

**Министерство образования и науки РФ**

**Томский государственный университет систем  
управления и радиоэлектроники**

**С.Л. Миньков**

**МИРОВЫЕ ИНФОРМАЦИОННЫЕ  
РЕСУРСЫ  
Лабораторный практикум**



**Томск 2012**

**Миньков С.Л.**

Мировые информационные ресурсы. Лабораторный практикум: учебное пособие. Изд. 2-е, испр.— Томск: ТУСУР, 2012. – 114 с.

Пособие предназначено для освоения приемов поиска информации в глобальной информационно-коммуникационной среде Интернет и разработки html-страниц статического web-сайта с использованием текстового редактора.

Рассчитано на студентов специальности 080801 «Прикладная информатика в экономике», изучающих дисциплину «Мировые информационные ресурсы», в том числе с применением дистанционных технологий.

© Миньков С.Л., 2012

## СОДЕРЖАНИЕ

Лабораторная работа 1 ПОИСК И ПОЛУЧЕНИЕ ИНФОРМАЦИИ В ИНТЕРНЕТ.....	4
ВВЕДЕНИЕ.....	4
1. ОБОЗРЕВАТЕЛЬ INTERNET EXPLORER .....	6
2. ПОИСК ИНФОРМАЦИИ В ИНТЕРНЕТ .....	14
3. ЗАГРУЗКА ИНТЕРНЕТ-ИНФОРМАЦИИ.....	31
Лабораторная работа 2 СОЗДАНИЕ WEB-САЙТА.....	44
ВВЕДЕНИЕ.....	44
1. СТРУКТУРА HTML-ДОКУМЕНТА.....	47
2. УПРАВЛЕНИЕ ТЕКСТОМ .....	53
3. ВСТАВКА РИСУНКОВ .....	62
4. ГИПЕРССЫЛКИ .....	64
5. ТАБЛИЦЫ .....	73
6. ФРЕЙМЫ.....	79
7. ТАБЛИЦЫ СТИЛЕЙ.....	87
8. ЗАДАНИЕ .....	100
Приложение 1. Таблица основных тегов языка HTML ..	103
Приложение 2. Цветовая палитра .....	109
Приложение 3. Темы для Web-сайтов.....	112
Приложение 4. Программное обеспечение, необходимое для выполнения лабораторных работ по дисциплине «Мировые информационные ресурсы» .....	114

# Лабораторная работа 1

## ПОИСК И ПОЛУЧЕНИЕ ИНФОРМАЦИИ В ИНТЕРНЕТ

***Цель работы:** освоить основные навыки работы со средством просмотра Web-страниц MS Internet Explorer, программами и сервисами Интернет, позволяющими осуществлять поиск и получение информации из Сети.*

### **ВВЕДЕНИЕ**

В конце XX века появилась и быстрыми темпами стала развиваться глобальная коммуникационная среда – Интернет. Это не только распределенная сеть, соединяющая сотни миллионов компьютеров по всему миру, но и новая форма общедоступного архива, основанная на гипертекстовом представлении мультимедийной информации.

Глобальные информационные сети становятся основой современной жизни во всех ее проявлениях: экономической, социальной, политической, культурной. Своей открытостью, доступностью, устойчивостью к сбоям Интернет способствует большей открытости, оперативной информированности, широкой гласности в обществе и бизнесе.

Чтобы умело пользоваться всеми возможностями Сети, необходимо освоить приемы навигации и поиска информации в Интернет.

Одним из самых популярных сервисов Интернета наряду с элек-

тронной почтой является World Wide Web (WWW) – «всемирная паутина». Это название достаточно точно определяет суть сервиса. WWW – гигантская сеть страниц (документов), связанных между собой гипертекстовыми ссылками, что действительно очень напоминает своими колоссальными размерами и структурой паутину (web), протянувшуюся по всему миру.

**Гипертекст** – это легкая в использовании и чрезвычайно мощная система связанных слов и фраз, позволяющая легко перемещаться по особым образом организованным страницам. Она связывает фразу или слово одной страницы с любой другой страницей, абзацем, фразой или словом (рис. 1.1).

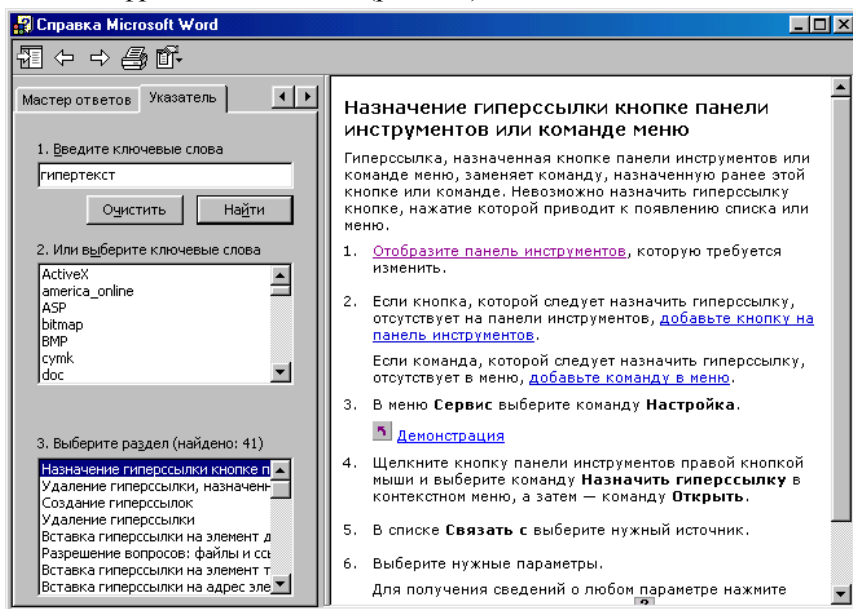


Рис. 1.1 – Справка MS Word, выполненная с использованием гипертекста (правое окно). Щелчок мышью по подчеркнутым объектам вызывает переход в другое окно Справки, или вызов справки-определения, или демонстрацию каких-либо действий.

Если развить идею гипертекста и включить в него графику, видео и звук, мы получим гипермедиа. **Гипермедиа** – среда, основанная, как и гипертекст, на взаимосвязях, в которой в качестве гиперссылок могут также выступать визуальные и аудиокомпоненты. Гипертекст и гипермедиа являются фундаментальными для WWW технологиями, а HTML – средство для работы с этими технологиями.

World Wide Web можно представить и как большую библиотеку Internet. Каждый узел «паутины» – Web-сайт подобен книге из этой библиотеки, а Web-страницы подобны страницам этих книг. Обычно путешествие по WWW начинается с определенного узла.

Первая страница представляет собой исходную точку для узла – нечто подобное обложке или содержанию книги.

Каждая страница, включая начальную страницу узла, имеет уникальный адрес в формате URL (Uniform Resource Locator).

Страницы Web являются гипертекстовыми документами, написанными в формате HTML (HyperText Markup Language), и имеют расширение .html или .htm.

Для передачи гипертекстовых документов используется сетевой протокол HTTP (HyperText Transfer Protocol).

URL для HTTP выглядит так:

`http://<host>`

где <host> – доменное имя или IP-адрес сервера.

Например, <http://192.252.19.1>, <http://www.asu.tusur.ru>.

## **1. ОБОЗРЕВАТЕЛЬ INTERNET EXPLORER**

### **1.1. Интерфейс Internet Explorer 6.0**

Программное средство, используемое для просмотра страниц Web, получило название браузер (browser).

Microsoft Internet Explorer представляет собой одно из наиболее

известных средств просмотра.

Подобно тому как Microsoft Word является программным средством для работы с текстом или Microsoft Excel – средством для работы с электронными таблицами, Internet Explorer является средством, предназначенным для выполнения переходов с одного информационного ресурса в Сети к другому и получения доступа к сведениям Web.

Внешний вид основного окна Internet Explorer приведен на рис.1.2.

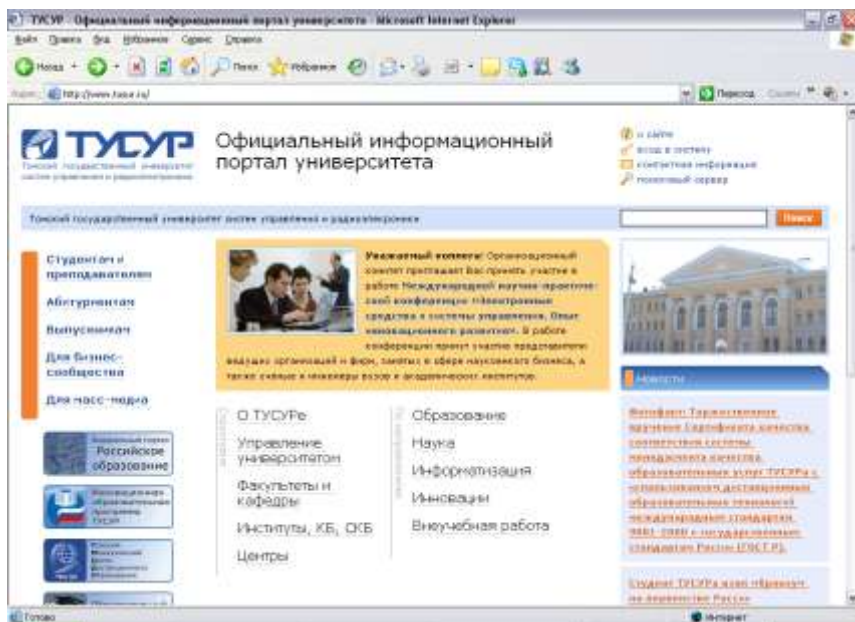


Рис. 1.2 – Основное окно Internet Explorer 6.0 с открытой главной страницей сайта [www.tsur.ru](http://www.tsur.ru)

Панель инструментов Internet Explorer (рис. 1.3) содержит ряд функций и команд для управления браузером, выполнения Web-переходов и обработки найденной информации. Строка адреса под

панелью инструментов отображает адрес текущего узла, к которому осуществляется доступ.

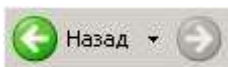
Для перехода на новый узел необходимо ввести адрес узла (в формате URL) непосредственно в поле адресной строки. После завершения ввода нажать на кнопку меню **Переход** или клавишу **Enter**.



Рис. 1.3 – Панель инструментов Internet Explorer 6.0.



При выполнении браузером обращений или при загрузке данных в верхнем правом углу экрана переливается эмблема Internet Explorer. Если этот процесс длится слишком долго, воспользуйтесь кнопкой «Остановка», чтобы прервать выполняемый запрос.



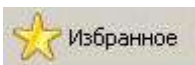
Кнопка для перехода назад позволяет вернуться к предыдущей странице, которая перед этим была отображена на экране. Кнопка для перехода вперед позволяет перейти к следующей, отображаемой ранее странице. Если кнопка для перехода назад еще не использовалась в текущем сеансе, кнопка для перехода вперед будет отключена.



Кнопка открытия начальной страницы позволяет вернуться к стандартной начальной странице, которая открывается автоматически при запуске браузера.



Кнопка поиска открывает раздел, содержащий встроенные в браузер средства выполнения поиска в Интернет.



Кнопка «Избранное» открывает раздел, позволяющий запомнить часто посещаемые Web-сайты, поместив ссылки на них в папку «Избранное». Всегда имеется возможность вернуться к любой ссылке из этой папки. Выбе-



рите опцию «Добавить избранное», и ссылка на текущую страницу сохраняется для последующего применения.

Главное меню браузера содержит пункты **Файл, Правка, Вид, Избранное, Сервис, Справка**, в основном имеющие тот же смысл, что и соответствующие пункты в офисных пакетах Microsoft.

Чтобы посмотреть и настроить разнообразные характеристики браузера, необходимо вызвать меню **Сервис |Свойства обозревателя**. Появившееся диалоговое окно (рис. 1.4) содержит семь закладок с различными характеристиками MS Internet Explorer 6.0.

Например, закладки **Безопасность** и **Конфиденциальность** позволяют настраивать параметры информационной безопасности при работе с Интернет, закладка **Дополнительно** позволяет настраивать параметры страницы, вид и стиль отображаемой информации (убрав галочку «Отображать рисунки», можно уменьшить время загрузки страницы, т.к. будет отображаться только текстовая информация).

В настоящее время распространяется седьмая версия MS Internet Explorer и появилась бета-версия Internet Explorer 8.

К другим популярным браузерам относятся Mozilla Firefox, Opera (Opera Software), Safari (Apple).

## 1.2. FTP-соединение

Интернет-сервисом является и FTP. Это сетевой протокол передачи файлов (File Transfer Protocol), пользуясь которым можно получить доступ к файлам, размещенным на удаленном FTP-сервере.

Для доступа к FTP-ресурсам существует большое количество специальных программ: FTP-клиентов, но такой доступ может быть осуществлен и посредством Internet Explorer.

URL для FTP выглядит так:

ftp://<user>:<password>@<host>:<port>/<url-path>,

где user – имя зарегистрированного пользователя,

<password> – его пароль,  
<host> – доменное имя или IP-адрес сервера,  
<port> – порт подключения;  
<url-path> – путь к файлу.

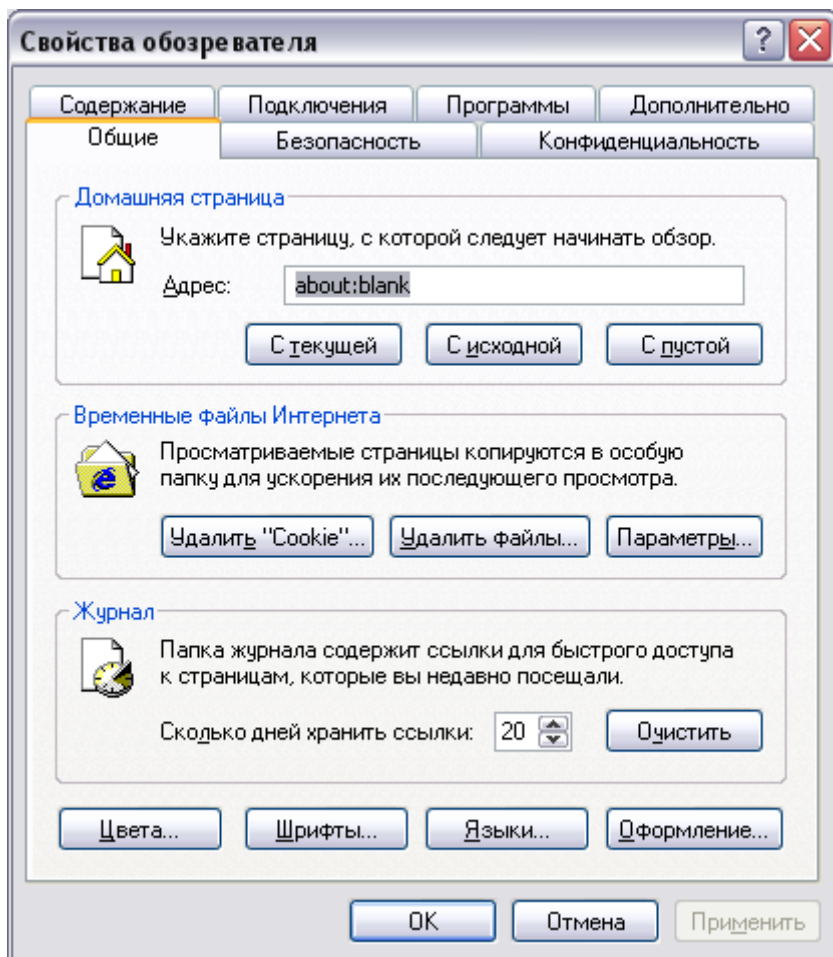


Рис. 1.4. Диалоговое окно **Свойства обозревателя**.

Зарегистрированный пользователь получает возможность считывать и записывать информацию в соответствующие папки на ftp-сервере. Незарегистрированного пользователя встречает предупреждающая информация (рис. 1.5).

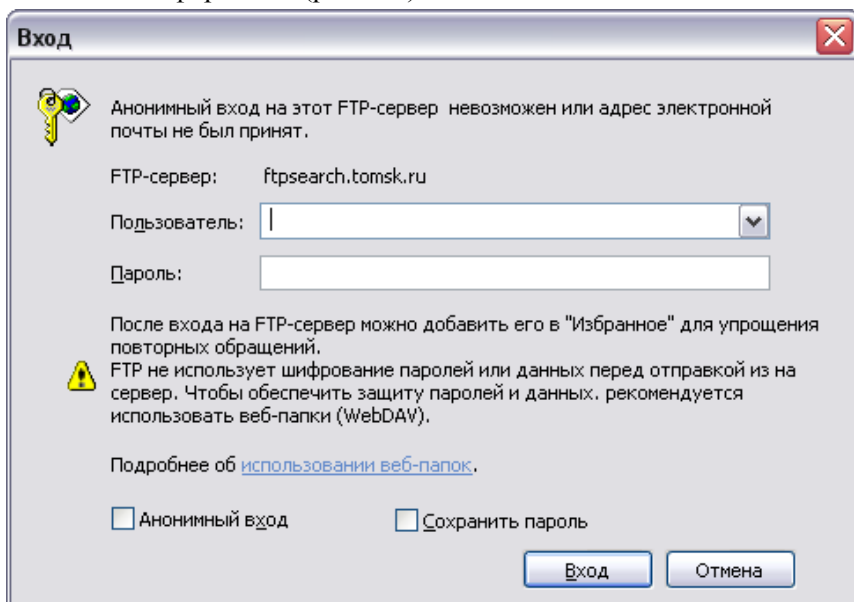


Рис. 1.5. Диалоговое окно авторизации на ftp-сервере.

Часто бывает возможным так называемый анонимный вход: в качестве имени пользователя достаточно указать anonymous, а в качестве пароля – свой почтовый адрес. При этом ваши права на удаленной машине будут урезаны до минимума – только скачивание информации.

На многие FTP-серверы возможен вход без предварительной регистрации; тогда синтаксис адреса упрощенный:

`ftp://<host>/<url-path>`

В качестве <host> можно указывать и IP-адрес.

Примеры адресов FTP:

<ftp://mail.dgs.ru>

<ftp://84.237.2.10/ustanovka.pdf> .

Итак, для того чтобы использовать Internet Explorer как FTP-клиент, необходимо в поле адресной строки указать URL для FTP. Правила формирования адреса были описаны выше, все остальное предельно просто. Если в качестве пути указан только путь к некоторому каталогу, а не сам файл (первый пример), то браузер покажет вам список файлов в этом каталоге. Если же путь указан вплоть до имени файла, то файл откроется на вашем компьютере прикладной программой в соответствии с его расширением.

### **1.3. Прoxy-сервер**

Прoxy-сервер — это сервер, который управляет доступом клиентских компьютеров в Сеть. Это связующее звено между компьютером локальной сети и глобальными сетевыми ресурсами.

Одна из важных функций проxy-сервера – кэширование скачиваемых файлов, т.е. когда вы посылаете ему свой запрос (например, набрав в Internet Explorer адрес интересующей вас WWW-страницы), проxy-сервер сначала ищет нужный вам файл у себя, и только если не находит его, то перенаправляет запрос по действительному адресу. Таким образом, если кто-либо (в том числе вы сами) один раз обратились к какому-то файлу в Internet, то при последующих обращениях этот файл не будет каждый раз запрашиваться с удаленного сервера, а будет выдаваться проxy-сервером. Это значительно снижает загрузку сети и ускоряет работу.

