так сказать исчислении у компьютера на самом деле не 0,1, а немного меньше, 0,099 и т.д. И в результате, выполняя все эти итерации на последнем шаге, компьютер имеет возможность еще одно прибавить вот это 0,1 и все равно переменная ***sum*** будет меньше 10. Чуть-чуть, но будет меньше. В результате прибавляется один лишний цикл.

Еще более надо быть осторожным, если в качестве условия, при использовании действительных чисел, брать знак равенства. Поскольку, если здесь, например, написать знак равно, то цикл просто-напросто вообще никогда не закончится. Поскольку целого числа 10, вот в этом выражения мы никогда не получим.

6.7.4.5. Оператор switch

Если нам в программе надо сделать выбор из многих вариантов, для этих целей в языке ***Java*** есть специальный оператор ***switch***. Он эквивалентен такому же оператору в языках ***C*** и ***C***++. Для того чтобы познакомиться с ними поближе, рассмотрим опять шаблон вот этой простейшей программы, которая выводит на консольное окно вот такую надпись. Сохраним ее на жестком



диске. Для этого щелкнем на кнопку сохранения. Создадим новую папку для нее отдельную. И введем имя для файла нашей программы. Как мы помним, он должен совпадать с названием нашего класса, ***My***7. Щелкнем на кнопку Сохранить. И теперь попробуем использовать оператор ***switch***. Для этого выведем сначала на наш экран диалоговое окно. Это можно сделать при помощи оператора ***showInputDialog***. Для этого возьмем сначала переменную типа ***String***. Пусть эта переменная будет ***input***. Равняется. Далее ***JOptionPane*** Точка. Теперь ***showInputDialog***. Скобка. И теперь внутри этой скобки в кавычках, напишем тот текст, который должен появиться в этом диалоговом окне. Просто напишем вариант 1,2,3,4. Закрыть скобку.

Теперь пусть будет кавычки. Закрыть скобку. Точка с запятой. Конечно же, чтобы использовать этот оператор, нам нужно импортировать соответствующий пакет. Для этого в первой строчке нашей программы напишем ***import***, далее пакет, ***javax***, внутри которого и находится средство поддержки диалогового окна. Далее ***swing***. Звездочка. Точка с запятой.

И теперь, преобразуем эту переменную строковую типа в целое число. Для этого определим новую переменную типа ***integer***. Пусть эта переменная будет ***i***1, равняется. Далее функция преобразования — ***integer***. Точка. ***parseInt***. Скобка и переменная ***input***, внутри которой и должно находиться число, которое определит наши дальнейшие действия. Теперь напишем ***switch***. Далее нам нужно указать в скобках ту переменную, на основе которой и будем производить переключение. Эта переменная у нас ***i***1. Далее фигурные скобки, внутри которых нам нужно написать все содержание нашего переключателя. Щелкнем теперь на клавише ***Enter*** и далее введем оператор ***case***. И далее 1. Это будет означать, что после этого двоеточия будут выполняться операторы, в случае, если у нас ***i***1 будет равен ровно 1. Т.е. мы выбрали первый вариант.

И теперь нам нужно написать те операторы, которые должны выполняться в этом случае. Для этого, просто-напросто скопируем эту строчку, выделим. Правая кнопка мыши. Copy. Вставим сюда. Правая кнопка. ***Paste***. И пусть на консольное окно выводится вариант 1. ***Variant one***. Далее кавычки. Закрыть скобку. Точка с запятой. И далее нам нужно выполнить и написать оператор ***break***, для того чтобы после выполнения вот этого оператора, который предназначен для варианта 1, переключатель прекратил свою работу и не выполнял остальные варианты.

Теперь выполним вариант 2. Для этого проще всего скопировать вот эту строку. Выделим. Правая кнопка мыши. Copy. Далее встанем сюда. Правая кнопка мыши и ***Paste***. Только вместо ***Variant one***, будит ***Variant two***. Теперь опять ***Enter***. Опять правая кнопка мыши и ***Paste***. На этот раз вместо варианта ***Variant two*** будет ***Variant three***, три. Но, соответственно, конечно же, нужно изменить здесь числа. Будет 2 и дальше 3. Теперь напишем, что должно происходить в варианте 4.

Вставим. ***Paste***. Это у нас будет вариант 4, поэтому здесь напишем ***four***. Теперь скопируем еще раз. Правая кнопка мыши и ***Paste***, ***case*** 5 и вариант будет ***five***. Теперь же нам нужно написать операторы, которые будут выполняться в том случае, если у нас ***i***1 не равен ни 1, ни2, ни 3, ни 4, ни 5. Поэтому напишем здесь ***default***, что должно происходить по умолчанию, если ***i***1 у нас не будет равен ни одному из существующих вариантов.

Опять напишем ***println***. Скопируем. Copy. Вставим сюда. Правая кнопка и ***Paste***. И в кавычках напишем ***No Variants***, нет вариантов. Кавычки. Закрыть скобку. Точка с запятой. И опять ***break***. Запустим теперь наше приложение. Сначала его скомпилируем. Для этого ***Tools***. Далее ***Compile Java***. После этого опять ***Tools***, ***Run Java Application***. Вот, появилось диалоговое окно с приглашением выбрать один из 5 вариантов. Но, выберем, например, вариант 4. ОК. И, как мы видим, появилась соответствующая надпись — ***Variant four***. Как мы видим, мы забыли написать ***system exit***.

В связи с этим, нормального выхода не получилось. Закроем наше консольное приложение и в конце напишем ***system exit***. Но и, конечно же, с кодом 0, что символизирует нормальное окончание процесса. Запустим опять наше приложение. ***Tools***. Сначала скомпилируем. ***Compile Java***. Опять ***Tools***, ***Run Java Application***. И на этот раз введем какое-либо совсем другое число, например, 88, которое у нас не присутствует в этом переборе возможных значении. ОК. И как мы видим, в этом случае появилась надпись ***No Variants***, вариантов нет. Нормальное окончание процесса.

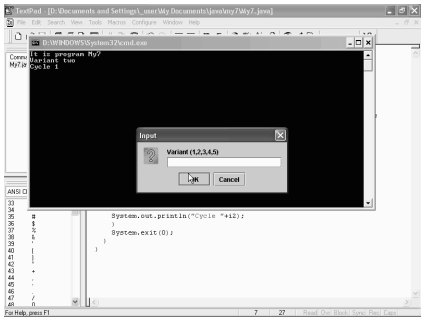
Закроем. Вернемся в обычный текстовый редактор. Попробуем теперь проэкспериментировать, и убрать один из операторов ***break***. Например, вот этот. После выполнения 4-ого варианта. Теперь опять скомпилируем. Для этого ***Tools***. Далее ***Compile Java***. После этого опять ***Tools***, ***Run Java Application***. И теперь, если мы введем вариант 4, и щелкнем на кнопку ОК, то мы можем видеть, что у нас получилось. А именно, выполним все операторы, которые предназначались и для варианта 4 и для варианта 5. Написалось и ***four*** и ***five***. Если же мы запустим это приложение еще раз и напишем 5, то, как мы видим, в этом случае выполняется оператор, который написан у нас для варианта 5 и только.

Т.е. в случае выбора 1,2,3 или 5, выполняется только 1 из соответствующих вариантов, а в случае выбора варианта 4, у нас выполняется и операторы 4 варианта и операторы 5. Это произошло, потому что мы ключевое слово ***break***, после оператора 4 варианта не написали. В принципе, если мы хотим разбить наши варианты на независимые друг от друга участки, то ***break*** обязательно надо писать после каждого варианта. Для того чтобы избежать вероятных ошибок. Отметим еще один момент. А именно, что вот эта переменная, которая у нас стоит внутри оператора ***switch***, обязательно должна быть целочисленной, всякие другие варианты были не предусмотрены. Здесь нельзя использовать ***String***, нельзя использовать другие числа и т.д., только лишь целое.

6.7.4.6. Прерывание потока управления. Оператор ***break***

Рассмотрим теперь такой оператор, как оператор Перехода. Оператор ***go to***, в отношении которого довольно часто возникают споры. И большинство программистов считают применение самого оператора ***go to*** признаком плохого тона в программировании. Именно поэтому разработчики языка ***Java***, Хотя и сохранили само зарезервированное слово ***go to***, но в сам язык его не включили. Правда, вместо оператора ***go to***, существует специальный оператор для обеспечения преждевременного выхода из цикла.

Это оператор ***break***, такой же, как мы применяли в случае переключателя. Например, напишем какой-либо цикл, при помощи которого выполним вот этот кусочек программы несколько раз. Для этого здесь щелкнем на клавише ***Enter***, далее напишем, оператор ***for***, и теперь внутри скобок создадим новую переменную целого типа. Пусть это переменная будет ***i***2.



Она меняется от 1 до, например, ***i***2 меньше 10. На каждом цикле пусть будет приращение на единицу. Закрыть скобки. Далее все остальные операторы поместим в фигурные скобки. А также, на каждом цикле на наше консольное окно выведем номер итерации ***i***2. Для этого скопируем вот этот ***println***, выделим его. Правая кнопка мыши. Copy. Вставим сюда. Правая кнопка и ***Paste***. Далее кавычки. Напишем цикл. Закрыть кавычки и номер ***i***2. Закрыть скобку. Точка с запятой. Вот теперь мы пока сделали цикл, при помощи которого просто-напросто 9 раз при ***i***2 равен от 1 до 9, будет выполняться вот этот цикл, цикл запроса номера варианта. А теперь представим себе, что нам, в каком-либо месте нашей программы,

например, вот в этом месте, проверить какую-либо величину и в зависимости от его значения, может потребоваться нам аварийный выход из вот этого цикла.

Например, напишем таким образом. Оператор ***IF***, далее проверим ***i***1. Если ***i***1 равняется, например, величине 55, тогда напишем оператор ***break***. Точка с запятой. Исправим теперь вот в операторе ***if*** знак равенства. Конечно же, здесь должен быть знак сравнения, а также вместо переменной ***i***, здесь должна быть переменная ***i***2, в этом операторе ***for***. Теперь скомпилируем эту программу. Для этого щелкнем на пункте ***Tools***. Далее ***Compile Java***. Теперь опять ***Tools***, ***Run Java Application***. Как мы видим, программа запустилась. Появилось вот это диалоговое окно. Введем здесь какое-либо число. Например, 2. ОК. Вот, как мы видим, это у нас вариант 2 отметился. И цикл номер 1. Теперь введем какое-либо другое число. Например, 4.

ОК. Как мы помним, в случае варианта 4, выполняется и оператор варианта 4 и варианта 5, поэтому напились две строки. ***Variant Four*** и ***Variant Five***. Теперь же, если мы введем какое-либо число, которое не входит в эти варианты, например, 33. ОК. Появилась надпись ***No variants***, а если же мы введем число 55, которое у нас являлось так сказать ключевым, для выхода из цикла, и щелкнем на кнопку ОК, то, как мы видим, у нас просто-напросто завершилось выполнение нашей программы. Что и было нами заложено. Закроем.

Кроме оператора ***break***, у языка ***Java***, существует еще один оператор, оператор ***continue***, который тоже наподобие оператора ***break*** прерывает нормальное выполнение нашей программы, но в отличие от оператора ***break*** не выходит целиком из цикла, а прерывает выполнение текущего цикла и передает управление на начало цикла. На выполнение следующей операции. Но, например, напишем здесь же, там где у нас был оператор ***break***, еще одну проверку ***if***, ***if i***1 равняется, например, 66, тогда выполним оператор ***continue***. Точка с запятой.

Теперь запустим эту программу. Для этого щелкнем на пункте ***Tools***. Далее ***Compile Java***. Теперь опять ***Tools***, ***Run Java Application***. Введем какое-либо число. Например, 3. Как видим, сейчас выполнился цикл номер 1. Если зададим число 5, выполнился цикл номер 2. А если же введем число 66, то, как мы видим, операторы следующего цикла не выполнились. Правда, выход из всего цикла не произошло. Опять появилось диалоговое окно. Если, сейчас опять введем какой-либо номер, например, номер 2, щелкнем на ОК, то, как мы видим, после цикла 2 мы перешли к циклу 4, цикл 3 у нас просто-напросто пропустился. Пропустился, благодаря оператору ***continue***.

Если же мы сейчас введем число 55 и щелкнем на ОК, то мы совсем выйдем из нашего цикла и соответственно из нашей программы, благодаря оператору

break. Закроем. Надо отметить, что при помощи оператора ***break*** мы можем прекратить выполнение только текущего цикла, самого вложенного. Что же делать, если мы хотим выйти из нескольких циклов сразу? Для этого существует в языке ***Java*** специальная разновидность оператора ***break***, ***break*** с меткой. Например, наряду с вот этим циклом ***for*** напишем еще один цикл. Опять напишем ***for***, ***int***.

Пусть это будет на этот раз переменная ***i***3. Пусть она опять меняется от 1 до ***i***3 меньше 10. Тут особо мудрить не стоит. Далее ***i***3 ++ на каждом шаге увеличиваем на единицу. Далее фигурные скобки, внутри которых поместим первый оператор ***for***. Теперь нам нужно еще вставить закрывающую скобку фигурную, которая встанет вот в это место. И далее, при распечатке выведем не только ***i***2, но и ***i***1. Поэтому напишем, таким образом, ***i***1, далее Плюс, кавычки, пара пробелов и плюс. Вот попробуем выполнить все это таким образом. Теперь же попробуем использовать оператор ***break*** с меткой. Для этого введем метку вот этому внешнему циклу. Например, пусть это будет следующая аббревиатура. Просто-напросто возьмем ***abcd***. В принципе это может быть любое имя, как мы составляем имя переменной, только дальше это должно быть двоеточие. И теперь вот в этом месте, где мы выполняли все проверки, выполним еще одну проверку ***if***. И если переменная ***i***1 у нас будет равняться, возьмем 77, в этом случае выполняется оператор ***break*** с меткой ***abcd***. Т.е. происходит выход всех циклов до метки ***abcd***.

Попробуем теперь выполнить эту программу. Для этого ***Tools***, ***Compile Java***. Сначала скомпилируем. Теперь ***Tools***, ***Run Java Application***. Запуск программы. Появляется вот это стандартное диалоговое окно. Наберем здесь единицу. ОК. Вот, как видно, у нас цикл 1,1. Возьмем теперь, например, 2. Выполняется цикл 1,2. Если же мы возьмем число 66, ОК. Как мы видим, мы пропустили один цикл. Если возьмем, например, число 4, видно, что мы после второго цикла перешли сразу на 4. Возьмем теперь число 55. ОК. Мы опять пропустили цикл, но на этот раз, если опять введем какое-либо число. Например, 2 и щелкнем на ОК, то можно видеть, что мы перешли к следующему циклу во внешнем цикле. Вот здесь, после единицы, появилась 2. А во внутреннем цикле у нас счет пошел сначала, с единицы.

Если мы теперь опять введем какое-либо число. Например, 5, то мы переходим к следующему по порядку циклу. А если же мы введем число 77. ОК. То на этот раз мы вышли из первого и из второго цикла. Мы перешли по метке, выполненной оператором ***break*** с меткой. Закроем теперь наше приложение и отметим, что в принципе такой оператор ***break*** с меткой и вот эту метку можно приписывать не только оператору цикла, но и любому составному оператору. Например, оператору ***if***, просто-напросто в круглых скобках, внутри которого находится некий составной оператор и т.д. Правда,

как мы сказали ранее, использование вот этих операторов ***break*** и ***continue***, операторов прерывания нормального выполнения программы, в принципе считается дурным тоном в программировании. И в надо воздерживаться от их неоправданного употребления, поскольку их использование может довольно сильно запутать программу и сделать ее совершенно нечитабельной.

6.7.4.7. Большие числа. Арифметические операции

При работе с программами, может возникнуть ситуации, когда обычной точности вычисления с целыми и действительными числами, которые есть по умолчанию в языке программирования ***Java***, нам может быть недостаточно. Именно для этих целей, в языке программирования ***Java*** есть специальные возможности, которые сконцентрированы в двух полезных классах, в пакете ***Java Math***, под названием ***BigInteger*** и ***BigDecimal***, которые позволяют оперировать с числами любой точности, состоящими из произвольного количества цифр. Для того чтобы начать рассмотрение этих вопросов, начнем работу с простейшего шаблона нашего приложения, которое выводит одну строчку в стандартное консольное окно. Сохраним этот шаблон в каком-либо файле. Для этого щелкнем на значке сохранения. Создадим папку, введем для нее имя. Пусть будет ***My***8. Далее ведем имя для нашего приложения. Как мы помним, оно должно совпадать с названием класса. Поэтому возьмем опять ***My***8. Причем первая буква М большая, так же как и в названии класса. И щелкнем на кнопку сохранения.

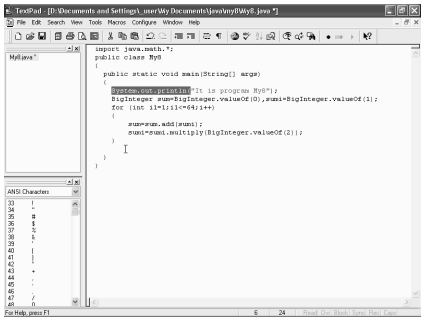
И теперь мы можем непосредственно начать работу по созданию нашего приложения. Для того чтобы мы могли использовать пакет ***Java Math***, мы должны его импортировать в нашу программу. Для этого в первой строчке нашего приложения напишем таким образом. ***Import***, далее ***Java***, ***Math*** и точка с запятой. Теперь решим задачу, связанную с большими числами. Попробуем подсчитать точное количество зерен, которое должен был выдать король древнему шахматисту за его изобретение шахмат.

При этом, как мы помним, на первую клетку нужно положить 1 зерно, далее 2, 4, 8 и т.д. до последней 64 клетки. Для того чтобы это подсчитать, введем две переменные больших чисел целого типа. Этот тип у нас будет ***BigInteger***. Пусть одна переменная будет ***sum***, в не***q*** будем хранить сумму всех зерен на всех досках. Проинициализируем его сначала. Начальное значение пусть будет

0. Для этого мы не можем написать прямо равенство ***sum*** и 0, это будет ошибкой. Эта операция не применима к большим числам.

При использовании больших чисел, надо использовать специальные операторы и для сложения, и для вычитания, и даже для присваивания. Поэтому нам просто обычное число 0 надо превратить в большое число. Поэтому напишем таким образом, ***BigInteger***, точка и ***valueOf***. Укажем в скобках число 0, которое нам нужно. Т.е. вот этим оператором ***BigInteger valueOf*** конвертировали обычное число 0 в число типа ***BigInteger***. Теперь введем еще одну переменную ***sumi***, в которой будем хранить текущее значение количества зерен на конкретном шахматном поле. Опять нам надо присвоить какое-либо значение, которым воспользуемся. Для этого напишем ***BigInteger*** и далее ***valueOf***. И на этот раз проинициализируем его числом 1. Поскольку, как раз с одного зерна все и начиналось. На первой клетке должно быть одно зерно. Далее напишем оператор цикла ***for***. Введем обычную переменную целого типа ***i***1, которая у нас будет меняться от 1 до 64. С приращением единицы. Поэтому напишем ***i***++.

Далее, внутри фигурных скобок напишем то, что у нас должно происходить внутри цикла. А внутри цикла мы проведем, во-первых, суммирование, присвоим нашей переменной ***sum*** значение, равное предыдущему его значению плюс число зерен на конкретной шахматной доске и для этого нам надо воспользоваться специальным оператором, который есть в операциях с большими числами. Мы здесь не можем просто написать сумму этих двух переменных. Такая запись выражения недоступна для операций с большими числами, поэтому надо воспользоваться другой записью, которая тут существует. ***Sum***, точка и далее надо воспользоваться специальным оператором ***Add***, добавить, ***c***кобка. И вот теперь мы можем написать ***sumi***, текущее значение зерен.



Теперь нам надо это выражение ***sumi*** увеличить в два раза, поскольку на следующей клетке у нас будет зерен в два раза больше, чем на предыдущей. Поэтому напишем таким образом. ***Sumi***, знак равенства и, точно так же, как мы не могли воспользоваться простыми операциями в предыдущем случае, здесь мы тоже не можем написать просто умножить на 2. Такая запись неприменима. Поэтому напишем несколько в другом виде. А именно, ***sumi*** точка и специальный оператор ***multiply***, который позволяет умножать на 2, не только на 2, а на любое другое

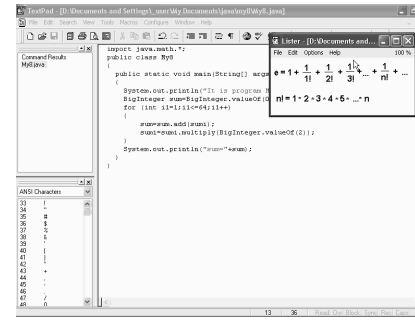
число. И здесь мы тоже не можем написать внутри скобок просто 2, поскольку число 2 не является большим числом.

Его надо конвертировать в большое число, для чего мы опять воспользуемся вот этой операцией ***BigInteger valueOf***. Для этого проще скопировать вот этот кусочек текста. Правая кнопка мыши, Copy. Далее вставим сюда. Правая кнопка, ***Paste***. Единственное отличие, вместо 1 здесь нам нужно написать 2. Далее закроем скобку, точка с запятой. Теперь попробуем вывести результат на печать. Для этого воспользуемся ***println***. Выделим вот этот кусок текста.

Далее правая кнопка мыши, Copy. Вставим сюда. Правая кнопка, ***Paste***, в скобках напишем ***sum***, знак равенства, плюс и само значение переменной. Закроем скобку, точка с запятой. Скомпилируем теперь эту программу и запустим. Для этого ***Tools***, ***Compile Java***. Далее ***Tools*** и ***Run Java Application***. Как видим, при компиляции нам компьютер выдал ошибку. Но в принципе достаточно понятно, что здесь произошло. При написании цикла ***for***, мы около ***i*** не написали 1, поскольку идентификатор у нас ***i***1. Поэтому переключимся на наш текст и припишем здесь недостающую здесь единицу. Теперь скомпилируем опять.

Для этого ***Tools***, ***Compile Java***, ***Tolls*** и ***Run Java Application***. На этот раз компиляция прошла успешно, поэтому запустим приложение на выполнение. Можно видеть, сколько зерен на самом деле нужно было выдать. Вот это большое число, которое, конечно же, простым ***Integer*** или ***LongInteger*** записать невозможно. Можно подсчитать, что здесь используется 20 цифр для записи этого выражения. Закроем теперь наше консольное окно.

6.7.4.8. Большие числа. Сравнение. Вычисление с большой точностью



Как мы могли видеть, для работы с большими числами, простые операции сложение, умножение, деление и т.д. не применимы. Для этого существует специальный класс операций. Это операции, которые возвращают сумму, разность, субстрат, произведение ***multiply***, ***devide*** частные и ***mod***, остаток от деления одного числа на другое. Также у больших чисел есть специальный оператор сравнения ***compare To***, который сравнивает два числа и возвращает значение 0, если: данное число больше второго, возвращает отрицательное число, если данное число больше, и возвращает положительное число в противном случае.

Последний оператор, оператор ***valueOf***, который возвращает большое число, значение которого равно переменной X, превращая ее в большое число для наших дальнейших операций. Этот оператор мы уже использовали в нашей программе. Операции с большими числами возможны, конечно же, не только с целыми, но и с действительными числами, числами с плавающей запятой.

Для этих целей приведем пример. Например, подсчитаем число ***e*** с большой точностью. Для этого воспользуемся вот такой формулой для подсчета числа ***e***. E — это есть сумма обратных величин факториалов натуральных чисел. А число факториал, это произведение от 1 до ***n***, всех натуральных чисел. Воспользуемся этой формулой. Поскольку возрастание ***n***, это число очень быстро убывает, то эта формула достаточно быстро сходится. Для контроля выведем сначала число ***e***, которое есть в этом пакете ***Math***, на принтер. И, чтобы не писать, выделим эту часть текста. Правая кнопка мыши, Copy. Затем вставим сюда. Правая кнопка мыши, ***Paste***.

И теперь напишем таким образом. Число ***E***, знак равенства, кавычки, плюс и теперь его представление из пакета ***Math***. Поэтому напишем ***Math***, точка и далее большую букву E. Закроем скобку, точка с запятой. Теперь попробуем это все подсчитать. Для этого введем переменную типа ***BigDecimal***, которая и является действительным числом с большой точностью. Назначим число ***e*** и введем для него начальное значение.

Как мы помним, прямо присвоить какое-либо значение тут мы не можем. И, так же как и в случае с натуральными числами, нам нужно воспользоваться специальным оператором ***valueOf***. Но, на этот раз возьмем не ***BigInteger***, а ***BigDecimal***, точка. Теперь ***valueOf***. И в скобках присвоим ему значение 0, запятая. Теперь в отличие от целого числа, здесь мы можем указать вместе с его значением и точность количества цифр, с которой мы хотим это число иметь. Возьмем, например, число 35. Мы хотим узнать точность числа e 35 знаков. Далее закроем скобку, запятая.

И возьмем еще одно число. Обозначим его ***e***1, в котором будем хранить число единица деленное на ***n*** факториал. Проинициализируем это число. Для этого скопируем вот этот кусок текста. Правая кнопка мыши, Copy. Далее вставим сюда. Правая кнопка, ***Paste***. Точка с запятой. Вот теперь у нас две переменные, типа ***BigDecimal*** уже есть.

Единственное, во втором из этих чисел запишем теперь единицу для начала. Поэтому напишем таким образом, ***e***1, знак равенства, далее ***e***1, точка ***add*** (добавить), и добавим единицу. Но нам нужно воспользоваться вот этим оператором ***BigDecimal valueOf***. Поэтому скопируем, выделим. Правая кнопка мыши, Copy. Вставим сюда. Правая кнопка, ***Paste***. И добавим здесь, конечно же, единицу. Теперь нам точность не нужна, в ***e***1 у нас точность уже зафиксирована, поэтому закроем все скобки и поставим точку с запятой.

Введем еще одну переменную, просто ***integer***. Пусть это будет ***i***1, которая нам понадобится для организации цикла. Для начала ее проинициализируем нулем, и далее напишем оператор цикла ***while***. Условием у нас пусть будет точность, и для этого напишем такое выражение. E1, прекращать мы будет вычисление, когда вот это число ***e***1, у нас будет меньше, чем 10 минус в 35 степени, поскольку мы как раз и хотим взять 35 знаков.