Помимо кэширующей функции, проxy-сервер играет также важную роль в обеспечении безопасности ресурсов локальной сети. При передаче всего трафика через проxy-сервер, внутренние IP-адреса остаются скрытыми от внешнего мира. Такая организация работы позволяет избавиться от необходимости иметь отдельное интернет-соединение для каждого компьютера. Это эквивалентно тому, что все компьютеры локальной сети при осуществлении свя-

зи с внешним миром работают, как один пользователь.

Кроме того, проху-сервер позволяет контролировать деятельность внутренних пользователей, например, предотвратить их доступ к нежелательным web-сайтам.

Структурно проху-сервер состоит из множества специфических посредников для конкретных приложений: посредника для web-страниц, для ftp-сервиса, для электронной почты и т.п.

### **Задание 1**

1. Запустить Internet Explorer и настроить проху-сервер, если ваша локальная сеть подключена к глобальной сети через сервер. Настройки (IP-адрес и порт) указываются в меню **Сервис | Свойства обозревателя | Подключения | Настройка LAN | Прокси-сервер**.

2. Установить собственное значение адреса стартовой страницы **Сервис | Свойства обозревателя | Общие | Домашняя страница**.

3. Создать в «Избранном» папку «Томск» и поместить в нее адрес сайта г. Томска: <http://www.tomsk.ru> и томских погодных страниц: <http://www.pogodavtomske.ru> и <http://www.gismeteo.ru/towns/29430.htm>.

4. Посетить главную страницу сайта ТУСУР <http://www.tusur.ru> и найти информацию об электронных ресурсах ТМЦДО. В библиотеке ТМЦДО найти литературу по информатике и Интернет-технологиям.

5. Перейти к электронному каталогу книг библиотеки ТУСУРа <http://www.lib.tusur.ru>. Найти в каталоге литературу, изданную преподавателями кафедры АСУ.

6. Посетить сайт библиотеки Максима Мошкова ([www.lib.ru](http://www.lib.ru)) или его томское зеркало (<http://moshkow.tomsk.ru>) и найти литературу по экономике.

7. Посетить любой томский FTP-сервер (список можно найти на сайте <http://sea.tomsk.ru>), например <ftp://tigra.wetnet.ru> , <ftp://unix2.asu.tusur.ru> , <ftp://188.204.10.5>, <ftp://ftp.ssti.ru> анонимным

пользователем. Описать структуру ftp-ресурса. Найти и сохранить любой текстовый файл, находящийся в корневом или ином каталоге.

8. Оформить отчет о выполненной работе, проиллюстрировав его скриншотами запросов и web-страниц, где вы побывали. Не забудьте с помощью какого-либо графического редактора (например, Microsoft Office Picture Manager, входящего в состав MS Office 2003 и выше) предварительно уменьшить их размер до разумного (~ 30-50 Кб).

## **2. ПОИСК ИНФОРМАЦИИ В ИНТЕРНЕТ**

### **2.1. Тематические каталоги**

Интернет – это огромное информационное хранилище (хотя некоторые называют Сеть огромной информационной свалкой). Но, в любом случае, поиск в пространстве Интернет необходимой информации – не тривиальная задача.

Как сориентироваться в столь масштабном информационном пространстве? На помощь приходят тематические каталоги, поисковые системы, системы метапоиска.

**Тематический каталог** представляет собой огромную базу данных URL-адресов сайтов самой различной тематики. Такая база данных при поиске информации позволяет пользоваться иерархическим деревом. Вы выбираете сначала общую тематику, удовлетворяющую вашему запросу информации, и далее конкретизируете, следуя подсказкам каталога. В конечном результате вы получаете список сайтов, содержащих информацию, соответствующую вашему запросу. При пользовании тематическим каталогом вы также можете пойти и по другому пути. Проанализируйте предполагаемое содержание запрашиваемой информации и выберите ключевые слова, которые обязательно встретятся в интересующих

вас материалах или их заголовках. Набираете эти слова через пробел в строке ввода на главной странице каталога и нажимаете Enter. Система попытается подобрать интересующую вас информацию и сообщит вам соответствующие ссылки.

Приведем примеры некоторых тематических каталогов.

### Томские ресурсы

**Каталог Томского интернета** – <http://catalog.tomsk.ru> (рис. 1.6)

**Каталог сайтов Томска ЦИС** – <http://catalog.tom.ru/>

**Каталог Tambo** – <http://catalog.tambo.ru>

**Томский городской портал** – <http://www.tomsk.ru>

**Томский каталог** – <http://metka.ru/>

**Справочник ru09.ru** – <http://www.tomsk.ru09.ru/catalog>

**Каталог сайтов TomskLink** – <http://link.tom.ru/>

**70.RU** – <http://70.ru>

**Каталог Интернет-ресурсов Томска и области** – <http://www.region70.ru/>

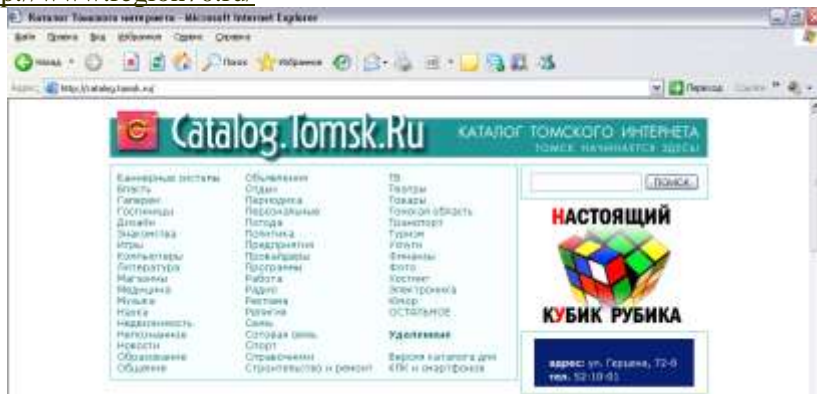


Рис. 1.6. Окно каталога Томского интернета

### Российские ресурсы

**Каталог@MAIL.RU** – <http://list.mail.ru> (рис. 1.7)

**АПОРТ-Каталог** – <http://catalog.aport.ru/rus/thememap.aspx>

**Russia on the Net** – <http://www.ru/rus>

**Интернет-каталог сайтов РУНЕТа** – <http://webcat.info>

**Яндекс.Каталог** – <http://yaca.yandex.ru>

**Каталог Рунета** <http://all-ru.net>

**Российская Информационная Сеть** – <http://www.rin.ru>

**Каталог сайтов** – <http://www.refer.ru>



Рис. 1.7. Окно интернет-каталога Каталог@MAIL.RU

### Всемирные ресурсы

**Yahoo!** – <http://www.yahoo.com> (рис. 1.8).

**My Excite** – <http://www.excite.com> Каталог поисковой системы Excite.

**Lycos** – <http://www.lycos.com/> Каталог Интернет-ресурсов.

**NKU Directory** – <http://www.nkud.com> Каталог Интернет-ресурсов.





Рис. 1.8 – Окно тематического каталога Yahoo!

## 2.2. Поисковые серверы

Поисковый сервер устроен иначе, чем тематический каталог. По сути, это сервер с огромной базой данных URL-адресов, который автоматически обращается к страницам WWW по всем этим адресам, изучает содержимое этих страниц, формирует и прописывает ключевые слова со страниц в свою базу данных (индексирует страницы). Более того, этот сервер обращается по всем встречаемым на страницах ссылкам и, переходя к новым страницам, проделывает с ними то же самое. Так как почти любая страница WWW имеет множество ссылок на другие страницы, то при подобной работе поисковая машина в конечном результате теоретически может обойти все сайты в Интернет.

Фактически, каждая поисковая система состоит из двух частей – поискового HTTP-робота и модуля обработки запроса пользователя. HTTP-робот, или поисковый движок, – это программа, которая

собирает доступную в виде текста информацию. Вся собранная информация записывается в index-файл. Поиск по запросу осуществляется другой программой, которая извлекает запрашиваемую пользователем информацию из index-файла.

Качество работы отдельно взятой поисковой системы сильно зависит от качества работы обоих программных модулей системы.

Поисковый запрос пользователя может состоять из одного или нескольких слов, в нем могут присутствовать знаки препинания. Составлять простые запросы можно и не вдаваясь в тонкости языка запросов. Так, если ввести в поисковую строку несколько слов без знаков препинания и логических операторов, будут найдены документы, содержащие все эти слова (причем на ограниченном расстоянии друг от друга). Однако знание и правильное применение языка запросов поисковой машины поможет сделать поиск быстрым и эффективным.

Все поисковые машины используют сходные принципы языка запросов. Ссылку на полное описание языка запросов для каждой поисковой машины можно найти на ее главной странице.

Ниже приведены основные операторы языка запросов на примере поисковой машины **Yandex** (<http://www.yandex.ru>).

Синтаксис запроса Yandex	
пробел или &	Логическое И (краткое - в пределах одного абзаца). <i>Пример:</i> легкая промышленность. <i>Результат:</i> все документы, где в пределах одного абзаца встречаются слова «легкая» и «промышленность»
&&	Логическое И (в пределах документа). <i>Пример:</i> легкая&&промышленность. <i>Результат:</i> все документы, где встречаются слова «легкая» и «промышленность»
, или	Логическое ИЛИ <i>Пример:</i> легкая,промышленность.

	<b>Результат:</b> все документы, где встречается либо слово «легкая», либо слово «промышленность»
~	Бинарный оператор И НЕ (в пределах одного абзаца) <b>Пример:</b> производство~газета. <b>Результат:</b> все документы, где в пределах одного абзаца встречается слово «производство», но не в словосочетании «газета производство»
~~	Бинарный оператор И НЕ (в пределах документа). <b>Пример:</b> производство~газета. <b>Результат:</b> все документы, где встречается слово «производство», но не в словосочетании «газета производство»
( )	Группирование слов. <b>Пример:</b> (швейное&&производство)~газета. <b>Результат:</b> все документы, где встречаются слова «швейное» и «производство», но не в словосочетании со словом «газета»
/ (число)	Расстояние в словах, где «число» – число слов между словами в запросе плюс единица. <b>Пример:</b> швейное/1 производство. <b>Результат:</b> все документы, где встречается словосочетание «швейное производство»

Поисковые машины и тематические каталоги сегодня имеют много общего. У каталогов присутствует возможность поиска информации по строке запроса с использованием логических операторов, а поисковые машины содержат свои собственные тематические каталоги. И тем не менее лучше всего эти поисковые сервисы проявляют себя в своей первоначальной категории.

Ниже приведен список основных поисковых серверов.

### *Томские ресурсы*

Томская поисковая система **Tambo** – <http://tambo.ru> (рис. 1.9).

Томская поисковая система – <http://get.tom.ru/>

Поиск по Тонету от Яндекс – <http://poisk.tomsk.ru/>

**Поисковый сервер по томским ftp-ресурсам** – <http://sea.tomsk.ru>.

**TomskNet FTPsearch** <http://ftpsearch.tomsk.ru>. Поисковый сервер по томским публичным ftp-ресурсам.



Рис. 1.9 – Окно поисковой системы Tambo.

### *Российские ресурсы*

**Яндекс** <http://www.yandex.ru> (рис. 1.10) и <http://www.ya.ru> (облегченная версия).

**Rambler** <http://www.rambler.ru>

**Апорт!** <http://www.aport.ru>

**@MAIL.RU** <http://mail.ru>

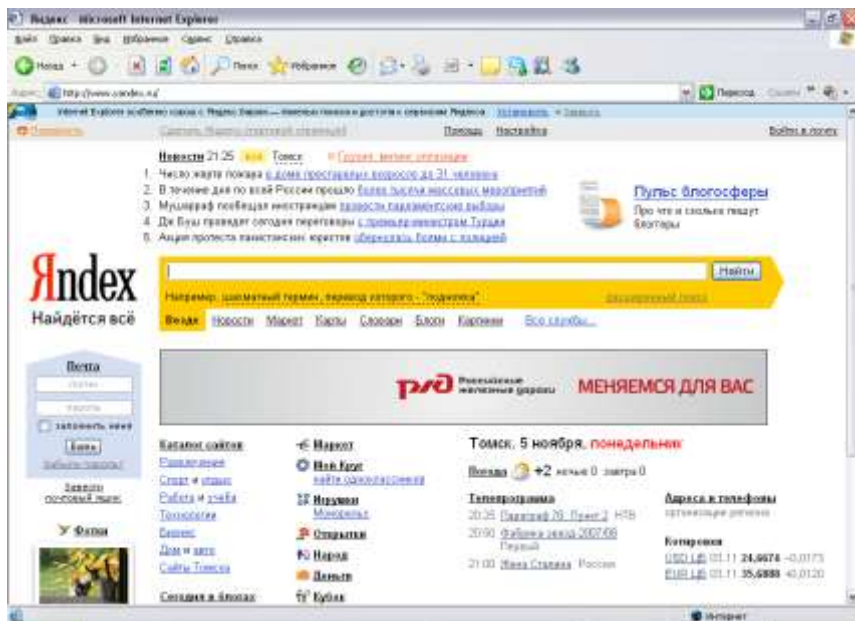


Рис. 1.10. Окно поисковой системы Яндекс.

### *Всемирные ресурсы*

**Google** [www.google.com](http://www.google.com). Google, Inc. – одна из самых новаторских Интернет-компаний. Уникальная технология поиска и инфраструктура позволяют быстро находить ответы даже на самые сложные запросы. Компания была основана в 1998 году и в короткий срок стала одной из самых популярных поисковых систем в Интернете. Есть и русскоязычный ресурс [www.google.ru](http://www.google.ru) (рис. 1.11).

**Alta Vista** <http://www.altavista.com>

**Excite** <http://www.excite.com>

**Mamma Metasearch** <http://www.mamma.com>

**Lycos** <http://www.lycos.com>

**HotBot** <http://hotbot.lycos.com>

**WebCrawler** <http://www.webcrawler.com>



Рис. 1.11 – Окно поисковой системы Google.

### 2.3. Метапоисковики

Увы, но ни одна поисковая машина, даже самая хорошая, не может похвастаться полнотой своей базы. Разные поисковые машины дают разные результаты при одних и тех же поисковых запросах. Как быть, если пользователь остался неудовлетворенным результатами поиска через какой-либо поисковый сервер? Повторить запрос через другой сервер? Затем через другой? А нельзя ли сразу организовать запрос через несколько серверов одновременно?

Для этого и созданы **системы метапоиска**, реализованные как в форме метапоисковых серверов, так и в форме программных продуктов.

#### *Метапоисковые серверы*

К серверным системам метапоиска относятся:

**MetaBot** <http://www.metabot.ru>

**AdClick** <http://www.adclick.ru>

**AllWeb** <http://www.allweb-search.ru>

**Nigma** <http://nigma.ru> (рис. 1.12)

**Turbo 10** <http://turbo10.com> (рис. 1.13)



стеризацию, обычно предлагают как минимум два списка результатов поиска. Это обычный список сайтов и список кластеров – ключевых слов, в которые были сгруппированы полученные результаты поиска. Выбор ключевого слова в этом списке приведет к отображению соответствующих этому слову ссылок на найденные сайты.

### ***Программы-метапоисковики***

**Метапоисковая утилита Search+** (Ю. Овчинников, Alone Co., <http://srchplus.chat.ru/searchplus.exe>, лицензия: freeware, размер: 1,11 Мб).

Эта программа предлагает поиск более чем по 60 различным категориям в 120 поисковых системах России и всего мира, причем категории и серверы не встроены в программу, а могут быть дополнены или изменены самим пользователем. При каждом запуске программа предлагает обновить через Интернет список категорий.

Ссылки, найденные программой, объединяются, сортируются и выводятся в главном окне. У каждой ссылки указывается адрес, заголовок, краткое описание, дата создания и степень соответствия (рис. 1.14). Количество ссылок с каждого поискового сервера задается в параметрах настройки.

Из дополнительных возможностей программы можно указать автоматическую паузу, обновление поиска, поиск в найденном, проверку ссылок с возможностью автоматического удаления «мертвых», сохранение и восстановление результатов поиска, так называемый «черный список».

Очень удобна возможность сохранения результатов поиска не только во внутреннем XML-совместимом формате, но и в популярных форматах CSV (Excel), HTML, IE URL.



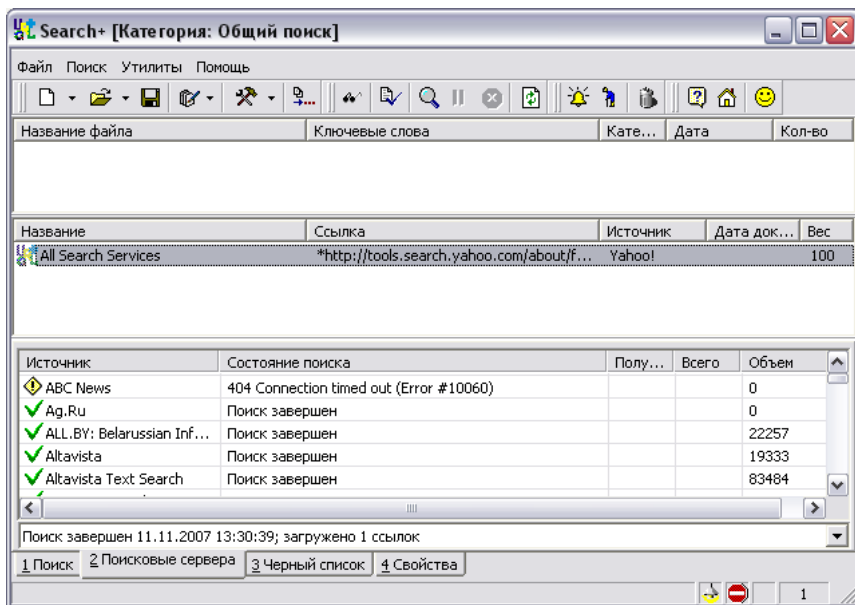


Рис. 1.14. Диалоговое окно Search+ с результатами поиска.

**Программа метапоиска Copernic Agent Professional** (Copernic Inc., <http://www.copernic.com>, коммерческий продукт, размер 5,8 Мб).

Программа ищет информацию о файлах или тексте, используя более шестисот поисковых машин и 129 категорий поиска, и работает с контентом многих крупных издательств и агентств, посылая запросы к CNN, MSNBC, New York Times, Forbes и т.д. Copernic Agent Pro позволяет создать индивидуальные настройки, сузив зону и критерии поиска, используя доступные движки. Найденные таким образом ссылки не дублируются и проверяются на жизнеспособность. Результаты поиска сохраняются в собственной базе программы, для возможности их последующего поиска и просмотра; при желании их можно сохранять в виде html-файлов.

Бесплатно распространяемая версия этой программы **Copernic**

**Agent Basic** (3,37 Mb) позволяет производить поиск с использованием 13 поисковых систем и 9 категорий (v. 6.12) и не допускает индивидуальные настройки (рис. 1.15).

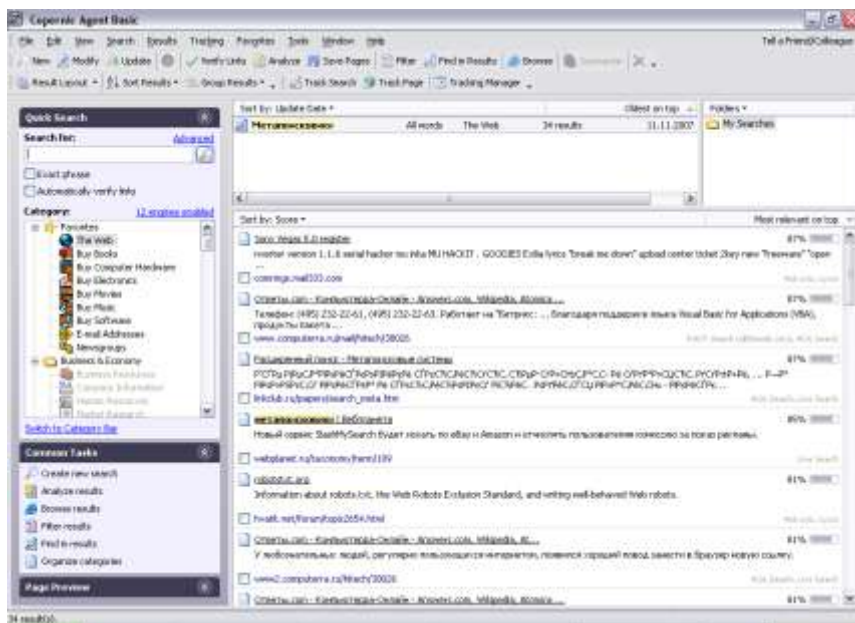


Рис. 1.15. Интерфейс метапоисковой системы Copernic Agent Basic.

**Система метапоиска Quintura Search** (Quintura Inc., <http://www.quintura.com>, лицензия: freeware, размер 5,52 Мб).

В основе программы Quintura Search лежат принципы естественного ассоциативного поиска. Начать работу с программой можно выбрав поисковую службу в списке **Искать в** и введя запрос в строку поиска. После обработки запроса слова, связанные по смыслу с запросом, выводятся на экран на левой панели окна Quintura Search (рис.1.16). Это визуальная карта программы. Визуальная карта указывает на возможные направления для дальнейшего поиска.



Подводя итог, можно сказать, что единой оптимальной схемы поиска в Интернет не существует. В зависимости от специфики необходимой информации, для ее поиска нужно использовать соответствующие поисковые службы. В принципе, конечно, можно всегда пользоваться какой-нибудь одной поисковой системой, но чем грамотнее подобраны поисковые службы и составлен запрос на поиск информации, тем качественнее будут результаты поиска. В заключение приведем кратко советы по работе с поисковыми системами.

- Проверяйте орфографию в написании слова.
- Используйте синонимы, если список найденных страниц слишком мал.
- Ищите больше чем по одному слову. Максимально сужайте предмет поиска.
- Не начинайте обычные слова с большей буквы, кроме имен собственных.
- Используйте ссылку «найти похожие документы», если один из найденных документов наиболее близок к искомому.
- При необходимости используйте язык запросов, чтобы сделать запрос более точным
- Обратите внимание, что контекст документа уже может содержать ответ, т.е. не потребуется заходить в сам документ.
- Используйте настройки поиска, чтобы задать удобные параметры описания документов и структуру страницы с результатами поиска.
- Для ускорения поиска отключите загрузку картинок в браузере или пользуйтесь облегченными версиями поисковых систем.

## **Задание 2**

1. В соответствии с выбранным вариантом, найти указанную

информацию на томских информационных ресурсах, используя тематические каталоги и поисковые машины Рунета.

*Примечание.* Можно использовать аналогичные ресурсы своего региона.

- 1) Найти расписание авиарейсов из Томского аэропорта.
- 2) Найти расписание движения ж. д. транспорта по станции Томск 1.
- 3) Найти расписание автобусов дальнего следования по автовокзалу г. Томска.
- 4) Найти цены на хостинг у нескольких томских провайдеров.
- 5) Найти электронные адреса пяти компьютерных фирм Томска. Какие фирмы предлагают кредит при покупке ПК?
- 6) Найти два-три предложения о работе в сфере программирования.
- 7) Найти два-три предложения о работе в бухгалтерской сфере.
- 8) Найти два-три предложения о работе в экономической сфере.
- 9) Каковы тарифы на коммунальные услуги в г. Томске?
- 10) Какие фонды хранятся в Государственном архиве Томской области?
- 11) Найти цены на услуги IP-телефонии в г. Томске двух поставщиков этой услуги.
- 12) Найти цены на статистическую информацию, распространяемую службами статистики Томской области.
- 13) Описать состояние социально-трудовой сферы Томской области за последний отчетный период.
- 14) Какие малые предприятия г. Томска получают финансирование из Фонда Бортника?
- 15) Найти информацию о системе поддержки малого предпринимательства в Томской области.
- 16) Найти информацию о деятельности в г. Томске публичных центров правовой информации.

17) Какова структура электронных информационных ресурсов в Томском центре научно-технической информации?

18) Привести статистику температуры, относительной влажности воздуха и атмосферного давления за прошедшую неделю в г. Томске.

19) Найти предложения фирм и предприятий г. Томска по экспорту товаров и внешнеторговый оборот Томской области за последний отчетный период.

20) По каким специальностям, связанным с экономической информатикой, осуществляется набор в вузах г. Томска?

21) Найти адрес, контактную информацию (телефоны, электронные адреса, имя главного редактора) пяти томских СМИ, имеющих свои Web-сайты.

22) Найти адрес, контактную информацию (телефоны, электронные адреса) пяти рекламных агентств г. Томска.

23) Найти адрес, контактную информацию (телефоны, электронные адреса) Томской службы занятости и описать виды ее деятельности.

24) Найти электронные адреса пяти томских фирм, занимающихся Web-дизайном.

25) Найти адреса и контактную информацию (телефоны, электронные адреса) пяти томских организаций, занимающихся аудиторской деятельностью.

26) В каких библиотеках города имеются книги Эдварда Йордона и Бьерна Страуструпа? Доступны ли их электронные варианты?

27) Какие банки в г. Томске оказывают услуги Интернет-банкинга? Каковы виды услуг и условия их предоставления?

28) Каковы условия предоставления сеансового доступа в Интернет (dial-up) у томских провайдеров?

29) В каких библиотеках города имеются книги лауреатов Но-

белевской премии по экономике В.В. Леонтьева и Л.В. Канторовича? Доступны ли их электронные варианты?

30) Описать структуру органов городского самоуправления вашего города. Найти фамилии депутатов городской Думы по тому избирательному округу, где вы проживаете.

2. Познакомиться с интерфейсом одного из описанных выше метапоисковых серверов, организовать свой тематический запрос, показать полученные результаты.

3. Установить одну из описанных выше метапоисковых программ и выполнить те же операции.

4. Оформить отчет о проделанной работе, приведя скриншоты страниц сайтов с найденной информацией и описание поисковых запросов.

### 3. ЗАГРУЗКА ИНТЕРНЕТ-ИНФОРМАЦИИ

#### 3.1. Менеджеры загрузок

Менеджеры загрузок (download managers) – это класс программ, значительно облегчающих процесс загрузки файлов с Интернет-серверов на персональные компьютеры пользователей. Основными преимуществами этих программ перед встроенными средствами обычного браузера считаются такие опции, как возможность докачки файлов с текущей позиции после обрыва связи, запуск заданий по расписанию и загрузка в несколько потоков. Они могут интегрироваться с браузером, автоматически снижать скорость загрузки, когда пользователь работает в Интернет, также большинство подобных программ обладает встроенным FTP-клиентом, позволяет хранить учётные данные для доступа к определённым серверам и многое другое.

Одной из популярных программ такого рода является **ReGet Deluxe** компании Reget Software (<http://www.reget.com>) (рис. 1.17).

На сайте производителя можно получить полнофункциональ-

ную демонстрационную версию программного продукта сроком на 30 дней.

ReGet Deluxe является очень удобным инструментом для опытных пользователей Интернета и тех, кому приходится часто загружать файлы.

ReGet Deluxe загружает (с возможностью докачки после обрыва) файлы как с файловых (FTP), так и с Web (HTTP)-серверов.

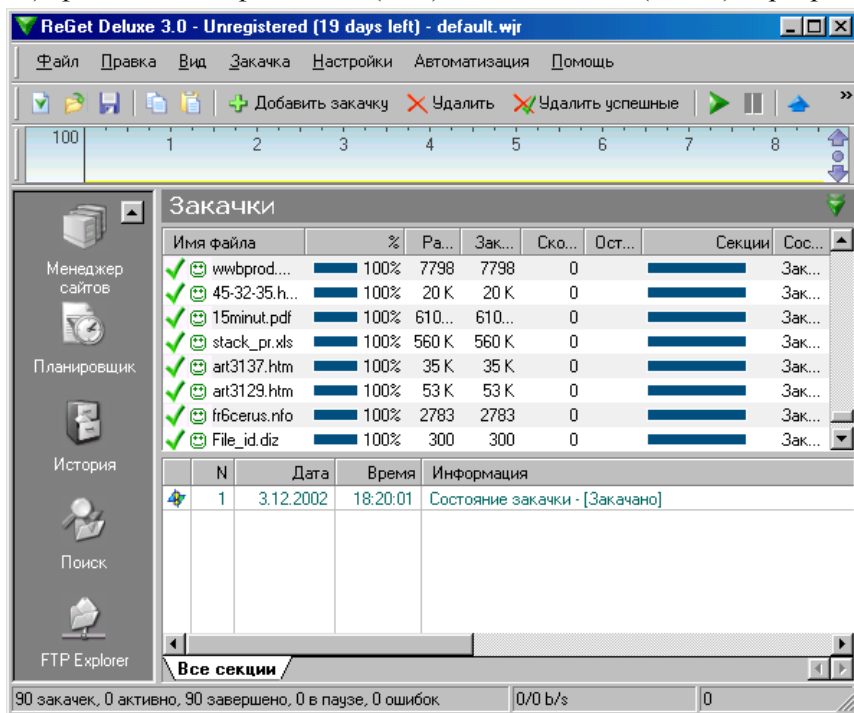


Рис. 1.17. Интерфейс менеджера загрузок ReGet Deluxe 3.0.

Она предпринимает неограниченное количество попыток загрузить нужный вам файл, пока попытки не увенчаются успехом, причем докачивает файлы после обрыва связи с того места, где закачка была прервана. Это позволяет сохранить время работы, так как загружать то, что уже есть на жестком диске, не придется.



ReGet Deluxe разбивает файл на части и загружает эти части одновременно, что позволяет увеличить скорость загрузки от 300% до 500% по сравнению со скоростью скачивания средствами Internet Explorer. Встроенный FTP-браузер отображает структуру удаленного FTP-сервера в удобном для чтения формате, позволяя пользователю выбрать, какие файлы или директории он хочет загрузить.

Чтобы ReGet Deluxe начал скачивать интересующий вас файл, необходимо добавить загрузку в очередь ReGet Deluxe.

Для этого, работая в браузере, кликнуть по ссылке на требуемый файл правой кнопкой мыши и в появившемся контекстном меню выбрать **Закачать при помощи ReGet Deluxe**. При этом поля **URL** и **Описание загрузки** заполнятся автоматически. В поле **Директория для сохранения** по умолчанию появляется путь C:\My Downloads. Его можно изменить. Если вы хотите сохранить файл под другим именем, введите желаемое имя в поле **Сохранить как....**

Другой способ – запустить Reget Deluxe и выбрать в его Панели инструментов кнопку **Добавить загрузку**. При этом открывается пустое окно и необходимо вручную задавать свойства загрузки. Но URL файла часто оказывается весьма длинным. Поэтому сначала рекомендуется скопировать его в буфер обмена, щелкнув по нему правой кнопкой и выбрав в меню **Копировать ярлык**, а затем нажать клавиши Ctrl+V, чтобы вставить его в соответствующее поле диалогового окна Свойства.

Примерно аналогичными возможностями обладает программа Download Master (компания Westbyte, [www.westbyte.com/dm](http://www.westbyte.com/dm)), распространяемая по лицензии «freeware» (рис. 1.18-1.20).

Программа обладает удобным и наглядным интерфейсом на русском языке, обеспечивает автоматическую загрузку файлов, графический контроль за процессом загрузки, имеет возможность автоматического распознавания типа файлов с помещением их в

указанные категории. Для повышения скорости загрузки файлов с использованием протоколов HTTP, HTTPS и FTP используется разбиение файла на потоки, которые закладываются одновременно. Программа также поддерживает докачку файла с текущей позиции после обрыва связи.

Для повышения удобства использования Download Master интегрируется в браузеры Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari, Opera, заменяя стандартный модуль загрузки.



Рис. 1.18. Интерфейс менеджера загрузок  
Download Master v5.6.1.1187

Чтобы начать загрузку, следует кликнуть по ссылке на требуемый файл правой кнопкой мыши и в появившемся контекстном меню выбрать **Скачать при помощи Download Master**.

В появившемся окне (рис. 1.19) следует выбрать категорию и каталог загрузки .

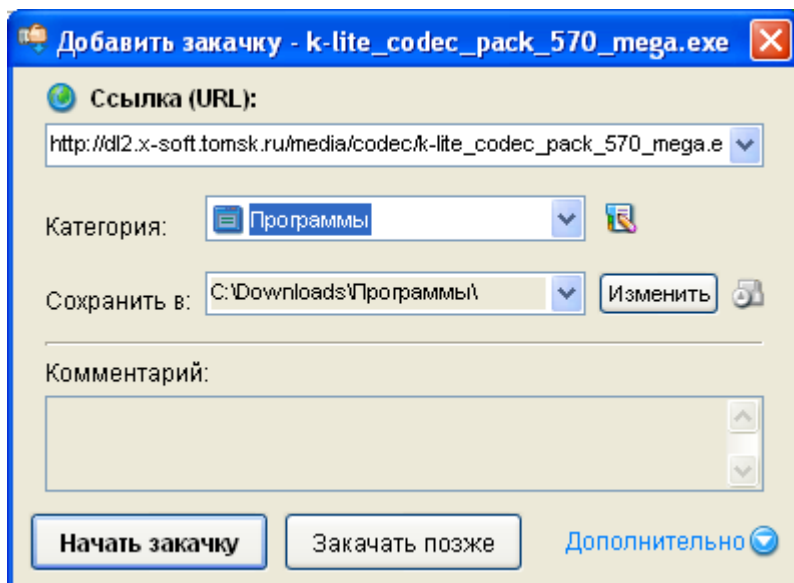


Рис. 1.19. Выбор категории и каталога загрузок в Download Master

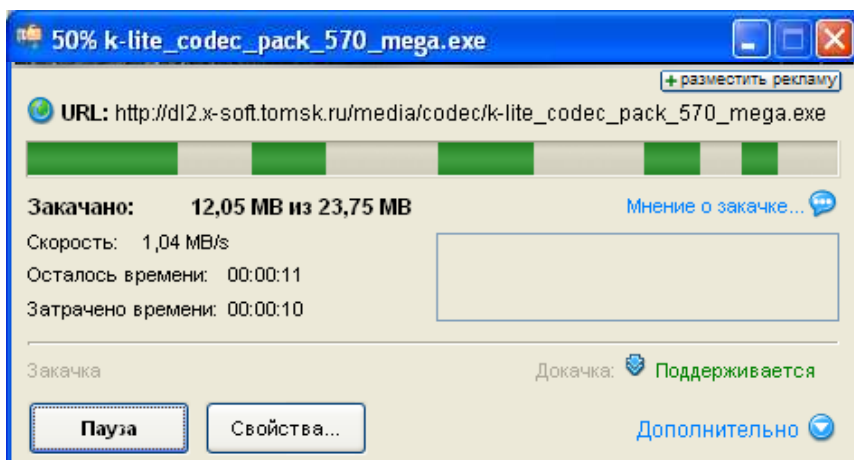


Рис. 1.20. Процесс загрузки в Download Master

Другие примеры менеджеров загрузок: Leech Get (рис. 1.21), GetRight, GoZilla, Get Down, и др.

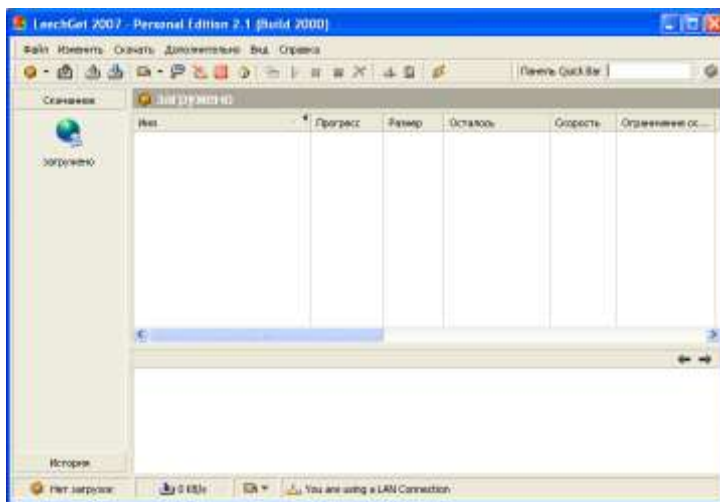


Рис. 1.21. Интерфейс менеджера загрузок  
LeechGet 2007 Personal Edition 2.1

### 3.2. Загрузка сайтов и отдельных Web-страниц

Желание скопировать хороший сайт к себе на жесткий диск время от времени возникает у каждого после обнаружения какого-нибудь шедевра веб-строительства. Несколько иные мотивы для копирования страниц и сайтов имеют люди, чей модем, телефонная станция или провайдер оставляют желать лучшего в смысле скорости соединения.

Ручное сохранение страниц – дело трудоемкое, да и не нужное, так как для этого существуют оффлайн-браузеры (offline browser), или «Web-качалки». Такие программы скачивают страницу, определенную директорию, целый сайт и даже группу сайтов, связанных ссылками, делая их доступными в отключенном от Интернета

состоянии.

Теоретически, оффлайн-браузер может скачать весь Интернет. Идя по ссылкам, без которых не обходится ни один сайт, программа будет копировать все новые и новые страницы. Для того чтобы этого не произошло, существуют специальные настройки. Так, можно загрузить все файлы, находящиеся в пределах одного домена или же определенной директории. Можно разрешить оффлайн-браузеру путешествовать не только по внутренним ссылкам, ведущим на страницы внутри сайта, но и по внешним, которые идут на страницы, расположенные на другом сервере. Для того чтобы не иметь всего содержимого Интернета на жестком диске своего компьютера, необходимо ограничить глубину загрузки по ссылкам. Так, например, страница А, связанная ссылкой со страницей В, расположена от нее на глубине, равной одной ссылке. Если же на странице В расположена ссылка на страницу С, то глубина между А и С становится равной двум ссылкам.

Затянуться процесс загрузки может и в связи с тем, что на сайте расположен какой-либо крупный архив файлов. Для того чтобы не перекачивать не нужные вам файлы, в оффлайн-браузере предусмотрены фильтры. С помощью них можно ограничивать загрузку определенных типов файлов. Так, можно отключить загрузку графических файлов или архивных. Или, напротив, можно настроить фильтр так, чтобы он скачивал только их. Так, например, нередко удобнее скачать только документы в формате .pdf, содержащие какие-либо инструкции, чем загружать полностью весь сайт.

Программные продукты, относящиеся к классу оффлайн-браузеров, достаточно многочисленны и разнообразны.

Среди бесплатно распространяемых оффлайн-браузеров можно назвать **WinHTTrack WebSite Copier** (Xavier Roche, <http://www.httrack.com>) (рис.1.22).