Поэтому напишем таким образом, ***e***1, далее оператор сравнения, ***compareTo***. Как мы помним, простые операторы больше — меньше мы здесь использовать не можем, поэтому напишем ***compareTo***, и скобки, внутри которых нам опять надо воспользоваться вот этой конструкцией ***BigDecimal valueOf***.

Поэтому выделим. Правая кнопка мыши, Copy. Вставим сюда. Правая кнопка мыши, ***Paste***. Точка с запятой нам не нужна, и вместо нуля напишем единицу. И тогда, вот это число, будет числом 1 умноженное на 10 в минус 35 степени. Закроем теперь еще одну скобку и далее, как мы помним, точно так же как и в случае с целыми числами, вот этот оператор ***compareTo*** выдает нам значение 0, если эти два числа равны, а значение больше 0 или меньше 0, в зависимости от того, какой из этих чисел больше. Поэтому возьмем больше или равно 0. Пока это верно, вычисления у нас будут происходить. Внутри фигурных скобок нам нужно написать, что же у нас будет происходить, так сказать, в теле цикла.

Увеличим значение ***i***1 на единицу, ***i***1++. Точка с запятой. Далее, к значению ***e*** надо добавить значение ***e***1. Для этого опять воспользуемся оператором ***e***, точка ***add***, добавить. Далее скобки и значение, которое у нас есть в переменной ***e***1. Закроем скобку, точка с запятой. Теперь нужно ***e***1 разделить на ***n***, для того чтобы там все время сохранялось число 1 деленное на ***n*** факториал. Поэтому напишем ***e***1, знак равенства ***e***1, точка и далее ***divide*** (делить), скобка. Теперь нам понадобится опять вот эта функция, преобразование обычных чисел в большое число. Выделим. Правая кнопка мыши, Copy. Вставим сюда. Правая кнопка мыши, ***Paste***. Делить нам, конечно же, нужно на вот это число ***e***1, на переменную нашего цикла ***i***1.

Закроем скобку, запятая. И теперь нам нужно указать, как будет происходить округление при нашей операции деления. Напишем таким образом, ***BigDecimal***, точка. И далее режим округления ***Round Half Up***, который означает обычное округление. Т.е. если последняя цифра меньше 5, то остальные цифры отбрасываются, и если следующая цифра больше или равна 5, последняя цифра увеличивается на единицу.

Далее закроем скобки, точка с запятой. Следующим оператором выведем все это на печать. Для этого воспользуемся операцией ***println***. Скопируем. Правая кнопка мыши, Copy. Вставим сюда. Правая кнопка, ***Paste***. Далее

выведем на печать оба числа, и единицу делить на ***n*** факториал, т.е. число ***e***1 и само число ***e***, подсчитанное нами.

Для этого напишем ***e***1, плюс. Введем еще пару пробелом между этими двумя числами, чтобы они отделялись друг от друга и далее само число ***e***, которое напишем таким образом, ***e***, закроем скобку, точка с запятой. В принципе, мы можем ввести и такой символ ***e*** со знаком равенства, чтобы это все у нас было видно более четко, что у нас тут происходит. Теперь скомпилируем эту программу и запустим ее на выполнение. Для этого ***Tools***, ***Compile Java***. Далее ***Tolls***, ***Run Java Application***. И вот можно видеть, что у нас получилось. Немножко растянем это окно, для того чтобы у нас все наши вычисления поместились и можем видеть, как у нас изменялось число ***e*** при вычислениях. И как у нас изменялся обратный факториал.

Как мы видим, он у нас все время мешался, пока не дошел до 0. А число ***e*** у нас получилось вот такое. Как мы видим, оно в принципе достаточно точно совпадает с числом ***e***, которое у нас находилось в пакете ***Math***. Первое, соответствующее количество цифр у нас совпадает, а далее после этого, мы дополнительно подсчитали еще достаточно много количества цифр.

Надо сказать, что, скорее всего, последняя цифры 1 или 2, тут вряд ли будет правильными из ошибок округления, но где-то 34, 33 цифры здесь будут абсолютно верными. Вот таким образом мы подсчитали число ***e***. Конечно же, такие вычисления при помощи обычных операций сложения, вычитания и т.д. с обычными целыми действительными числами, даже с двойной точностью недоступны. Закроем теперь это окно. И вернемся в наш обычный редактор.

6.7.4.9. Одномерные массивы. Создание. Инициализация. Число элементов

Очень часто при написании программ встречаются такие структуры данных, как массивы, которые представляют собой некий последовательный набор данных, в котором хранятся величины одинакового типа. Причем надо сказать, что к отдельному элементу массива, доступ осуществляется с помощью целого индекса. Для того чтобы познакомиться с этим понятием поближе, начнем работу с вот этого простейшего приложения, которое, как всегда, выводит на наш консольный экран одну строчку вот такого текста. Сохраним этот файл.

Для этого щелкнем на кнопке сохранения. Создадим для него отдельную папку. Щелкнем на кнопке Открыть и далее введем имя для нашего файла. Ну, пусть будет вот такое. Как мы помним, это имя должно совпадать с именем класса. Щелкнем на кнопке Сохранить. И вот предварительное действие мы уже завершили и теперь займемся с нашими массивами. Первое, что надо сделать при работе с массивами, это, конечно же, его объявить. Объявление происходит достаточно просто. Надо просто указать тип элементов, содержащийся в массиве. Затем пара пустых квадратных скобок и имя переменной. Например, сделаем таким образом, определим массив целых чисел. Для этого нам нужно указать, что это ***int***, затем квадратные скобки и далее само имя этой переменной массива. Ну, пусть имя будет ***a***1. Точка с запятой. Вот после этого объявления мы уже можем с переменной ***a***1 работать.

Однако надо отметить, что этот оператор, который мы только что написали, лишь объявляет переменную, но не инициализирует ее настоящим массивом. А для того чтобы именно инициализировать массив, нам здесь еще нужно написать дополнительно его начальную инициализацию, которую напишем таким образом, знак равнения, затем ***new int***, поскольку у нас массив целых числе и, далее в квадратных скобках укажем размер нашего массива. Пусть, например, это будет массив, состоящий из 10 чисел. И массив, состоящий из 10 чисел, нумеруется числами от 0 до 9, т.е. от 0 до вот этого числа (-1). Поэтому, если мы захотим вот этот массив ***a***1 заполнить какими-либо данными, то для этого нам нужно будет написать следующие коды.

Например, напишем, оператор цикла ***for***, далее создадим переменную, допустим, ***i***1, которая у нас будет меняться от 0 до ***i***1 меньше 10, точка с запятой, и ***i***1++ прибавление на единицу. И далее напишем такой оператор, ***a***1 от ***i***1. Вот этому элементу массива присвоим какое-либо значение. Например, пусть это будет квадрат от ***i***1. Т.е. ***i***1 умножить на ***i***1. Точка с запятой. Вот, после этих нескольких строчек, у нас уже будет полностью готов массив ***a***1, заполненный квадратами соответствующих первых 9 натуральных чисел. Как видите, еще одно обстоятельство при работе с массивами, если у нас программа достаточно большая, мы можем не помнить в середине или в конце нашей программы, сколько же элементов у нас было в массиве ***a***1.

Для того чтобы это узнать, оказывается, есть специальная функция ***u***1, которая называется ***length***. Поэтому, вот здесь, когда мы инициализировали наш массив, вместо 10 мы можем написать такую конструкцию, ***a***1 точка ***length***, длина массива ***a***1. И теперь мы здесь к числу 10 привязаны уже не будем. Размер массива достаточно указать один раз при его инициализации, а дальше пользоваться вот этой величиной ***length***. Надо помнить, что в языке ***Java***, изменять размер массива невозможно. Размер массива создается только один раз при его инициализации.

А теперь же попробуем распечатать наш массив и вынести его на наш экран. Для этого опять воспользуемся вот этим оператором цикла, который проще всего скопировать, выделим его, щелкнем правой кнопкой мыши, Copy. Затем встанем вот в это место. Вновь щелкнем правой кнопкой мыши и ***Paste***. ***Enter***. И теперь скопируем оператор ***println***. Выделим его, вызовем контекстное меню, Copy. Встанем сюда. Правая кнопка мыши и ***Paste***. И теперь напишем здесь, что мы хотим распечатать. Ну, распечатаем сначала номер элемента массива ***i***1. Плюс, затем несколько пробелов, затем плюс, дальше элемент массива ***a***1 с номером ***i***1. Закроем скобку, точка с запятой.

Запустим теперь наше приложение. Для этого сначала скомпилируем его. Развернем меню ***Tools***. Далее ***Compile Java***. Компиляция прошла успешно. Теперь в меню ***Tools*** выберем команду ***Run Java Application***. Запустим наше приложение и можно видеть, что у нас получилось. Мы выписали целые числа и рядом их квадраты. Закроем наше приложение. Конечно, в принципе то же самое можно было сделать гораздо проще, не прибегая к массивам. Просто написать ***i***1 и рядом его квадрат, но понятие массива, конечно же, используется не только для таких простых случаев. Определить массив можно и по-другому, сразу заполняя его значениями.

Для этого напишем таким образом. Напишем ***int***, поскольку мы опять хотим создать массив из целых чисел, затем квадратные скобки, для того чтобы указать, что это у нас будет массив. Далее название этого массива. Пусть это будет массив ***a***2, знак равнения и далее в его фигурных скобках уже укажем сразу же элементы этого массива. Пусть это будет, например, 1,2,3,4, далее введем обратный отчет 4,3,2 и 1. Далее закроем фигурную скобку. Точка с запятой. Т.е. мы сразу же определили массив, сразу же определяется его размерность, как число элементов в этом массиве, и его отдельные элементы. Теперь, например, этот массив можно распечатать. Для этого скопируем вот эти две строчки, чтобы не писать их заново. Выделим их. Щелкнем правой кнопкой мыши. Далее Copy.

Теперь встанем вот сюда. Правая кнопка мыши, ***Paste*** и теперь все, что нам нужно сделать, это изменить ***a***1 на ***a***2. У нас уже массив ***a***2. Точно, то же самое при распечатке. Распечатать мы хотим массив ***a***2. Теперь запустим наше приложение. Развернем меню ***Tools*** и для начала выберем команду ***Compile Java***. Далее воспользуемся командой ***Run Java Application***. И вот, можно видеть, что второй массив у нас состоит из 8 элементов с индексами от 0 до 7, и он распечатался у нас вот в этом месте. Идут цифры 1,2,3,4, а затем в обратном порядке. Закроем это консольное окно. И надо отметить еще одно обстоятельство. А именно, то, что при определении массива, вот эти фигурные скобки мы можем писать не только после ключевого слова, определяющего вид нашего массива, но и после имени массива.

Например, если мы здесь удалим квадратные скобки, указывающие на то, что это массив, и введем их после имени переменной, то это абсолютно одно и тоже. Эти две записи идентичны.

То же самое можно сделать и при определении первого массива. Вот эти две скобки, можно было перенести после названия массива ***a***1. Давайте скомпилируем сейчас нашу программу, выберем команду ***Compile Java***, затем ***Run Java Application***, то можно заметить, что у нас ничего не изменилось. На экране мы получили абсолютно один и тот же результат, одно и тоже окно. Как мы видим, от перестановки наших фигурных скобок, ничего не изменилось. Закроем наше приложение и вернемся в наш текстовый редактор. При применении массивов, правомочно использовать оператор присваивания, оператор равенства.

Например, если мы напишем следующую строчку, определим еще один массив, назовем его, например, ***a***3, и этому массиву присвоим значение ***a***2. Точка с запятой. Но при этом надо помнить, что массив ***a***3 и ***a***2, это на самом деле один и тот же массив, который ссылается на один и тот же набор данных. И поэтому, если мы здесь изменим какой-либо элемент какого-либо из этих массивов, например, ***a***2 от 2, например, второй элемент с индексом 2. Присвоим ему, например, значение 100. Точка с запятой. А элементу массива ***a***3 с номером, например, 5 присвоим значение 50, на самом деле, в случае первого и второго присвоения изменяется значение обоих массивов, поскольку это опять-таки один и тот же массив. Попробуем это проверить.

Для этого распечатаем оба массива. Скопируем эти две строчки. Щелкнем правой кнопкой мыши, Copy. Далее встанем сюда. Правая кнопка и ***Paste***. Теперь изменим распечатку ***println***. Сделаем ее немного другой. После ***a***2 распечатаем еще значение массива ***a***3. Для этого сначала введем пару пробелов. Затем +, далее ***a***3, тоже от ***i***1. Скомпилируем. Воспользуемся командой ***Compile Java***, далее ***Run Java Application***. И можно видеть, что и массив ***a***2 и массив ***a***3, у нас абсолютно одинаковые. Этот элемент с индексом 2 у нас в обоих элементах стал 100, и элемент с индексом 5, тоже в обоих индексах стал 50. Хотя мы изменяли сначала элемент ***a***2, а затем элемент ***a***3. Но изменились оба массива, поскольку это один и тот же набор данных. Закроем окно программы и вернемся в обычный режим редактора.

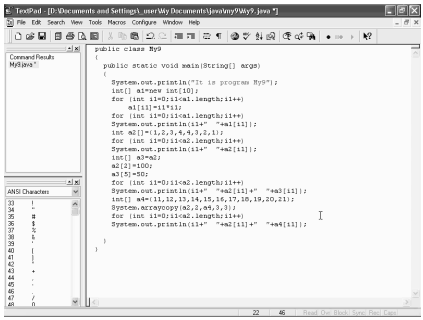
6.7.4.10. Одномерные массивы. Копирование. Сортировка

Если же у нас встал вопрос, чтобы создать еще один массив, элементы которого будут равны, например, массиву a2, так сказать, провести копирование, при котором будут создаваться два идентичных набора данных. Для этого в языке ***Java*** существует специальный оператор ***array.copy***. Конечно же, при использовании этого оператора надо иметь в виду, что массив, в

который мы копируем элементы из другого массива, должен быть по размерам не меньше, или такой же или больше.

Например, создадим еще один массив. Опять целого типа. Пусть он называется a4. Возьмем его размер, например, чуть больше. Определим его опять в явном виде. Напишем таким образом. Пусть это будет 11,12,13,14 и т.д. Ну вот, пожалуй, элементов здесь хватит. Явно видно, что массив a4 больше, чем массив a2, и аналогично a3.

И далее скопируем элементы из массива a2 в массив a4. Теперь напишем таким образом ***System***, далее точка. Теперь оператор ***array.copy***. Откроем скобку. Теперь нужно указать из какого массива мы хотим копировать. Например, мы копируем из массива a2, запятая. Теперь надо указать с какого индекса мы хотим копировать элементы. Например, начнем копировать из элемента с индексом 2. Далее надо указать, так сказать, массив приемник. Это у нас a4, запятая. Теперь надо указать индекс, начиная с которого в массиве a4 будут располагаться эти элементы, но пускай это будет у нас число 3, запятая. И далее надо указать количество копируемых элементов. Например, скопируем 3 элемента. Закроем скобку, точка с запятой.



Еще раз надо отметить, что во избежание ошибок и аварийного выхода, массив a4 должен быть достаточно большим, чтобы в нем поместились все копируемые элементы, начиная с того номера, который мы здесь указываем. А теперь же, после копирования, попробуем распечатать массив a2 и a4. Для этого опять воспользуемся вот этими двумя строчками. Выделим их. Щелкнем правой кнопкой мыши и Copy. Далее вставим сюда. Правая кнопка и ***Paste***. И вместо a3 тут напишем a4, поскольку нас теперь интересует именно этот массив. Правда, надо отметить, что при распечатке вот таким образом, у нас из 4 распечатываются не все элементы, а только такое же количество элементов, как и в массиве a2, поскольку мы здесь задали распечатать количество элементов, равное a2 ***length***. Т.е. то, которое есть в массиве a2. Запустим теперь наше приложение. Для этого развернем меню ***Tools***. Сначала скомпилируем ее.

Далее запустим, используя команду ***Run Java Application***. И вот можно увидеть, что у нас получилось. А именно, скопировалось у нас три элемента. Причем копирование у нас произошло со сдвигом. Т.е. вот это число 100, которое было на втором месте в первом массиве, перешло на третье место в

массиве a4, вот в этом. То же самое произошло со следующим элементом. Вот эта 4 тоже перешла сюда, но со сдвигом. И следующая 4 тоже. А все остальные элементы массива a4 остались такими же, как и были. Вот в начале 11,12,13 и в конце 17,18. Эти элемента массива не изменились. Изменились только вот эти три, которые стали равны соответствующим трем элементам из массива источника. Закроем теперь наше консольное окно и опять вернемся в наш обычный текстовый редактор.

При работе с массивами, иногда возникает необходимость по сортировке элементов массива и для этого существует специальный оператор ***sort***. Например, как мы помним, у нас уже есть массив a2, попробуем теперь его элементы скопировать в новый массив, который сейчас создадим. Для этого создадим массив целого типа ***int***. Далее введем квадратные скобки. Определим новый массив. Пусть это будет массив a5. Знак равнения.

И далее напишем таким образом, ***new int*** и далее укажем размер нашего массива. Размер укажем, как длину массива a2. ***a***2 точка далее ***length***. Закроем скобку, точка с запятой. Теперь у нас a5 и a2, оба массива одинаковой длины. Теперь скопируем все элементы a2 в a5. Для этого вновь воспользуемся оператором ***System.arraycopy***. Так и напишем. Далее откроем скобку. Массив источник у нас будет a2. Запятая. Копировать мы будем из элемента с индексом 0. Запятая. Далее у нас массив приемник будет a5. Запятая. Опять мы копируем, начиная с индекса 0.

И далее количество копируемых элементов. Мы возьмем все элементы, поэтому напишем a2 ***length***. Закроем скобку. Точка с запятой. Теперь же, после того, как мы все элементы скопировали, попробуем элементы массива a5 отсортировать. Для этого надо написать таким образом ***Arrays***, точка ***sort***. И внутри скобок название массива a5. Закроем скобку. Точка с запятой. Теперь попробуем массивы a2 и a5 распечатать. Для этого воспользуемся опять вот этой частью ***println***, вместе с соответствующим циклом.

Выделим эти строки. Щелкнем правой кнопкой мыши, Copy. Далее встанем сюда. Правая кнопка и ***Paste***. И теперь единственное, здесь нам надо изменить, вместо a4 написать распечатку массива a5 и кроме того, чтобы отделить эту задачу от всех предыдущих, введем какую-либо строчку. Для этого скопируем вот этот оператор. Щелкнем правой кнопкой мыши, Copy. Встанем сюда. Правая кнопка мыши и ***Paste***. И внутри кавычек напишем просто-напросто вот такую разделительную линию. Далее закроем кавычки. Закроем скобку. Точка с запятой.

Теперь, для того чтобы эта программа корректно работала, нам нужно так сказать узаконить использование вот этой функции ***Arrays.sort*** из класса ***Arrays***. Для этого в нашей программе мы должны импортировать соответствующий пакет ***util***. Поэтому напишем таким образом, ***import***, далее

java, точка, ***util***, точка, звездочка, точка с запятой. Теперь мы можем скомпилировать программу. Для этого развернем меню ***Tools***, ***Compile Java***. Компиляция прошла успешно. Теперь ***Tools*** и ***Run Java Application***. И можно видеть результат выполнения нашей программы. Как мы видим, у нас и массив a2 и массив a5 состоят из одних и тех же наборов чисел, но массив a5 у нас в отличие от массива a2, расположен по своему возрастанию. Т.е. отсортирован.

Начинается вот с единицы и кончается именно на том, значении которое равно 100. В отличие от массива a2, где все эти элементы расположены так сказать в рабочем беспорядке. Закроем окно нашего приложения и опять вернемся в наш текстовый редактор. В языке ***Java*** для массивов существует также возможность поиска номера элемента с определенным значением. Например, напишем таким образом. Скопируем вот эти три строчки. Щелкнем правой кнопкой мыши, Copy. Затем вставим их сюда, ***Paste***. Следующая наша задача будет опять отделена вот таким прочерком. И далее напишем таким образом. На этот раз ***i***1 возьмем от 1 до 10. И будем искать вот эти числа от 1 до 10, т.е. ***i***1, присутствуют ли они в массиве a2.

Для этого вот это все уберем, и напишем так. ***Arrays***, поскольку этот метод сортировки есть именно в этом классе, точка. Далее ***binarySearch***. Далее откроем скобку. Теперь нам нужно указать массив, в котором мы будем искать. Пусть это будет массив a2, запятая и далее, что мы будем искать. Искать мы будем элемент с значением ***i***1. Закроем скобку. А теперь скомпилируем программу. Развернем меню ***Tools***, ***Compile Java***. Компиляция прошла успешно. Теперь вновь меню ***Tools***, ***Run Java Application*** и можно увидеть, что у нас получилось. Как мы видим, выводится номер элемента, значение которого равно вот этим числам. А если такого элемента нет, тогда отображается отрицательное число. Вот, как мы видим, в массиве a2, вот это у нас массив a2 был распечатан в предыдущем примере. Элемента со значением 0, нет. А элемент с значением 1, есть. И у этого элемента, со значением 1, индекс 0. Что мы тут и получили.

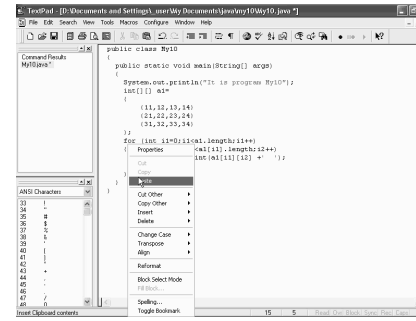
А элемент, значение которого равно 2, его номер у нас единица, что у нас и отображается. Элемента в массиве a2, который равнялся бы 3, нет, а элемент, который равнялся бы 4, у нас есть, у него индекс 3. Вот тройку мы здесь и получаем. И далее, элементы со значением 5, 6, 7, 8 и 9 отсутствуют. Тут таких элементов нет. И в результате здесь все отрицательные числа. Закроем теперь окно нашей программы.

6.7.4.11. Многомерные массивы. Создание. Инициализация

Таблицы и другие более сложные упорядоченные структуры данных, тогда нам не обойтись без многомерных массивов, которым для доступа к элементам массива нужно применить несколько индексов. Для того чтобы рассмотреть

вопросы, связанные с многомерными массивами, создадим шаблон нового приложения, рассмотрим, как всегда, вот этот простейший шаблон, который выводит на консольное окно одну строчку. Сохраним его в нашем жестком диске. Для этого щелкнем на кнопку сохранения, создадим новую папку для него. Пусть это будет ***my***10. Открыть. Введем новое имя для нашего приложения. Как мы помним, оно должно совпадать с именем класса ***My***10 и щелкнем на кнопке Сохранить. И вот, после этих предварительных действий, мы можем заняться массивами. И первое, что нам необходимо, конечно же, сделать это, проинициализировать многомерный массив.

Ну, например, это можно сделать таким образом. Создадим, например, двумерный массив, состоящий из целых чисел. Для этого нам нужно определить тип, т.е. написать ***int***, далее квадратные скобки. Причем, если мы напишем одну пару квадратных скобок, то это, конечно же, будет одномерный массив, а если напишем две пары квадратных скобок, то мы, таким образом, проинициализировали, двумерный массив. Пусть этот двумерный массив имеет имя a1 и определим его прямым образом. Для этого можем написать внутри фигурных скобок элементы этого массива. Ну, например, таким образом, 11,12,13,14 — это у нас первый ряд этого массива, затем следующий, пусть это будет 21,22,23,24 и т.д. 31,32,33 и 34. И закроем фигурную скобку. Точка с запятой. Вот, таким образом мы проинициализировали многомерный массив, размером 3x4. Это у нас, двумерный массив.

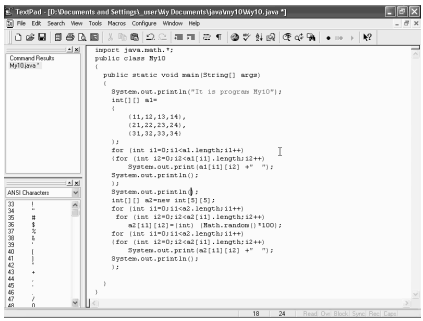


Попробуем его теперь распечатать на нашем консольном окне. Для этого нам, конечно же, потребуется оператор цикла. Напишем таким образом, ***for***, ***int*** ***i***1, который меняется от 0, далее точка с запятой, и от 0 до ***i***1 меньше, чем длина нашего массива, так сказать, по вертикали, т.е. вот это наружное. Потом напишем a1, точка ***length***. Точка с запятой. И далее ***i***1++, чтобы на каждом шагу было приращение на единицу. Далее нужно закрыть скобку. А теперь фигурные скобки, внутри которых напишем еще один цикл ***for***, на этот раз возьмем опять переменную тип ***int***, целую. И пусть это будет ***i***2, которая меняется от 0 до ***i***2 меньшем, чем, на этот раз, возьмем длину нашего массива, так сказать, в другом направлении.

Поэтому возьмем a1, возьмем его элемент под номером ***i***1 и на этот раз используем именно его длину. ***length***, точка с запятой, ***i***2++ опять, приращение

на единицу. Закроем скобку. И теперь воспользуемся функцией печати. Для этого нам нужно использовать оператор ***println***, который скопируем из этого места. Щелкнем правой кнопкой мыши, Copy. Вставим сюда. Правая кнопка мыши, ***Paste***. Поскольку кроме печати нам ничего не нужно, фигурные скобки вставлять не обязательно. Теперь же напечатаем то, что мы хотим распечатать. Именно вот эти числа из массива a1. a1, ***i***1 первый индекс и ***i***2 второй индекс. Далее плюс. И введем несколько пробелов, для того чтобы эти числа у нас не прилипли друг к другу. Закроем скобку. Точка с запятой. Кроме этого, воспользуемся лучше не оператором ***println***, а просто ***print***, для того чтобы элементы внутреннего массива, у нас отображались на одной строке. А между циклами наружного индекса, индекса ***i***1 сделаем перевод строки.

Поэтому щелкнем здесь правой кнопкой мыши, ***Paste***. Вставим этот оператор и закроем скобки. Здесь нам нужен только перевод строки. Теперь исправим еще одну ошибку, которая у нас закралась. Между элементами массива при определении нам нужно было вставить запятые в этих местах, для того чтобы вот эти три элемента отделялись друг от друга. Исправим также одинарные кавычки на двойные, поскольку для вставки текста при распечатке и вообще при использовании, нам нужно использовать именно двойные кавычки. Теперь скомпилируем нашу программу. Для этого развернем меню ***Tools***, ***Compile Java***. Далее ***Run Java Application***. И можно видеть результат выполнения нашей программы. Мы вот эту нашу матрицу, наш двумерный массив распечатали на консольном окне. Закроем его. Вернемся в наш текстовый редактор.