Рис. 1.22. Интерфейс оффлайн-браузера  
**WinHTTrack WebSite Copier.**

Программа имеет удобный и интуитивно понятный интерфейс (в том числе, русскоязычный). Поочередно переходя от одного диалогового окна к другому, пользователь производит настройку загрузки:

- 1) задается имя загрузки и тематическая категория (что бывает удобно при множественных загрузках);
- 2) определяется папка, куда будет помещен загруженный сайт (по умолчанию C:\Мои Web Сайты);
- 3) вводится URL и параметры загрузки (рис. 1.23).

Загруженные страницы просматриваются с помощью обычного браузера.

Другие примеры оффлайн-браузеров: Offline Explorer Pro, Teleport Pro, WebSite Extractor, SuperBot, BlackWidow и др.

Большинство оффлайн-браузеров относятся к типу shareware (условно-бесплатное программное обеспечение), т.е. бесплатно предоставляются на ограниченный период (~ 30 дней) для ознакомления, после чего требуют пройти регистрацию и заплатить некоторую сумму.

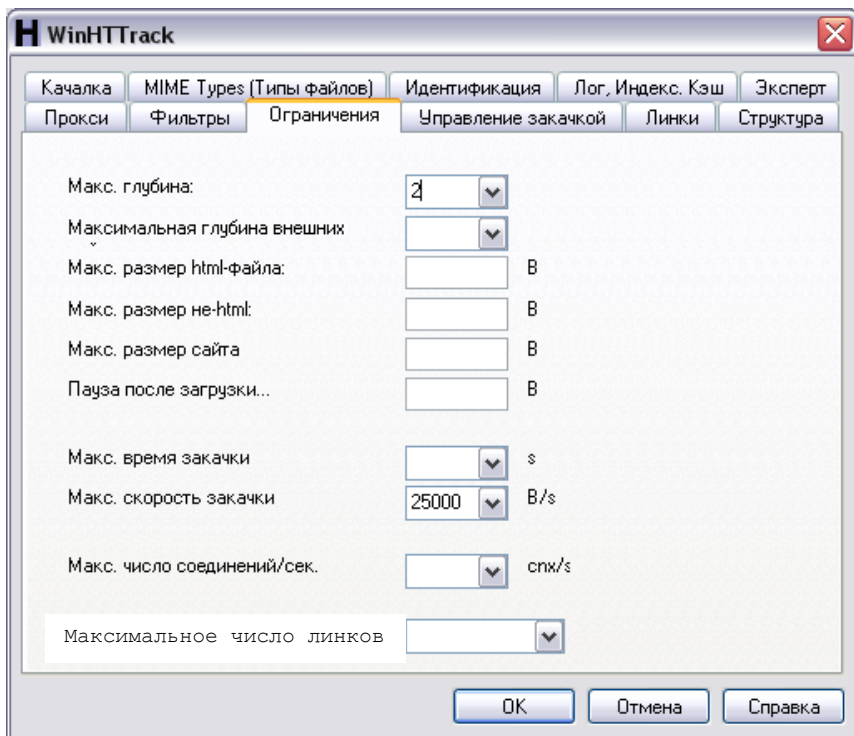


Рис. 1.23. Диалоговое окна офлайн-браузера WinHTTrack WebSite Copier с параметрами загрузки.

Отдельные Web-страницы – полностью или частично – можно сохранять с помощью обычного браузера. В Internet Explorer это можно выполнить четырьмя способами (рис.1.24).

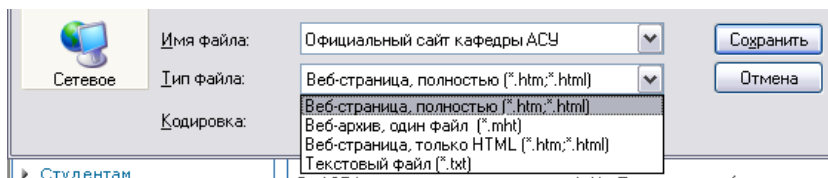


Рис. 1.24. Выбор способа сохранения Web-страницы «Официальный сайт кафедры АСУ (<http://www.asu.tusur.ru>)

1. Выбираем меню **Файл | Сохранить как... | Тип файла | Веб-страница, полностью** и определяем каталог сохранения. После нажатия кнопки **Сохранить** появляется информационное окно о процессе сохранения (рис. 1.25) и в выбранном каталоге помещается htm-файл и папка, имеющие одинаковые названия, например файл «Официальный сайт кафедры АСУ.htm» и папка «Официальный сайт кафедры АСУ.files».

В папке находятся файлы с элементами графического оформления Web-страницы (графические файлы, css-таблицы и др.). Удаление папки вызывает удаление и htm-файла.

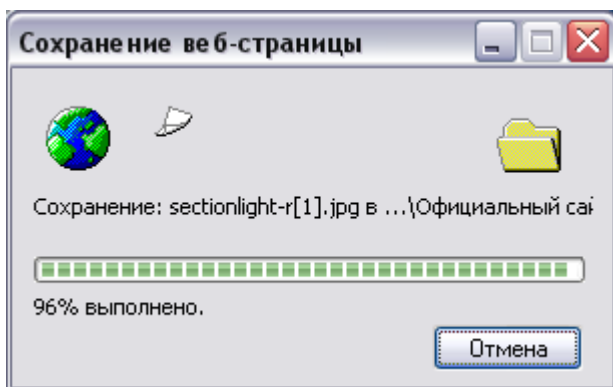


Рис. 1.25. Информационное окно сохранения Web-страницы.

2. Выбираем меню **Файл | Сохранить как... | Тип файла | Веб-архив, один файл**. В этом случае в выбранном каталоге помещается один файл с расширением .mht (Microsoft Hypertext Archive – Гипертекстовый архив Microsoft), в котором сохраняется выбранная web-страница со всеми деталями оформления. При наведении на любую гиперссылку на этой странице в информационной строке (внизу слева) появляется соответствующий адрес, по которому можно перейти (если, конечно, компьютер подключен к



сети в данный момент :-))).

3. Выбираем меню **Файл | Сохранить как... | Тип файла | Веб-страница, только HTML**. В этом случае в выбранном каталоге помещается один файл с расширением .htm, в котором сохраняется выбранная web-страница без деталей графического оформления. На месте картинок появляются квадратики с красными крестиками.

4. Выбираем меню **Файл | Сохранить как... | Тип файла | Текстовый файл**. В этом случае в выбранном каталоге помещается один файл с расширением .txt, в котором сохраняется только текстовая информация с выбранной web-страницы.

Все эти способы находят применение в зависимости от того, с какой степенью детализации надо получить информацию с выбранной web-страницы.

Кроме того, существуют программы независимых производителей, которые при установке встраиваются в Internet Explorer и также позволяют сохранять информацию web-страниц (так называемые программы-аддоны<sup>1</sup>). Например, **SaveChm** (Я. Кириллов, <http://yarix.by.ru>), работающая с IE6 и IE7.

При ее использовании в выбранном каталоге помещается файл с расширением .chm (Compiled HTML Help – Скомпилированный HTML-файл справки), в котором сохраняется выбранная web-страница со всеми деталями оформления. Этот файл открывается с помощью программы Microsoft HTML Help, входящей в набор стандартных программ Microsoft. В интерфейсе этой программы содержится меню (рис. 1.26), с помощью которого можно управлять содержимым файла.

---

<sup>1</sup> Add-on (англ.) – в информатике: встраиваемая утилита, увеличивающая возможности какой-либо программы.

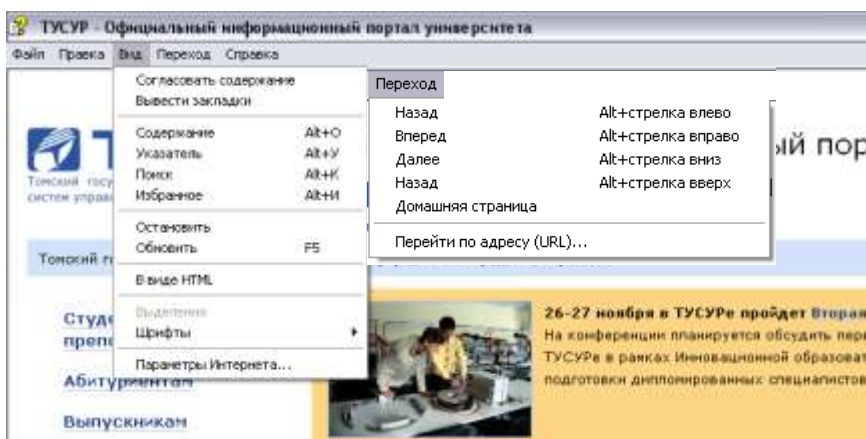


Рис. 1.26. Web-страница, сохраненная в формате .chm. Открыты пункты меню **Вид** и **Переход**.

### Задание 3

1. Открыть главную страницу любого Web-сайта и сохранить ее, пользуясь возможностями браузера Internet Explorer (четыре способа). Описать отличия этих способов.

2. Открыть страницу томского тематического каталога <http://catalog.tomsk.ru> и обратиться к категории «Компьютеры». Запустить программу WebSite Copier и скачать Web-сайт, адрес которого находится в списке категории «Компьютеры» под номером, совпадающим с номером вашего варианта. Предварительно указать каталог, в котором будет сохраняться скачиваемая информация, ввести IP-адрес и порт прокси-сервера (если необходимо), установить глубину скачивания не более 2. Для этого нужно найти и изменить соответствующие установки в меню настроек. После работы программы WebSite Copier в указанном каталоге будет создан сайт-зеркало. Открыть его (файл index.html), описать, какие ресурсы сайта сохранились в зеркале.

3. Через сайт <http://ftpsearch.tomsk.ru> на томских ftp-ресурсах найти файл с названием «freeware» под номером, соответствующим вашему варианту, Запустить Download Master (или другой менеджер зачек), указать каталог, в котором будет сохраняться скачиваемая информация, и скачать найденный файл, используя менеджер зачек.

4. Перейти на сайт Центра автоматизированных информационных ресурсов Томской области <http://rcair.tomsk.ru> , зайти в раздел «Нормативные документы» или «Проекты информатизации» и, пользуясь менеджером зачек Download Master, скачать один из нормативных документов, находящихся в этих разделах.

5. Оформить отчет о проделанной работе со скриншотами web-страниц, приложив скачанные сайты и файлы.

## Лабораторная работа 2

# СОЗДАНИЕ WEB-САЙТА

***Цель работы:** получить навыки разработки html-страниц статического web-сайта с использованием текстового редактора.*

### ВВЕДЕНИЕ

**Web-страница** — это документ, организованный в виде гипертекста с включениями текста, графики, звука, видео или анимации и снабженный уникальным адресом (URL). В Интернет просмотр web-страниц осуществляется посредством специальной программы-браузера. Браузер интерпретирует код HTML, содержащийся в файле, и согласно командам форматирования собирает готовую Web-страничку. Web-страница может иметь статическое или динамическое построение.

**Web-сайт** — это совокупность web-страниц с повторяющимся дизайном, объединенных по смыслу и физически находящихся на одном **web-сервере**, обеспечивающем предоставление информации в службе глобального соединения. Web-сервер хранит и предоставляет во внешнюю сеть данные, организованные в виде web-страниц. Web-сервер отвечает за обработку запросов клиентов к web-сайту и исполнение CGI-, JSP-, ASP-, PHP- и других приложе-

ний<sup>1</sup>.

Можно назвать несколько основных видов Web-сайтов.

Во-первых, это «**Интернет-визитка**» **фирмы** – аналог обычной визитной карточки, информирующей о владельце, только информации можно представить значительно больше, чем умещается на обычной визитке размером 5 на 9 см. Примерами таких сайтов может быть любой сайт небольшой компании.

Во-вторых, это **корпоративный сайт** – дальнейшее развитие «Интернет-визитки», сайт с мощной информационной поддержкой деятельности фирмы. Примеры: [www.gazprom.ru](http://www.gazprom.ru), [www.beeline.ru](http://www.beeline.ru).

В-третьих, это средство поддержки **электронного бизнеса**. К таким сайтам относятся всевозможные интернет-магазины и электронные торговые площадки. Например: [www.ozon.ru](http://www.ozon.ru), [www.rusmet.ru](http://www.rusmet.ru).

В-четвертых, это средство формирования **Интернет-сообщества**. Вокруг таких сайтов формируется сообщество людей, которые, общаясь между собой, сами участвуют в развитии сайта. Примерами являются [www.wikipedia.com](http://www.wikipedia.com), [www.livejournal.com](http://www.livejournal.com), [www.developing.ru](http://www.developing.ru).

Ну, и в-пятых, это отличное средство **самовыражения**. Эту цель преследуют многие домашние странички и проекты, которые создаются в первую очередь для удовлетворения интересов их ав-

---

<sup>1</sup> CGI (Common Gateway Interface) - спецификация взаимодействия веб-сервера и внешней программы, которую веб-сервер запускает для обработки запроса. Эта программа, вне зависимости от своей природы, называется CGI-приложением, или CGI-скриптом.

JSP (JavaServer Page) – интерфейс доступа к базам данных или приложениям на языке Java; позволяет формировать динамические страницы на web-сервере.

ASP (Active Server Pages)-файл – это html-страница со встроенными сценариями, которые запускаются сервером после получения URL этой html-страницы. ASP-файлы способны выполнять web-серверы компании Microsoft.

PHP (PHP: Hypertext Preprocessor) – скриптовый язык программирования, созданный для генерации HTML-страниц на веб-сервере и работы с базами данных.

Источник: Глоссарий.ru (<http://www.glossary.ru>).

торов. Мало кому интересны фотографии любимого хомячка или список ссылок на любимые сайты, кроме самого владельца сайта. Таких проектов в Сети очень много. Их можно увидеть на сайтах бесплатного хостинга: [www.narod.ru](http://www.narod.ru), [www.boom.ru](http://www.boom.ru) и других. Тем не менее отметим, что такое любительское сайтостроительство может служить хорошей стартовой площадкой для серьезных сетевых проектов.

Пусть такой стартовой площадкой послужит для вас эта лабораторная работа.

Пока мы не будем вдаваться в подробности увлекательного мира профессионального web-программирования, а займемся созданием простого статического web-сайта, состоящего из нескольких web-страничек, написанных на языке HTML в текстовом редакторе.

Рекомендуется использовать не стандартный Блокнот (Notepad) Windows, а текстовый редактор Notepad++.

Notepad++ – это бесплатный редактор текстовых файлов, отличная замена стандартного Блокнота Windows с поддержкой синтаксиса большого количества языков программирования: C, C++, Java, C#, XML, HTML, PHP, Javascript, Pascal, Fortran, Perl, Python, VB и т.д. Он ориентирован для работы в операционной системе MS Windows, позволяет подсвечивать синтаксис в файлах, поддерживает поиск и замену по тексту, работу одновременно с несколькими файлами, а также поддержку функции drag & drop<sup>1</sup>.

Отметим также, что существует большое количество визуальных редакторов web-страниц. Эти редакторы обладают богатым набором инструментальных средств и предоставляют пользователю возможность создавать великолепно выглядящие страницы, ис-

---

<sup>1</sup> Последнюю версию Notepad++ можно скачать по адресу <http://notepad-plus.sourceforge.net/ru/site.htm>

пользуя многочисленные шаблоны и библиотеки картинок, как правило, входящие в комплект поставки редактора.

Одними из самых популярных визуальных редакторов являются пакеты Microsoft FrontPage ([www.microsoft.com/frontpage](http://www.microsoft.com/frontpage)), Macromedia HomeSite, Macromedia Dreamweaver ([www.macromedia.com/software/dreamweaver](http://www.macromedia.com/software/dreamweaver)), NetObjects Fusion ([www.netobjects.com](http://www.netobjects.com)).

Несомненно, визуальные редакторы позволяют значительно ускорить создание типовых страниц и программирование некоторых элементов, например сложных таблиц, написание которых от руки уходит много времени. Но, для того чтобы как следует разобраться со структурой HTML-документов и основными элементами HTML, лучше создать несколько страниц, не пользуясь этими сложными средствами. Это позволит в будущем как следует приглядывать за работой визуальных редакторов и корректировать код. Как правило, люди, знающие HTML «в лицо», в дальнейшем практически не пользуются визуальными редакторами.

## 1. СТРУКТУРА HTML-ДОКУМЕНТА

Итак, web-страница – это текстовый документ, размеченный с помощью специальных элементов HTML, которые называются **теги**, или html-теги. Такие страницы часто называют html-страницами. Они имеют расширение .html или .htm<sup>1</sup>. На экран html-теги не выводятся, они только указывают браузеру, как отображать содержимое документа.

Вы всегда можете посмотреть html-код любой страницы в браузере, вызвав на этой странице контекстно-зависимое меню и щелк-

---

<sup>1</sup> Браузеры поддерживают оба расширения. Поэтому решите для себя, какое расширение вы будете использовать.

нув по строке «Просмотр HTML-кода». Можно выбрать аналогичную строку в главном меню браузера (опция «Вид»).

В браузер встроен интерпретатор языка HTML. К сожалению, интерпретаторы, встроенные в различные браузеры, работают неодинаково и одна и та же html-страница может отображаться в них по-разному. Это приводит к необходимости тестирования сайта на разных браузерах, а уж затем – размещение его на хост-сервере.

Отметим еще один важный момент: HTML определяет логическую структуру документа и не является языком программирования. В нем нет логических последовательностей. Это именно язык разметки документов (текста).

Теги html бывают двух типов – **контейнерные** и **одиночные** и заключаются в угловые скобки `<имя_тега>`. Регистр записи имени тега не имеет значения.

Контейнерные теги состоят из пары – открывающий тег `<имя_тега>` и закрывающий тег `</имя_тега>`. Закрывающий тег завершает действие открывающего, например теги `<html>` и `</html>` начинают и заканчивают html-страницу.

Одиночный тег не требует закрывающего тега, например тег `<br>` начинает новую строку.

Итак, откроем Блокнот и наберем следующий текст с тегами (рис.2.1):

```
<html>
<head>
<title>Моя первая Web-страница</title>
</head>
<body>
ЗАГОЛОВОК <br>
Содержимое страницы
</body>
</html>
```



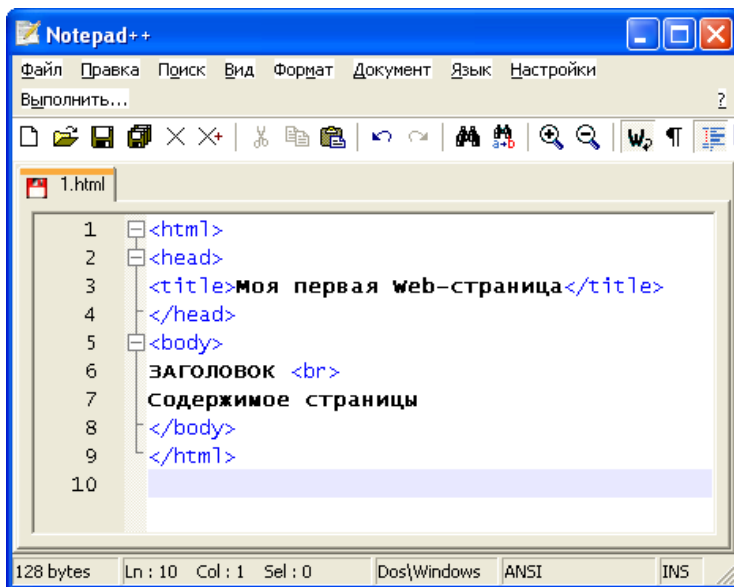


Рис. 2.1. Код html в Notepad++

Сохраним этот текст в файле с именем 1.html в папке Мой сайт, выбрав Тип файлов All types (".") (рис.2.2).

Закроем Блокнот и перейдем в папку Мой сайт. Сохраненный файл как html-документ по умолчанию будет открываться браузером, например MS Internet Explorer (рис. 2.3).

Теперь вся дальнейшая работа будет заключаться в добавлении все новой и новой информации в этот пока унылый информационный пейзаж по следующему алгоритму:

- 1) щелкнуть правой кнопкой мыши по полю web-страницы и вызвать «Просмотр HTML-кода»;
- 2) внести изменения в html-код;
- 3) сохранить изменения и закрыть (желательно) Блокнот;
- 4) обновить web-страницу (кнопка F5 на клавиатуре или кнопка Обновить в Панели инструментов браузера) и проанализировать

изменения;

5) повторить с позиции 1.

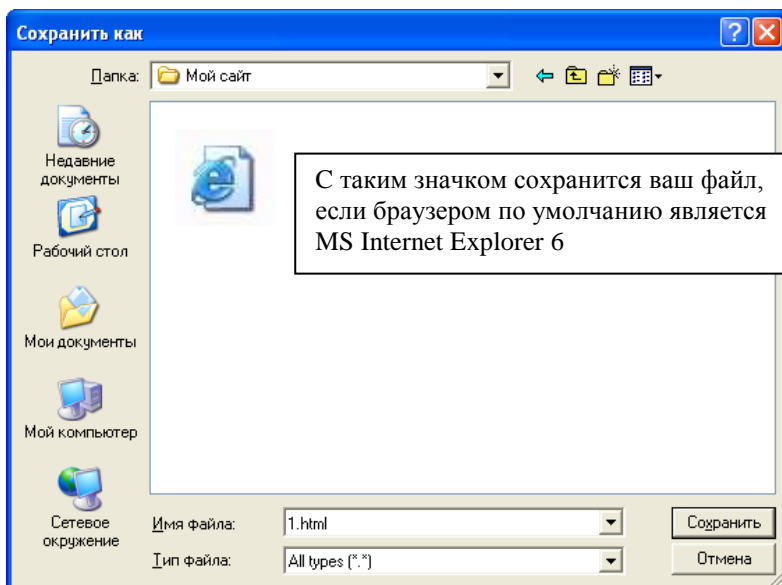


Рис. 2.2. Первое сохранение html-страницы



Рис. 2.3. Отображение html-страницы в браузере

Рассмотрим структуру html-документа.

Документ состоит из двух частей: заголовок, выделяемый тегами **<head>** и **</head>**, и тело, выделяемое тегами **<body>** и **</body>**. В заголовке содержится информация о документе – название, мета-информация, ссылки на внешние и внутренние таблицы стилей, сценарии JavaScript и т. д. В теле находится само содержимое документа – то, что выводится в окне браузера: текст

(здесь: ЗАГОЛОВОК Содержимое страницы), картинки, таблицы и т. д.

**Комментарий о комментариях.** Как любой язык, HTML позволяет вставлять в тело документа комментарии, которые сохраняются при передаче документа по сети, но не отображаются браузером.

*Синтаксис комментария:*

<!-- Это комментарий -->

*Комментарии могут встречаться в документе где угодно и в любом количестве.*

Название документа (Моя первая Web-страница) выделяется тегами <title> и </title>. Это название отображается в заголовке окна браузера, открывающего документ, и в файле закладок «Избранное», если он поддерживается Web-браузером.

Для расширения или модификации действия тега в него добавляют атрибуты, которые могут принимать различные значения. Наборы допустимых атрибутов для конкретного тега описаны в спецификации языка HTML (см., например, <http://www.citforum.ru/internet/html40>). Некоторые из них приведены в Приложении 1.

Правила записи атрибутов и значений:

- атрибуты следуют после имени тега через пробел;
- атрибуты отделяются друг от друга пробелами;
- порядок следования атрибутов произволен;
- атрибуты не нужно повторно описывать в закрывающем теге;
- значения атрибутов записывают после знака равенства в кавычках "..." (кавычки можно опускать, если значение атрибута состоит из одного слова или числа).

В тег <body> могут входить атрибуты, определяющие параметры страницы, такие, как цвет фона, фоновый рисунок, цвет и раз-

мер основного шрифта и гиперссылок и т. п. Цвета этих элементов web-страницы указываются в шестнадцатеричном исчислении. Но существует и возможность словесного указания цвета (см. Приложение 1)<sup>1</sup>.

Включим в тег <body> следующие атрибуты и зададим им значения:

```
<body bgcolor="#00FFFF" text=red>
```

Атрибут **bgcolor** задает цвет фона, а **text** – цвет текста.

После сохранения и обновления страницы на бирюзовом фоне вы увидите красные буквы.

Для большинства документов хорошо выглядит в качестве фона ненавязчивый полупрозрачный рисунок (обои). Для того чтобы оформить страничку фоновым рисунком, в теге <body> нужно указать атрибут **background**="имя\_файла". В этом случае графический файл с фоновым рисунком должен находиться **в том же каталоге, что html-страница**. В качестве фонового рисунка могут быть использованы только файлы формата GIF, JPG (JPEG) и PNG.

Пример использования этого атрибута:

```
<body background="back.jpg">
```

Регистр символов в имени файла имеет значение. Расширение указывать обязательно!

Большое значение имеет выбор рисунка: и цветовая насыщенность, и размер. На ярком рисунке будет не виден текст, а маленький рисунок будет выложен плиткой на web-странице.

Большие коллекции фоновых рисунков можно найти на сайтах <http://fonegallery.narod.ru> , [www.textures.boom.ru](http://www.textures.boom.ru).

Кстати, небольшой рисунок можно увеличить до необходимых размеров (с некоторой потерей качества), воспользовавшись графическими редакторами. Например, в состав MS Office 2003 и

---

<sup>1</sup> К сожалению, в черно-белом варианте издания этого пособия сами цвета не видны.

поздних выпусков входит Microsoft Office Picture Manager. Открыв им картинку, выберите в главном меню опцию **Рисунок → Изменить размер**. В правой части появившегося окна задайте или выберите необходимый размер. Сохраните файл.

Рекомендуется все графические файлы, входящие в состав html-документа, хранить в отдельной папке, например images. Тогда в соответствующих атрибутах надо указать **весь путь**:

```
<body background="images/back.jpg">
```

Рекомендуется также названия всех файлов (текстовых, графических) вашего сайта писать строчными буквами по причине разного восприятия различными серверами регистров букв.

## 2. УПРАВЛЕНИЕ ТЕКСТОМ

### 2.1. Заголовки

Для задания заголовков в документе используются контейнерные теги `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` (в порядке убывания размера). По умолчанию браузеры выводят заголовки жирным начертанием.

Пример:

```
<h1>Заголовок</h1>
```

Заголовки можно выравнивать по левому краю (по умолчанию), по правому краю и по центру. Для этого используется атрибут **align** с параметрами соответственно **left**, **right**, **center**.

Пример:

```
<h1 align=center>Заголовок</h1>
```

Заголовок будет отцентрирован.

### 2.2. Абзацы (параграфы)

Для разбивки текста на абзацы используется контейнерный тег

**<p>**. Встретив этот тег, браузер пропускает строку и начинает текст с новой строки. Это заметно облегчает восприятие текста, нежели сплошной текстовый блок – «кирпич».

Например,

`<p> Текст абзаца</p>`

Текст абзацев можно выравнивать по левому краю (по умолчанию), по правому, по центру и по ширине.

Для этого используется атрибут **align** с параметрами **left**, **right**, **center**, **justify**.

Например, запись

`<p align=justify > Текст абзаца</p>`

выравнивает абзац по ширине.

Для выравнивания текста или встроенных картинок также можно использовать контейнерный тег **<center>**.

### 2.3. Цитата

Контейнерный тег **<blockquote>** предназначен для обозначения в документе цитаты из другого источника. Текст, обозначенный тегом **<blockquote>**, отступает от левого края документа на 8 пробелов.

Пример:

`<blockquote>Текст цитаты</blockquote>`

### 2.4. Списки

Списки бывают трех видов: неупорядоченные, упорядоченные и списки определений.

**Неупорядоченные списки** создаются тегами **<ul>** **</ul>**. Внутри помещается весь список. Отдельные элементы списка отмечаются тегом **<li>** (закрывающий тег не требуется). У команд **<ul>** и **<li>** можно указать атрибут **type**, который принимает одно из трех зна-

чений: **disc**, **square**, **circle**.

Пример.

В html-коде:	На странице браузера:
<pre>&lt;ul&gt;   &lt;li&gt; Первый пункт   &lt;li&gt; Второй пункт   &lt;li&gt; Третий пункт &lt;/ul&gt;  &lt;ul type=circle&gt;   &lt;li&gt; Первый пункт   &lt;li type=square&gt; Второй пункт   &lt;li&gt; Третий пункт &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>• Первый пункт</li> <li>• Второй пункт</li> <li>• Третий пункт</li> </ul> <ul style="list-style-type: none"> <li>○ Первый пункт</li> <li>▪ Второй пункт</li> <li>○ Третий пункт</li> </ul>

**Упорядоченные списки** создаются тегами **<ol>** и **</ol>**.

У тега **<ol>** есть два атрибута **type** и **start**. Атрибут **type** задает тип нумерации:

1 – арабские числа 1, 2, 3, ...;

a – прописные буквы a, b, c, ...;

A – заглавные буквы A, B, C, ...;

I – маленькие римские числа i, ii, iii, ...;

I – большие римские числа I, II, III, ...

Атрибут **start** указывает, с какого номера начинать список.

У тега **<li>** можно указать тоже два атрибута: **type** и **value**. С помощью **value** можно задать номер элемента в списке.

Пример.

В html-коде:	На странице браузера:
<pre>&lt;ol type=i start=5&gt;   &lt;li&gt; Первый пункт</pre>	<p>v. Первый пункт</p>

<code>&lt;li&gt;</code> Второй пункт	vi. Второй пункт
<code>&lt;li type=1&gt;</code> Третий пункт	7. Третий пункт
<code>&lt;/ol&gt;</code>	
<code>&lt;ol type=I&gt;</code>	
<code>&lt;li&gt;</code> Первый пункт	I. Первый пункт
<code>&lt;li&gt;</code> Второй пункт	II. Второй пункт
<code>&lt;li value=10&gt;</code> Третий пункт	X. Третий пункт
<code>&lt;/ol&gt;</code>	

**Списки определений.** Для создания списка используются теги `<dl>` и `</dl>`. Элемент списка создается двумя тегам `<dt>` и `<dd>`. Они не требуют закрывающих тегов. Первый тег задает термин, второй – определение. Для большей наглядности термин можно выделить жирным шрифтом.

Пример.

В html-коде:

```
<dl>
<dt> <b> WWW (World Wide Web) </b>
<dd> Всемирная Паутина – основная служба в сети Интернет, позволяющая получать доступ к информации на любых серверах, подключенных к сети
<dt> <b> DNS (Domain Name System) </b>
<dd> Система доменных имен – распределенная служба формирования имен узлов, используемая в сети Интернет, устанавливающая соответствие между именами узлов и доменов, с одной стороны, и IP-адресами, с другой стороны.
<dt> <b> HTTP (HyperText Transfer Protocol) </b>
<dd> Протокол передачи гипертекста – базирующийся на TCP/IP протокол передачи гипертекста, обеспечивающий доступ к документам на web-узлах.
</dl>
```



На странице браузера:

**WWW (World Wide Web)**

Всемирная Паутина – основная служба в сети Интернет, позволяющая получать доступ к информации на любых серверах, подключенных к сети

**DNS (Domain Name System)**

Система доменных имен – распределенная служба формирования имен узлов, используемая в сети Интернет, устанавливающая соответствие между именами узлов и доменов, с одной стороны и IP-адресами, с другой стороны.

**HTTP (HyperText Transfer Protocol)**

Протокол передачи гипертекста – базирующийся на TCP/IP протокол передачи гипертекста, обеспечивающий доступ к документам на web-узлах.

## 2.5. Выделение текста

Для выделения части текста **полужирным** шрифтом используются теги контейнерного типа

**<b> Текст </b>**,

*курсивом* – теги

**<i> Текст </i>**,

текстом, имитирующим шрифт пишущей машинки, – теги

**<tt> Текст </tt>**.

Тег **<sup>** определяет верхний индекс, а тег **<sub>** – нижний, например, формула  $I_i = x_i^2 + a_{i,j}$  в html-коде будет выглядеть так:

**<i>I<sub>i</sub>=x<sub>i</sub></i><sup>2</sup>+<i>a<sub>i,j</sub></i>**

Обратите внимание, какой длинный и трудно читаемый (да и трудоемкий в исполнении) получился код. Поэтому вставка сложных формул в текст обычно осуществляется картинкой.

Текст можно выделить, используя режим бегущей строки, внеся

таким образом некий «динамизм» в статичный сайт<sup>1</sup>.

Для этого используется контейнерный тег **<marquee>** с различными атрибутами (поддерживается только в MS IE).

Пример.

```
<marquee align="center" bgcolor="#99CCFF" height="10"
width="800" direction="left" behavior="scroll" loop="2"
scrollamount="2" >Текст бегущей строки</marquee>
```

Атрибут **align** определяет расположение строки относительно страницы (см. выше), **bgcolor** задает цвет фона строки, **height** – высоту строки, **width** – ширину строки, **direction** – направление движения (left, right, up, down), **behavior** – поведение строки (scroll – обычная прокрутка, slide – прокрутка с остановкой, alternate – движение от края до края), **loop** – число прокруток строки, **scrollamount** – скорость движения строки (от 1 до 10).

## 2.6. Управление шрифтами

Для управления шрифтовым оформлением web-страниц предназначен контейнерный тег **<font>**. Атрибуты этого тега определяют размер, цвет и название шрифта. Текст, выделенный тегами **<font>** **</font>**, выглядит иначе, чем это задано по умолчанию или в теге **<body>**.

Пример.

```
<font face="Arial" size="6" color="#8080C0">Текст</font>
```

---

<sup>1</sup> Вообще-то, чтобы сделать html-страницу динамичной, надо использовать Java-скрипты...

Атрибут **face** задает название шрифта (по умолчанию Times New Roman). Обычно в Web используются немного разных видов шрифтов (именно они однозначно воспринимаются всеми браузерами). Это: Times New Roman, Arial, Courier New, Tahoma, Verdana.

Если же вы хотите сделать витиеватый заголовок – выполните его в графическом редакторе и вставьте картинкой.

Атрибут **size** определяет размер шрифта. Шрифт может иметь размер от 1 до 7. Можно прямо указать размер шрифта цифрой или указать смещение относительно базового значения (по умолчанию 3) в положительную или отрицательную сторону, например задание

size=+2

определит размер 5.

Атрибут **color** определяет цвет текста аналогично заданию цвета фона в теге <body>.

## 2.7. Спецсимволы

Как ввести в текст символы, которые отсутствуют на клавиатуре, например греческие буквы, или символы ×, ©, ®, или неразрывный пробел (в Word – сочетание клавиш Shift+Ctrl+Пробел), или знаки <, >, ", которые входят в состав тегов? Для этого используются особые последовательности символов (escape-последовательности), начинающиеся с символа & («амперсанд»).

Например, группа из 5...8 escape-последовательностей **&nbspnbsp** используется для получения абзацных отступов.

В таблицах 1 и 2 указаны некоторые escape-последовательности.

Таблица 1. Наиболее распространенные escape-последовательности

Название	Символ	&-ия
Прямая кавычка	"	&quot;;
Амперсанд	&	&amp;;
Неразрывный пробел (невидимый символ)	°	&nbsp;;
Цент	¢	&cent;;
Фунт	£	&pound;;
Копирайт	©	&copy;;
Зарегистрированный торговый знак	®	&reg;;
Градус	°	&deg;;
Плюс-минус	±	&plusmn;;
Знак умножения	×	&times;;
Знак параграфа	§	&sect;;

Эти сочетания символов просто размещаются в нужном месте текста, ничем более не выделяясь, и будут интерпретированы браузером как соответствующий символ. Точка с запятой, входящая в состав escape-последовательности, может опускаться.

Таблица 2. Греческий алфавит в HTML

Буква		Название	Обозначение в HTML	
прописная	строчная		прописная	строчная
Α	α	альфа	&#913;	&#945;
Β	β	бета	&#914;	&#946;
Γ	γ	гамма	&#915;	&#947;
Δ	δ	дельта	&#916;	&#948;
Ε	ε	эпсилон	&#917;	&#949;
Ζ	ζ	дзета	&#918;	&#950;
Η	η	эта	&#919;	&#951;
Θ	θ	тета	&#920;	&#952;
Ι	ι	йота	&#921;	&#953;
Κ	κ	каппа	&#922;	&#954;
Λ	λ	лямбда	&#923;	&#955;
Μ	μ	мю	&#924;	&#956;
Ν	ν	ню	&#925;	&#957;
Ξ	ξ	кси	&#926;	&#958;
Ο	ο	омикрон	&#927;	&#959;
Π	π	пи	&#928;	&#960;
Ρ	ρ	ро	&#929;	&#961;
Σ	σ	сигма	&#931;	&#963;
Τ	τ	тау	&#932;	&#964;
Υ	υ	ипсилон	&#933;	&#965;
Φ	φ	фи	&#934;	&#966;
Χ	χ	хи	&#935;	&#967;
Ψ	ψ	пси	&#936;	&#968;
Ω	ω	омега	&#937;	&#969;

## 2.8. Линия

Одиночный тег **<hr>** вставляет в текст горизонтальную разделительную линию.

Атрибуты тега:

**width** определяет длину линии в пикселях или процентах от ширины окна браузера;

**size** определяет толщину линии в пикселях;

**align** определяет выравнивание горизонтальной линии: **=left** – по левому краю документа, **=right** – по правому краю документа, **=center** – по центру документа (используется по умолчанию);

**noshade** определяет способ закрашки линии как сплошной (не требует указания значения). Без данного атрибута линия отображается объемной;

**color** определяет цвет линии.

## 3. ВСТАВКА РИСУНКОВ

Для размещения картинок на web-страницах существует одиночный тег **<img>**. Требования к графическому файлу – те же, что и для фонового рисунка (см. п. 1).

В html-коде нужно указать

```

```

Атрибуты этого тега:

**src** – обязательный параметр. Указывает адрес (URL) файла с картинкой; ошибка в его написании (что очень часто встречается у начинающих) приводит к тому, что картинка упорно не хочет отображаться в браузере. Поэтому будьте внимательны;

**width** и **height** задают ширину *n* и высоту *m* изображения (в пикселях) соответственно. Имейте в виду, что если указанные зна-

чения не совпадают с реальным размером изображения, изображение масштабируется (порой с заметной потерей качества);

**hspace** и **vspace** определяют отступ картинки (в пикселях) по горизонтали (*h*) и вертикали (*v*) от других объектов документа. Просто необходимо при обтекании изображения текстом;

**border** определяет ширину рамки (*b*) вокруг изображения в пикселях;

**align** определяет способ выравнивания изображения в документе. Если его не указать, картинка раздвигает строки текста, расположенные до и после нее и располагается слева (отцентрировать можно тегами `<center>` `</center>`).