Создадим теперь еще один массив. Для этого, пусть это будет опять массив целых чисел ***int***. Опять будет двумерный, поэтому вот эту пару квадратных скобок напишем два раза. Пусть эта переменная будет ***a***2, которую определим при помощи оператора ***new***. ***new***, далее ***int*** и определим размерность этого массива. Ну, пусть это будет на этот раз просто 5x5. Точка с запятой. Теперь заполним, какими-либо значениями вот этот массив. Для этого опять воспользуемся вот этим двойным циклом. Скопируем вот эти две строчки. Выделим их. Правая кнопка мыши. Copy. Теперь вставим сюда. Правая кнопка мыши и ***Paste***. Теперь нам немножко это все надо переделать. Вместо ***a***1 написать ***a***2. И в первом цикле и во втором. Теперь напишем определение самой переменной. Вот эта фигурная скобка нам не нужна. Далее ***Enter***. И далее напишем таким образом. ***a***2 от ***i***1, это первый индекс. И внутри квадратных скобок ***i***2, это

второй индекс, пусть равняется случайному числу, которое будет лежать в пределах от 0 до 100.

Для этого нам нужно написать следующую строку. ***int***, это число нам нужно перевести к типу ***integer***. Далее ***Math***, точка, ***random***. Далее скобки и умножить на 100, поскольку число мы хотим взять от 0 до 100. Дело в том, что вот этот оператор ***Math.random*** выдает нам случайные действительные числа от 0 до 1. А при умножении на 100, мы получаем соответственно от 0 до 100. А приведение к целому типу выдает целое число соответствующего диапазона от 0 до 100. Для того чтобы мы могли пользоваться вот этим классом ***Math***, конечно же, нам нужно сначала провести его импорт. Поэтому напишем ***import***, далее ***java***, точка ***Math***, звездочка. Точка с запятой. И теперь попробуем многомерный массив ***a***2 вывести на печать. Для этого опять воспользуемся вот этой готовой заготовкой. Выделим вот эти несколько строчек.

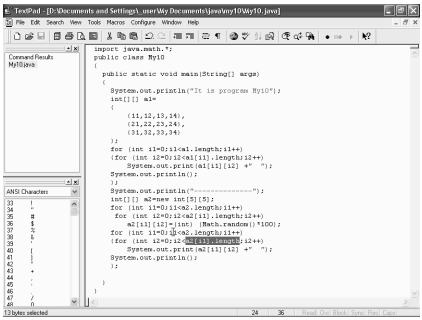
Далее правая кнопка мыши и Copy. И вставим в это место. Вставим сюда. Правая кнопка и ***Paste***. И теперь везде ***a***1 поменяем на ***a***2. В этом месте, в этом и соответственно вот в этом месте. Единственное, чтобы вот эти первая и вторая маленькие подзадачи у нас не путались между собой вставим между ними еще один оператор ***println***. Выделим вот этот фрагмент текста.

Щелкнем правой кнопкой мыши, Copy. Далее вставим сюда. Правая кнопка и ***Paste***. И введем опять разделительную линию. Внутри кавычек напишем несколько минусов. Вот таким образом. Теперь развернем меню ***Tools***. ***Compile Java*** — скомпилируем. Компиляция прошла успешно, и поэтому запустим наше приложение. ***Run Java Application***. И вот можно видеть результат выполнения нашей программы. Мы получили матрицу 5x5, состоящую из случайных чисел. Закроем это консольное окно. Щелкнем на вот этом крестике. И выйдем в наш обычный текстовый редактор.

6.7.4.12. Многомерные неровные массивы

Рассмотрим теперь одну особенность тех строчек кода, которые мы только что написали. А именно вот в этих двумерных циклах мы каждый раз проверяли длину, так сказать, этой строчки массива ***A***. Хотя в принципе, независимо от вот этой переменной ***i***1, эта длина у нас все время одна и та же, поскольку у нас размерность массива 5x5. Почему это так сделано? Дело в том, что в языке ***Java***, в отличие от других языков программирования, могут существовать так называемые неровные массивы. Т.е. массивы, у которых так сказать количество элементов в строке разное. Это все происходит оттого, что в языке ***Java*** на самом деле существуют только одномерные массивы, а двумерные массивы являются имитацией тех массивов. Просто-напросто многомерные массивы имитируются, как массивы массивов.

Например, возьмем вот этот массив 5x5, который мы только что разбирали и превратим его в неровный массив. Для этого скопируем вот эти строчки.



Выделим их. Щелкнем правой кнопкой мыши, Copy. Теперь вставим сюда. Правая кнопка мыши и ***Paste***. И теперь пусть этот массив у нас называется ***a***3, и зададим мы для начала только одну его размерность. Пусть это будет 5 элементов, а по второму направлению, размерность пока задавать не будем. Далее в следующей строчке, где у нас начинается цикл, определим цикл, конечно же, массива ***a***3. Далее напишем здесь фигурные скобки, внутри которых и будет находиться задание нашего массива. Внутренний следующий цикл. А внутри этих фигурных скобок, до следующего цикла ***for*** зададим и длину этого массива по другому направлению. Теперь определим размерность массива ***a***3 по другому направлению. Поэтому напишем таким образом. ***a***3 от ***i***1 равняется ***new int*** и затем размер массива, который нам нужно написать внутри квадратных скобок. Размер массива опять возьмем случайным образом, вот этим оператором ***Math*** и ***random***. Поэтому скопируем вот этот фрагмент текста, выделим его, щелкнем правой кнопкой мыши, Copy.

Далее вставим сюда. Правая кнопка и ***Paste***. Единственное, возьмем размерность этих случайных чисел не 100, а меньше. Например, возьмем до 8, чтобы у нас все это поместилось в одну строчку. Далее закроем квадратную скобку. Точка с запятой. Теперь нам нужно переделывать каждый раз ***a***2 на ***a***3 во всех дальнейших строчках. Вот это у нас была инициализация вот этого массива. А в следующих строчках у нас происходила его распечатка. Поэтому здесь тоже напишем ***a***3. Вот таким образом у нас уже исправлены все строчки кода. В конце каждой строки выведем также размерность. Вот эту величину. Скопируем ее. Правая кнопка, Copy. Вставим сюда. Правая кнопка и ***Paste***. Единственное, перед этой информацией, поместим еще в кавычках следующий текст. Напишем просто, ***n*** и знак равнения.

И попробуем выполнить и скомпилировать эту программу. Для этого развернем меню ***Tools***. Далее ***Compile Java***. Теперь, после того, как компиляция прошла успешно, вновь развернем меню ***Tools*** и ***Run Java Application***. И вот можно видеть, что у нас получилось. У нас получился вот такой абсолютно неравномерный массив. Первая строчка у нас состоит из 4 элементов. Вторая строчка из 7. Третья тоже из 7.

И последние две строчки — по одному элементу. Закроем теперь это консольное окно и вернемся в наш текстовый редактор. Неровные массивы можно определить не только вот таким образом, как мы только что делали, но

и при помощи явного определения. Так, как это делали вот в этих строчках. Попробуем это сделать. Определим для разнообразия на этот раз не массив из целых чисел, а, например, массив из переменных типа ***char***, т.е. символов.

Напишем ***char***, далее две пары квадратных скобок, поскольку опять возьмем многомерный массив и определим имя этого массива. Ну, пусть будет ***a***4, и далее напишем таким образом. Знак равнения. И теперь внутри фигурных скобок нам нужно написать элементы этого массива. Пусть это будет таким образом. Возьмем буквы, которые расположены у нас на первой строчке нашей клавиатуры. Это будет буква ***q***, затем буква ***w***, и далее по порядку. Возьмем первые несколько букв. Вот первую строчку из этих букв мы уже ввели. Теперь запятая, введем следующую строчку. Опять фигурные скобки и продолжим тем же способом. Закроем фигурную скобку. Опять запятая и следующая последняя, третья строчка, которая будет состоять из таких букв. Эта буква ***z***, далее ***x***, ***c*** и т.д. Закроем фигурную скобку, закроем также вторую фигурную скобку. Точка с запятой.

Теперь попробуем это вывести на экран, для того чтобы у нас было полное представление о том, что же мы здесь ввели. Поэтому скопируем вот эти строчки, выделим их. Щелкнем правой кнопкой мыши и Copy. Далее встанем сюда. Правая кнопка мыши и ***Paste***. И теперь переправим, конечно же, ***a***3 на ***a***4 во всех местах, где это у нас встречается. Все остальные действия, которые здесь есть, в принципе, вполне пригодны. Теперь попробуем посмотреть, что же у нас получилось.

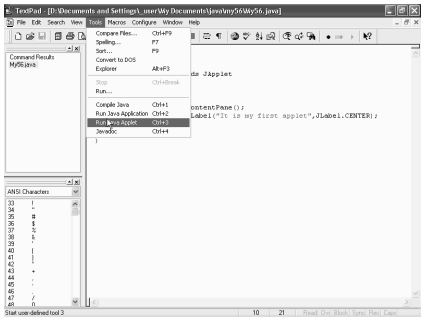
Единственное, опять-таки, чтобы отделить одну задачу от другой, вставим между ними вот эту строчку. Выделим ее. Правая кнопка мыши и Copy. Вставим сюда. Правая кнопка мыши и ***Paste***. Теперь скомпилируем эту программу. Для этого развернем меню ***Tools***. Далее ***Compile Java***. Далее опять меню ***Tools*** и ***Run Java Application***. И вот можно увидеть, что у нас получилось. У нас получился вот такой неравномерный массив, в первой строчке которого у нас 10 элементов, начиная с буквы ***q*** и кончая буквой ***p***, во второй строке у нас 9 элементов от ***a*** до ***l***, и в третьей 7 элементов от ***z*** до ***m***, так как это все у нас расположено на контактной клавиатуре. Закроем это приложение. Щелкнем на вот этом крестике и выйдем в наш обычный текстовый редактор.

6.7.5. Апплеты

6.7.5.1. Создание простого апплета

Все рассматриваемые нам раньше программы и приложения, никакого отношения к ***html*** страницам и к браузерам, которые работают с Интернетом, не имели. Но особую популярность язык ***Java*** приобрел, как язык, который помогает "оживить" ***Internet***. Дело в том, что на языке ***Java*** можно создавать

особые программы, которые называются аплетами, и которые загружаются из Интернета и выполняются браузерами. И в этом случае программа, написанные на языке ***Java*** становится как бы частью ***html*** страницы, которая для этого выделяет некий прямоугольник на своей странице. Для того чтобы на ней отображалось все то, что мы обычно выводим в нашем фрейме, написанном на языке ***Java***. Попробуем создать такой аплет самостоятельно.



Первое, что для этого нужно сделать, это загрузить при помощи директивы ***import*** пакеты ***awt*** и ***swing***. Напишем таким образом: ***import***, далее ***java***, точка ***awt***, точка звездочка, точка с запятой.

Теперь опять введем еще одну директиву ***import***, на этот раз ***javax***, точка ***swing***, точка звездочка, точка с запятой. Теперь же, после того, как мы эти написали импорты, необходимо написать наш основной класс, с которого будет начинаться работа нашего аплета. Ну, пусть это будет, например, таким образом, ***public***, далее ***class***, и далее введем имя для нашего класса. Ну, например, пусть это будет вот такое имя, My56. И далее этот класс должен поддерживать в качестве родительского класса, класс ***JApplet***. Поэтому напишем ключевое слово ***extends*** и далее ***JApplet***. Щелкнем на клавише ***Enter***.

Введем фигурные скобки, внутри которых мы должны написать описание всего нашего класса. Первое, и необходимое, что нам нужно ввести в нашем аплете, это метод ***init***, с которого и начинается работа всех аплетов, так же, как и с метода ***main*** начиналась работа всех консольных программ. А для аплетов, как мы еще раз сказали, это метод ***init***.

Поэтому опять напишем ***public***. Тип возвращаемого значения ***void***, поскольку никакого возвращаемого значения тут не предусмотрено и далее метод ***init***, инициализация. Далее введем скобки. ***Enter***, фигурные скобки, внутри которых мы теперь и должны написать какой-либо код для нашего аплета. Ну, самое простое, возьмем и просто-напросто на нашем аплете, на этом прямоугольнике, внутри ***html*** страницы, поместим какую-либо метку. В принципе с аплетом можно обращаться точно так же, как мы ранее обращались с и фреймом. Единственная разница в том, что задавать размеры нам здесь не нужно, поскольку размеры уже задаются внутри ***html*** страницы.

Поэтому первое, что нам нужно сделать, это создать контейнер. Для этого просто напишем ***Container***, далее, пусть он называется ***pane***, далее, как обычно, равняется и ***getContentPane***. Далее скобки, точка с запятой. Теперь

же, после того, как мы получили контейнер, поместим на нем, как мы сказали метку. Напишем, как обычно ***JLabel***. Далее, пусть эта переменная называется ***label***. Ну, как обычно введем знак равнения, ***new***, опять ***JLabel***. Нам нужен конструктор этой метки. Далее откроем скобку. И теперь внутри скобок напишем тот текст, который должен появляться на этой метке. Напишем просто-напросто "***It is my first applet***". Закроем кавычки, далее запятая. И теперь укажем центральное расположение этой метки. Для этого напишем ***JLabel***, точка, и ***CENTER***. Вот теперь закроем скобку. Точка с запятой.

Ну и теперь эту метку нам надо добавить на наш контейнер. Сделаем это, как обычно: ***pane***, точка ***add***, откроем скобку и добавить нашу метку ***label***. Закроем скобку. Точка с запятой. Вот у нас уже полностью готово все то, что должно определять наш аплет. Попробуем теперь его сохранить в каком-либо файле. Для этого щелкнем на кнопке сохранения. Создадим для этого проекта новую папку. Щелкнем вот на этой кнопке. Напишем таким образом. Щелкнем на кнопке ***Open***. Введем имя для нашего файла. Пусть оно будет вот таким ***My***56, и щелкнем на кнопке ***Save***, предварительно изменив тип нашего файла, который, как мы видим, сейчас стал ***html***. А нам нужно, конечно же, чтобы это был файл с расширением ***java***. Щелкнем теперь на кнопке ***Save*** (сохранить). И вот мы опять в нашем текстовом редакторе.

Теперь же, попробуем посмотреть, как же работает это приложение. Ну, для начала нам нужно его скомпилировать. Для этого скомпилируем его, как обычно. Развернем меню ***Tools***, далее ***Compile Java***. Компиляция прошла успешно. А теперь же попробуем посмотреть этот аплет. Для этого развернем меню ***Tools*** и здесь, в этом меню, у нас есть специальная команда ***Run Java Applet***, которая на самом деле оперирует специальные программы ***Applet Viewer***, которые входят в стандартную поставку ***Java***. Щелкнем на этой команде. И вот, мы можем видеть результат выполнения программы ***Applet Viewer***, внутри которой загрузился наш аплет и, конечно же, вот эта надпись на его поверхности. В принципе программа ***Applet Viewer*** просто-напросто, так сказать, стимулирует работу браузера, показывает именно тот кусочек, который относится к данному аплету.

Закроем теперь это окно. Щелкнем вот на этом крестике. И попробуем запустить его еще раз. Для этого вновь развернем меню ***Tools***, далее опять ***Run Java Applet*** и, как мы видим, на этот раз у нас появилось другое окно. А именно, окно, в котором указывается ***html*** файл. Этот файл находится у нас в нашей текущей директории, в котором как раз и находится ссылка на наш аплет. Если мы сейчас щелкнем на кнопке ***Yes***, то, конечно же, в принципе мы получаем все то же самое.

Дело в том, что вот этот файл, который мы только что видели, это тот файл, который текстовый редактор ***TextPad***, создал внутри нашей директории.

Закроем теперь это приложение. И теперь попробуем рассмотреть этот ***html*** файл, и из чего он состоит. Для этого щелкнем на кнопке открытия. Далее переключимся на открытие ***html*** файлов. Щелкнем на вот этом треугольнике. Выберем ***html*** файлы. И вот этот временный ***html*** файл, который только что и был автоматически создан. Щелкнем на кнопке ***Open***, чтобы рассмотреть его. И вот, можно видеть из чего он состоит.

В принципе, все кто знакомы с тегами ***html*** страниц и с языком их написания, могут видеть, что это просто-напросто самая простая страница. Она только существует и в которой вот эта вторая строчка, эта именно та строка, которая касается нашего аплета. В начале создается заголовок, тело и т.д. Всего этого просто-напросто нет. И есть просто-напросто один аплет, который задается ключевым словом ***Applet***. Далее указывается название аплета. Вот этот My56 точка ***class***. И задается ширина и высота прямоугольника, внутри которого этот аплет должен отображаться. Попробуем теперь рассмотреть, как вот эта ***html*** страница выглядит, так сказать, в ***Internet Explorer***-е.

Для этого щелкнем на пункте меню ***View*** и далее ***In Web Browser***. Но, прежде чем мы запустим просмотр этой страницы в ***Internet Explorer***-е, лучше всего, конечно же, сохранить его под каким-нибудь именем, учитывая, что это в принципе, временный файл. Поэтому развернем меню ***File***, далее команда ***Save As***. Выберем имя для нашего приложения. Ну, назовем просто ***Myhtm*** и щелкнем на кнопке ***Save*** (сохранить). А теперь, как мы сказали, развернем меню ***View*** и далее выберем команду ***In Web Browser***. Щелкнем на ней. Вот теперь загрузился наш ***Internet Explorer***, внутри которого, как мы видим, отображается серый прямоугольник. Это и есть наш аплет, внутри которого и размещается наша надпись.

Закроем теперь его. Щелкнем на вот этом крестике. И вот мы опять в нашем текстовом редакторе, в котором надо отметить, что тот же самый запуск показа в ***Explorer***-е можно создать, нажав на вот эту пиктограмму.

Теперь попробуем немножко приукрасить нашу ***html*** страницу. Введем, например, какой-либо заголовок. Для этого щелкнем здесь на клавише ***Enter***. И далее, чтобы нам было проще писать ***html*** теги, из которых и состоит наша ***html*** страница. Щелкнем здесь вот на этом треугольнике и перейдем на систему ***html*** тегов. Они у нас появились вот в этом левом нижнем углу нашего экрана. Сдвинемся теперь ниже. Выберем какой-либо шрифт. Например, вот этот наклонный, ***Italic***. Щелкнем здесь дважды.

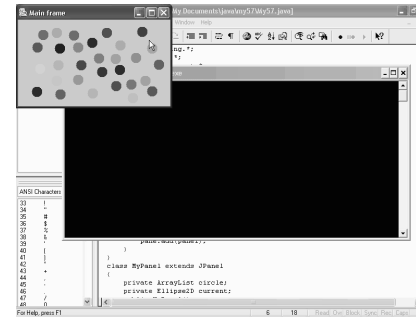
Вот, теперь появилась возможность написания наклонного шрифта. Сдвинемся еще ниже и напишем этот заголовок, как заголовок второго уровня. Щелкнем здесь дважды. И теперь введем здесь какой-либо текст. Например, тот же самый, который мы написали в нашем аплете "***It is my first applet***".

Теперь введем здесь знак перевода строки и посмотрим, как теперь все это выглядит в нашем браузере. Щелкнем на вот этой кнопке, и мы можем видеть результат работы нашей ***html*** страницы внутри браузера. Вот наверху, это у нас простой заголовок, это все написано, так сказать, средствами ***html*** страницы, тегами.

А вот в этом прямоугольнике у нас находится наш аплет, в котором в принципе тот же самый текст. Пока это не слишком впечатляет, но мы, наверное, уже помним, что внутри нашего аплета мы в принципе можем сделать практически все, что угодно. Поместить кнопки, разные компоненты, элементы, в том числе и обработчики событий, любую графику и практически все то, что можно выполнять при помощи обычных приложений. Т.е. вот этот аплет будет вполне "живой", в отличие от практически статического текста, из которого состоит ***html*** страница. Закроем теперь наше приложение. И после того, как мы закрыли наш браузер, мы опять попали в наш текстовый редактор.

6.7.5.2. Преобразование приложения в апплет

Рассмотрим теперь, каким образом, если у нас уже есть приложение ***Java***, написанное нами ранее, которое, как обычно, у нас состоит из главного класса, в котором есть метод ***main***, т.е. консольное приложение, каким образом его превратить в аплет. Для этого начнем работу с вот этого приложения, которое мы создавали ранее и при помощи которого мы, щелкая на кнопку мыши, можем рисовать на поверхности нашего аплета разного рода круги. Сохраним этот листинг в каком-либо файле. Переделаем этот класс. Напишем вместо My35, например, My57, по порядку следования наших файлов и сохраним его в соответствующем месте.



Щелкнем на кнопке сохранения. Создадим для этого проекта новую папку. Назовем ее таким же именем. Щелкнем на кнопку ***Open***. Теперь введем имя для нашего файла. Конечно, оно должно повторять имя нашего главного класса. Поэтому оно должно быть вот таким и щелкнем на кнопку ***Save***, конечно же, выбрав предварительно расширение ***Java***. ***Save***, теперь мы его сохранили, Теперь скомпилируем и посмотрим, как она выглядит. Для этого конечно же, сперва надо скомпилировать ее. Развернем меню ***Tools***, выберем команду ***Compile Java***. Компиляция прошла успешно. Теперь запустим наше приложение. Вновь меню ***Tools***, далее ***Run Java Application***. Вот это окно появилось перед нами и,

если мы будет щелкать на нем, то в соответствующем месте, там, где мы щелкаем, появляются разноцветные кружочки, которые в принципе мы можем двигать.

Например, мы можем схватить вот этот кружочек, потащить его в какое-либо другое место. Можем какой-либо кружочек удалить, например, щелкнув дважды, вот этот кружочек удалился. Можем перетащить кружочки еще куда-нибудь. Можем щелкнуть и добавить кружочки. В общем вот таким образом, мы можем расцветить наше окно. Вот этот удалим. Ну, вот так и работает наше приложение. Мы его уже ранее создавали. Ну, в принципе, конечно, прямо видно, что это самое обычное консольное приложение, какие мы ранее написали довольно много. Теперь же попробуем его переделать в аплет. Закроем это приложение.

Консольное окно тоже закроем. И мы опять в нашем текстовом редакторе. Теперь же, для того чтобы переделать все это наше приложение в аплет, сделаем следующие процедуры. Ну, для начала, нашему классу My57 напишем, что он является наследником класс ***Applets***, поэтому напишем ***extends*** и далее ***JApplet***, далее этот метод ***main*** нам абсолютно не нужен, поэтому поставим перед ним и перед всеми его строчками комментарии. Также совершенно излишним является вот этот класс ***MyFrame***.

Поэтому напишем таким образом, косая черта и далее звездочка, и в конце этого класса звездочка и закрытые. Опять косая черта. Теперь вот это все у нас закомментировано. Теперь нам опять придется позаботиться о нашем классе My57, который на данном этапе является, так сказать, совершенно пустым. Поэтому щелкнем здесь на клавише ***Enter*** и далее напишем метод ***init***, который обязательно должен быть внутри аплета.

Напишем, конечно же, ***public***, далее тип возвращаемого значения ***void*** и наш метод ***init***, инициализация. Далее скобки. ***Enter***. Фигурные скобки, внутри которых поместим вот эти три строчки, в которых мы внутри нашего класса ***MyFrame*** создавали нашу панель, на основе класса ***MyPanel***. Поэтому выделим эти три строчки. Далее щелкнем правой кнопкой мыши и Copy. Встанем сюда. Правая кнопка мыши и ***Paste***. Как мы видим, из всех строчек создания нашего фрейма ***MyFrame***, мы скопировали вот эти три строчки, где у нас создается контейнер, панель и т.д., а вот строчки, в которых задается титульная строка и строчка, в которой задаются размеры нашего фрейма, у нас здесь не используется. Что и понятно. Поскольку размеры окна задаются внутри браузера. Ну а титульная строка, там вообще абсолютно лишняя. Титульная строка у ***html*** страницы своя. И поэтому вот эта строчка является лишней.

Теперь же попробуем запустить наш аплет. Для начала, нам, конечно же, нужно скомпилировать его. Для этого развернем меню ***Tools***, далее ***Compile***

Java. Компиляция прошла успешно. Теперь запустим наш аплет. Для этого вновь развернем меню ***Tools***. Далее ***Run Java Applet***. И вот аплет появился перед нами. Если мы сейчас будет щелкать на его поверхности, то, как мы видим, мы рисуем на нем наши кружочки. Можем опять схватить любой из них, передвинуть, т.е. наш аплет работает так, как и должен работать по своей логике создания.

А теперь же закроем его. Щелкнем на вот этом крестике. Найдем ***html*** страницу, которую создал наш текстовый редактор. Для этого щелкнем на кнопке открытия. Переключимся теперь на ***html*** страницы и выберем вот этот ***html*** файл. Щелкнем на нем дважды. Вот, видна строка, которая содержит описание нашего аплета. Ну, в принципе дальнейшим переделыванием этой страницы мы заниматься не будем. Просто-напросто сохраним ее под каким-либо именем. Для этого развернем меню ***File***, далее ***Save As***.

Введем для него имя ***My_applet*** и щелкнем на кнопке ***Save*** (сохранить). Теперь же, щелкнем на вот этой кнопке запуска просмотра в ***web*** браузере. И вот, наша ***html*** страница вместе с аплетом, появилась перед нами. Ну, опять-таки мы можем щелкать и на поверхности этого аплета, который уже находится внутри нашей ***html*** страницы, появляются все наши кружочки. Мы можем их двигать, удалять, т.е. все работает совершенно идеально. Закроем теперь нашу ***html*** страницу. И переключимся опять на наш файл ***Java***.

Теперь же обратим внимание на все те строчки, которые мы закомментировали. Они сейчас, конечно же, в принципе не работают. И если мы хотим оставить это приложение только в таком виде, как некий аплет, который запускается только внутри нашей ***html*** страницы, то на этом в принципе можно и поставить точку. Но, конечно же, в принципе можно сделать так, чтобы это ***Java*** приложение запускалось бы и как аплет и как отдельно стоящее приложение с консольным окном. Для этого в принципе достаточно просто-напросто удалить комментарии со всех мест, в которые мы их вставили. При этом метод ***main*** станем просто-напросто методом вот этого класса My57. То, что он является наследником класса ***JApplet***, делу абсолютно не мешает. А класс ***MyFrame***, который нужен для создания соответствующего окна, тоже будет присутствовать.

Попробуем это сделать. Удалим вот эти комментарии. Теперь удалим комментарии с нашего класса ***MyFrame***. Теперь же, попробуем опять все это скомпилировать. Для этого развернем меню ***Tools***, далее ***Compile Java***. Компиляция прошла вполне успешно. Теперь посмотрим, как она работает в режиме приложения. ***Tools***, далее ***Run Java Application***. И вот, наше приложение появилось перед нами. Мы опять можем наводить на него кружочки. Все в принципе работает прекрасно, и мы можем видеть, так

сказать, типичное консольное приложение, с отдельно стоящим фреймом. Закроем теперь его. Закроем и консольное окно тоже.

Теперь переключимся на нашу ***html*** страницу. Вот эту, ***My_applet***. И попробуем опять запустить уже вот эту ***html*** страницу. И посмотрим, как все это работает в ней. Щелкнем на вот этой кнопке для просмотра этой страницы в ***web*** браузере. Как мы видим, наш аплет тоже появился на нашей ***html*** странице и тоже выглядит совершенно нормальным. Т.е. можно видеть, что у нас один листинг ***Java***, так сказать, обслуживает две цели. Запускает нашу программу и в режиме аплета и в режиме консольного приложения.

В каких-то случаях это иногда бывает полезно. Особенно для отладки наших приложений. Поскольку, если мы запускаем наше приложение в режиме консольного окна, то мы в каких-то местах можем вставлять какие-либо ***println***-ы, при помощи которого промежуточные результаты можем выводить на наши консольные окна и контролировать выполнение кода нашей программы. Ну, а затем, когда все у нас будет готово, мы можем запустить его уже в ***Web*** браузере, как аплет и все у нас после этого будет выполняться идеально. Закроем теперь нашу ***Web*** страницу и мы опять в нашем текстовом редакторе.

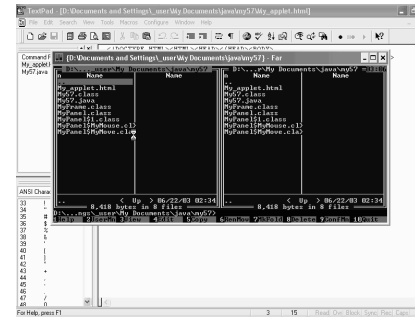
*6.7.5.3. Архивы **jar***

Щелкнем на значке открытия файла. И в типе файлов переключимся на все файлы, ***All files***, и можно видеть все файлы, которые у нас находятся в текущей директории. Среди них, конечно же, все классы, которые находятся в нашем проекте. Как мы видим, их достаточно много. А именно 6 штук, что необходимо для выполнения и нашего ***Java*** проекта и в качестве приложения, и в качестве аплета. Конечно же, надо учитывать, что если мы пересылаем эти все файлы по Интернету, то каждый файл требует отдельной загрузки.

Отдельного связывания для своей пересылки. Что, конечно же, увеличивает время, которое нам необходимо для полной пересылки и загрузки нашего ресурса. Именно для этих целей, в языке ***Java*** есть специальный прием, который называется Упаковка в ***Jar*** файлы. Т.е., вот эти все файлы, с расширением ***Class*** и не только они, вообще все файлы, которые необходимы для работы нашего приложения или нашего аплета, можно упаковать в один ***Jar*** файл, который на самом деле является обычным архивов.

В связи с этим, он занимает сам по себе гораздо меньше места, ну и представляет собой, конечно же, один файл. Это, гораздо лучше, чем вот эти 6 файлов, которые у нас есть в нашем случае. Попробуем сейчас упаковать все эти наши файлы, которые нам необходимы для работы в наш архив. Откроем это окно. Щелкнем на кнопку ***Cancel***. Мы опять в нашем текстовом редакторе. И вот перед нами файловый редактор, в котором можно видеть всю нашу директорию, и как она выглядит. А для того чтобы теперь упаковать нам все

эти классы в один файл, достаточно набрать в командной строке следующую строку, ***Java*** имя нашего файла, который занимается упаковками и который поставляется вместе со стандартной поставкой ***Java***.



Далее укажем несколько ключей C, B и ***F***, которые означают, что опция C означает, что создается новый или пустой архив и добавляется в него файлы. B — означает, что генерируется отчет о файле, а ***F*** — что в качестве второго аргумента у нас будет имя ***Jar*** файла. Напишем таким образом. Пусть он называется просто-напросто ***My*** точка ***Jar***. Далее нам нужно указать, что мы хотим упаковать, а именно, упакуем все файлы с классами. Для этого напишем так, звездочка точка и класс. Щелкнем теперь на клавише ***Enter***. Можно увидеть вот этот новый файл ***My***.***Jar***, который создан для нас и в котором сейчас у нас запакованы все наши файлы с классами. Можем щелкнуть на кнопку ***Ctrl F***1 и ***Ctrl F***2, для того, чтобы очистить окно и можем увидеть, как выполнялась наша командная строка ***Jar***. Можно увидеть подробно, как у нас были упакованы отдельные файлы и, что из этого получилось. Теперь же, щелкнем на ***Ctrl F***1 и ***F***2 опять, чтобы появились вот эти окна с файлами. И теперь для чистоты эксперимента откроем внутри этого каталога, новый каталог. Для этого щелкнем на клавише ***F***7, назовем этот каталог как-нибудь. Ну, пусть это будет просто ***My***. Щелкнем на клавише ***Enter*** и внутрь этого каталога скопируем всего лишь два файла.

Это файл с архивом ***My***.***Jar*** и наш ***html*** файл, в котором находится у нас наша ***html*** страница. Выделим их. Далее щелкнем на клавише ***F***5, ***Enter***. Вот эти два файла скопируем и теперь попробуем только при помощи этих двух файлов запустить наш аплет. Но, сначала, нам придется отредактировать вот этот ***html*** файл. Щелкнем на кнопку ***F***4 для этого. Мы открыли этот файл в режиме редактирования и далее впишем здесь такой текст. ***ARCIVE***, далее равняется, кавычки, внутри которых напишем имя нашего архива, который, как мы помним, называется ***My*** точка ***Jar***.

Теперь сохраним этот файл. Для этого щелкнем на клавише ***Save***, на ***F***2. Закроем теперь этот файл. Вот мы теперь опять в нашем файловом менеджере. Теперь же запустим наш ***html*** файл. Для этого достаточно, просто-напросто дважды щелкнуть на вот этой строчке. Можно увидеть, что у нас получилось. А именно, запустился наш аплет, который, конечно же, выполняет все те действия, которые мы в него заложили. А именно, конечно же, рисует эти

кружочки. Причем, как мы можем видеть, мы добились этого всего лишь двумя файлами. Один файл это, конечно же, ***html*** файл. И второй файл это файл с аплетом. Закроем теперь эту ***html*** страницу.

Теперь же попробуем сделать так, чтобы вот этот упакованный архив ***My***.***Jar*** запускался и в режиме программы, а не в режиме аплета. Как мы помним, у нас эта возможность есть, поскольку в нашем листинге ***Java*** в нашем основном классе, мы писали оба метода, которые нам нужны. И метод ***Init***, который нам нужен, для того чтобы это был аплет и метод ***main***, который нам нужен, для того чтобы у нас запускалась бы консольная программа. Но в этом случае нам придется написать еще один файл для того, чтобы все работало хорошо. А именно, нам нужно создать файл манифеста, в котором мы должны написать название нашего главного класса. Создадим его. Создадим новый файл. Для этого щелкнем на шрифт ***F***4, теперь наберем имя для нашего файла, пусть это будет Манифест. Точка. ***MF***.

Enter. Теперь нам нужно набрать то, что здесь должно быть. А именно, ***MyClass***, главный класс. Далее двоеточие, пробел и далее название нашего главного класса. Это, конечно же, ***My***57. Теперь отметим еще одно обстоятельство, а именно, нам обязательно надо щелкнуть здесь на клавише ***Enter***, для того чтобы ввести перевод строки после окончания вот этой. Иначе, наш манифест будет работать неправильно. Он не укажет имя вот этого класса. Поэтому работать не будет.

Теперь же щелкнем на кнопку ***Save***, сохранить. Далее ***Esc***, и теперь опять наберем нашу командную строку для запаковки наших файлов ***Class***. Для этого, как обычно, мы только что делали, ***Jar***, далее ключи, на этот раз они будут такими, ***C***, ***F*** и M. Первый ключ C — это создание нового архива. ***F***- Это задание имени ***Jar*** файла в качестве второго аргумента. А M- это добавление файла с манифеста.

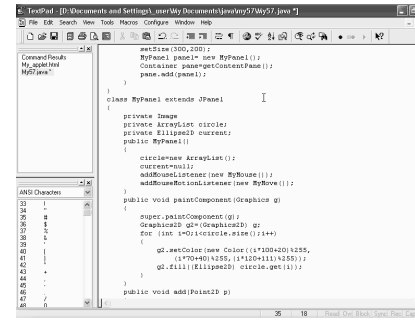
Ну, сначала укажем наш архивационный файл. Это, конечно же, ***My***.***Jar***, пробел, теперь напишем имя нашего файла с манифестом. Это у нас будет Манифест точка и ***MF***. И в конце нам нужно указать файлы, которые мы хотим упаковать. Это, конечно же, звездочка точка класс. Все файлы с расширением ***Class***. Щелкнем теперь на клавише ***Enter***. Компьютер успешно сархивировал все, что мы хотели. И теперь вот этот файл с архивом ***My***.***Jar*** скопируем в эту нашу отдельно стоящую директорию. Для этого щелкнем на клавише ***F***5, ***Enter***, конечно же, ***OverWrite***, нам нужно записать этот файл поверх старой версии архива.

И теперь попробуем запустить этот файл ***My***.***Jar*** в режиме командной строки. Для этого наберем его внизу. ***My*** точка ***Jar***, и щелкнем на клавише ***Enter***. И вот, можно видеть результат на нашем экране. А именно, вот это приложение появилось перед нами, в котором мы, как обычно, можем

рисовать все наши кружочки. Т.е. этот ***Jar*** архив работает и в режиме приложения. Закроем теперь это приложение и мы опять попали в наш менеджер.

6.7.5.4. Упаковка ресурсов в архивы ***jar***

Рассмотрим теперь такой вопрос. Что же нам надо сделать, если у нас в нашем приложении используются разного рода ресурсы, а не только классы. Как их тоже упаковать в наш упакованный файл и как в дальнейшем их вызывать. Для этого чтобы рассмотреть эти вопросы, изменим наше приложение таким образом, чтобы на нем рисовать не только кружочки, но и какие-либо изображения. Например, иконки. Сдвинемся для этого при помощи ползунка ниже и там, где у нас происходит определение класса ***Panel***, введем еще одно поле класса ***Image***.



Для этого напишем, таким образом, ***private***. Далее пусть это будет класс ***ImageIcon***, поскольку загружать будем иконки, и пусть это у нас будет массив, состоящий из нескольких элементов. Назовем этот массив ***im***, далее равняется, и прямо здесь же определим все элементы этого массива. Фигурные скобки, далее щелкнем на клавише ***Enter***.Теперь ***new***, ***ImageIcon***, далее скобка и внутри кавычек напишем имя этой иконки. Пусть это будет ***ball***1 точка ***gif***. Этот файл мы уже заранее поместили в нашу текущую папку. Закрыть кавычки. Закрыть скобку. Далее запятая. Теперь введем еще несколько таких же иконок. Для этого выделим вот это. Правая кнопка мыши и Copy.

Вставим сюда. Правая кнопка мыши и ***Paste***. Вместо ***ball***1 у нас будет ***ball***2, далее запятая. ***Enter***. Следующая иконка. Правая кнопка мыши и ***Paste***. Здесь единицу переправим на тройку. Далее закрыть фигурные скобки. Точка с запятой. А теперь, эти загруженные иконки используем внутри нашего приложения, там, где мы отрисовываем наш рисунок. Для этого сдвинемся чуть ниже. Щелкнем здесь на клавише ***Enter***.

Далее выведем определение "Эллипс 2***D***", из отрисовки нашей окружности. Для этого выделим вот этот текст и потащим его чуть выше. Здесь же напишем, таким образом, Эллипс 2***D***. Пусть этот объект будет просто "***E***", равняется, в конце, конечно же, точка с запятой и далее вот здесь напишем "***E***", чтобы возвратит к своему первоначальному виду, чтобы целостность вот этого куска программы не нарушалась.

Теперь сделаем так, чтобы вот эта окружность рисовалась только в одном случае, а в остальных случаях рисовались бы наши иконки. Для этого введем

здесь оператор ***IF***, далее скобка, далее ***I*** по модулю возьмем, например, 4 и в одном случае из вот этих 4 будем рисовать вот эту окружность. Закрыть скобку. И далее вот этот оператор ***g***2.***fill***. А в противном случае, щелкнем здесь на клавише ***Enter***, введем здесь ***else***, и далее в противном случае будем рисовать наши иконки, которые находятся в нашем массиве ***im***. Поэтому напишем, таким образом, ***g***2 точка ***DrawImage***, обрисовать рисунок, далее скобка. Рисунок нам нужно получить из массива ***im***. Квадратные скобки, внутри которых напишем номер ***i*** и далее деление по модулю на 4, опять-таки. Закрыть квадратные скобки, точка и его метод ***getImage***, поскольку на самом деле у нас здесь находится объект типа ***ImageIcon***, а в ***DrawImage*** нам нужно использовать именно ***Image***.

И в результате получаем, что если ***i*** по модулю в 4 у нас равняется, 0, 1 или 2, будут отрисовываться вот эти иконки из этих трех изображений, а если она равняется 3, тогда в этом случае у нас должна отрисовываться окружность по старому стилю. Ну, конечно же, здесь должно быть равно не 4, а 3. Теперь же запятая, ***Enter***. Далее нам понадобятся координаты X и ***Y*** нашего эллипса. И именно в этом месте мы и будем рисовать нашу иконку. Ну, сначала, конечно, нам надо преобразование типа, к типу ***integer***. Далее "***e***", точка ***GetX***, скобки, запятая. Теперь то же самое по координате ***Y***, преобразование к целому типу, поскольку ***GetX*** и ***GetY*** нам возвращает действительные числа, а не целые. "***E***" точка ***GetY*** на этот раз. Далее скобки. Запятая. Теперь получим ширину и высоту нашего эллипса, для того чтобы наше изображение, так сказать, растягивалось, чтобы покрывать целиком это место.

Опять-таки преобразование к целому типу ***int***. Далее "***E***", точка ***getWidth***, получить ширину. Скобки, запятая. Опять преобразование к целому типу ***int***. "***E***", наш эллипс, на этот раз ***getHeight***, получить высоту, скобки, запятая, и далее напишем 0, поскольку следить за процессом загрузки мы, конечно же, не будем. Далее закрыть скобку, точка с запятой. Запустим теперь наше приложение и посмотрим, что у нас получилось. Для этого ***Tools***, далее ***Compile Java***. Сначала, конечно же, скомпилируем. Компиляция прошла успешно. Теперь ***Tools***. Далее ***Run Java Application***, запустим наше приложение.

Вот оно появилось перед нами и, щелкая на нем, мы на этот раз можем видеть, что рисуются то иконки, то вот такие просто окружности и разного цвета. Иконки у нас представляют собой вот такие шарики. И таким образом, мы можем заполнять наш фрейм. Можем удалить какие-либо из них. Ну, все остальное по-старому, за исключением того, что наряду с окружностями, у нас сейчас выводятся на экран и вот такие иконки.

Закроем теперь наше приложение. Закроем и консольное окно тоже. И мы опять в нашем текстовом редакторе, в котором отметим, что теперь в комплект

нашего приложения, наряду с файлами, с расширением Класс, еще теперь входят и вот эти три гидовских файла. И, конечно же, при архивации нам это обязательно надо учесть. Для того чтобы учесть это обстоятельство немножко изменим тот процесс, при помощи которого мы загружаем вот этот файл.

Для этого здесь напишем немножко по-другому. А именно, ***MyPanel*** точка. Далее Класс, точка, и далее ***getResource***, далее скобка, конечно же, если мы тут вставляем скобку, нужна еще одна закрывающая скобка. И теперь объясним, что же здесь происходит. Тут происходит следующее обстоятельство. Мы указываем нашему компилятору, что вот этот ресурс, нам надо искать в той же директории, что и класс ***MyPanel***. Просто-напросто вот этим методом ***getResource*** создается ***URL*** адрес вот этого файла. Скопируем теперь вот этот кусочек текста, чтобы его заново не писать. Выделим. Правая кнопка мыши и Copy. Встанем сюда. Правая кнопка мыши и ***Paste***. Встанем сюда тоже. Правая кнопка мыши и ***Paste***. И теперь вставим еще две закрывающие соответствующие скобки. Скомпилируем теперь приложение наше и посмотрим, что получилось. Для этого ***Tools***, далее ***Compile Java***.

Компиляция прошла успешно. Теперь ***Tools***, далее ***Run Java Application***, запустим наше приложение. Ну, как мы видим, щелкая, опять получаем все то же самое, ничего в нашем приложении, конечно же, не изменилось. Поскольку мы пока еще работаем с ним, так сказать, в обычном режиме, а не в упакованном. Теперь же закроем его. Закроем и консольное окно тоже. Теперь попробуем все наши классы вместе с вот этими ресурсами, изображениями упаковать в один файл и там запустить, поскольку доступ к вот этим трем файлам изображений сделали мы при помощи указания на место нашего класса, то сейчас мы это уже правомочны сделать.

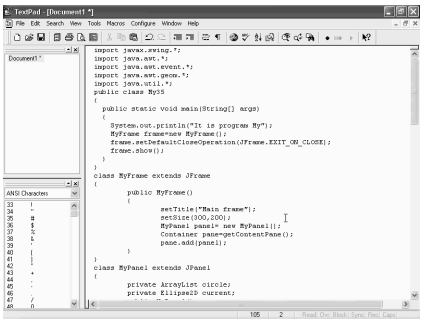
Вот теперь мы опять запустили наш файл в менеджер, в котором теперь и проведем архивацию наших файлов в файл ***My***.***Jar***. Для этого напишем опять ***Jar*** в командной строке. Далее его опции ***CFM***, как мы писали ранее. Мы, конечно же, упаковываем вместе с манифестом, для того чтобы можно было его запускать с командной строки. Далее My точка ***Jar***, имя нашего файла после упаковки. Далее имя файла с манифестом. Это у нас манифест. Точка ***MF***. Этот файл нами уже был приготовлен ранее.

Теперь те файлы, которые надо упаковать. Это, конечно же, звездочка точка класс. Все файлы с классами и далее все файлы с изображениями. Звездочка точка ГИФ, щелкнем теперь на клавише ***Enter***. Архивация прошла вполне успешно. А теперь сдвинемся ниже и там, где у нас находится файл с архивом, ***My***.***Jar***, как мы видим, его размер уже увеличился, поскольку туда вставились и три файла с изображениями. Его теперь скопируем в правое окно. Скопируем туда, где у нас находится специальная папка, в которой всего два файла, файл архива и файл с ***Web*** страницей. Щелкнем теперь на клавише ***F***5,

скопировать, ОК, дальше ***Owerwrite***, конечно же, заменить, и теперь перейдем на правую сторону и попробуем запустить этот файл сначала с командной строки.

Для этого напишем ***My.Jar*** и щелкнем на клавише ***Enter***. И вот, как мы видим, у нас запустилось наше приложение. И. щелкая на нем, мы можем опять-таки рисовать то изображение вот этих шариков, то соответствующие кружочки. Закроем теперь это приложение и попробуем то же самое запустить в режиме аплета, в режиме ***Web*** страницы. Для этого щелкнем теперь на ***Enter***, находясь на ***Web*** странице. Вот наша ***Web*** страница загрузилась перед нами и, щелкая на нем, мы опять-таки можем рисовать все наши кружочки и шарики. Мы можем их сдвигать в какую-либо сторону. Ну, все, как обычно. Закроем теперь наш аплит. Для этого щелкнем вот на этом крестике и перейдем на наш текстовый редактор.

6.7.5.5. Всплывающие окна в апплетах Создание кнопки



Как мы видели, аплет обычно встраивается в ***html*** страницу в виде некого прямоугольника, и внутри этого прямоугольника аплет можем выполнять все свои действия. Очень часто это бывает достаточно неудобно. И более удобным было бы, если вместо вот этого прямоугольника, у нас была бы одна кнопка, при нажатии на которую появлялось всплывающее окно, фрейм, внутри которого мы уже могли бы полностью развернуться, выполняя все действия при помощи языка ***Java***. В принципе это тоже не очень сложная задача, которая вполне разрешима. Попробуем это сделать.

Для этого, рассмотрим вот это приложение, которое мы уже рассматривали не раз. Оно создает окно фрейм, внутри которого мы можем рисовать разноцветные кружочки, щелкая на поверхности фрейма мышкой. Переделаем название класса. Пусть будет по порядку, My58. Сохраним его в соответствующем файле. Для этого щелкнем на значке сохранения. Введем новую папку для этого проекта. Пусть это будет ***m***y58. Щелкнем на кнопку ***Open***. Введем имя для нашего файла. Оно, конечно, должно совпадать с именем нашего класса. Тип файла возьмем ***Java***. И щелкнем на кнопку ***Save***, сохранить. И вот мы опять в нашем текстовом редакторе. Теперь реализуем нашу идею следующим образом. Во-первых, класс My58 должен поддерживать класс ***JApplet***, быть его наследником.

Поэтому ***extends***, далее ***JApplet***. Класс ***main*** нам уже не нужен. Поэтому удалим вот эту строку с текстом. ***Public*** оставим, как есть, ***void***, и далее, нам обязательно нужен метод ***init***. Далее скобки. Теперь ***println*** можно удалить. Удалим и вот эти две строчки. Выделим их и щелкнем на значке удаления, ***Delete***. Определение нашего фрейма зададим, как поле, в классе My58. Для этого щелкнем здесь на клавише ***Enter*** и далее напишем таким образом. Пусть это будет ***private***. Далее ***MyFrame***, имя у него будет просто ***frame***. Точка с запятой. Задание этого фрейма оставим, а ***MyFrame*** нужно удалить, щелкнем теперь на клавише ***Enter*** и создадим кнопку. Кнопку создать достаточно просто. Воспользуемся методом ***JButton***. Пусть эта кнопка называется просто ***button***.

Далее, как обычно, знак равенства, ***new***, ***JButton***. Это именно та кнопка, появление которой мы и хотим на браузере. Далее скобки, внутри которых напишем тот текст, который должен появляться на этой кнопке. В кавычках напишем ***My applet***. Закроем скобку, точка с запятой. Теперь эту кнопку нам надо добавить на наш контейнер. Поэтому напишем таким образом ***getContentPane***. Отдельно контейнер создавать не будем, а прямо на полученный контейнер добавим нашу кнопку.

Далее его метод ***add***, и внутри скобок напишем нашу кнопку ***button***. Закроем скобку, точка с запятой. Попробуем теперь запустить наш аплет и посмотреть, что же у нас получилось, хотя бы чисто внешне. Сначала все это надо скомпилировать. Для этого ***Tools***, далее ***Compile Java***. Компиляция прошла успешно. Теперь ***Tools***, далее ***Run Java Applet***, и вот аплет появился перед нами, который заключается в том, что внутри вот этого достаточно большого прямоугольника как раз и находится кнопка с надписью ***My applet***.

Мы можем щелкнуть на нее. Правда, сколько бы мы не щелкали, ничего существенного не происходит, поскольку обработчика события на эту кнопку, мы пока еще не создали. Закроем теперь этот аплет и откроем ту временную ***html*** страницу, которую для нас создал ***TextPad***. Щелкнем на значке открытия. Далее переключимся на ***html*** файлы. Откроем вот этот файл. Щелкнем на нем дважды. Вот он появился перед нами. Сохраним его под каким-либо именем. Для этого ***Files***, ***Save As***. Выберем для него какое-либо имя.

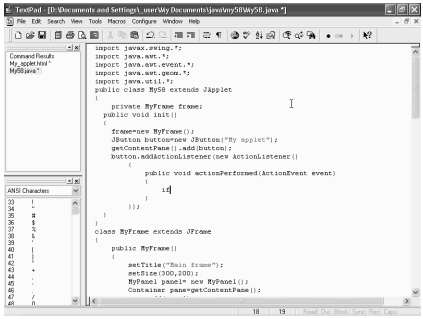
Например, пусть это будет ***My_applet***. Щелкнем на кнопке ***Save*** (сохранить). И теперь изменим тот прямоугольник, который у нас ***html*** страница отводит под наш аплет. Как видим, у нас отводится 400x300 единиц, что, конечно же, для кнопки очень большое число. Мы вполне можем задать 100 единиц по горизонтали и 20 единиц по вертикали. Для кнопки этого будет вполне достаточно. И далее, до этой кнопки, поместим такую надпись, например, ***It is button to start my applet***.

Теперь щелкнем на вот этой кнопке просмотра в ***Web*** браузере и посмотрим, как все это выглядит там. И вот, можно видеть, что у нас получилось. У нас загрузилась наша ***html*** страница, внутри которой находится вот эта кнопка ***My applet***. Если мы на ней щелкнем, пока ничего не происходит. Но, кнопка есть. Закроем теперь нашу ***Web*** страницу. И вот мы опять находимся в нашем текстовом редакторе, в котором переключимся на наш листинг ***Java***.

6.7.5.6. Всплывающие окна в апплетах. Обработчик событий

Теперь нам необходимо написать обработчик для нажатия на эту кнопку ***My applet***, создать слушателя. Для этого напишем таким образом. Наша кнопка ***button***, точка, далее ***addActionListener***, добавляем слушателя. Далее, как обычно, скобка,***new***, создадим внутренний безымянный класс для описания этого слушателя, опять ***ActionListener***, далее скобки, ***Enter***, фигурные скобки, внутри которых нам необходимо описать слушателя. Закроем скобки, точка с запятой. Теперь, внутри фигурных скобок напишем необходимый метод ***actionPerformed***, который является обязательным условием для интерфейса ***ActionListener***. Поэтому ***Enter***, пусть будет ***public***, тип возвращаемого значения ***void***, поскольку никакого возвращаемого значения нет, и далее ***actionPerformed***.