Некоторые параметры атрибута align:

**left** – выравнивает изображение по левому краю документа, прилегающий текст обтекает изображение справа;

**right** – выравнивает изображение по правому краю документа, прилегающий текст обтекает изображение слева.

Атрибут **alt** определяет текст, отображаемый браузером при наведении курсора мыши на изображение или появляющийся на его месте, если браузер по каким-то причинам не может вывести изображение.

Примеры.

`<center></center>` – рисунок расположен по центру страницы, рамка отсутствует, размеры 100×30 пикселей, подпись к рисунку отсутствует;

`` – рисунок размером 120×90 пикселей обтекается текстом (рисунок слева, текст справа) на расстоянии 5 пикселей от рамки толщиной 3 пикселя, при наведении появляется текст «Зимнее утро».

## 4. ГИПЕРССЫЛКИ

### 4.1. Текстовые гиперссылки

Для того чтобы объединить несколько страниц в один сайт, необходимо связать их гиперссылками. Это и есть основной принцип построения WWW.

Гиперссылка может быть присвоена любому элементу текста, расположенного на странице. Для организации гиперссылки существует контейнерный тег `<A>`. Между открывающим и закрывающим тегами располагается текст, который будет отображен подчеркнутым в окне браузера. Именно этот текст и будет объектом, которому назначена гиперссылка на какой-то другой объект вашего сайта или глобальной сети<sup>1</sup>.

Рассмотрим атрибуты этого тега.

**href** – определяет находящийся между начальным и конечным тегами текст как гипертекстовую ссылку (URL, или линк) на объект, указанный в значении данного параметра.

Возможные варианты задания значения:

**href="http://www.tusur.ru"** – создает ссылку на web-сайт или www-документ;

**href="ftp://ftp.tomsk.ru/pub/"** – создает ссылку на ftp-сайт или расположенные на нем папки и файлы;

**href="mailto:abc@mail.ru"** – запускает почтовую программу-клиент с заполненным полем имени получателя;

**href="page1.html"** – указывает на документ, находящийся в том же каталоге, что и тот, с которого происходит вызов. Если же этот

---

<sup>1</sup> Поэтому не рекомендуется выделять подчеркиванием обычный текст на web-страницах – он же не станет от этого гиперссылкой, а рука пользователя потянется щелкнуть мышью по нему.



документ находится в другом каталоге, то требуется указать полный путь (т.е. те же правила, что и для атрибута `src` тега `<img>`).

Например, линк `<a href="docs/title.html">Документация</a>` будет ссылаться на файл `title.html` в подкаталоге `docs` (относительно текущего).

`href="#top"` или `href="#bottom"` задает возврат на начало или конец web-страницы, что бывает полезно, если web-страница не входит целиком на экран монитора.

`href="#имя"` создает гиперссылку на часть текущего документа. Здесь имя – это помеченная область текущего документа, например отдельное слово-указатель, уникальное для данного документа.

Для того чтобы пометить эту область, используется тоже тег `<a>`, но с атрибутом **name**.

Атрибут **name** помечает находящуюся между начальным `<a>` и конечным `</a>` тегами область документа как возможный объект для ссылки. В качестве значения нужно написать любое слово-указатель, уникальное для данного документа.

Например,

`<a name="Chapter1">Глава 1</a>`.

Теперь на область «Глава 1», помеченную именем «Chapter1», можно ссылаться при помощи линка

`<a href="#Chapter1">Глава 1</a>`.

Таким способом можно организовать гипертекстовое оглавление большого текстового документа, разбитого на разделы, главы, пункты, подпункты.

Еще один атрибут тега `<a>` – **target** указывается для того, чтобы определить, в каком окне должен открываться документ, на который ведет гиперссылка. Если параметр `target` не указан, новый документ будет открыт в том же окне браузера. Если указать параметр `target` с зарезервированным именем `"blank"`, документ, на ко-

торый указывает гиперссылка, будет открыт в новом окне, что рекомендуется делать при ссылках на другие сайты. Этот атрибут используется и во фреймах, но об этом чуть позже.

Для всех ссылок на странице можно задать цвета:

**link** – определяет цвет просто ссылки;

**alink** – определяет цвет активной ссылки (нажатой);

**vlink** – определяет цвет уже посещенной ссылки.

Это – атрибуты тега `<body>`, в нем они и размещаются. Их цветовые параметры – в таблице 1, например:

```
<body link="#0000FF" alink="#6495ED" vlink=red>
```

## 4.2. Картинка-гиперссылка

В качестве гиперссылки может использоваться и рисунок. Принцип ссылки тот же, что и для текстовой гиперссылки, только между открывающим и закрывающим тегами `<a>` помещают не текст, а тег `<img>` вместе со всем его содержимым (см. п. 3).

Пример.

```
<a href="http://fdo.tusur.ru" target="blank"><img src= "logo.jpg" width=50 height=50 border=2 alt="Щелкни, чтобы перейти на сайт ФДО"></a>
```



Этим самым картинке размером 50×50 пикселей, находящейся в файле с названием logo.jpg, назначена гиперссылка на сайт факультета дистанционного обучения ТУСУР. При наведении на нее стрелка курсора трансформируется в изображение ладони с вытянутым указательным пальцем и появляется надпись «Щелкни, чтобы перейти на сайт ФДО». Страница будет открываться в новом окне браузера.

## 4.3. Навигационная карта

А можно ли назначить различным частям одного графического изображения различные гиперссылки? А как разбить картинку (например, фотографию) на отдельные участки так, чтобы при подведении к ним курсора появлялись разные подписи?

Для этого картинку надо превратить в **навигационную карту**.

Прежде чем начинать работу, поместите ту картинку, которую собираетесь превратить в карту, на html-страницу и посмотрите, удовлетворяют ли вас ее размеры. Подгоните размеры под требуемые, но не изменением параметров `height` и `width`, а с помощью графического редактора (см. п.1).

Затем вставляем в тег `<img>` картинки за атрибутом `src` атрибут `usemap="#имя"`, где **имя** – это название, которое вы даете карте. После этого начинается работа по описанию активных областей карты. Для этого используется контейнерный тег `<map>`.

Описание начинается с тега `<map name="имя">`, а заканчивается тегом `</map>`. Между ними помещается одиночный тег `<area>` описания каждой активной области изображения:

`<area shape="форма" coords="координаты" href="адрес" alt="текст">`.

Атрибут **shape** определяет ту форму, с помощью которой выделяется отдельный объект на картинке. Параметр **форма** может принимать одно из следующих значений:

**circle** – окружность,

**rect** – прямоугольник,

**poly** – многоугольник.

Атрибут **coords** определяет координаты выделенного объекта.

Параметр **координаты** задается списком координат, которые отсчитываются **в пикселях** от левого верхнего угла картинки.

Для окружности этот список состоит из координат центра окружности (x, y) и ее радиуса.

Для прямоугольника это координаты (x, y) левого верхнего и правого нижнего углов прямоугольника.

Для многоугольника это координаты (x, y) каждого из углов многоугольника.

Все координаты перечисляются через запятую.

Атрибут **href** определяет **адрес** гипертекстовой ссылки, реализуемой при нажатии на выделенный объект.

Атрибут **alt** определяет появляющийся при наведении на объект **текст**.

И тут возникает вопрос: а как рассчитать точно координаты нужной части изображения? Конечно, это можно сделать вручную, зная размер картинки в пикселях, но уж очень трудоемкий этот процесс.

На помощь приходят специальные программы – WYSIWYG-редакторы навигационных карт, которые значительно упрощают и ускоряют разметку навигационных областей<sup>1</sup>. Но сразу отметим, что сгенерированный ими HTML-код все равно может потребовать доведения его «до ума» вручную.

Укажем некоторые такие программы.

1. **GeoHTML 2.1** (<http://www.fegi.ru/geohtml/>). Относится к классу freeware. Дистрибутив имеет размер 1,2 Мб.

2. **MapEdit** (<http://www.boutell.com/mapedit/>). Относится к классу shareware. Без регистрации и оплаты она работает 30 дней или позволяет обработать 100 графических файлов.

Возможности построения навигационных карт заложены и в визуальных редакторах HTML, например Macromedia Dreamweaver 8 (<http://www.adobe.com/products/dreamweaver/>), Microsoft FrontPage 2003 (<http://www.microsoft.com/Rus/Office /FrontPage/default.msp>), предназначенных как для профессиональных Web-дизайнеров, так

---

<sup>1</sup> WYSIWYG (What You See IS What You Get) – Что видишь, то и получишь.



и для начинающих пользователей и позволяющих создавать как отдельные страницы, так и сложные Web-сайты.

Рассмотрим работу редактора карт на примере программы **GeoHTML**.

После запуска программы выбираем **File | New Map...** Окно **New Map** («Новая карта») служит для создания новой карты. Оно содержит две страницы **Open an Image** («Открыть изображение») и **Look In Document** («Найти в документе») (рис. 2.4).

Страница **Open An Image** используется для открытия файла с диска (поле **Source**). Изменяя свойства **Width** и **Height**, можно задать ширину и высоту изображения. После нажатия кнопки «Ok» изображение появляется в поле программы и готово к редактированию.

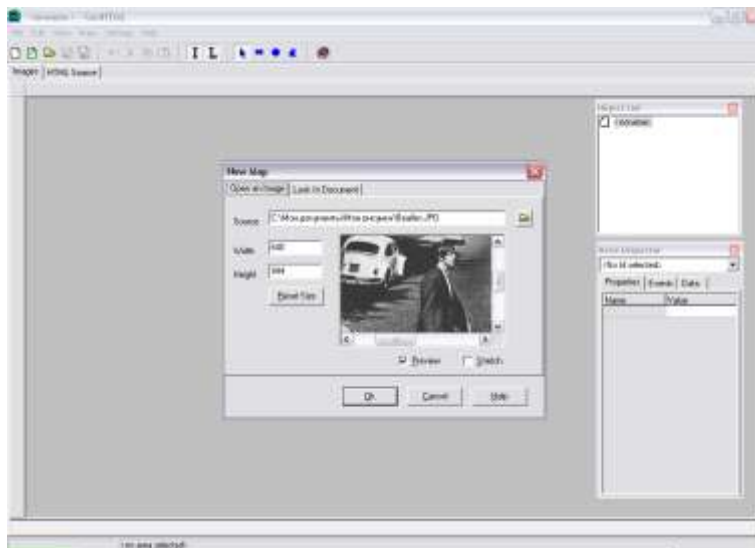


Рис. 2.4. Главное окно программы **GeoHTML** с выбранным графическим файлом.

Щелкая по изображению формы, выбираем «Прямоугольник», «Окружность» или «Многоугольник». Стрелкой осуществляется

переход от одного объекта выделения к другому. При работе с формой «Прямоугольник» ставите курсор в левый верхний угол предполагаемой области выделения и ведете его вправо-вниз. При работе с формой «Окружность» ставите курсор в центр объекта и начинаете двигать его по радиусу, увеличивая обхват. При работе с формой «Многоугольник» курсором обводите объект, щелкая мышью на углах поворота. Необходимо обязательно замкнуть ломаную линию! Объекты выделения покрываются синей сеткой (рис. 2.5).

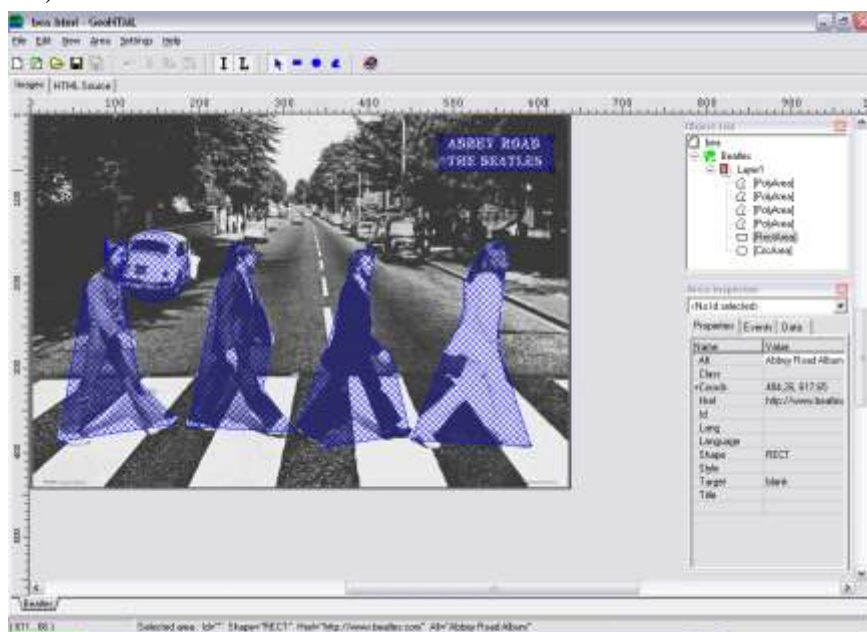


Рис. 2.5. Окно редактора GeoHTML с выделенными объектами.  
В нижнем правом углу – окно **Area Inspector**.

Чтобы отменить выделение, нужно выбрать стрелкой объект, щелкнуть правой кнопкой мыши и в контекстном меню выбрать **Delete Area**.

К выделенной области привязывается определенная информация. Это делается при помощи окна **Area Inspector**. Атрибуты разделяются на **Свойства (Properties)** и **События (Events)**.

Свойство **Shape** – тип области. Свойству автоматически присваивается одно из трех значений: RECT (область прямоугольная), CIRCLE (область круглая) или POLY (область многоугольная). **Coords** – координаты области. Содержит целочисленные значения координат, записанные через запятую, и зависит от геометрического типа (свойства Shape) области. Единицей измерения этих координат являются пиксели (точки) экрана. **Href** – гипертекстовая ссылка. **Target** – задает окно, в котором откроется документ, указанный в Href. **ID** – идентификатор области. Используется для обращения к свойствам области из кода сценария. **Alt** и **Title** – определяют выпадающий текст-подсказку (в зависимости от браузера – следует проверить!). **Language** – описание языка сценария, используемого в событиях области (например, «javascript»). **Lang** – описание ISO-языка для области (например, «text/javascript»). **Style** – определяет строку таблицы стилей для области. **Class** – определяет класс области, используемый в таблицах стилей (CSS).

Атрибуты из раздела **События** содержат операторы языков сценария (JavaScript или VBScript) и предназначены для продвинутых пользователей. Коды сценария выполняются, когда происходит соответствующее атрибуту событие.

После того, как определены все объекты, необходимо открыть закладку **HTML Source**, скопировать сгенерированный программой html-код, начиная с тега <img> и кончая тегом </map>, и вставить в нужное место на html-странице вашего сайта. При необходимости подкорректируйте путь к рисунку.

Таким образом, выбранный рисунок превратился в навигационную карту.

Пример.

Для фотографии, изображенной на рис. 2.5, был сгенерирован следующий код:

```
<IMG SRC="Мои рисунки/Beatles.jpg" USEMAP="#Beatles"
WIDTH=640 HEIGHT=444 BORDER="0" alt="Abbey Road Street">
<MAP NAME="Beatles">
  <AREA SHAPE="CIRCLE" HREF="http://www.avto.ru"
TARGET="blank" ALT="Автомобиль" COORDS="147,180,41">
  <AREA SHAPE="RECT" HREF="http://www.beatles.com"
TARGET="blank" ALT="Abbey Road Album" COORDS="484,26,
617,65">
  <AREA SHAPE="POLY" HREF="#" ALT="Джон Леннон"
COORDS="538,157, 514,198, 513,197, 511,240, 489,284, 450,385,
592,392, 552,271, 564,155, 545,150, 532,163, 538,157">
  <AREA SHAPE="POLY" HREF="#" ALT="Ринго Старр"
COORDS="387,160, 374,196, 345,292, 315,369, 350,394, 449,377,
427,284, 401,259, 400,196, 406,180, 400,156, 386,163, 387,160">
  <AREA SHAPE="POLY" HREF="#" ALT="Пол Маккартни "
COORDS="240,152, 218,213, 187,369, 315,376, 275,325, 285,262,
259,194, 266,165, 252,151, 238,157, 240,152">
  <AREA SHAPE="POLY" HREF="#" ALT="Джордж
Харрисон" COORDS="90,148, 89,188, 64,201, 64,201, 34,387,
176,365, 129,316, 110,214, 108,178, 113,152, 88,149, 90,169, 86,171,
90,148">
</MAP>
```

Видно, что были сформированы шесть областей выделения (автомобиль, надпись в правом верхнем углу и четыре фигуры), причем первым двум областям назначены гипертекстовые ссылки. Первая область сформирована окружностью, вторая – прямоугольником, остальные – многоугольником.

Представляете, как было бы нелегко это исполнить вручную!



Отметим, тем не менее, что подпись, которая должна появиться при наведении курсора вне зон выделения (Abbey Road Street), была задана в атрибуте alt тега `<img>` вручную.

## 5. ТАБЛИЦЫ

Таблицы, пожалуй, самый распространенный элемент web-дизайна.

При создании html-страниц таблицы используются не только для традиционных целей – табличного размещения текстовой и числовой информации, но и для компоновки элементов дизайна страницы – текста, графики. Например, в таблицу с невидимыми границами размещаются необходимые элементы: навигация, текст, картинки, таблицы и т.д. Пользователь может даже не догадаться, что на самом деле это таблица.

Сегодня практически любая Web-страница создана именно таким образом.

Для создания таблицы используется контейнерный тег `<table>`. Внутри тегов `<table>...<table>` располагается описание всех рядов и ячеек.

Для того чтобы границы таблицы были видимы, в теге `<table>` задают атрибут **`border=n`**, где *n* – ширина границы в пикселях.

Таблицы в HTML формируются **построчно**.

Сначала – первый ряд со всем его содержимым, затем второй и т. д.

Каждый ряд формирует контейнерный тег `<tr>`. Тег `<tr>` начинает ряд в таблице, тег `</tr>` завершает ряд.

Внутри описываются все ячейки ряда. Для формирования ячеек первой строки (она обычно является строкой заголовков столбцов) используется контейнерный тег `<th>`. Внутри этих ячеек текст по умолчанию отцентрирован и выделен полужирным шрифтом.

Ячейки всех остальных строк формируются тегами `<td>...</td>`. По умолчанию текст в ячейке смещен влево и отцентрирован по вертикали. Ячейки таблицы обязательно должны содержать какие-либо символы. Если хотите оставить пустой, поместите туда неразрывный пробел `&nbsp;`.

Для формирования подписи к таблице используется контейнерный тег `<caption>`. Он должен присутствовать внутри тегов `<table>...</table>`, но снаружи описания какой-либо строки или ячейки. По умолчанию `<caption>` имеет атрибут **align=top**, но может быть явно установлен в **align=bottom**. Атрибут align определяет, где – сверху или снизу таблицы – будет поставлена подпись. Подпись всегда центрирована в рамках ширины таблицы.

Пример.