Далее скобка, внутри которой нам нужно указать тип возвращаемого параметра. ***ActionEvent***, и его имя пусть будет тоже ***event***. Закроем скобку, ***Enter***, фигурные скобки, внутри которых мы и должны написать то действие, которое нам нужно, а именно, вызов нашего фрейма. Во-первых, сделаем так, чтобы при нажатии на эту кнопку у нас фрейм появлялся.



А при повторном нажатии, исчезал с экрана. Для этого нам при помощи ***if*** надо проверить текущее состояние этого объекта, этого экрана. Является ли он видимым или нет. Для этого напишем так, ***frame***, точка. Далее ***isVisible***, является ли видимым. Закроем скобки, и если этот фрейм у нас виден, тогда надо его сделать невидимым. Поэтому ***frame***, точка, далее ***setVisible***, сделать его невидимым. Конечно же, в этом случае нам внутри скобок надо указать параметр ***false***. Закроем скобку, ***Enter***, точка с запятой. И если это у нас неверно, т.е. окно невидимое, только ***else***, и в этом случае нам наш фрейм нужно вывести на поверхность экрана, т.е. выполнить его метод ***show***. Закроем скобки, точка с запятой. Скомпилируем теперь наш аплет.

Для этого ***Tools***, далее ***Compile Java***. Скомпилируем. Компиляция прошла успешно. Теперь переключимся на нашу ***html*** страницу. Вот она появилась перед нами, и попробуем ее запустить, посмотреть ее в ***Web*** браузере. Щелкнем на этом значке. Вот, наша ***Web*** страница появилась перед нами, и как мы видим, здесь опять находится вот эта кнопка ***My applet***. Если мы на ней щелкнем, то перед нами появляется вот этот фрейм, в котором мы можем рисовать все необходимые нам кружочки.

Попробуем его теперь сдвинуть чуть-чуть. Вот такое расположение аплета и нашей ***html*** страницы позволяет судить о том, находится ли фрейм на экране после того, как мы переключимся на нашу ***html*** страницу. Вот мы переключаемся и, как мы видим, вот этот фрейм все равно остается у нас на экране, на заднем плане. А если же мы сейчас щелкнем на кнопку ***My applet***, то, как мы видим, вот с этого заднего плана нашего экрана аплет исчезает, т.е. он становится полностью невидимым.

Если мы еще раз щелкнем, аплет появился перед нами с теми же самыми кружочками, в том же самом состоянии, щелкнем еще раз на ***My applet***, он исчез, и таким образом, он может появляться и исчезать. Нажимая один раз, мы его убираем с экрана, а во второй раз мы его активизируем. Теперь отметим еще одно обстоятельство, ту надпись, которая у нас появляется вот в этом окне, в ее нижней части, а именно, сейчас мы видим надпись ***Java Applet Window***. Надо отметить, что в более ранних вариантах браузеров сообщение этого окошка было более угрожающим.

Раньше там писалось или "окно аплета неизвестного происхождения", и даже "окно подозрительного аплета". Дело в том, что обычные приложения, написанные на других языках программирования, таких как ***C***, ***Pascal*** и др. в принципе, являются достаточно опасными, поскольку при запуске приложения из непроверенных источников, теоретически они могут достаточно серьезно повредить наш компьютер и данные на нем, т.к., что они могут делать внутри нашего компьютера, проконтролировать совершенно невозможно. Но в отношении языка ***Java***, это абсолютно неверно.

Дело в том, что поскольку язык ***Java*** является интерпретатором, то он запускается под управлением специального менеджера безопасности, который, если не существует специальной электронной подписи под нашим фреймом, разрешает запуск нашего приложения на языке ***Java*** нашего аплета только в так называемой песочнице, при которых правку этого приложения достаточно мало. Во-первых, он ни при каких обстоятельствах не может запускать какие-либо выполняемые модули на нашем локальном компьютере. Он не может взаимодействовать ни с одним компьютером, кроме своего сервера, откуда он был загружен.

Также надо отметить, что аплет не может считывать никакие данные из локальной файловой системы или же записывать их туда. Также аплеты не способны распознавать никакую информацию о локальном компьютере, за исключением нескольких таких, например, как номер версии языка ***Java***, символы, используемые для разделения файлов, такие как запятые, точка с запятой, двоеточие и т.д. Т.е. в принципе такие достаточно безобидные вопросы. А такие достаточно скрываемые от посторонних взоров вопросы, типа адреса электронной почты, ***Web*** страницы и т.д., они полностью недоступны для языка ***Java***. Закроем нашу ***html*** страницу. Щелкнем на вот этом крестике. И вот мы опять в нашем текстовом редакторе.

*6.7.5.7. Создание передаваемых параметров на **html** странице*

Аплеты, которые мы используем на наших ***Web*** страницах, не являются совершенно изолированными объектами, а могут взаимодействовать со своим родителем, со своей ***html*** страницей. И эта задача, задача взаимодействия, решается с помощью дескриптора языка, ***html***, под названием ***Param***, при помощи которого можно передавать параметры от ***html*** страницы внутрь нашего аплета. Для того чтобы подробнее рассмотреть эти вопросы, начнем работу с вот этого шаблона приложения, который является простейшим аплетом, и содержит наш основной класс My59. Он является дочерним классом класса ***JApplet***, внутри которого реализован, как мы видим, метод ***init***, с которого начинается работа всех аплетом, и в этом методе создается панель на основе класса ***MyPanel***. И далее происходит отрисовка вот этого текста.

Сохраним теперь этот листинг в каком-либо файле. Щелкнем для этого на значке сохранения. Создадим, как обычно, для него новую папку. Пусть это будет у нас таким образом. И щелкнем на кнопку ***Open***. Теперь введем имя для нашего документа. Конечно, оно должно повторять имя нашего класса, поэтому должно быть вот таким. Тип нашего файла, конечно, должен быть ***Java***, теперь щелкнем на кнопку ***Save***, сохранить.

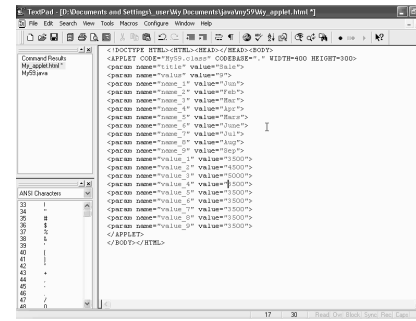
И вот, мы опять в нашем текстовом редакторе. Попробуем теперь решить такую задачу, при помощи параметров, которые мы передадим аплету в наших ***html*** страницах, попробуем построить какую-либо гистограмму. Первое, что мы сейчас сделаем, посмотрим, как выглядит вот этот аплет, который у нас перед глазами. Сначала скомпилируем его. Для этого развернем меню ***Tools***, и далее ***Compile Java***.

Компиляция прошла успешно. Теперь ***Tools***, и далее ***Run Java Applet***, запустить аплет. И вот, аплет появился перед нами. Пока ничего впечатляющего нет. Тут только одна единственная строчка текста. Закроем теперь его и откроем ту ***html*** страницу, которую создал наш текстовый редактор в качестве временного файла. Для этого щелкнем на значке

открытия, переключимся на ***html*** файлы, и откроем вот этот файл. Щелкнем на нем дважды и сохраним его под каким-либо именем. Для этого ***File***, далее ***Save As***, выберем имя для этого файла.

Пусть это будет ***My_applet***. Щелкнем на кнопку ***Save***, сохранить. А теперь, сформируем внутри этой ***html*** страницы передаваемые параметры для аплета. Для этого щелкнем здесь на клавише ***Enter*** и далее напишем таким образом. Знак больше — меньше, внутри которого обычно должны располагаться любая единица, информация в ***html*** страницах. Далее напишем ключевое слово ***param***, параметр. Далее введем имя для нашего параметра ***name***, и внутри кавычек напишем таким образом. Пусть это будет та строка, которая должна появляться наверху нашей гистограммы.

Поэтому это будет ***title***, и далее нам нужно указать его значение, ***value***, которое будет равняться строчке, которую мы напишем внутри кавычек. Напишем просто ***Sale***, продажа. Представим себе, что мы хотим сорганизовать гистограмму, которая будет показывать продажу каких-то товаров по месяцам. Поэтому щелкнем здесь на кнопку ***Enter*** и далее введем, во-первых, количество значений, которые мы хотим получить в нашей гистограмме. Для этого опять знак больше — меньше, внутри которого опять ***param***, параметры, далее имя нашего параметра, знак равенства. Наверху тоже напишем знак равенства, который мы здесь забыли написать, и далее это у нас будет ***values***, количество значений. Теперь его значение, его ***value***, знак равенства. Возьмем, например, 9 значений.



Теперь, чтобы не писать это еще раз, выделим вот эту строчку, далее правая кнопка мыши и Copy, встанем сюда, правая кнопка мыши, ***Paste***. Следующее имя у нас будет ***name***_1. Далее возьмем по порядку, и значением будет та надпись, которая должна соответствовать этому элементу.

Пусть ему соответствуют показатели за январь. Напишем таким образом, январь. Теперь скопируем эту строчку. Для этого выделим. Правая кнопка мыши, Copy. Вставим. Правая кнопка мыши, ***Paste***. Вместо января, у нас будет февраль. Встанем опять сюда. Правая кнопка мыши, ***Paste***. Это будет следующий месяц, март.

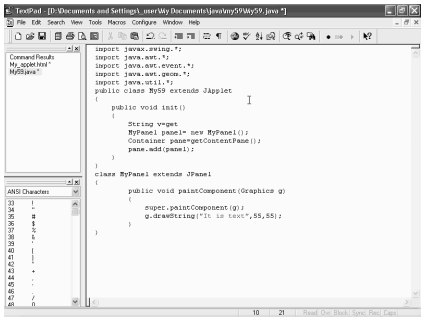
Встанем сюда. Правая кнопка мыши, ***Paste***. Очередной месяц будет май. Следующий месяц, который мы сюда вставим при помощи правой кнопки мыши, будет июнь. Встанем сюда. Правая кнопка мыши, ***Paste***. Следующий

месяц июль, после которого идет последний, 9 месяц в нашем списке, это месяц август. Теперь соответствующим образом изменим номера этих параметров. После 1, будет 2, далее 3, 4, 5. Мы пропустили апрель, вставим его вот сюда, между мартом и маем. Активизируем команду ***Paste***.

Теперь встанем сюда. Правая кнопка мыши, ***Paste***, это будет последний месяц сентябрь.Продолжим вот эту нумерацию. После третьего, будет четвертый параметр, далее пятый, шестой, седьмой, восьмой и последний девятый. Теперь сформируем цифры, которые будут характеризовать количество продаж в эти месяцы. Для этого вставим сюда строчку. Правая кнопка мыши, ***Paste***. Вместо ***name***_1, напишем, ***value***_1. Пусть это значение равняется какому-то числу, которое как раз и будет характеризовать количество продаж за январь. Например, пусть будет число 3.500. Теперь скопируем эту строчку. Выделим. Правая кнопка мыши, Copy. Встанем сюда. Правая кнопка мыши, ***Paste***. Вставим несколько раз. Теперь после ***value***_1, будет ***value***_2, затем 3, 4, 5, 6, 7, 8 и последнее, девятое значение.

Изменим и числа, которые будут значениями. После 3.500 пусть будет 4.500, далее пусть будет 5.000, следующее число пусть будет 7.500, после которого пусть будет чуть меньше, 6.500. Далее опять 5.500, 8.000, 8.500 и 7.500. Теперь у нас наша ***html*** страница полностью готова, переключимся на листинг программы ***Java***, в котором и попробуем обработать все эти значения, и построить на их основе гистограмму. Щелкнем вот здесь.

6.7.5.8. Чтение параметров в апплете



Теперь, внутри этого метода ***init***, попробуем прочитать все эти значения. Для этого, первое, что сейчас сделаем, это прочитаем количество значений в нашем массиве. Для этого ***String***. В принципе, все передаваемые параметры по определению являются строковыми. Пусть эта переменная у нас будет ***v***, знак равенства и ***getParameter***, получить параметр. Далее скобка, в кавычки напишем ***values***. Закроем кавычки, скобку, точка с запятой. Теперь первое, что сделаем, предусмотрим выход из нашей программы, если эта переменная является нулевым. ***if***, ***v***, равное нулю, ***null***. В этом случае мы возвращаемся и прекращаем выполнение аплета.

А в другом случае преобразуем это число к целому типу. ***int***, далее ***n***, равное ***integer***, преобразование из строковой в целую.Для этого нам нужно воспользоваться методом ***parseInt***. Далее скобка и наша переменная ***v***, точка с

запятой. Теперь создадим два массива. Первый массив действительных чисел ***double***. Пусть это будет ***values***, значение.

Далее, как обычно, ***new***, ***double***, действительные числа в количестве ***n*** штук. ***N*** мы уже установили. Теперь определим еще один массив. На этот раз массив типа ***String***, строковых переменных, в котором будет сохранять имена, ***names***, названия месяцев. Далее, как обычно, знак равенства, ***new***, ***String***, и массив из ***n*** чисел. Точка с запятой. А теперь организуем цикл. Для этого введем переменную этого цикла, которое изначально определим, как целое число ***i***, далее ***for***, скобка, начальное значение ***i*** равно 0. Теперь, как обычно, ***i*** меньше, чем ***n***. И за одну итерацию ***i*** увеличиваем на единицу, ***i***++, закроем скобку, ***Enter***, далее фигурные скобки, внутри которых определим, что должно происходить в цикле.

А именно, мы должны прочитать наши значения. ***Values***, от ***i***, ***i***-тое значение, знак равенства, преобразование к действительному числу, ***double***, точка, ***parseDouble***, скобка, и далее нам нужно получить параметр, при помощи метода ***get Parameter***. Имя нашего параметра будет ***value***. Затем, знак плюса, его номер, который возьмем, как ***i***+1, поскольку нумерация должна идти от единицы, а не от 0, как у нас происходит в цикле. Далее закроем скобки, точка с запятой. Теперь получим массив имен, ***names***, от ***i***, знак равенства, далее, чтобы много не писать, скопируем вот этот кусок текста, выделим, правая кнопка мыши, Copy. Встанем сюда. Правая кнопка мыши, ***Paste***.

Здесь у нас будет вместо ***value***, а ***name***, и в конце поставим точку с запятой. А теперь все эти значения, которые мы только что вытащили, передадим классу ***MyPanel***, в качестве параметров. Для этого встанем вот сюда и напишем таким образом. Передадим значение, ***values***, запятая, далее передадим имена, ***names***, и затем, через запятую передадим название нашей гистограммы, которую тоже получим при помощи вот этого метода ***getParameter***. Поэтому правая кнопка мыши, Copy, скопируем.

Встанем сюда. Правая кнопка мыши, ***Paste***. И нам надо получить параметр, который связан с ***title***. Закроем кавычки, скобки. Теперь сдвинемся скроллингом вниз и напишем конструктор для класса ***MyPanel***, который должен принимать в качестве значений вот эти значения. Поэтому щелкнем на кнопку ***Enter*** и далее напишем таким образом. Конструктор должен быть ***public***, далее имя, ***MyPanel***, имя конструктора точно должно повторять имя класса.

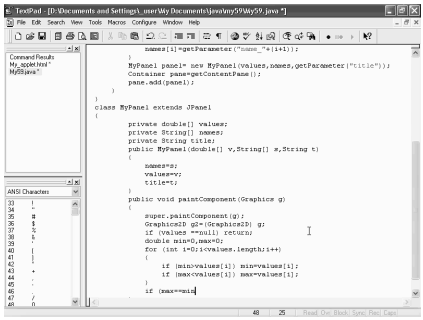
В скобках напишем таким образом, первый параметр у нас значение, которое имеет тип массива ***double***. Пусть это у нас будет ***v***, запятая. Следующий параметр, это массив типа ***String***, строковые значения, пусть это у нас будет ***s***, запятая и последнее, это наименование нашей гистограммы,

просто строка. И пусть это у нас будет, например, переменная ***t***, ***title***, далее закроем скобку, ***Enter***. Фигурные скобки, внутри которых проведем осваивание вот этих значений. Для этого создадим соответствующие поля. Они должны быть ***private***. Первый из них, значение, он у нас должны быть типа ***double***. Название будет ***values***, точка с запятой.

Следующее поле опять с кодом доступа ***private***. Оно у нас должно быть типа ***String***, строковой переменной. Конечно же, массив, и это у нас будет имена, ***names***, точка с запятой. Следующее поле, ***private***, и оно будет у нас просто ***String***. Строковая переменная, которую назовем просто ***title***, точка с запятой. Теперь проведем присваивание. Конечно же, ***names*** должно равняться ***s***, точка с запятой. Далее ***values***, значения, должны равняться ***b***. А титульная строка ***title***, должна равняться переменной ***t***, точка с запятой. Теперь у нас есть все данные внутри этого класса, и мы можем на их основе построить внутри вот этого метода ***paintComponent*** соответствующую гистограмму.

6.7.5.9. Передача параметров апплетам. Построение гистограммы на их основе

Теперь вот эту строку с отрисовкой текста удалим, она нам уже не нужна. И напишем таким образом. Проведем преобразование типа ***Graphics*** к ***Graphics2D***. Для этого напишем таким образом. ***Graphics2D***, назовем этот объект ***g2***, знак равенства и преобразование типа к типу ***Graphics2D*** от нашего графического объекта ***g***. Теперь вычислим минимальное и максимальное значение среди наших значений ***values***, для того чтобы вы могли подобрать соответствующий масштаб нашей гистограммы.



Во-первых, проверим при помощи ***if***, не является ли случайно значение нашей переменной ***values*** пустым, ***null***, и если это верно, тогда прекращаем выполнение нашей программы. ***Return***, в нее введем две переменные типа ***double***. Первая — пусть будет у нас минимальным значением, ***min***, вторая — максимальным, ***max***. Обоим значением присвоим для начала значение 0. Теперь щелкнем на кнопку ***Enter***, и введем оператор цикла ***for***. ***For***, далее, как обычно, ***int***, ***i***, равное нулю, начальное значение, точка с запятой. Далее ***i*** меньше, чем размер нашего массива, который мы возьмем таким образом ***values.length***, точка с запятой.

И далее приращение за одну итерацию ***i*** единицу, ***i***++. Далее ***Enter***. Фигурные скобки, внутри которых проведем такое сравнение. ***If***, если у нас значение ***min*** больше, чем текущее значение ***values***, от элемента ***i***, в этом

случае нашей переменной ***min*** как раз это значение и присвоим. ***Values*** от ***i***, точка с запятой. Теперь выделим эту строчку. Правая кнопка мыши, Copy. Встанем сюда. Правая кнопка мыши, ***Paste***. И теперь, для того чтобы определить максимальное значение ***max*** сделаем все с точностью до наоборот. Знак больше изменим на знак меньше, и ***min*** изменим на ***max***.

Теперь проведем еще одну проверку ***if***. Если у нас максимальное значение равняется минимальному, то в этом случае нет смысла строить график особого значения, прекратим выполнение нашего аплета, ***return***, точка с запятой, ***Enter***. Теперь вычислим несколько постоянных, которые нам понадобятся в дальнейшем. В частности получим ширину нашего аплета.

Для этого напишем так. Это у нас целое число. Пусть будет ***w***, знак равенства и ***getWidth***, получить ширину, точка с запятой. Теперь ту же самую операцию проведем с высотой, ***int h***, знак равенства и далее ***getHeight***, точка с запятой, ***Enter***. Теперь вычислим коэффициент масштабности. Он у нас будет типа ***double***. Назовем его просто ***scale***, знак равенства, он у нас будет равняться следующей величине. Во-первых, высота ***h***, вычесть какое-то количество единиц, которое мы отведем под наш заголовок, допустим, 20 единиц. Закроем скобку. И разделить все это на (***max-min***), разность между максимальным и минимальным значениями наших значений.

Теперь ***Enter***. Вычислим ширину одного единственного столбика. Это у нас будет ***int***. Назовем эту величину ***bar***. Далее знак равенства, общая ширина аплета, деленная на количество элементов в нашем массиве. Поэтому ***values***, точка, ***length***, точка с запятой. А теперь приступим непосредственно к отрисовке наших столбиков. Скопируем вот этот оператор цикла. Выделим его. Правая кнопка мыши, Copy. Вставим сюда. Правая кнопка мыши, ***Paste***. Вставим сюда еще одну закрывающую фигурную скобку, а внутри нее напишем таким образом. Для начала получим координаты левого верхнего угла соответствующего прямоугольника нашей гистограммы. Для этого напишем таким образом, ***double***, получим сначала ***x***1, знак равенства, получим его таким образом, ***i*** умножить на ширину нашего столбика, которая равна переменной ***bar***.

Затем сместим все это на единицу вправо, +1, точка с запятой. Теперь получим его координату ***Y***, это у нас будет опять ***double***. Далее ***y***1, знак равенства, и напишем таким образом. ***Max***, максимальное значение, минус текущее значение ***values*** от ***i***, и все это нам нужно умножить на наш масштаб ***scale***, и далее точка с запятой. Далее ***Enter*** и теперь получим прямоугольник, ***Rectangle***, ***Rectangle2D***. Назовем его просто ***rect***, далее знак равенства, как обычно, ***new***, ***Rectangle2D***, далее точка. Надо воспользоваться его разновидностью, которая рисует при помощи ***double***, при помощи действительных чисел такой точности.

Далее, как обычно, ***x***1, ***y***1, запятая. Теперь ширина нашего столбика ***bar***. Сделаем его чуть меньше. ***Bar***-2, запятая, и высота, которая будет равняться текущему значению ***values*** от ***i***, умноженное на коэффициент ***scale***. Закроем скобку, точка с запятой. Теперь отрисуем этот прямоугольник. Для этого ***g***2, точка. ***setPaint***. Зададим цвет для отрисовки. Для этого ***Color***, точка. Возьмем, например, цвет красный. ***Red***, точка с запятой. Теперь нарисуем закрашенный прямоугольник. Для этого ***g***2, точка, ***fill*** и наш прямоугольник ***rect***. Закроем скобку, точка с запятой.

Теперь зададим цвет для контура, ***g***2, точка, далее ***setPaint***. Зададим для этого черный цвет. ***Color***, точка, ***black***. Закроем скобку, точка с запятой. Теперь отрисуем сам контур. Для этого ***g***2, точка. Контур рисуется при помощи ***draw***, скобка, и прямоугольник ***rect***. Закроем скобку, точка с запятой. Теперь сдвинем все наши столбики при помощи вот этой переменной ***y***1 на какое-то количество единиц вниз. Например, напишем плюс 15, и в этом случае у нас сверху от нашей гистограммы, как раз и останется просвет в 15 единиц. А снизу просвет в 20 -15, т.е. 5 единиц, чтобы наши графики совсем на нижний край нашего аплета не садились. Скомпилируем теперь наш аплет. Для этого ***Tools***, далее ***Compile Java***.

Компиляция прошла успешно. Теперь переключимся на наш ***html*** файл и посмотрим его в браузере. Для этого щелкнем на вот этой кнопке ***View in Web Browser***, и можно видеть результат того, что у нас получилось. А именно, мы на основе тех данных, которые у нас были на ***Web*** странице построили вот такой график внутри нашего аплета. Закроем теперь, переключимся на наш листинг. И вот мы опять в нашем текстовом редакторе.

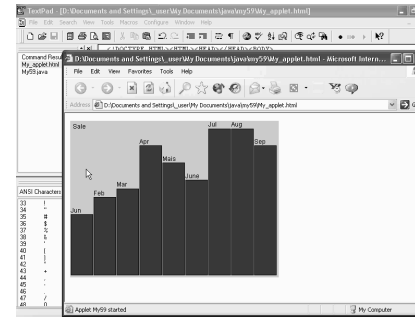
6.7.5.10. Анимированные гистограммы на основе параметров из Web—страниц

Выведем теперь на поверхность нашего аплета название наших столбиков, т.е. январь, февраль и т.д. Для этого сдвинемся скроллингом ниже. Щелкнем здесь на клавише ***Enter*** и далее напишем таким образом, ***g***2 точка, далее метод, далее откроем скобки, внутри которых нам сначала нужно написать ту строку, которую мы хотим вывести. А именно ***names***, затем ее значение ***i***, запятая. Теперь координаты, где мы должны вывести эту строку.

Далее напишем таким образом. Во-первых, проведем преобразование типа к ***float***, поскольку именно числа такого типа нужны для нашего метода ***drowString*** и далее возьмем ***x***1, запятая. Опять преобразование типа к ***float*** и ***y***1, и чуть-чуть сместим его наверх. Например, минус 4 единицы. Закроем скобку. Точка с запятой. Щелкнем теперь здесь на клавише ***Enter*** и выведем на поверхность нашего аплета название нашего рисунка, нашей гистограммы. Для этого опять воспользуемся вот этим методом ***drowString***. Проще всего эту строчку скопировать. Выделим ее. Щелкнем правой кнопкой мыши и Copy.

Встанем сюда. Правая кнопка мыши и ***Paste***. И теперь, конечно же, выведем название ***title***.

И выведем это название просто-напросто в левом верхнем углу нашей гистограммы. Для этого вот это все уберем. Нажмем на клавишу ***Delete*** и далее чуть-чуть отступим от левого края. Например, на 5 единиц и сверху, например, на 15 единиц. Закроем скобку. Точка с запятой. Скомпилируем теперь наш аплет и посмотрим, что у нас получилось. Для этого развернем меню ***Tools***, далее ***Compile Java***. Компиляция прошла успешно. Теперь перейдем на наш ***html*** файл и посмотрим все это дело в браузере. И вот, можно видеть внешний вид нашего аплета ***с*** гистограммой, уже со всеми ее надписями. Закроем теперь этот файл. Перейдем опять на наш листинг. И вот мы опять в нашем текстовом редакторе. Учтем теперь, что все-таки наша гистограмма у нас нарисована на языке ***Java*** и попробуем ее немножко оживить. Для этого сдвинемся скроллингом выше и внутри конструктора ***MyPanel*** создадим новый таймер, при помощи которого попробуем чуть-чуть санимировать нашу гистограмму. А именно, чтобы наши столбики появлялись не все вместе, а с какими-то интервалами по времени.



Ну, напишем таким образом: ***Timer***. Пусть у нас будет называться вот так, две буквы ***tt***. Далее знак равнения, как обычно ***new***, далее ***Timer***. Вызываем его конструктор, откроем скобку. И первое, что нам нужно задать, это интервал, зададим его 1000, т.е. 1000 милли секунд или 1 секунда, и далее нам нужно задать обработчик события, который наступает через каждую секунду, по истечении времени таймера. Ну, это у нас должно быть, конечно же, ***new ActionListener***. Этот класс опишем прямо здесь, как внутренний класс, внутренний безымянный.