Строки html-кода

```
<table border="1">
  <caption>Таблица </caption>
  <tr>
    <th>Заголовок столбца 1</th><th>Заголовок столбца 2</th>
    <th>Заголовок столбца 3 </th>
  </tr>
  <tr>
    <td>Содержимое ячейки 4</td><td>Содержимое ячейки
5</td>
    <td>Содержимое ячейки 6</td>
  </tr>
  <tr>
    <td>Содержимое ячейки 7</td><td>Содержимое ячейки
8</td>
    <td>Содержимое ячейки 9</td>
  </tr>
```

&lt;/table&gt;

определяют простую таблицу следующего вида

Таблица		
Заголовок столбца 1	Заголовок столбца 2	Заголовок столбца 3
Содержимое ячейки 4	Содержимое ячейки 5	Содержимое ячейки 6
Содержимое ячейки 7	Содержимое ячейки 8	Содержимое ячейки 8

Рассмотрим основные атрибуты табличных тегов.

Атрибут **align**. Если он используется внутри тегов <tr>, <th> или <td>, то управляет положением данных в ячейках по горизонтали, если используется внутри тега <table> – управляет размещением таблицы на странице (аналогично изображению <img>).

Может принимать значения **left** (слева), **right** (справа) или **center** (по центру).

Атрибут **valign**. Он используется внутри тегов <tr>, <th> и <td> и определяет вертикальное размещение данных в ячейках. Может принимать значения **top** (вверху), **bottom** (внизу), **middle** (по середине).

Атрибут **nowrap** говорит о том, что данные в ячейке не могут логически разбиваться на несколько строк и должны быть представлены одной строкой.

Атрибут **colspan** указывает, какое количество ячеек будет объединено **по горизонтали** для указанной ячейки. По умолчанию равен 1.

Атрибут **rowspan** указывает, какое количество ячеек будет объединено **по вертикали** для указанной ячейки. По умолчанию равен 1.

Умелое управление атрибутами colspan и rowspan позволяет создавать сколь угодно сложные таблицы.

Рассмотрим на примере алгоритм их создания.

Запишем html-код, описывающий таблицу вида:


Продлим пунктиром все линии этой таблицы до ее краёв. Получим таблицу из четырех строк и шести столбцов.


Теперь начинаем писать код. Сначала откроем таблицу и укажем толщину рамки:

```
<table border=1>
```

Каждая горизонтальная черта разделяет строки таблицы. Ячейка, обведенная сплошными линиями, – это одна пара `<td>...</td>`. Количество столбцов в такой ячейке записываем в `colspan`, а количество строк – в `rowspan`.

**Если верхний левый угол ячейки находится не в рассматриваемой строке, то при записи кода эту ячейку игнорируем.**

Открываем первую строку:

```
<tr>
```

Первая ячейка не делится ни на столбцы, ни на строки. Значит, так и запишем:

```
<td>Ячейка 1</td>.
```

Вторая ячейка делится на 4 столбца и 2 строки, поэтому записываем:

```
<td colspan=4 rowspan=2>Ячейка 2</td>.
```

```
</tr>
```

Строка закрывается, т.к. в ней ячейки закончились.

Переходим ко второй строке:

```
<tr>
```

Первая её ячейка делится на три строки.

```
<td rowspan=3>Ячейка 3</td>
```

Вторую ячейку мы уже записали, так что ее игнорируем (ее верхний левый угол находится в предыдущей строке).

Закрываем строку:

```
</tr>.
```

Переходим к третьей строке:

```
<tr>.
```

Первую ячейку игнорируем, вторая и третья делятся на две колонки. Пишем:

```
<td colspan=2>Ячейка 4</td>
```

```
<td colspan=2>Ячейка 5</td>
```

```
</tr>
```

Переходим к четвертой строке:

```
<tr>.
```

Первую ячейку игнорируем, вторая не делится, третья делится на две колонки, четвертая не делится. Пишем:

```
<td>Ячейка 6</td>
```

```
<td colspan=2>Ячейка 7</td>
```

```
<td>Ячейка 8</td>
```

```
</tr>
```

</table>

Запишем весь код полностью, добавив центрирование таблицы и центрирование текста в ячейках:

```
<table align=center border=1>
  <tr align=center >
    <td>Ячейка 1</td>
    <td colspan=4 rowspan=2>Ячейка 2</td>
  </tr>
  <tr align=center >
    <td rowspan=3>Ячейка 3</td>
  </tr>
  <tr align=center >
    <td colspan=2>Ячейка 4</td>
    <td colspan=2>Ячейка 5</td>
  </tr>
  <tr align=center >
    <td>Ячейка 6</td>
    <td colspan=2>Ячейка 7</td>
    <td>Ячейка 8</td>
  </tr>
</table>
```

Внешний вид таблицы кроме атрибута border определяется еще несколькими атрибутами.

Атрибут тега <table> **cellspacing** задает расстояние  $n$  между ячейками (в пикселях) в таблице:

```
<table cellspacing= $n$ >.
```

Атрибут тега <table> **cellpadding** задает расстояние  $n$  между границей (рамкой) ячейки и ее содержимым (в пикселях). Этот атрибут определяет данное расстояние для всех ячеек в таблице:

```
<table cellpadding=  $n$  >.
```

В теге `<table>` атрибут **width** задает ширину таблицы  $n$ , **height** – ее высоту  $m$ :

```
<table width= $n$  height= $m$ >.
```

Если размер таблицы не задан, то он будет минимально возможным для того, чтобы уместить ее содержимое. Если текста много, то когда ширина таблицы достигнет границ программы просмотра, браузер перенесет текст на следующую строку.

В теге `<td>` атрибут **width** задает ширину ячейки  $n$ , **height** – высоту  $m$ :

```
<td width= $n$  height= $m$ >
```

Уже известные нам атрибуты **bgcolor** и **background** определяют цвет фона или фоновый рисунок для всех ячеек таблицы:

```
<table background="путь к картинке">;
```

для строки таблицы:

```
<tr bgcolor=цвет>;
```

для ячейки таблицы:

```
<td bgcolor=цвет>;
```

причем последующий цвет перекрывает предыдущие.

Цвета могут быть заданы шестнадцатеричными RGB-значениями или стандартными названиями (приложение 2).

## 6. ФРЕЙМЫ

### 6.1. Страница-контейнер

Английское слово *frame* («фрейм») означает рамку, остов, оправу, кадр, словом, структурную единицу различных объектов.

В информатике фрейм – элемент языка HTML версии 3.0 и выше. Фреймы позволяют разделить web-страницу на несколько независимых окон и в каждом из них размещать отдельную web-страницу. При этом допускаются ссылки из одного окна в другое окно. Обычно фреймы применяются для организации меню, посто-

янно находящихся на экране, тем самым связывая отдельные страницы в web-сайт. Чем мы сейчас и займемся.

Любой сайт, содержащий страницы с фреймами, начинается с написания специальной **странички-контейнера**, которая сама не показывается, но содержит в себе указания для организации фреймовой структуры (на сколько фреймов делить страницу и как) и ссылок на участвующие файлы.

Принципиальная особенность этой странички-контейнера – она не содержит тегов `<body>...</body>`. Их заменяют теги **`<frameset>... </frameset>`**.

Рассмотрим пример.

Наберем в текстовом редакторе Notepad++ html-код и сохраним его, как это описано в п.1, с именем **index.html**:

```
<html>
  <head>
    <title>Frame container</title>
  </head>
  <frameset rows="10%,*">
    <frame src="up.html" name="logo" scrolling=no noresize>
  </frameset cols="15%,*">
    <frame src="left.html" name="menu" scrolling=no noresize>
    <frame src="right.html" name="content" noresize>
  </frameset>
</frameset>
</html>
```

Рассмотрим действие новых тегов и их атрибутов.

Контейнерный тег **`<frameset>... </frameset>`** описывает количество и размеры фреймов.



Атрибут **rows** задает горизонтальное разбиение страницы. Его параметры – это цифры и знаки, разделенные запятыми. Каждое значение определяет ширину области и может задаваться в процентах, пикселях и при помощи знака звездочки, обозначающей «все оставшееся пространство». Аналогично, параметр **cols** задает вертикальное разбиение страницы.

Таким образом, в примере вся область сначала разделена тегом `<frameset rows="10%,*">` на два фрейма по горизонтали (верхняя часть составляет 10% высоты области), а затем нижний фрейм разделен, в свою очередь, тегом `<frameset cols="15%,*">` по вертикали также на два фрейма, причем левая часть составляет 15% ширины области. Таким образом, создана трехфреймовая структура страницы.

Кроме указанных атрибутов в теге `<frameset>` используются атрибуты **frameborder=yes (no)** – указывает браузеру, отображать ли рамку у фреймов или нет; **border=*n*** – определяет ширину *n* рамки между фреймами в пикселях; **bordercolor="#RRGGBB"** – задает цвет рамки, которая разделяет отдельные фреймы.

Контейнерный тег `<frame>...</frame>` определяет источник информации для соответствующего фрейма.

Атрибут **src** задает URL страницы, которая будет помещена во фрейм. Обычно эти страницы и страница-контейнер находятся все вместе в одной папке, поэтому адрес состоит только из имени файла и его расширения.

В примере верхнему фрейму назначен файл **up.html**, нижнему левому фрейму – **left.html**, нижнему правому фрейму – **right.html**. Именно их содержимое появится на экране при запуске файла **index.html**.

Атрибут **name** позволяет задать каждому фрейму уникальное имя, по которому к нему можно будет обращаться. Ниже это будет использовано при создании гипертекстового меню.

Параметры атрибута **scrolling** позволяют управлять появлением полосы прокрутки: **no** – полосы прокрутки не будет ни при каких обстоятельствах; **yes** – полоса прокрутки будет всегда; **auto** – полоса прокрутки появится только тогда, когда она нужна (значение по умолчанию).

Атрибут **noresize** позволяет создавать фреймы без возможности изменения размеров. По умолчанию размер фрейма можно изменить при помощи мыши так же просто, как и размер окна Windows. Данную возможность **noresize** отменяет. Если у одного фрейма установлен атрибут **noresize**, то у соседних фреймов тоже не может быть изменен размер со стороны данного.

Итак, теперь если вы откроете файл `index.html`, то перед вами появится трехфреймовая страница примерно такого вида (рис. 2.6) .

Такой вид можно наблюдать, если у вас нет файлов с именами `up.html`, `left.html`, `right.html` или вы ошиблись в их именах. Введя в код страницы-контейнера имена `html`-файлов, созданных вами в процессе работы над заданием, вы получите их на экране монитора в соответствующем фрейме.

**Внимание!** Чтобы открыть страницу-контейнер на изменение `html`-кода, выберите в главном меню браузера опцию **Вид → Просмотр HTML кода**. Обновление – как обычно.

Но для реализации поставленной цели – организовать оглавление сайта, постоянно находящееся на экране, – еще необходимо создать `html`-страницу с гипертекстовым меню.

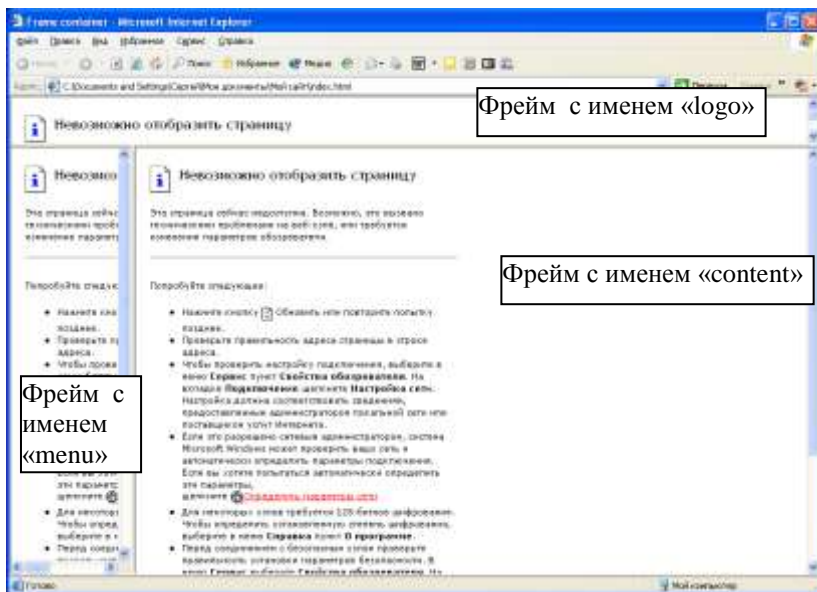


Рис. 2.6. Структура страницы с тремя фреймами

## 6.2. Страница меню

Пусть созданные вами html-страницы имеют имена 1.html, 2.html, 3.html и title.html.

Создадим простейший файл menu.html со страничкой, содержащей меню<sup>1</sup>.

```
<html>
<head>
<title>Menu Page</title>
</head>
<body>
<a href="1.html" target="content">Страница 1</a><br>
```

<sup>1</sup> Затем, естественно, вы ее «облагородите»...

```
<a href="2.html" target="content">Страница 2</a><br>
<a href="3.html" target="content">Страница 3</a><br>
</body>
</html>
```

Действие тегов, входящих в код, вы уже знаете.

Первый тег `<a>` атрибутом **href="1.html"** определяет гиперссылку на html-страницу с именем 1.html. Атрибут **target="content"** определяет, в каком фрейме открывать эту страницу. Ранее, на странице-контейнере index.html это имя было назначено нижнему правому фрейму. Значит, там и откроется html-страница с именем 1.html.

Отметим, что атрибут target может принимать и другие значения:

`target="_blank"` определяет, что документ, полученный по ссылке, будет отображаться в новом окне браузера;

`target="_self"` определяет, что документ, полученный по ссылке, будет отображаться в том же фрейме, в котором находится ссылка;

`target="_top"` определяет, что документ, полученный по ссылке, будет отображаться на всей поверхности окна вне зависимости от наличия фреймов. Использование данного параметра удобно в случае вложенных фреймов.

Второй тег `<a>` атрибутом **href="2.html"** определяет гиперссылку на html-страницу с именем 2.html, которая тоже откроется в нижнем правом фрейме. И так далее.

Кстати, не забудьте сразу подкорректировать код странички-контейнера! В соответствии с назначенными выше именами файлов ее основное содержимое будет выглядеть так:

```
<frameset rows="10%,*">
  <frame src="title.html" name="logo" scrolling=no noresize>
</frameset cols="15%,*">
  <frame src="menu.html" name="menu" scrolling=no noresize>
```

```
<frame src="1.html" name="content" noresize>
```

Таким образом, при запуске файла index.html в верхнем фрейме окажется содержимое файла title.html, в левом фрейме появятся подчеркнутые строчки-гиперссылки

Страница 1

Страница 2

Страница 3.

а в правом фрейме появится содержимое файла 1.html.

Файл index.html является стартовым файлом созданного web-сайта – так и принято по умолчанию в Интернет.

### 6.3. Плавающие фреймы

Контейнерный тег **<iframe>** позволяет создавать так называемые плавающие фреймы, находящиеся внутри обычного документа. В них разрешается встраивать любые другие независимые документы. В теге **<iframe>** поддерживаются те же параметры, что и в обычном фрейме. Кроме того, можно использовать следующие параметры: **width**, **height**, **hspace**, **vspace**, **align**, значение которых совпадает с соответствующими параметрами тега **<img>**.

Пример (рис.2.7).

```
<iframe src=vstavka.html width=500 height=200 hspace=10  
vspace=10 align=center> </iframe>
```

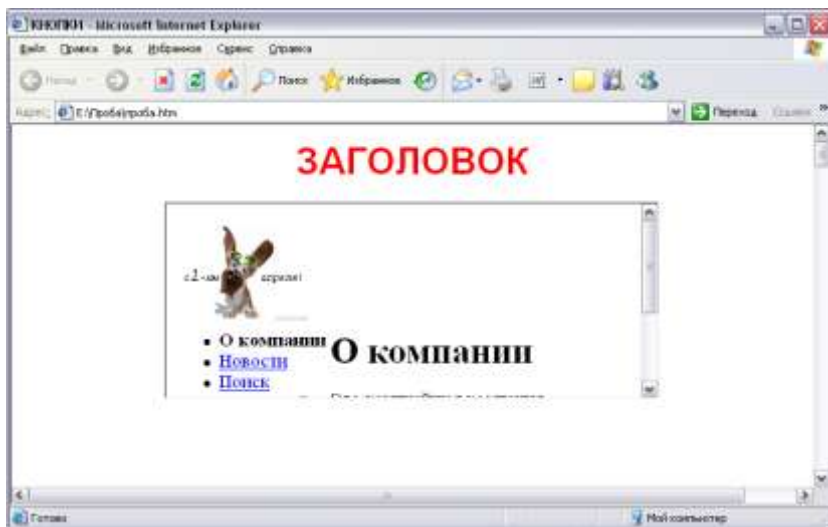


Рис. 2.7. Структура страницы с плавающим фреймом

#### 6.4. Фреймы: pro и contra

Использование фреймов заметно упрощает навигацию по сайту. На самом деле, очень удобно – на одной части страницы размещаешь меню, на другой – выкладываешь содержимое.

Но в последнее время использование фреймов становится все менее популярным решением. В качестве довода приводят неправильную обработку страниц, содержащих фреймы, некоторыми поисковыми машинами. Когда пользователь находит такой документ через поисковый сервер, то он попадает лишь на один из фреймов, но не видит истинного отображения страницы, что вызывает его полную дезориентацию. Да и сами попробуйте начать просмотр вашего сайта не со страницы с фреймами, а, например, со страницы 1.html. Ну, и где она, красивая фреймовая структура?

Эту проблему, конечно, можно обойти, поместив на каждой странице ссылку на страницу с фреймами. Но можно поступить по-

другому: использовать **таблицы** для произвольного расположения элементов на странице и организации навигации по сайту.

Ведь как можно интерпретировать созданную выше трехфреймовую структуру в табличных терминах? Как таблицу из двух строк и двух столбцов, причем две верхних ячейки объединены. Создадим на каждой странице такую структуру в виде таблицы, и фреймы будут не нужны. Правда, будет много повторов (меню, заголовочная часть)...

В качестве информации к размышлению приведем слова Артемия Лебедева, авторитетного в РУНЕТе web-дизайнера ([www.artlebedev.ru](http://www.artlebedev.ru)): «Многие начинающие веб-дизайнеры страшно боятся использовать фреймы, потому что они слышали, что это не хорошо. И не задумываются над вопросом: «А почему, собственно?». Решение простое – если вам надо, используйте. Но помните одну вещь – если сайт большой, то для организации всей информации фреймы лучше не использовать. Они пригодятся для решения небольших конкретных задач интерфейсного характера».

## 7. ТАБЛИЦЫ СТИЛЕЙ

Многие рассмотренные выше теги имели атрибуты, с помощью которых можно было менять стили отображаемых объектов.

**Стиль документа** – это правило, которое определяет внешний вид того или иного элемента в документе: вид шрифта, цвет и размер объектов (символов, границ), формат текста и т.п.

Когда в составе сайта немного страниц, то эти html-возможности вполне устраивают разработчика. А если страниц много или очень много? Представьте себе, что, задав на всех страницах один и тот же фон или цвет ячеек таблиц, вы впоследствии решите их изменить. Очень долгая и нудная работа.

С появлением каскадных таблиц стилей (CSS – Cascading Style Sheets) этот процесс стал намного проще.

Идея CSS очень проста. В html-документе прямо ставится указание на то, как должен выглядеть тот или иной элемент, а при использовании CSS такие указания выносятся в отдельный блок, который может либо включаться в документ, либо читаться из внешнего файла.