Далее напишем фигурные скобки. Далее, чтобы не забыть, открывающую скобку и точку с запятой. Ну, а теперь, внутри этих фигурных скобок напишем тот метод, который обязательно должен быть в интерфейсе ***ActionListener***, а именно ***actionPerformed***. Поэтому напишем ***public***. Далее тип возвращаемого значения ***void***, поскольку возвращаемого значения никакого нет, и далее ***actionPerformed***.

Далее откроем скобку. Потом надо указать тип передаваемого параметра. Это у нас будет ***ActionEvent***. И далее сам этот параметр пусть будет ***event***. Закроем скобку. Щелкнем на клавише ***Enter***. Далее введем фигурные скобки,

внутри которых теперь нам нужно прописать, что же мы хотим сделать с вот этим интервалом в 1 секунду. Теперь зададим два поля внутри класса ***MyPanel***, они, конечно же, должны быть ***private***. Далее пусть они будут типа ***int*** (целые) и назовем их, например, ***n*** и далее ***ni***. Точка с запятой. Теперь же, вот в эту переменную ***n*** поместим количество элементов в нашем массиве ***v***. Для этого напишем таким образом, ***n*** равняется, ***v*** точка ***length*** (длина массива). И даже возьмем это число чуть побольше. Например, плюс 5. Точка с запятой.

Для чего это сделано? Дело в том, что, когда мы будем выводить столбики, будем выводить их так сказать по циклу до этого числа ***n***, и поскольку ***n*** у нас больше, чем количество элементов в нашем массиве, то мы, вот эти последние 5 шагов, просто-напросто получим неподвижное изображение нашей гистограммы на экране. Ну, это сделано, для того чтобы, так сказать, полностью отрисованная гистограмма задерживалась бы на нашем экране на несколько секунд дольше.

Теперь зададим следующим образом. Зададим начальное значение для ***ni***. Пусть оно будет равно 0. А теперь же напишем так. Наша переменная ***ni***, знак равнения, далее внутри скобок напишем ***ni***+1 и деленное по модулю на ***n***. Точка с запятой. Теперь, после того, как мы это число задали, нам надо перерисовать наш аплет. Для этого, конечно, напишем ***repaint***. Далее введем скобки. Точка с запятой. А теперь сдвинемся скроллингом ниже. И там, где у нас отрисовывается вот этот полный прямоугольник, напишем таким образом.

Щелкнем здесь на клавише ***Enter***, далее ***if***, проведем проверку, и если у нас, наша переменная ***i*** меньше, чем значение ***ni***, только в этом случае будем проводить отрисовку нашего прямоугольника, закрашивание его целиком. А во всех же остальных случаях, пусть рисуется только контур, при помощи оператора ***g***2 ***draw***. Теперь же, прежде чем мы скомпилируем наше приложение, сдвинемся скроллингом выше и если мы использовали наш таймер, его, конечно, нужно запустить. Для этого щелкнем на клавише ***Enter***, и далее напишем таким образом. Наш таймер ***tt***, точка и далее его метод ***Start***. Далее введем скобки. Точка с запятой. Теперь же сдвинемся скроллингом еще выше, туда, где у нас находятся все директивы ***import*** и отметим, что наш класс ***Timer*** находится в двух пакетах, в пакете ***java.util*** и в пакете ***javax.swing***.

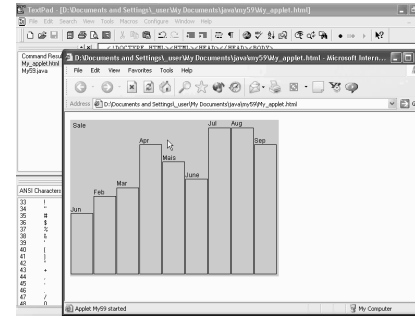
И в связи с этим, конечно же, если мы запустим наш компилятор, то компьютер сразу же выдаст нам ошибку. Для того чтобы это исключить, мы можем, конечно, непосредственно указать тот класс, который мы хотим использовать, не в виде пакета, а прямо в виде класса. Но мы можем поступить и по-другому. Учитывая, что пакет ***java.util*** мы не использовали внутри нашего аплета, мы можем его удалить. Щелкнем на клавише ***Delete***. И теперь скомпилируем наше приложение. Для этого развернем меню ***Tools***, далее

Compile Java. Компиляция прошла успешно. Теперь переключимся на наш ***html*** файл и запустим просмотр нашего аплета в браузере, в обозревателе.

Щелкнем на этой кнопке, и вот можно видеть, как выглядит это все на практике. Как мы видим, через каждую секунду у нас появляется новый столбик. Теперь в течении нескольких секунд мы видим всю фигуру и далее все столбики опять заполняются соответствующим цветом. Закроем теперь нашу ***html*** страницу. Щелкнем на вот этом крестике. И перейдем на наш листинг. И вот мы опять в нашем текстовом редакторе.

6.7.5.11. Создание цветных анимированных гистограмм на основе параметров из Web страниц. Создание нескольких вариантов страниц

Попробуем теперь сделать так, чтобы все столбики в нашей гистограмме выводились бы с разными цветами. Для этого сдвинемся скроллингом ниже. И создадим новый массив, массив цветов. Для этого щелкнем на клавише ***Enter***, далее, конечно же, ***private***. Далее тип этой переменной ***Color***. Это должен быть массив. Поэтому напишем квадратные скобки, и далее пусть это будет переменная ***c***. Далее точка с запятой. Теперь проинициализируем эту переменную ***c***. Для этого щелкнем здесь на клавише ***Enter*** и напишем таким образом, "c" равняется, далее ***new Color***, и в квадратных скобках укажем размер массива ***n***. Далее закроем квадратные скобки. Точка с запятой.



Теперь определим этот цвет внутри нашего таймера, наш текущий цвет. Для этого щелкнем здесь на клавише ***Enter*** и зададим таким образом, "***c***" от ***ni***, от текущего столбика. Пусть равняется, далее зададим новый цвет, ***new Color***, откроем скобку, и теперь зададим новый цвет, как случайную величину. Для этого напишем таким образом, преобразование типа к целому типу ***int***, далее откроем скобку, укажем генератор случайных чисел ***Math.random***, далее введем скобки, далее умножим на 255, поскольку максимальная яркость одного цвета как раз и составляет вот эту величину. Закроем скобку, запятая.

Теперь то же самое нам нужно сделать для остальных двух оставшихся компонентов цвета, поэтому выделим вот этот текст, далее правая кнопка мыши и Copy. Встанем сюда. Правая кнопка мыши и ***Paste***. Щелкнем на клавише ***Enter***. Вставим еще раз для последнего компонента цвета, правая кнопка мыши и ***Paste***. И теперь закроем скобку. Точка с запятой. Теперь же, после того, как вот это массив "***c***", который задает цвета, у нас

проинициализирован и задан, сдвинемся скроллингом ниже и там, где мы раньше задавали красный цвет для нашего столбика, зададим просто-напросто цвет "***c***" от ***i***. Скомпилируем теперь наше приложение и посмотрим, что у нас получилось. Для этого развернем меню ***Tools***, далее ***Compile Java***. Компиляция прошла успешно.

Теперь просмотрим наш аплет внутри ***html*** страницы. Переключимся туда и теперь запустим его в режиме просмотра ***Web*** страницы. Запустим нашу ***html*** страницу. Щелкнем вот на этой кнопке. И теперь можно видеть, что у нас получилось. Как мы видим, наша гистограмма последовательно заполняется соответствующими столбиками с разными цветами. И вот, после того, как все цвета стали заполненными, все началось сначала. Вот таким образом и работает наша гистограмма. Закроем теперь нашу ***html*** страницу. Щелкнем вот на этом крестике. И, перейдя на наш листинг ***Java***, мы опять попадаем в наш текстовый редактор. Используем теперь то обстоятельство, что наша ***html*** страница передает данные в наш аплет и попробуем создать несколько вариантов ***html*** страницы с разными передаваемыми в наш аплет данными.

Перейдем для этого на ***html*** страницу. И далее сохраним вот этот файл под каким-либо другим именем. Для этого развернем меню ***File***, ***Save As***, назовем этот файл ***My_applet***2. Щелкнем на кнопку ***Save*** (сохранить). Вот теперь эта ***html*** страница у нас сохранена под вот этим именем. Теперь щелкнем на кнопке открытия ***Open*** и откроем первоначальную ***html*** страницу ***My_applet***. Щелкнем на кнопку ***Open***. Вот теперь они у нас обе загружены и одну из них немножко переделаем. Например, пусть наши значения будут чуть в меньшем количестве, ну, например, всего 7 значений.

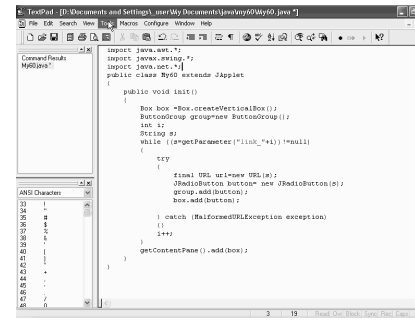
Поэтому последние 2 строки удалим. То же самое сделаем с числами со значениями. И введем какие-либо другие числа вот здесь. Например, сделаем таким образом. Изменим немного все эти значения. И далее попробуем посмотреть, как же все это будет выглядеть в действительности. Для этого запустим теперь этот аплет в обеих наших ***html*** страницах. Сначала сохраним вот эту ***html*** страницу. Щелкнем на кнопке сохранения.

Теперь, прежде чем мы запустим наш аплет, переделаем здесь количество наших элементов, вместо 9, здесь, конечно, должно быть 7, и посмотрим теперь, после сохранения, как все это будет выглядеть у нас в ***web*** обозревателе. Щелкнем вот на этой кнопке. И вот можно видеть процесс работы нашего аплета. Как мы видим, и в этом случае у нас рисуется вот этот график, последовательным заполнением вот этих столбиков разными цветами. Как мы видим, на этот раз эта гистограмма выглядит уже несколько по-другому. Закроем теперь вот эту гистограмму, эту ***html*** страницу. Переключимся теперь на вторую ***html*** страницу и, запустив ее, посмотрим, как она выглядит.

И, как мы видим, мы возвратились к своему первоначальному виду. Т.е. вот таким образом, мы при помощи одного аплета вывели на ***html*** страницу две абсолютно разные гистограммы с разными данными и разным количеством элементов в них. Закроем теперь нашу ***html*** страницу. Щелкнем вот на этом крестике. И вот мы опять в нашем текстовом редакторе.

6.7.5.12. Апплет с закладками. Создание

Рассмотрим теперь, как при помощи наших аплитов можно управлять ***html*** страницами. Для этого создадим, конечно же, простейший аплет и сначала напишем две директивы ***Import***, поскольку, конечно же, пакет ***Java.awt*** и ***Java.Swing*** нам точно понадобятся. Первый из них, это напишем ***Java*** точка ***AVT***. Точка звездочка, точка с запятой. И второй импорт возьмем ***Javax*** точка и пакет ***Swing***. Точка звездочка, точка с запятой. А теперь же, напишем наш основной класс, который у нас будет реализовывать аплет. Поэтому напишем, таким образом, ***public***, далее, конечно же, ключевое слово ***Class***, имя нашего аплита. Пусть будет, например, ***My***60.



И далее, он должен быть наследником класса ***JApplet***. Далее ***Enter***. Фигурные скобки, внутри которых надо обязательно, конечно же, написать тот метод, с которого у нас обычно начинается работа аплета. Это метод ***Init***, начальная инициализация. Поэтому ***public***. Далее тип возвращаемого значения ***void***, поскольку возвращаемого значения нет. И, как мы сказали, метод ***init***. Далее скобки, ***Enter***, и фигурные скобки. Теперь же сохраним этот аплет в каком-либо файле. Для этого щелкнем на значке сохранения. Выберем новую папку для нашего аплита. Щелкнем на вот этом значке, напишем вот такое имя. И щелкнем на кнопку ***Open***. Теперь выберем имя для нашего файла.

Конечно, оно должно совпадать с именем нашего класса, поэтому пусть будет таким и, конечно же, тип этого файла должен быть ***Java***. Теперь же щелкнем на кнопку сохранения, ***Save***. Вот мы опять в нашем текстовом редакторе. Теперь же внутри этого аплита создадим группу переключателей, при помощи которых будем вызывать различного рода ***html*** страницы. Поэтому щелкнем здесь на клавише ***Enter*** и первым образом создадим переключатели, скомпонуем их вертикальным образом и для этого создадим элемент типа ***Box***, который у нас будет управлять вертикальной компоновкой. Назовем его тоже также ***box***, только с маленькой буквой, и далее равняется ***box***

точка, и воспользуемся его методом ***createVerticalBox***, при помощи, которого и создается вертикальное положение наших компонентов, которые мы на нем разместим. Далее скобки. Точка с запятой.

Теперь же, создадим группу, в которую и объединим наши переключатели. Для этого, как обычно, ***ButtonGroup***, группа кнопок, назовем ее просто ***group***. Далее равняется, ***new***, опять то же самое ***ButtonGroup***, теперь нам нужно вызвать его конструктор. Далее скобки. Точка с запятой. А теперь создадим две переменные. Первая, это переменная типа ***integer*** — ***i***, которую мы используем в дальнейшем для цикла и переменную типа ***String***, пусть это будет у нас ***S***, которую используем для хранения класса и имени нашей ***html*** страницы. Причем, конечно же, эти адреса мы можем задавать прямо внутри этого аплета, а можем сделать так, чтобы они передавались у нас, как параметры из нашей ***html*** страницы. Ну, так и поступим.

Организуем теперь цикл. Причем цикл организуем при помощи оператора ***wile***, поскольку, конечно же, заранее мы не знаем количество параметров, которые мы получим из нашей ***html*** страницы. Далее скобки, внутри которых, прочитав в нашей переменной "***S***" значение из соответствующего параметра. Для этого нам нужно воспользоваться методом ***getParameter***.

Далее скобка. Далее нам нужно указать в кавычках имя нашего параметра. Ну, пусть, например, это будет ***link***, далее закрыть кавычки и номер ***i***, плюс ***i***. Закрыть скобку. Еще раз закрыть скобку. И теперь выясним, не равняется ли это значение 0, пустому элементу. Далее не равно и далее 0. Закрыть скобку. Т.е. цикл у нас будет происходить до тех пор, пока у нас каждый раз метод ***getParameter*** что-либо считывает из нашей ***html*** страницы. А как только он ничего не сможет прочитать, в этом моменте цикл и завершится.

Далее фигурные скобки, внутри которых нам и нужно написать, что же должно происходить внутри этого цикла и первым образом напишем, оператор ***try***, для того чтобы мы могли обработать соответствующую исключительную ситуацию, которая может возникнуть, когда мы будем выяснять, является ли эта строка "***S***" корректным адресом. Поэтому фигурные скобки. Далее закрывающая фигурная скобка. Затем, как обычно, оператор ***catch*** и далее внутри скобок мы должны написать ту исключительную ситуацию, ради которой мы и написали оператор ***try***, который мы обрабатываем. Это у нас будет ***MalformedURLException***.

Имя пусть у него будет ***exception***. И далее напишем, что должно происходить. Введем скобки. Введем в цикл увеличение переменной ***i*** на единицу. Далее точка с запятой. И теперь заполним вот наш оператор ***try*** внутри вот этих фигурных скобок соответствующими строчками. Теперь же, первое что сделаем, это преобразуем нашу строчку "***S***", которую мы прочитали из нашей ***html*** страницы при помощи оператора ***getParameter*** в ***URL*** значений,

т.е. в соответствующий адрес. Или в Интернете или на нашем жестком диске. Для этого, во-первых, сделаем его ***final***, поскольку в дальнейшем нам придется этот объект использовать внутри внутреннего класса, который нам потребуется для создания слушателя, обработчика событий.

Далее тип. Это, конечно же, ***URL***. В качестве названия этой переменной возьмем тоже ***URL***, только с маленькими буквами. Далее равняется. ***new***, как обычно, ***URL***, опять. Соответствующий конструктор и внутри скобок напишем нашу строчку "***S***". Закрыть скобку. Точка с запятой.

А теперь же, если у нас так сказать, мы уже создали вот этот адрес, создадим и кнопку для него. Для этого, как обычно, ***JRadioButton***, переключатель, назовем его просто ***Button***, кнопка. Далее равняется. Как обычно, ***new***, ***JRadioButton***, вызываем соответствующий конструктор, далее скобки, внутри которых мы должны написать то имя, которое должно появляться на этой кнопке. Ну, имя на этой кнопке, конечно же, будет вот эта переменная "***S***". "***S***", закрыть скобку. Точка с запятой. Теперь эту кнопку нам, во-первых, нужно добавить в соответствующую группу переключателей, поэтому ***group***, точка ***add***, добавляя вот эту кнопку ***button***, закрыть скобку. Точка с запятой.

И теперь эту же кнопку нам нужно добавить и в нашу панель ***Box***, которая, как располагает их вертикальным образом. ***Box***, точка ***add***. Далее опять-таки внутри скобок наша кнопка текущая ***Button***. Закрыть скобку. Точка с запятой. Теперь пока оставим здесь, как есть вот в этом месте. В дальнейшем нам надо было бы написать обработчик события нажатия на вот этот переключатель. Но сейчас это делать не будем, а напишем просто-напросто создание нашего контейнера, внутри которого и поместим вот эту панель ***Box***. Для этого щелкнем здесь в этом месте на ***Enter*** и создадим контейнер ***getContentPane***. Далее скобки, и внутри него при помощи его метода ***add***, поместим нашу панель ***Box***, как мы сказали. Закрыть скобку. Точка с запятой.

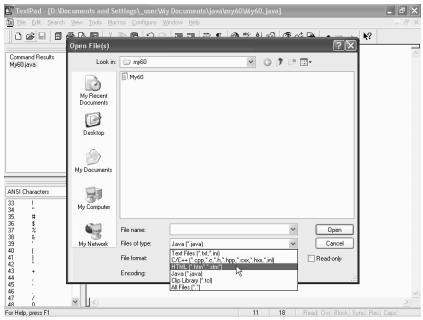
Попробуем теперь скомпилировать наше приложение и посмотрим, как этот аплет действует. Для этого ***Tools***, далее ***Compile Java***. Сначала, конечно же, скомпилируем. Как мы видим, компьютер выдал нам ошибки, которые в принципе понятно, в чем заключаются. А именно, компилятор абсолютно не воспринял все, что касается ***URL*** адресов и соответствующих исключений, которые тоже в принципе связаны с этими же действиями. Ну, конечно, судя по всему, нам не достает соответствующей директивы ***Import***, возможность использоваться соответствующий пакет.

Поэтому переключимся на наш листинг, и введем здесь еще одну директиву ***Import***, при помощи которого разрешим использование пакета ***Java*** точка ***net***, которая как раз и включает в себе всякого рода сетевые операции. Далее точка с запятой. Попробуем опять скомпилировать наше приложение. Для этого

Tools, далее ***Compile Java***. Компьютер опять выдал нам ошибку, которая заключается в том, что переменная ***i*** не проинициализирована.

Переключимся на наш листинг. И, правда, заметим, что мы при создании нашей переменной ***i*** ее не проинициализировали. Если наш цикл ***while*** запуститься, то, конечно же, он не будет знать с чего же начать нашу работу, чему же будет равняться ***i***. Поэтому проинициализируем. Равняется 1, напишем. Теперь скомпилируем. Для этого ***Tools***, далее ***Compile Java***. На этот раз компиляция прошла успешно.

6.7.5.13. Апплет с закладками. Передача адресов для закладок в качестве параметров



Теперь же, запустим этот аплит. Для этого щелкнем на кнопку ***Tools*** и далее ***Run Java Applet***. Вот компьютер запустил нам наш аплит, причем, поскольку ***html*** файла пока еще у нас нет, в котором мы бы разместили этот аплит, он его создал сам временным образом. Но, поскольку он создал просто-напросто самый простой код для создания аплита. В нем, конечно же, никаких параметров нет, то в результате мы получили вот такой пустой аплит, в котором пока еще никаких переключателей нет. Закроем теперь этот аплит. Щелкнем на крестике. Найдем теперь тот временный ***html*** файл, который нам компьютер создал.

Щелкнем на значке открытие файла, переключимся на ***html*** файлы, вот этот файл. Выделим его. Щелкнем на кнопку ***Open***. Вот можно видеть, как он выглядит, и теперь сохраним его в каком-либо файле с нормальным названием, а не с вот этим временным. Для этого щелкнем на значке ***File***, далее ***Save As***, сохранить, как, и теперь напишем имя для нашего файла. Ну, пусть, например, будет просто-напросто ***Page***, и щелкнем на кнопку ***Save*** (сохранить). Теперь же, введем вот этот ***html*** файл в наши параметры. Для этого щелкнем здесь на клавише ***Enter***, далее значки больше меньше, внутри которых нам нужно вписать все наши параметры и вообще все теги. Ключевой слово ***Para***, параметр, затем имя ***Name***, нашего параметра, далее равняется, имя у нас будет ***Ling*** и первый такой ***ling*** у нас будет с номером один, и далее значение, ***value***.

Конечно же, ***Ling*** нам нужно взять в кавычки, далее теперь значение, которое равняется. Опять кавычки и напишем таким образом. Введем какое-либо имя ***html*** файла, которое в дальнейшем будем открывать при помощи

нашего переключателя. Или же надо ввести какой-либо ***URL*** адрес в Интернете. Ну, чтобы сейчас нас было проще, ***URL*** адреса вводить не будем, а просто напишем прямо адрес. И пусть этот файл будет просто-напросто ***p***1 точка и с расширением ***html***. ***Html***, далее закрыть кавычки, и косая черта. Теперь же, повторим этот параметр еще несколько раз.

Для этого выделим. Правая кнопка мыши. Сору. Вставим сюда. Правая кнопка мыши и ***Paste***. Это у нас будет вторая ссылка, и страница тоже вторая. Встанем сюда. Правая кнопка мыши и ***Paste***. Это ссылка у нас будет третья. И страница тоже третья. Встанем сюда. Правая кнопка мыши и ***Paste***. Это у нас пусть будет 4. И страница тоже четвертая. Ну, пожалуй, пока хватит. Сохраним теперь этот файл. Щелкнем на кнопке сохранения. Теперь же переключимся на наш основной файл, и учтем одно обстоятельство. А именно, поскольку мы ввели вместо ***URL*** адресов прямо ***html*** файлы, здесь нам придется немножко переделать.

А именно, нам придется переделать определение нашего ***URL*** адреса. Вместо вот такого определения, при помощи строчки ***S***, мы сначала укажем адрес нашей текущей директории, которую напишем таким образом. ***getCodeBase***. И далее скобки. Теперь опять скомпилируем наше приложение. ***Tools***, далее ***Compile Java***. Компиляция прошла успешно. Теперь переключимся на нашу страницу и попробуем ее запустить. Щелкнем на вот этом значке просмотра в обозревателе. И вот, перед нами появилась наша ***html*** страница с нашим аплитом. Правда, как мы видим, он опять абсолютно пустой. Что-то у нас получилось не так. Закроем теперь нашу ***html*** страницу и попробуем разобраться.

И если мы внимательно присмотримся к нашему ***html*** файлу, то можно заметить, в чем тут дело. Дело в том, что мы все вот эти наши параметры, так сказать, поместили вне нашего блока с аплитом. Вот закрытие аплита происходит гораздо раньше, чем мы определяем наши параметры. Поэтому нам это надо исправить. Схватим за вот этот выделенный кусок текста и потащим его вниз, вот в это место.

Теперь же сохраним. И опять попробуем запустить в ***Web*** браузере, в обозревателе. И на этот раз, как мы видим, все получилось так, как мы и хотели. Появились вот эти все наши переключатели. Правда, нажимая на них, мы, конечно же, пока никакой реакции не получаем. Для этого нам придется еще написать строчки кода с нашим слушателем, обработчиком события и т.д. Закроем теперь, перейдем в наш ***Java*** листинг.

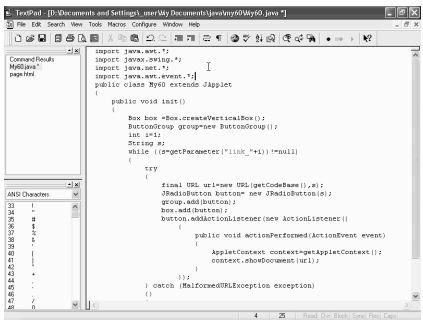
Создадим здесь обработчик нажатия на наш переключатель. Ну, тут как обычно. Наша кнопка ***Button***, далее точка, ***addActionListener***, далее скобка. Теперь напишем ***new***, создадим новый объект этого класса ***ActionListener***, далее конструктор, далее скобки. Создадим непосредственно в этом месте

внутренний класс этого типа. Для этого фигурные скобки, открывающие и закрывающие, далее закрывающая скобка и точка с запятой.

Ну, а теперь, внутри вот этих фигурных скобок нам нужно прописать все методы, которые необходимы для интерфейса ***ActionListener*** и это, конечно же, метод ***ActionPerformed***. Поэтому напишем, таким образом, ***public***, далее тип возвращаемого значения, ***void***, поскольку никакого возвращаемого параметра нет.

Далее, как мы сказали, ***ActionPerformed***. Далее скобка. Тип передаваемого параметра ***ActionEvent*** и сам передаваемый параметр тоже пусть будет ***event***. Закрыть скобку. ***Enter***. Далее фигурные скобки. И теперь получим контекст нашего аплета. Для этого создадим объект, типа ***AppletContext***. Пусть этот объект называется просто ***Context***. И для того, чтобы его получить, напишем так, равняется и метод ***getAppletContext***. Далее, после того, как мы его получили, щелкнем на клавише ***Enter***. Затем внутри этого контекста покажем, конечно же, другой документ. Для этого используем его метод ***Showdocument***, показать документ. Далее скобки, внутри которых нам теперь, надо указать непосредственно имя этого документа, его адрес, в качестве которого, конечно же, возьмем вот это ***URL***. Напишем его сюда. Далее закрыть скобку. Точка с запятой.

Теперь, прежде чем мы скомпилируем наше приложение, добавим еще директиву ***Import***, которая необходима, во-первых, для того чтобы мы могли использовать интерфейс ***ActionListener***, для этого нам нужна директива ***Import***. И далее ***Java***, точка ***awt*** и точка ***event***, поскольку именно вот в этом пакете сконцентрированы все наши возможности по работе с событиями. Далее добавим еще одну директиву ***Import***, при помощи которой, добавим возможности работы с аплитами. Для этого ***Java***, точка аплит. И точка звездочка. Точка с запятой.



А теперь скомпилируем наше приложение. Для этого ***Tools***, далее ***Compile Java***. Компиляция прошла успешно. Теперь переключимся на нашу ***html*** страницу ***Page*** и в предположении, что вот эти ***html*** страницы ***p***1, ***p***2, ***p***3 и ***p***4 у нас созданы независимо от того, что мы сейчас делаем (этим вопросом сейчас заниматься не будем) и помещены в нашу текущую директорию в этом предположении запустим нашу основную ***html*** страницу ***Page***, в котором у нас и находится наш аплит. Щелкнем. Вот она появилась перед нами, выберем первый,

например, переключатель и, как мы видим, мгновенно мы перескочили на вот эту страницу ***It is page*** 1. Вернемся обратно. Мы опять попали в том же самый аплит. Выберем вторую страницу.

Вот она перед нами. Если мы вернемся и выберем, например, четвертую страницу, то она точно также открывается перед нами. Как мы видим, весь наш вот этот аплит и все эти переключатели, работают совершенно нормально. Закроем теперь эту ***html*** страницу, и мы опять в нашем текстовом редакторе. Попробуем теперь сделать так, чтобы все это у нас работало в ***html*** странице с фреймами, т.е. в том случае, если у нас наша ***html*** страница главная, разделена на несколько частей. Но в нашем случае возьмем просто-напросто две части, в одной из которых отобразим наш аплит, а во второй, мы будем отображать, щелкая на наш переключатель, те страницы, которые при этом должны появляться.

Ну, страница с фреймами, которая будет управлять всем этим делом, у нас записана заранее. Попробуем ее открыть. Для этого щелкнем на значке открытия файла и откроем ***html*** страницу с названием ***My***. Щелкнем на кнопку ***Open***. Вот она появилась перед нами. И в нем, как мы видим, с левой стороны во фрейме под именем ***left*** отображается наша страница ***Page***, в котором у нас находится аплит. А с правой стороны во фрейме с названием ***right*** у нас отображается просто некая ***html*** страница с названием ***p*** точка ***html***. Рассмотрим. Все это выглядит на данном этапе. Для этого щелкнем на значке Отобразить в обозревателе, в ***Web*** браузере. И вот внешний вид нашей ***html*** страницы мы можем видеть. И как мы и обещали, с левой стороны у нас находится аплит, а с правой стороны — некая ***html*** страница.

Правда, если мы выберем какой-либо из этих переключателей. Ну, возьмем, например, второй, то вот эта страница ***Page***2, отображается не с правой стороны, а прямо на месте исходной ***html*** страницы. Конечно, мы можем вернуться и проделать все то же самое. Но цель у нас сейчас другая. При помощи вот этих переключателей чтобы менялся вот этот правый фрейм. Закроем теперь наше окно. Запомним вот это имя правого фрейма ***right*** и переключимся на наш листинг. И теперь, в том месте, где мы показывали наш документ, где использовали метод ***Showdocument***, напишем, таким образом, ***URL***, далее запятая, и именно в этом месте мы и должны указать имя фрейма, внутри которого мы хотим отобразить нашу страницу. Для этого кавычки и далее, как мы сказали, там должно быть ***right***, как мы запомнили.

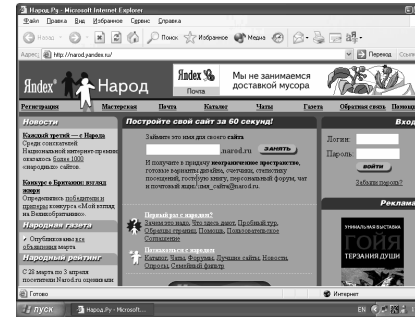
Скомпилируем теперь опять вот это наше приложение. Для этого ***Tools***, далее ***Compile Java***. Теперь же переключимся на нашу ***html*** страницу с закладками, на вот эту. Щелкнем и посмотрим ее в ***Web*** браузере в обозревателе. Щелкнем. Вот она появилась перед нами, щелкнем теперь на какой-либо странице, например, на вот этой третьей, и как мы видим, вот эта

надпись ***It is page*** 3, появилась в правом фрейме. Переключимся на какой-либо другой, например, на первый и, как мы видим, номер страницы поменялся на первый. Т.е. таким образом, мы можем менять фрейм с правой стороны от нашего окна. Закроем теперь наше приложение и мы опять в нашем текстовом редакторе.

Раздел 7. Публикация в WEB

7.1. Регистрация сайта и загрузка файла на сайт

Мы с Вами начинаем новую тему, которая называется Публикация Веб. После того, как Вы создали свои собственные страницы, пришла пора опубликовать результат Ваших усилий на Веб. Что для этого необходимо? Во-первых, Вы должны купить у какого-либо из провайдеров имя Вашего домена, то есть, то название, которое будет задавать пользователь в адресной строке. Кроме того, нужно будет купить, так называемый ***Hosting***, то есть, место хранения Ваших страниц на серверах Вашего провайдера. Вы, вероятнее всего, ужаснетесь, так как все упирается в купить, купить, купить.



Однако Вы можете опубликовать Ваши страницы и совершенно бесплатно. Большое количество серверов в Интернете предоставляет такую услугу. Взамен этого они помещают на Ваших страницах рекламу. Попытаемся опубликовать что-нибудьв Интернете. Для этого запустим какой-нибудь из браузеров, например ***Internet Explorer***. Мы заранее знаем, что услуги такого бесплатного предоставления веб-***Hosting***-а предлагает ***www.narod.ru***... Попробуем поработать здесь. Перед нами открылась страница ***Narod.ru***. На самом деле домен называется ***narod.yandex.ru***. Именно на этом сервере мы с Вами сейчас зарегистрируем наш домен. Прямо с первой же головной страницы нам предлагают занять имя для своего сайта. Имя будет состоять из какого-либо названия плюс ***narod.ru***. И мы получим в придачу неограниченное пространство, готовые варианты дизайна, счетчик, статистику посещения, гостевую книгу и тому подобное.

Давайте же, попробуем зарегистрироваться здесь. Допустим, мы хотим, чтобы наша страница называлась ***sergey_zverev.narod.ru***. Нажимаем на кнопку ЗАНЯТЬ. Открылась регистрация нового ***Account***-а на Яндексе. Нас просят аккуратно и добросовестно заполнить приведенную здесь регистрационную форму. Давайте, мы с Вами аккуратно заполним ее. Сначала введем пароль... Теперь подтвердим его... Контрольный вопрос нужен на тот случай, если мы забудем пароль для нашей веб-страницы. Здесь у нас, в общем-то, выбор

небольшой: наше прозвище в школе, дата рождения бабушки и так далее и Ваш любимый номер телефона.

Выберем, к примеру, любимое блюдо, хотя тот пароль, который мы с Вами ввели, никак не относится к любимому блюду. Какое же выбрать любимое блюдо? Напишем что-нибудь стандартное, например, Щи. Бессмысленный вопрос — и бессмысленный ответ. Наш ***E-mail*** (если он у нас имеется). ***E-mail*** у нас, безусловно, имеется: ***Karapeta@mail.ru***. Наш день рождения. Он никого не интересует, поэтому мы напишем, например, 1 января 2000 года (что, безусловно, не так). Персональные данные. Обращения. Уважаемый. Здесь напишем, к примеру, ***Sergey***, а здесь — ***Zverev***. Псевдонима у нас нет. Пол — мужской. Страна. Странау нас — Россия. Город — Москва. Сферу занятия мы никому не выдадим. Формат писем — ***html***. Прекрасно. Теперь нажмем на эту кнопку ОК.

Нас с прискорбием извещают о том, что ***Login sergey_zverev*** уже занят. Мы можем выбрать нечто другое. Посмотрим, что нам предлагают. Можно, например, выбрать нечто типа ***zver_sergey***. К сожалению, пароль нам придется ввести заново... и подтвердить его... Посмотрим, что еще у нас здесь есть. Вроде, все остальное сохранилось. Прекрасно. Еще раз нажимаем на кнопку ОК. Оказывается, мы здесь не подставили... Сейчас мы с Вами подставим. Здесь надо заполнить. Опять-таки — пароль... Нажать Подставить (уже после того, как введен пароль) и еще раз нажать на кнопку ОК. Сейчас у нас есть имя нашего будущего сайта. Это — ***zver_sergey.narod.ru***. Подписаться ли на общие новости сайта или на новости народной газеты? Мы с Вами ни на что не хотим подписываться, а просто нажимаем на кнопку Сохранить. Перед нами открылась, так называемая, управляющая страница.

Проект наш называется ***zver_sergey.narod.ru***. Теперь мы имеем следующую возможность. Мы можем создать сайт, можем редактировать и управлять имеющейся информацией, можем индексировать в поисковых системах и так далее. Начнем с самого начала. Мы загрузим файлы на сайт.

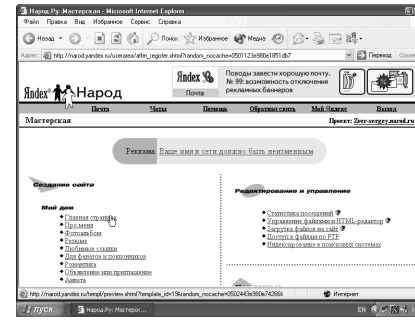
Что мы для этого делаем? Здесь открывается страница, с помощью которой мы можем загрузить файлы на наш сайт. Для этого достаточно нажать на кнопку Обзор, выбрать здесь любой файл и загрузить его. Как это делается? Выбирается кнопка Обзор, затем выбирается любой из файлов на нашем жестком диске (в данном случае открыт наш Рабочий стол) и нажимается кнопка Открыть. И, если сейчас мы нажмем на кнопку Загрузить файлы, данный файл будет загружен на наш веб-сервер. Попробуем сделать это. Кроме того, здесь имеется предупреждение о том, что объем каждого файла не должен превышать 5 мегабайт. В данном случае мы имеем маленький файл. Это тот самый файл ***Doc***, который мы с Вами создавали на одном из предыдущих уроков по созданию веб-страниц. Сейчас этот файл загружается

на наш сервер. Нам сообщают о том, что файл ***Doc***2.***htm*** загружен в корневую папку. Он уже там. Если мы нажмем на кнопку Назад, то сможем попасть в предыдущее окно. Мы с Вами не нажали на кнопку Назад, поэтому сейчас сервер ***narod.ru*** все делает самостоятельно. Итак, один файл у нас там уже есть. Мы можем также проверить свободное место.

Для этого достаточно щелкнуть по этой ссылке. Нам сообщается о том, что всего доступно вот такое пространство. Занято — всего 1 килобайт, а свободно 102399 килобайтов. Вернемся назад и посмотрим, что мы, все-таки, делаем. Итак, как закачать файл на сервер ***zver_sergey.narod.ru***, мы с Вами уже поняли.

Мы научились работать с этим окном, которое называется Загрузка файлов в разделе управления файлами. На этом завершим текущий урок, а на следующем уроке мы продолжим работу с этим сервером ***narod.ru***.

7.2. Редактирование опубликованного сайта



На прошлом уроке мы с Вами начали изучать тему Публикация в Веб. Продолжим работу с этим очень интересным веб-сервером. Перейдем по ссылке Мастерская и рассмотрим все имеющиеся здесь возможности. Начнем с подраздела, который называетсяСоздание сайта. Перейдем на главную страницу. Здесь мы можем выбрать вариант шаблона Главная страница сайта. На самом деле публикация в Веб может производиться двумя вариантами. Первый вариант. Вы можете прямо на сервере с помощью его возможностей и его инструментов создать свой собственный веб-сайт. Второй вариант. Вы можете загрузить прямо на сервер уже имеющиеся страницы, созданные Вами заранее с помощью ***Netscape Composer***-а вручную, или с помощью еще какого-нибудь ***HTML***-редактора.

Сейчас мы с Вами начинаем с первого варианта. С помощью инструментов, которые имеет данный сервер ***narod.ru***, мы пытаемся создать свою страницу. Сейчас мы выберем тот вариант шаблона, который отвечает нашим требованиям. Первый файл должен называться ***index.html***. Мы уже знаем, что открытие страницы обычно начинается с этого файла ***index.html***. Поэтому мы с Вами не собираемся сейчас менять его.

Давайте же, выберем один из вариантов оформления. Пусть это будет, к примеру, Сиреневый. Теперь нажмем на кнопку Дальше. Этим самым мы

выбрали нашу самую первую страницу, которая будет открываться при наборе строки адреса ***zver_sergey.narod.ru***. Кроме того, мы сейчас это будем проверять. В мастерской нам сообщают, что мы создаем главную страницу ***index.html***. О чем сайт? Это персональный сайт. Здесь мы можем ввести текст ссылки и так далее, и выдать ***E-mail***. Начнем. О чем сайт?

Допустим, что сайт Обо мне любимом... Текст. Введем, к примеру, СергейЗверев приветствует Вас... Этого пока достаточно. Теперь мы с Вами будем выбирать какую-нибудь ссылку. Пусть у нас на странице будут две ссылки. Одна из них будет вести в гостевую книгу, а другая — на какой-нибудь следующий сайт. Этот сайт у нас так и будет называться — Ссылка... А здесь мы латинскими буквами так и введем... Прекрасно. Дополнительная информациянам не нужна. Мы просто нажмем на кнопку Готово. Сейчас нам показывают примерную версию нашей страницы.

Она называется Обо мне любимом. Текст — Сергей Зверев приветствует Вас. Далее следует информация: Ниже следуют ссылки на различные разделы моей странички. Это у нас — ссылка. К сожалению, этого файла у нас сейчас нет. Поэтому, давайте, приостановим это, потому что нам сообщили, что такой страницы нет. Еще тут есть Напишите мне. Здесь указан наш ***E-mail***. Если мы сейчас щелкнем по этой строке, откроется наш почтовый клиент (в данном случае это ***Outlook Express***). Если мы выберем здесь Не делать эту программу по умолчанию, то у нас откроется вот эта возможность прямо сейчас написать письмоавтору этой страницы. Изменения мы сохранять не будем, и посмотрим, что представляет собой эта ссылка.

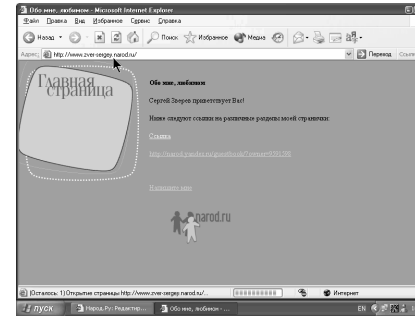
Мы это сделаем в новом окне. Контекстное меню мы вызвали нажатием правой клавиши мыши. Как Вы помните, здесь у нас должна иметься ссылка на гостевую книгу. Посмотрим, так ли это. Да, это гостевая книга сайта ***sergey_zverev***. Мы даже можемдобавить туда сообщение. А пока вернемся на нашу страницу и посмотрим, что еще мы с Вами можем отредактировать. Кроме всего, в этой копии браузера мы можем посмотреть, что у нас реально получается. Введем ***www.zver_sergey.narod.ru***...

После того, как мы ввели этот адрес в адресную строку браузера, мы окажемся на той странице, которую мы только что создали. Причем, это нормальный интернетовский адрес: он доступен с любой точки захода в Интернет. Вот она — Сергей Зверев приветствует Вас. А теперь вернемся в ***narod.ru*** и посмотрим, что еще мы можем здесь сделать. Это — главная страница, и мы здесь редактируем главную страницу ***Index.html***. Сейчас мы с Вами не собираемся редактировать главную страницу ***Index.html***, а просто хотим создать вторую страницу, которая будет называться Ссылка. Сейчас в этой левой части мы видим все возможности создания наших сайтов, то есть, все тематики. Мы можем просто создать нашу домашнюю страницу, которая

имеет главную страницу, информацию про меня, фотоальбом, резюме и так далее, и любимые ссылки, и анкету.

Мы можем ориентироваться на Бизнес или на Обращение. Итак, мы продолжаем изучение нашей темы. Выполним следующую страницу, которая называется Про меня. Сейчас нам достаточно выбрать вариант шаблона, который наиболее полно отвечает нашим требованиям. Делаем это точно так же, как и в прошлый раз. Мы выбираем, например, вот такой таинственный. Нашему файлу зададим имя Ссылка... Он будет иметь расширение ***html***. Нажмем на кнопку Дальше. Сейчас мы можем определенным образом настроить дизайн. Как мы это сделаем? Показывать ли заголовок про меня? Мы сбросим этот флажок. Выберем цвет фона. Пусть это будет вот такой сиреневый цвет. Приветствие. Введем, например,

Привет всем... Цвет текста. Выберем вот такой синий. Шрифт — Жирный. Курсив. Выберем размер побольше, например, 32. Фотографий у нас в данный момент нет, но с помощью кнопки Обзор мы с Вами можем выбрать любое изображение, которое нам сейчас доступно. К сожалению, ничего подходящего, по-моему, даже в Моих рисунках, нет. Здесь имеются образцы рисунков. Выберем, к примеру, водяные линии. Текст Вашей странички. Напишем следующее: Сюда Вы перешли по ссылке... Вот так. Цвет текста у нас будет вот такой зеленый. На этой странице точно таким же образом можно организовать до пяти ссылок на другие страницы, но мы с Вами сейчас не будем этого делать. Нажмем на кнопку Готово. Именно так будет выглядеть та страница, которую мы только что создали. Привет всем. Именно синими буквами. То есть, именно так, как мы с Вами и устанавливали. Кроме того, сейчас открывается та страница, которую мы загрузили в качестве фотографии. Информация у нас осталась здесь сбоку. Эта фотография сейчас открывается. Посмотрим, что у нас здесь получилось.



Если мы обновим эту страницу и попробуем перейти на только что созданную страницу с помощью этой ссылки (пробуем), то у нас ничего не получится. Почему это произошло? Дело в том, что возможность перехода мы с Вами поставили на Ссылкабез ***html*** (без ***.html***). А здесь наша страница называется "Ссылка, точка, ***html***".Вернемся в Мастерскую и получим с помощью этого простого инструмента сервера ***narod.ru*** вполне работающую страницу.

Перейдем на нашу главную страницу, выберем что-нибудь еще раз и нажмем на кнопку Дальше. Нам сообщают, что у нас уже имеется страничка с таким именем ***Index.html*** и пока не предоставляют нам возможность редактирования данной страницы. Но мы попробуем нажать на кнопку Дальше и посмотрим, может, все-таки, нам предоставят эту возможность.

Нет, такую возможность нам не предоставляют. Вернемся в Мастерскую и попробуем отредактировать нашу страницу. Редактировать нашу страницу мы можем с помощью этой группы, которая расположена справа. Управление файлами, ***html***-редактор. Это именно редактирование и управление. Мы щелкаем мышью вот сюда.

Сейчас мы видим все те страницы, которые имеются на нашем сервере. У нас имеются страницы по шаблону, ***index*** и ***ssilka***.Имеется та страница, которую мы с Вами загрузили совершенно отдельно и еще имеется картинка. Предположим, нас интересует вот эта самая страница по шаблону. Здесь устанавливаем флажок и нажимаем на кнопку Свойства. Сейчас у нас имеются данные об этом самом файле ***Index.html***. У нас имеется имя и описание, имеется счетчик, его каталог, текстовый редактор и редактор по шаблону. Выполним щелчок мыши по Текстовому редактору.

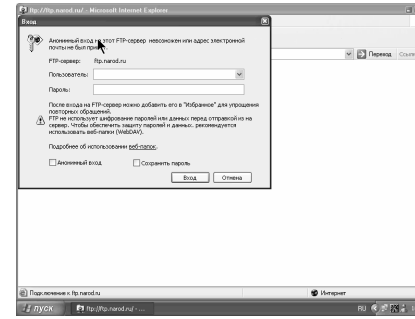
Посмотрим, что мы при этом получим. Мы здесь сейчас получаем в виде ***html*** (а мы уже знакомы с этим языком) вот эту нашу страницу с помощью текстового редактора. Если мы вызовем Редактор по шаблону, то мы увидим тот самый редактор, с помощью которого мы и создавали эту страницу. Попытаемся отредактировать это в ***HTML***. Мы уже знаем, как это делается. Мы помним, что у нас была ссылка, которая называется ***HREF***-ссылка, а должна была быть ссылка ***HREF-ssilka.html***.

Нажмем на кнопку Сохранить. Изменения сохранены. Мы можем продолжить редактирование, а можем и просто закрыть окно. Давайте, закроем это окно и посмотрим, что у нас здесь получилось. В другой копии приложения мы вернемся назад, то есть, фактически, мы возвращаемся на головную страницу. Сейчас она откроется. Будет открыта страница ***www.narod.ru***. Теперь нажмем на эту самую ссылку ***html***. Причем, если Вы обратите внимание, на статусной строке уже указывается правильная ссылка. Если же мы сейчас щелкнем здесь (а файл ссылка.***html*** на веб-сервере у нас уже имеется), то мы перейдем в правильном направлении. У нас открылась та страница, которую мы с Вами создали во второй раз.

Имеется приветствие, имеется пара ссылок. Именно сюда мы перешли по ссылке. И сейчас здесь постепенно будет открываться нужная фотография, которая называется ***Picture.jpg***. На этом завершим наш урок. Итак, нам с Вами удалось создать вполне нормальную страницу, которая уже опубликована и находится в Интернете. Сейчас у нас данная страница открывается. Текст

расположен вот здесь справа. Может, это не самый удачный вариант дизайна, но представление о том, как публикуются страницы в Веб, Вы получили.

7.2. Доступ к файлам по FTP



На прошлом уроке мы с Вами научились создавать веб-страницы с помощью встроенного редактора сервера ***narod.ru***. А теперь мы рассмотрим некоторые другие возможности работы с этим самым сервером. Перейдем в Мастерскую и посмотрим, какие еще возможности поддерживает данный сервер. Одной из самых интересных возможностей, которую поддерживает данный сайт, является доступ к файлам по ***FTP***. Мы с Вами уже рассматривали, как работают с ***FTP***. Сейчас предположим, что у нас уже имеются заранее готовые страницы. Давайте, минимизируем оба окна нашего браузера. У нас имеется определенное количество страниц, которые мы с Вами создавали на предыдущих уроках (вот они все). Допустим, что мы хотим загрузить их на данный сервер. Для этого мы щелкаем по ссылке Доступ к файлам по ***FTP***. Здесь у нас сейчас имеется информация на этот счет. Во-первых, нам объясняют, для чего нужен ***FTP*** и, что это такое. Для того, чтобы воспользоваться протоколом ***FTP***, необходима специальная программа, так называемый, ***FTP***-клиент. Мы знаем, что ***FTP***-клиентом может быть и Интернет-браузер, и такая программа, как ***CuteFTP*** и так далее. ***FTP***-клиент также встроен в некоторые ***HTML***-редакторы. И вот здесь объясняется, как именно настроить ***FTP***-сервер. Попытаемся с Вами сделать это. Допустим, что у нас имеется***Netscape Composer***.

Попробуем настроить его для работы с ***FTP***. Публикация в ***Web*** осуществляется с помощью кнопки ***Publish***. Здесь мы дадим определенную информацию, после чего наша страница спокойнозакачается в Интернет. Откроем какую-нибудь страницу. Предположим, нас интересует закачка вот этой страницы, которая называется ***MyPage***. Это та самая первая страница, которую мы с Вами сооружали в Интернете. Она так и называется ***MyPage***. Теперь нажимаем на кнопку ***Publish***. Здесь мы должны задать имя сайта. Допустим, имя сайта называется ***Sergey***... А вот здесь мы должны дать адрес публикации. Его мы задаем посредством вот этой страницы с Помощью. ***FTP***-адрес у нас так и называется ***ftp.narod.ru***.

Давайте, скопируем этот адрес и перейдем на нашу страницу (вот она). Это у нас — ***FTP***-адрес. ***HTTP***-адрес нашей страницы так и называется ***http***://***zver_sergey***.***narod***.***ru***. Причем, здесь надо исправить и вставить сюда ***www***. Теперь нам нужно ввести сюда имя пользователя и пароль. Вернемся на эту страницу. Здесь нам сообщается о том, что имя пользователя и пароль — это наш логин в Яндекс паспорте. И, соответственно, — наш пароль. Вводим здесь ***Usename*** — ***zver_sergey***..., а ***Password*** у нас — большая тайна... Исправим здесь, мы не совсем правильно задали...

Теперь у нас начинается загрузка ***MyPage***.***htm*** на страницу ***ftp***.***narod***.***ru***. Нам сообщают о том, что эти несколько файлов уже успешно загружены туда. Проверим, что у нас здесь получилось. Если здесь уже с самого веб-сервера мы вызовем страницу, которая называется ***MyPage***.***htm***, то, по идее, она у нас должна открыться. Сейчас посмотрим, откроется она или нет. Вот, эта страница у нас открылась уже с самого Интернет-сервера. Иначе говоря, эта страница сейчас находится прямо в Интернете. О чем это говорит? Это говорит о том, что мы с помощью программы ***Netscape Composer*** успешно закачали все это на ***FTP***.

Попробуем сейчас сделать следующее. Закроем ***Netscape Composer***. Как это делать с его помощью, мы с Вами уже знаем. Попробуем теперь сделать это с помощью обычного браузера. Поступим следующим образом. Введем здесь ***ftp***:***ftp***.***narod***.***ru***. Мы сейчас заходим прямо на ***FTP***. Правда, следует отметить, что нам сейчас необходимо сделать следующее. Нам сейчас необходимо ввести имя пользователя и пароль.

Нам сообщается о том, что анонимный вход на этот ***FTP***-сервер невозможен. Наш ***Login*** — ***zver***.***sergey***, а пароль — он для того и пароль, чтобы его не разглашать. Сейчас мы подключаемся прямо к ***FTP*** данного Интернет-сервера. На нем мы видим все те страницы, которые мы туда загрузили. Попробуем сделать следующее. Мы свернем это окно браузера. Выберем здесь все имеющиеся файлы вот таким образом. Правая клавиша мыши. Копировать. А затем сюда мы эти файлы Вставим. Пусть все наши файлы будут закачаны на этот ***FTP***-сервер ***narod***.***ru***.

В чем состоит идея того, что мы делаем? С помощью сервера ***narod***.***ru*** мы показываем способы публикации в ***Web***. Первый способ заключался в опубликовании в ***Web*** с помощью встроенных инструментов данного сервера. Второй способ заключался в опубликовании в ***Web*** наших страниц с помощью ***Netscape Composer***-а и его соответствующих настроек для работы с ***FTP***. А сейчас мы с Вами просто без каких-либо дополнительных возможностей с помощью обычного браузера публикуем наши страницы на ***Web***. Как видите, сейчас все наши файлы копируются в нашу папку на сервере ***narod***.***ru***. Но здесь нужно помнить о том, что открытие страницы начинается с файла,

который называется ***index.html***. Сейчас, после того, как данные файлы закачаются на наш ***FTP***, мы с Вами попробуем переименовать эти самые документы и, соответственно, открытие домена ***zver.sergey.narod.ru*** каждый раз будет производиться с разного файла.

Сейчас нам сообщают о том, что папка уже содержит файл с именем ***MyPage.html***. Мы можем заменить имеющийся файл, а можем и не заменять его. Мы нажмем на кнопку Нет и, тем самым, заменять его, соответственно, не будем. То же самое касается и другого файла, который называется ***Doc***2.***html***, и который мы загрузили на предыдущих уроках. И точно также мы нажимаем на кнопку Нет. Итак, все файлы уже закачаны на наш ***FTP***.

Как видите, здесь уже имеется файл ***index.html***. Откроемвторую копию приложения ***Internet Explorer***. Введем сюда ***zver_sergey.narod.ru***. Сейчас у нас открывается вот эта страница, которую мы создали с помощью встроенных инструментов данного сервера. Перейдем на наш ***FTP*** и попытаемся переименовать какую-нибудь из этих страницв ***index.html***. Переименуем, например, ***MyPage.htm***. Здесь мы вводим ***index***... и, соответственно, ***html***. А затем нажимаем на ***Enter***.

Если теперь мы еще раз обновим данную страницу, то у нас откроется уже та страница, которую мы с Вами назвали ***index.html***. Вы видите, что доменное имя при этом остается неизменным. На этом мы завершим данный урок. Мы с Вами научились работать уже напрямую с ***FTP*** и публиковать страницы там. На следующем уроке мы научимся регистрировать наши страницы в поисковых системах.

Предметный указатель

A

Amount, 269

B

BASE, 126, 154
BASEFONT, 118
Blend Options, 276
BLOCKQUOTE, 120, 121
Blur, 240, 269
BODY, 4, 113, 115, 116, 130, 144, 146
Browser, 239, 268, 288, 428, 445, 516, 536

C

cellpadding, 132
cellspacing, 132
cite, 121
class, 111, 115, 441, 447, 448, 514, 516
CODE, 117, 118
Colors, 164, 166, 168, 170, 174, 249, 269, 396, 397, 407, 416
colspan, 131, 145
Compile Java, 442, 443, 444, 447, 451, 453, 457, 458, 459, 460, 461, 463, 464, 467, 468, 469, 471, 472, 473, 474, 477, 478, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 492, 493, 494, 495, 496, 499, 502, 504, 505, 507, 508, 510, 511, 512, 513, 515, 518, 519, 520, 524, 525, 527, 529, 531, 536, 537, 539, 540, 543, 544, 545, 546, 548
Ctrl, 21, 22

D

Delete Settings, 270
Dither, 257, 269, 286
div, 101
Document Color Mode, 265
Domain Name System, 82
Drag, 21
Duplicate Layer, 274

E

Edit Output Settings, 253, 270
Embedded file, 279, 283
Enter, 15
Eyedropper Tool, 266

F

Flash embedded in HTML, 286, 288
FONT, 118, 187
FORM, 114, 139, 140, 142, 152
FRAME, 145, 165
FRAMESET, 144, 145

G

Generalized Markup Language, 105
GIF, 6, 59, 73, 88, 89, 90, 91, 92, 93, 134, 135, 234, 235, 237, 238, 239, 246, 247, 250, 251, 254, 255, 256, 258, 259, 262, 263, 264, 265, 267, 268, 269, 270, 271, 279, 283, 284, 285, 286, 287, 311
GML, 105

H

Hand Tool, 265
HTML, 4, 5, 7, 8, 31, 32, 33, 34, 81, 85, 87, 94, 96, 97, 98, 99, 102, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 115, 116, 120, 122, 124, 126, 128, 130, 131, 132, 134, 137, 138, 139, 142, 143, 144, 145,

146, 147, 149, 151, 152, 154, 156, 237, 242, 244, 246, 255, 265, 266, 268, 270, 271, 272, 275, 277, 278, 279, 283, 286, 287, 288, 289, 352, 354, 356, 359, 365, 390, 392, 406, 407, 408, 409, 412, 421, 423, 425, 426, 428, 429, 430, 431, 432, 442, 444, 445, 446, 451, 551, 554, 555
HTML 3.2, 107
HTML 4, 107, 112
HTML and Images, 271, 272
Hyperlink, 412, 413, 414, 419, 422
Hypertext, 87

I

IFRAME, 147
Image map, 136
IMG, 114, 120, 133, 134, 135, 136, 137, 138, 140
INPUT, 140, 141, 142, 151, 153
Interlaced, 269
Internet, 7, 16, 31, 32, 58, 59, 82, 87, 102, 103, 107, 108, 115, 220, 239, 240, 242, 258, 268, 271, 279, 280, 281, 282, 287, 288, 295, 296, 306, 316, 317, 319, 320, 321, 323, 324, 325, 326, 328, 329, 330, 331, 332, 334, 335, 336, 337, 339, 341, 342, 345, 347, 349, 350, 351, 352, 354, 355, 358, 359, 361, 362, 364, 365, 366, 368, 371, 372, 374, 375, 376, 377, 378, 402, 403, 413, 420, 429, 436, 437, 445, 514, 516, 549, 557
Internet Explorer, 7, 16, 82, 87, 107, 108, 115, 239, 240, 242, 268, 271, 287, 288, 295, 296, 306, 316, 317, 319, 320, 321, 323, 324, 325, 326, 328, 329, 330, 331, 332, 334, 335, 336, 337, 339, 341, 342, 345, 347, 349, 350, 351, 352, 354, 355, 359, 361, 362, 364, 365, 366, 368, 371, 372, 374, 375, 376, 377, 378, 402, 403, 413, 420, 429, 436, 437, 445, 516, 549, 557
IP-адрес, 79, 82, 83

J

JPEG, 59, 88, 89, 91, 92, 154, 173, 182, 234, 235, 237, 238, 239, 241, 242, 244, 246, 247, 251, 254, 265, 268, 269, 271, 279, 283, 285, 287, 311, 402

K

KBD, 117

L

Looping, 275
Lossy, 269

M

Macromedia Flash, 5, 157, 275, 276, 277, 278, 291
Make Text Web Compatible, 279, 280, 287
MAP, 137, 155
Matte, 258, 269

N

nbsp, 120, 121
Netscape Navigator, 87, 99, 107, 115, 316, 317, 318, 319, 320, 323, 324, 325, 328, 329, 331, 335, 377
NOBR, 131

O

Ok, 15
Only Web Colors, 266
Optimize to File Size, 270
OPTION, 143, 151

P

parent, 146, 413
Pencil Tool, 274

Preferences, 8, 94, 158, 400, 401, 402, 403, 428, 447
Progressive, 240, 251, 269, 402
Publish To PDF, 289, 290

R

Read Only, 275
Release to Layers (Build), 278
Release to Layers (Sequence), 274, 276, 278
Repopulate Views, 270
rowspan, 131, 145
Run Java Applet, 444, 447, 515, 516, 519, 527, 531, 544
Run Java Application, 443, 444, 447, 451, 453, 457, 458, 459, 463, 464, 465, 469, 471, 472, 473, 474, 477, 478, 480, 483, 484, 485, 486, 487, 488, 489, 490, 492, 493, 494, 495, *496, 499, 502, 504, 505, 506, 508, 510, 511, 512, 513, 518, 520,* 524, 525

S

Save Settings, 270
self, 146, 413
Shift, 21, 22
shy, 121
Slice Select Tool, 266
small, 297
STRIKE, 118
STRONG, 114, 116
SUB, 120, 121, 151
SUP, 120, 121

T

table, 130, 188, 247
TEXTAREA, 142
TFOOT, 130, 131
TH, 130, 132
THEAD, 130, 131
Toggle Slices Visibility, 266
top, 130, 132, 136, 146, 420
Transparency, 251, 256, 257, 258, 269

U

Uniform Resource Locator, 83, 125
Unlink All Slices, 270

W

W3C, 112
Web, 3, 4, 5, 6, 7, 8, 10, 31, 32, 33, 34, 72, 73, 75, 77, 78, 80, 88, 90, 95, 98, 99, 102, 104, 106, 107, 157, 186, 220, 234, 236, 239, 240, 241, 242, 243, 244, 245, 246, 255, 256, 257, 258, 259, 262, 264, 265, 266, 267, 269, 270, 271, 272, 273, 277, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 300, 313, 314, 316, 329, 331, 345, 378, 380, 383, 390, 393, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 410, 411, 412, 414, 416, 417, 418, 420, 421, 422, 423, 425, 426, 428, 429, 430, 432, 445, 516, 520, 526, 528, 529, 530, 536, 539, 540, 545, 547, 548, 555, 556
Web Image Optimizer, 283, 284, 285, 286
Web Snap, 269
Web-сайт, 4, 5, 31, 33, 77, 90, 102, 157
Web-страница, 31, 32, 220, 242, 244, 246, 258
WorldWideWeb, 4, 80, 83
WWW, 4, 31, 80, 81, 82, 84, 99

X

XML, 8, 107, 392, 404, 430, 431,432

XSL, 430, 431

Z

Zoom Level, 267

Zoom Tool, 266

А

атрибут элемента FORM, 139

Б

Браузер, 81, 87, 120, 317, 335, 379, 403

В

Векторная графика, 51, 60, 61, 62, 63, 195

Видео, 294

Всемирная паутина, 80

Г

Гиперссылка, 124

Гипертекст, 319, 320

гипертекстовая разметка, 98

Д

Даглас Энгельбарт, 84

дизайн, 3, 29, 31, 32, 34, 35, 37, 38, 41, 42, 43, 44, 56, 63, 72, 76, 85, 129, 138, 189, 390, 391, 392, 415, 553

Дизайн сайта, 86

докер Bookmark Manager, 283

докер Object Properties, 281, 283

З

Заголовки, 99

Закрывающий тег, 122, 123, 137

И

Изменение рисунка, 7, 304, 305

Интернет, 4, 7, 42, 53, 78, 79, 80, 81, 82, 86, 87, 88, 99, 117, 234, 279, 281, 313, 314, 315, 317, 318, 321, 322, 323, 329, 334, 335, 336, 337, 345, 348, 349, 350, 355, 356, 357, 368, 370, 371, 377, 383, 386, 389, 391, 392, 400, 401, 406, 411, 412, 417, 424, 425, 426, 429, 433, 552, 555, 556

К

кадрирование, 4, 74, 75, 76

Клиент, 138

код, 14, 85, 94, 95, 97, 98, 104, 108, 109, 110, 112, 113, 137, 202, 237, 242, 244, 245, 315, 365, 397, 430, 431, 432, 433, 435, 437, 447, 448, 459, 476, 483, 514, 544

комплектующие сайта, 86

компьютерная графика, 51, 60, 62, 63, 64, 67

контур, 55, 60, 63, 167, 168, 169, 170, 175, 176, 178, 184, 191, 192, 198, 214, 220, 415, 536, 538

Л

Литература в WEB, 8, 383

М

Мастер автономного просмотра избранного, 347, 366

Мастер расписания синхронизации, 371

Меняющиеся кнопки, 308, 311

модели CMYK, 69, 242, 243, 245, 259

модель RGB, 68, 69, 243

Н

неразрывный пробел, 120, 121

О

Открывающий тег, 118

Отправление открытки, 8, 389

Отчет о конфиденциальности, 365

П

Параметры синхронизации, 370
Параметры страницы, 296, 297, 357
переносы, 121
персональный сайт, 552
пиксель, 54, 57, 135, 237, 259
подменю Insert Internet Object, 283
Предварительный просмотр, 184, 358, 390
предпечатная подготовка, 76

Р

развлекательные сайты, 313
разметка, 124
разрешение, 57, 58, 157, 303, 398
растровая графика, 54, 58, 59, 61
растровый формат, 91

С

Сайт, 83, 126, 313, 380
Свойства коллекции фотографий, 299, 303, 304, 305, 306, 307
Свойства страницы, 312
Свойства фильма в формате Flash, 292, 293
Сервер, 138
Системы цветов HSB, 70
содержание сайта, 82
Списки, 5, 122
Список, 149, 151, 174, 187, 188, 283, 284, 288, 291, 300, 301, 307
Ссылка, 125, 126, 127, 128, 155, 309, 552, 553
Страница, 83, 84, 291, 292, 298, 302, 305, 306, 337

Т

Таблицы, 5, 115, 128, 133, 508
Таблицы стилей, 115, 133
Тело документа, 4, 113
Теодор Нельсон, 84
Тим Бернерс-Ли, 84, 106
трехмерная графика, 56, 65

Ф

ФАЙЛ, 244, 246
Формы обратной связи, 138
фрактал, 66
фрактальная графика, 66

Ц

цветоделение, 70, 72, 76
цветокоррекция, 74, 76

Ч

Чарльз Голдфарб, 105

Э

эффект вспышки, 231

е-Издательство «Мультимедиа Технологии и Дистанционное Обучение»

Основные направления деятельности е-Издательства ММТиДО — разработка и создание мультимедийных обучающих программ (учебников и пособий). За истекшие годы создано более 160 наименований ***CD-ROM*** для России, Западной Европы и США. По иностранным языкам (серии ***Gold*** и ***Platinum***), по компьютерной грамотности, по школьным и некоторым специальным дисциплинам (серия ***TeachPro***).

ММТиДО является одним из лидеров в РФ в области разработки мультимедийных обучающих программ, предназначенных для индивидуального и группового обучения. Курс французского языка — «***Français d`Or 2000***» — выбран Министерством образования РФ для поставки в 40 тысяч городских и сельских школ России по Президентской программе. ММТиДО по заданию Минобразования разработало курс "Компьютерная грамотность и дизайн", "Немецкий язык, 5-9 классы" и "Организация и технология механизированных работ в животноводстве".

Совместно с МЧС РФ создана обучающая программа «***TeachPro*** ОБЖ, 10 класс», которая удостоена золотой медали на международной выставке в Сеуле — «***SIIF 2002***». Программа имеет Гриф Министерства Образования РФ.

Большинство ведущих вузов России (МЭСИ, РАГС, СГА, МГУ, МПГУ, РУДН, ВЗФИ, РТА, МГОУ, МПСИ, МГСУ, ЛГУ и др.), а также сотни школ и лицеев г. Москвы используют обучающие программы ММТиДО в образовательном процессе. Курсы серии ***TeachPro*** по Математике, Физике и Химии, разработанные совместно с СГА, получили золотую медаль и Диплом ВВЦ (№82 от 29.10.02) на выставке "Современная образовательная среда 2002 г. ".

ММТиДО успешно выполнил государственные заказы Госкомстата РФ по разработке и созданию мультимедийных обучающих программ для временных переписных работников – «***TeachPro*** Перепись 2002», за что несколько сотрудников ММТиДО Указом Президента РФ от 14.10.02 награждены медалью "За заслуги в проведении всероссийской переписи населения", а также «***TeachPro*** Сельскохозяйственная перепись 2006».

Курс "***TeachPro Internet Trading***" по мнению самого авторитетного американского профессионального издания в области трейдинга — "***Technical Analysis Of Stocks And Commodities***" "не имеет равных на всем американском рынке". (См. ***www.terratrade.ru/vcollege.html***).

В разное время наши заявки на конкурсы, проводимые Минобрнауки, поддерживали: Фельдштейн Д.И. — вице-президент РАО, Бондырева С.К. — академик РАО, Гулидов А.Д. — заместитель гендиректора Росстата, Песоцкий Ю.С. — гендиректор «Росучприбор", Солдаткин В.И. — гендиректор "Российского государственного. института открытого образования» и др. ведущие специалисты.

Наши партнеры — ЦБ РФ, Госдума РФ, Мэрия Москвы, МЧС, Госкомстат, Русский Алюминий, Тюменская Нефтяная Компания, 1С, Лукойл, РБК и др. известные организации.

Мультимедийные самоучители на CD-ROM Серия «TeachPro™»

«***TeachPro***™» — это серия интерактивных компьютерных учебников. При использовании этих программ обучаемому не требуется иметь никакой справочной или учебной литературы, не понадобится обращаться к преподавателю или более опытному пользователю, заниматься на дорогостоящих курсах, — полный цикл лекций и практические занятия позволят самостоятельно освоить данный предмет.

Обучение в системах ***TeachPro***™ построено следующим образом: компьютер, выступая в роли опытного педагога, использует приемы реальной работы с учащимися. Каждый цикл лекций предоставляет исчерпывающий набор знаний, умений и навыков, достаточных для работы с изучаемым предметом. Занятия осуществляются в форме диалога, таким образом проверяется объем изученного материала. Тесты в конце каждого урока позволят оценить и закрепить знания учащегося. Для группового обучения в локальных сетях «ММТ и ДО» предлагает сетевые версии обучающих программ ***TeachPro***™.

Мультимедийные самоучители серии «TeachPro™» по программным продуктам Microsoft.

Серия представляет собой набор интерактивных курсов, предназначенных для обучения работе с программами пакета ***Microsoft***. Учебники будут в равной степени полезны как новичкам, так и опытным пользователям. Люди, не имеющие навыков работы с компьютером, смогут быстро освоить выбранный продукт; опытные же пользователи наверняка узнают множество приемов по работе с системой.

Занятия организованы таким образом, что, прослушав только первые лекции, вы уже сможете начать реальную работу с изучаемой программой. Продолжая занятия, вы, максимально автоматизируя свою работу, будете постепенно расширять круг решаемых задач, использовать новые инструменты, создавать все более серьезные и объемные проекты.

На практике программные продукты ***Microsoft*** применяются почти повсеместно. Но, как уже говорилось, используются далеко не так широко, как могли бы. И дело здесь не только в недостатке знаний у пользователей. В этом повинны, в частности, многочисленные справочники и руководства (всех видов, мастей и названий), которыми усыпаны книжные прилавки. Авторы данных курсов уполномочены заявить со всей ответственностью — нам не встретился ни один учебник, который позволил бы качественно освоить любой из продуктов ***Microsoft***, не имея начальной подготовки. Хорошие комплексные знания предмета можно получить, занимаясь с группой на курсах, но, во-первых, далеко не каждые курсы дают действительно полные и хорошие знания, а, во-вторых, это весьма дорогое удовольствие, и отнимает оно много времени (одна дорога туда и обратно чего стоит).

Курсы серии «***TeachPro***™» совмещают в себе качество знаний, получаемых при занятиях в хорошей группе с опытным преподавателем, с индивидуальной работой с учебником и справочным руководством. Вы будете поражены, убедившись, как легко и быстро усваиваются новые знания, методы и приемы работы.

Мультимедийные самоучители по программным продуктам Microsoft на CD-ROM

TeachPro Word 2003	*Самоучитель по Microsoft Word 2003.*
TeachPro Excel 2003	*Самоучитель по Microsoft Excel 2003.*
TeachPro Access 2003	*Самоучитель по Microsoft Access 2003.*
TeachPro PowerPoint 2003	*Самоучитель по Microsoft PowerPoint 2003.*
TeachPro Outlook 2003	*Самоучитель по Microsoft Outlook 2003.*
TeachPro FrontPage 2003	*Самоучитель по Microsoft FrontPage 2003.*
TeachPro Visio 2003	*Самоучитель по Microsoft Visio 2003.*
TeachPro Publisher 2003	*Самоучитель по Microsoft Publisher 2003.*
TeachPro OneNote 2003	*Самоучитель по Microsoft OneNote 2003.*
TeachPro Internet 2003	*Самоучитель по Microsoft Internet 2003.*
TeachPro Office 2003	*Сборник самоучителей по Word 2003, Excel 2003, Access 2003, Outlook 2003, PowerPoint 2003, FrontPage 2003, Visio 2003, Publisher 2003 и OneNote 2003.*
TeachPro Windows XP	*Самоучитель по Microsoft Windows XP.*
TeachPro Word 2002	*Самоучитель по Microsoft Word 2002.*
TeachPro Excel 2002	*Самоучитель по Microsoft Excel 2002.*
TeachPro Access 2002	*Самоучитель по Microsoft Access 2002.*
TeachPro PowerPoint 2002	*Самоучитель по Microsoft PowerPoint 2002.*
TeachPro Outlook 2002	*Самоучитель по Microsoft Outlook 2002.*
TeachPro FrontPage 2002	*Самоучитель по Microsoft FrontPage 2002.*
TeachPro Win&Office XP	*Сборник самоучителей по Windows XP, Word 2002, Excel 2002, Access 2002, Outlook 2002, PowerPoint 2002, FrontPage 2002 и Internet 2002.*

Самоучители для школьников и абитуриентов

TeachPro ОБЖ	*Основы безопасности жизнедеятельности для 10 класса.*
TeachPro Физика 11 класс	*Физика по программе для 11 класса.*
TeachPro Физика	*Физика для абитуриентов для 7-11 классов.*
TeachPro Математика	*Математика для абитуриентов для 7-11 классов.*
TeachPro Химия	*Химия для абитуриентов для 7-11 классов*
TeachPro Решебник по Физике	*Решебник по физике для 7-11 классов.*
TeachPro Решебник по Математике	*Решебник по математике для 7-11 классов.*
TeachPro Решебник по Химии	*Решебник по химии по 7-11 классов.*
TeachPro Информатика	*Информатика для абитуриентов.*

Самоучители по языкам программирования

TeachPro C++	*Самоучитель C++.*
TeachPro JAVA	*Самоучитель JAVA.*
TeachPro Visual Basic .Net	*Самоучитель Visual Bacic. Net.*
TeachPro Visual Basic for Applications	*Самоучитель Visual Basic for Applications*

Самоучители по графическим программам

TeachPro Flash MX	*Самоучитель по Flash MX.*
TeachPro AutoCad 2005	*Самоучитель по AutoCad 2005.*
TeachPro Adobe Premiere Pro 1.5	*Самоучитель по Adobe Premiere Pro 1.5.*
TeachPro Adobe Photoshop CS	*Самоучитель по Adobe Photoshop CS.*
TeachPro Adobe Photoshop 6.0	*Самоучитель по Adobe Photoshop 6.0*
TeachPro Adobe Illustrator CS	*Самоучитель по Adobe Illustranor CS.*

TeachPro Adobe Illustrator CS2	*Самоучитель по Adobe Illustranor CS2*
TeachPro 3D Studio Max6	*Самоучитель по Adobe по 3D Studio Max6.*
TeachPro CorelDraw 12	*Самоучитель по Adobe по CorelDraw 12.*
TeachPro Web-дизайн	*Самоучитель по Web-дизайну.*
TeachPro Quark XPress 6.0	*Самоучитель по Quark XPress 6.0.*
TeachPro Adobe Photoshop	*Самоучитель по Adobe Photoshop 6.0.*
TeachPro Macromedia Studio MX 2004	*Самоучитель по Macromedia Studio MX*
TeachPro Adobe PageMaker 7.0	*Самоучитель по Adobe PageMaker 7.0*
TeachPro Macromedia Flash MX 2004	*Самоучитель по Macromedia Flash MX*
TeachPro Adobe InDesign CS2	*Самоучитель по Adobe InDesign CS2*
TeachPro Компьютерная Графика 2006	*Самоучитель по Компьютерной Графике*
TeachPro MathCAD 13	*Самоучитель по MathCAD 13*
TeachPro ArchiCAD 9	*Самоучитель по ArchiCAD 9*
TeachPro Alias Wavefront Maya 6	*Самоучитель по Alias WaveFront Maya 6*
TeachPro Macromedia Dreamwaver MX 2004	*Самоучитель по Macromedia Dreamwaver MX 2004*
TeachPro Macromedia Director MX 2004	*Самоучитель по Macromedia Director MX*

Самоучители по другим приложениям

TeachPro Утилиты	*Самоучитель по утилитам.*
TeachPro Утилиты 2006	*Самоучитель по утилитам 2006.*
TeachPro Excel	*Учебник. Проектирование производственного на производстве программного обеспечения в среде Microsoft Excel.*

TeachPro Лекции по	*Самоучитель математики для юристов. математике для юристов*
TeachPro Internet Trading	*Самоучитель по операциям с ценными бумагами в Internet (на английском языке).*
Евангелия	*Мультимедийное издание.*
TeachPro Программы	*Самоучитель компьютерных программ для профессионального перевода профессиональных переводчиков.*
TeachPro Основы бизнеса	*Самоучитель по Основам бизнеса*
TeachPro Маркетинг	*Самоучитель по маркетингу товаров и услуг*
товаров и услуг	
TeachPro Концепция	*Самоучитель по концепции совр. естествознания*
совр. естествознания	
TeachPro Словарь 7 языков	*Мультимедийное издание.*
TeachPro Механизация Работ в животноводстве	*Самоучитель по механизации работ животно-водстве*
TeachPro БНП по немецкому языку	*Мультимедийное издание.*
TeachPro Препринт	*Мультимедийное издание.*
TeachPro Компьютер и Периферия	*Мультимедийное издание.*
TeachPro e-коммерция и	*Самоучитель по e-коммерции и e-подписи*
e-подпись	

TeachPro™

WEB-дизайн

Под редакцией:

Юлии Федоровны Катхановой
Галины Семеновны Гринберг
Григория Антонова
Ивана Соловьева
Романа Суховерхова

Ответственный редактор В.Ф. Гончаров
Художественный редактор Ю.М. Славнова
Компьютерная верстка К.А. Савичев

Издательская лицензия ИД № 06489

Гигиенический сертификат
№77.99.02.953.П.000415.02.02 от 28.02.2002

Наши координаты:

111024, Москва, Шоссе Энтузиастов, д. 20А
тел. (495) 673-7692, (495) 673-7693
факс (495) 673-5975
http://www.mmt-dl.ru
sales@mmt-dl.ru — отдел продаж
support@mmt-dl.ru — служба технической поддержки
webmaster@mmt-dl.ru — замечания и предложения по сайту
info@mmt-dl.ru — общая информация о наших продуктах
mmt@mmt-dl.ru — другие вопросы