Такой простой подход сразу дает массу преимуществ. Прежде всего, значительно облегчается изменение внешнего вида сайта или отдельных его элементов – достаточно изменить определение соответствующего стиля в единственном CSS-файле, и эти изменения распространятся на весь сайт. Второе преимущество – сокращение размеров документов. Ну, и третье – таблицы стилей позволяют отделить логическую структуру страниц от визуального представления (оформления). Ведь язык HTML изначально был разработан для разметки структуры документа.

Более того, в новой спецификации HTML 4.0 было рекомендовано использовать для оформления именно каскадные таблицы стилей. При этом многие теги и атрибуты (собственно говоря, почти все, относящиеся к оформлению) были отменены. Это, конечно, не означает, что всем нужно немедленно прекратить их использование, они будут еще долго поддерживаться, но к рекомендациям WWW-консорциума следует прислушаться (<http://www.w3.org/TR/REC-CSS1/>).

Существует **три способа добавления CSS в html-документ**:

- 1) подключение внешней стилевой таблицы;
- 2) использование внутренней стилевой таблицы, действующей только в данном документе;
- 3) указание стиля непосредственно внутри тега.

## 7.1. Внешняя таблица стилей

Внешняя стилевая таблица создается с помощью текстового ре-



дактора, например того же Notepad++. Только нужно сохранить этот файл с расширением .css (например, style.css).

В этом файле объявляются правила стилей для разных объектов. Они записываются с самой первой строки блокнота, ничем особым не выделяясь.

Например,

```
H1 {color:blue}
```

Это правило говорит о том, что все заголовки первого уровня (выделяемые тегами `<h1>... </h1>`) будут иметь синий цвет<sup>1</sup>.

Каждое правило состоит из двух частей:

- **селектор** (здесь: **H1**) определяет тег в html-документе (заголовков, параграф, таблица и ее элементы, ссылки, и т. д.), для которого назначается стиль;
- **определение** (здесь: **color:blue**) устанавливает стиль, применяемый к данному селектору (тегу).

Определение, в свою очередь, тоже состоит из двух частей: свойства (color) и значения (blue). Они записываются в фигурных скобках и отделяются друг от друга двоеточием. Определения перечисляются через точку с запятой. Селекторы тоже могут перечисляться, отделяясь запятой.

Например:

```
H1, H2 {color:red; font-family:Arial}
```

В данном примере задается одинаковое оформление для заголовков первого и второго уровней. Они будут иметь красный цвет и шрифт Arial.

Некоторые свойства и их значения приведены в таблицах 3, 4, 5.

В таблицах стилей также могут применяться **контекстные селекторы**. Они позволяют задать стилевое оформление для тега, который находится внутри другого тега.

Пусть в CSS заданы стили

H4 {color:grey}

H4 I {color:green} .

Два подряд (через пробел) записанных селектора H4 и I и есть контекстный селектор.

Тогда в строке

<h4> Наследование <i> родительских </i> свойств </h4>

текст, заключенный между тегами <i>...</i>, будет отображаться зеленым цветом, в то время как остальной текст между тегами <h4>...</h4> будет серого цвета. И это будет выполняться для любых участков текста, где встретится такая комбинация.

Таблица 3. Параметры шрифта

Свойство	Описание	Значения	Примеры и примечания
font-family	Задаёт шрифт(ы) или семейство шрифтов	Названия шрифтов в порядке приоритетности. Перечисляются через запятую.	BODY { font-family: Verdana, Arial, sans-serif} В данном примере страница будет оформляться шрифтом Verdana. Если этого шрифта на машине пользователя нет, то браузер будет искать Arial, а если нет и его, то используется любой

<sup>1</sup> В CSS принято названия тегов писать прописными буквами.

			шрифт из семейства sans-serif
font-style	Задаёт способ начертания шрифта	normal, italic, oblique	H1 { font-family: Helvetica; font-style: italic; font-weight: bold }
font-weight	Задаёт толщину шрифта	bold (400) normal (700) Числовые значения от 100 до 900, с шагом, кратным 100. Можно употреблять относительные значения bolder и lighter	
font-size	Задаёт размер шрифта	Точный размер: px – пиксели, pt – пункты (1pt = 1/72 in), in – дюймы, mm – миллиметры. Относительное изменение: larger, smaller	H1 { font-weight: bold; font-size: 30pt }

Таблица 4. Параметры текста

Свойство	Описание	Значения	Примеры и примечания
text-decoration	Задаёт параметры начертания: подчёркивание, надчёркивание, зачёркивание	none, underline, overline, line-through	A:visited { text-decoration: none }
text-align	Задаёт горизонтальное выравнивание текста	left, right, center, justify	P {text-align: justify}
text-indent	Задаёт абзацный отступ	px – пиксели, pt – пункты, mm – миллиметры, em – символ максимальной ширины	P {text-align: justify; text-indent: 5em}
vertical-align	Задаёт отображение текста по вертикали; верхние ( $2^{10}$ ) и нижние ( $H_2O$ ) индексы	top, middle, bottom, super, sub,	TD {vertical-align: top} <P>H<SPAN STYLE="vertical-align: sub;">2</SPAN>O</P> В браузере – $H_2O$

Итак, наберем в блокноте строки:

BODY {font-family: Verdana}

H1, H2 {color:red; font-family:Arial}

P {text-align: justify; text-indent: 3em}

TD {background-color: yellow}  
и сохраним под именем style.css. Таблица стилей создана. Как теперь ее применить?

Таблица 5. Параметры цвета и фона

Свойство	Описание	Значения	Примеры
color	Задаёт цвет текста	Название цвета – red; шестнадцатеричное значение составляющих цвета – #FF0000	H1 {color:red} A:link {color:#8B4513}
background-color	Задаёт цвет фона	TD { background-color: #F5F5DC }	
background-image	Задаёт фоновое изображение	BODY { background-image: URL (picture /back.gif) }	
background-repeat	Задаёт способ повторения фонового изображения	repeat – повторять по вертикали и горизонтали. no-repeat – не повторять. repeat-x – повторять по горизонтали. repeat-y – повторять по вертикали	BODY { background-image: URL (picture/back.gif); background-repeat: repeat-x }

## 7.2. Применение внешней таблицы стилей

Чтобы применить созданную таблицу стилей, вставляем в заголовочную часть **каждого html-документа**, в котором хотим при-

менять стили (т.е. внутрь тегов `<head>...</head>`), одиночный тег **<link>** с соответствующими атрибутами и значениями:

```
<link href="style.css" rel="stylesheet" type="text/css">
```

Атрибут **href="style.css"** указывает путь к файлу со стилями.

Атрибут **rel="stylesheet"** указывает, является ли подгружаемые таблицы стилей постоянными, предпочитаемыми или альтернативными. В данном случае установлены постоянные таблицы стилей для документа (подробнее об атрибуте rel и его значениях – в спецификации HTML 4.0 (<http://www.citforum.ru/internet/html40/>)).

Атрибут **type="text/css"** – тип подгружаемой информации, в данном случае указывается, что это CSS.

Теперь браузер отобразит текст шрифтом **Verdana**, заголовки первого и второго уровня – шрифтом **Arial** красного цвета, текст параграфов (тот, который охвачен тегами `<p>...</p>`) будет выровнен по ширине страницы, а абзацный отступ будет длиной в 3 символа. Фон ячеек таблиц будет желтым.

Но описанной выше стилевой таблице не хватает гибкости. Как быть, например, если хочется один параграф выполнить в одном стиле, а другой – в другом? Для большей гибкости при оформлении страниц можно создать **свои собственные классы**, дать им имена и задать для них стили.

Классы задаются двумя способами.

**Способ 1.** Привязка класса к определенному тегу.

Например, в CSS записывается:

```
P.oba {text-align: justify; font-style: italic}
```

```
P.sprava {text-align: right; color:red }
```

Это означает, что задано два класса стилей, которые можно применять только к параграфам.

Здесь **oba** и **sprava** — названия классов. Названия можно давать любые, но лучше осмысленные. Обратите внимание на синтаксис

(между названием тега и названием класса ставится точка).

Чтобы теперь использовать эти классы, нужно добавить атрибут **class** в начало параграфа с соответствующим значением, например

```
<p class="oba">Параграф оформлен с помощью класса oba</p>.
```

Или

```
<p class="sprava">Параграф оформлен с помощью класса sprava</p>.
```

**Способ 2.** Создание произвольного класса, который можно присваивать любому тегу на странице.

В CSS помещают стиль без упоминания тега:

```
.cvet1 {color: #EE82EE}
```

На html-странице помещают атрибут class:

```
<h6 class="cvet1">Цвет заголовка задан с помощью класса .cvet1</h6>
```

или

```
<body class="cvet1">Цвет текста всей страницы задан с помощью класса .cvet1</body>
```

При работе с классами сначала рекомендуется определить стиль по умолчанию для всего документа. Этот стиль по умолчанию задается для тега `<body>`. А уже потом по мере необходимости добавлять стили для разных тегов.

Из примеров можно понять, что атрибут class нужно «прикрепить» к какому-то тегу. Часто он используется с тегами параграфа `<p>....</p>`. Но есть еще один подобный контейнерный тег – `<div>`:

```
<div class="oba">Абзац</div>
```

Это тег-разделитель, он определяет фрагмент текста, который можно наделить какими-либо стилевыми свойствами, расположив его между тегами `<div>` и `</div>` и задав стиль. Новый тег `<div>`,

так же как и тег `<p>`, начинает новый абзац, но в **отличие от тега `</p>` после тега `</div>` пропуск строки не производится**, поэтому текст выглядит «как обычно».

Но возникает вопрос, а как применить стиль к отдельному слову (группе слов) в тексте документа? Тег `<div>...</div>` не используешь, поскольку в этом случае следующий после выделения текст начнется на новой строке.

В таких ситуациях вместо тега `<div>` следует использовать контейнерный тег **`<span>`** :

```
<span class="cvet1">выделено три слова</span>
```

Этот тег, кстати, бессмысленно применять без атрибутов, поскольку тогда вообще не будет никакого эффекта. Как правило, он применяется с атрибутами `class`, `style` (о последнем речь пойдет ниже).

Кстати, кроме стилевых свойств элементов HTML и классов, можно определять свойства так называемых **идентификаторов**. Дело в том, что у каждого элемента HTML может быть атрибут **`id`** – его уникальное имя в данном документе (этот атрибут играет большую роль при создании динамических страниц). Для того чтобы определить в таблице стилей свойства элемента, имеющего определенный идентификатор `id`, следует записать, например, так:

```
#cool {color: white; background-color: black}
```

Тогда элемент по имени **`cool`** (например `<div id="cool">`) будет наделен указанными свойствами (в данном случае белым цветом символов на черном фоне). Только не забывайте, что в каждом HTML-документе каждый идентификатор должен быть уникальным, то есть не может быть двух и более элементов с одинаковым значением атрибута `id`.

### 7.3. Внутренняя таблица стилей



Внутренняя таблица стилей внедряется непосредственно в html-документ при помощи тега **<style>**, в котором размещается код таблиц стилей (**<style>**код таблицы стилей**</style>**), и действует только внутри одного документа.

Обратите внимание, тег **<style>**, как и тег **<link>**, размещается внутри тега **<head>...</head>**, а не **<body>**!

Пример.

```
<head>
```

```
...
```

```
<style type="text/css">
```

```
BODY {background-color: #BABAAO; color: blue}
```

```
P.oba {text-align: justify; font-style: italic}
```

```
P.sprava {text-align: right; color:red}
```

```
H1, H2 {color:red; font-family:Arial}
```

```
</style>
```

```
...
```

```
</head>
```

Атрибут `type="text/css"` здесь имеет тот же смысл, что и в теге **<link>**, а правила записи стилей – те же, что были рассмотрены выше для внешних таблиц.

Еще один пример.

Как оформить в таблицах стилей разные цветовые **настройки для гиперссылок**? Гиперссылка может быть в одном из четырех состояний: непосещенная; посещенная; активная (в момент нажатия на нее); реагирующая (при наведении на нее курсора мыши).

В стилевых таблицах ссылки оформляются с помощью псевдоклассов селектора **A**:

```
<style type="text/css">
```

```
A:LINK {color: #8B4513} /* непосещенная ссылка*/ ;  
A:VISITED {color: #2E8B57} /* посещенная ссылка*/ ;  
A:ACTIVE {color: #F4A460} /* активная ссылка*/ ;  
A:HOVER {color: Olive; text-decoration: none} /* в момент наведе-  
дения*/ .  
</style>
```

Аналогично записывается и для внешней стилевой таблицы.

Порядок следования стилей должен быть именно таким, иначе работать не будет. Пробелы между селектором и названием псевдокласса не допускаются.

Заодно здесь показано, как в таблицах стилей оформляются комментарии: они охватываются символами «слэш-звездочка» и «звездочка-слэш».

Обратите внимание на свойство `text-decoration: none` – при наведении мыши текст будет неподчеркнутым (по умолчанию для ссылок указано свойство `text-decoration: underline`).

Если нужно в одном документе задать разное оформление для разных ссылок, то используются классы, которые комбинируются с псевдоклассами, например:

```
A.menu:LINK {свойство:значение; ...}  
A.menu:VISITED {свойство:значение; ...}
```

В этом примере создан класс `menu`.

Это имя затем указывается в атрибуте `class` тега `<a>` на html-странице:

```
<a href="путь" class=menu>текст ссылки</a>
```

#### 7.4. Указание стиля внутри тега

Третий способ внедрения CSS в html-документ заключается в указании стиля непосредственно внутри тега при помощи атрибута

**style** (не путать с тегом!), например:

```
<h1 style="color:red; font-family:Arial">...</h1> .
```

Сразу отметим, что существует иерархия стилей: стиль, заданный во внутренней таблице, отменяет стиль, заданный во внешней таблице, а стиль, заданный внутри тега, отменяет табличные стили. Поэтому они и получили название «каскадные таблицы стилей».

Этот способ используется мало, так как теряет все достоинства таблиц стилей. Ведь применяется он только к одному конкретному тегу.

В заключение отметим, что освоить азы технологии создания web-страниц и web-сайтов можно, просматривая HTML-код любых сайтов (строка **Просмотр HTML-кода** в контекстно-зависимом меню, открываемом при щелчке правой кнопкой мыши в окне браузера) и разбираясь, как работает та или иная строчка кода. В Интернет также можно найти массу электронных учебников как для начинающих пользователей, так и руководств для продвинутых «юзеров», см., например, <http://www.htmlbook.ru/>, <http://postroika.ru/>, <http://htmlbook.city.tomsk.net/>, <http://html.manual.ru/> и многие другие.

## 8. ЗАДАНИЕ

Используя описанные выше теги HTML, создать при помощи текстового редактора Notepad++ тематический Web-сайт по вопросам дисциплины «Мировые информационные ресурсы». Перечень тем приведен в Приложении 3. Информацию по выбранной теме найти в Интернет (закрепив заодно полученные в лабораторной работе 1 навыки поиска Интернет-информации).

### **Требования к содержанию сайта:**

1) на главной странице поместить название работы, информацию об авторе (ФИО, город, учебная группа, вуз), информацию о руководителе работы (ФИО, должность), название дисциплины, по которой выполнена работа;

2) информацию по теме поместить на разных страницах сайта в соответствии со структурой материала;

3) информационное наполнение сайта (контент) должно достаточно полно раскрывать тему, в то же время не следует перегружать сайт детальными подробностями – лучше сделайте соответствующую ссылку;

4) упоминаемые в тексте названия организаций, учреждений, предприятий дать гиперссылками на соответствующие сайты; которые должны открываться в новом окне браузера;

5) на отдельной странице привести гиперссылки на электронные ресурсы, которые были использованы при выполнении работы (сайт с указанием конкретного ресурса).

### **Требования к оформлению сайта:**

1) организовать систему навигации по страницам сайта (гипертекстовое меню переходов на тематические разделы (страницы) сайта, гипертекстовые переходы со страницы на страницу («линейная навигация»), гипертекстовое оглавление отдельных страниц с

большим по объему текстом, гипертекстовые переходы на начало страницы);

2) использовать фреймы для создания трехколонной структуры страницы, например:

Здесь «шапка» страницы	
Здесь меню	Здесь основная часть

Меню выполнить в виде гипертекстового перечня названий страниц сайта;

3) при оформлении «шапки» страницы использовать бегущую строку;

4) использовать однородную цветную заливку страницы, а также использовать картинку в качестве фона страницы;

5) при оформлении текста на страницах:

- a) использовать разный шрифт, цвет, размер, нижний и верхний индексы, выделение курсивом, полужирным, выравнивание слева, справа, по центру;
  - b) большой по объему текст разбить на пункты, подпункты, абзацы;
  - c) вставить списки (нумерованные и нenumерованные);
  - d) использовать спецсимволы, например `&nbsp;`; `&copy;`; `&quot;`; `&sect;`; `&laquo;`; `&raquo;`. Обозначить свои авторские права на созданный web-сайт: © Иванов И.И., 2006;
  - e) использовать таблицы для форматирования текста;
  - f) использовать таблицу каскадных стилей CSS;
- б) вставить графические объекты (рисунки, фотографии, графики) и оформить интерактивную подпись к ним. Использовать рисунок как гиперссылку. Использовать различные режимы обтекания картинки текстом. Все используемые рисунки поместить в отдельную папку;

7) с помощью прилагаемой программы GeoHTML сформировать на одном из рисунков навигационную карту.

## Приложение 1. Таблица основных тегов языка HTML

Основные теги документа	
<code>&lt;html&gt;&lt;/html&gt;</code>	Указывает программе просмотра страниц, что это HTML документ
<code>&lt;head&gt;&lt;/head&gt;</code>	Определяет место, где помещается различная информация, не отображаемая в теле документа. Здесь располагается тег названия документа <code>&lt;title&gt;</code> , теги стилевых таблиц <code>&lt;link&gt;</code> и <code>&lt;style&gt;</code> и теги для поисковых машин <code>&lt;meta&gt;</code>
<code>&lt;body&gt;&lt;/body&gt;</code>	Определяет видимую часть – «тело» документа
<code>&lt;title&gt;&lt;/title&gt;</code>	Помещает название документа в оглавление программы просмотра страниц
<code>&lt;body bgcolor = "?"&gt;</code>	Устанавливает цвет фона документа. ? – значение цвета в виде RRGGBB или текстовом виде. Например: FF0000 – красный (Red) цвет
<code>&lt;body text = "?"&gt;</code>	Устанавливает цвет текста документа. ? – значение цвета в виде RRGGBB или текстовом виде. Например: 000000 - черный цвет (Black)
<code>&lt;body link = "?"&gt;</code>	Устанавливает цвет гиперссылок. ? – значение цвета в виде RRGGBB или текстовом виде. Например: 00FF00 - зеленый цвет (Green)
<code>&lt;body vlink = "?"&gt;</code>	Устанавливает цвет гиперссылок, на которых вы уже побывали. ? – значение цвета в виде RRGGBB или текстовом виде. Например: 333333 (Gray) - серый цвет
<code>&lt;body alink = "?"&gt;</code>	Устанавливает цвет гиперссылок при нажа-

	тии
<b>Теги форматирования текста</b>	
<code>&lt;pre&gt;&lt;/pre&gt;</code>	Обрамляет предварительно отформатированный текст
<code>&lt;h1&gt;&lt;/h1&gt;</code>	Создает САМЫЙ БОЛЬШОЙ заголовок
<code>&lt;h6&gt;&lt;/h6&gt;</code>	Создает самый маленький заголовок
<code>&lt;b&gt;&lt;/b&gt;</code>	Создает жирный текст
<code>&lt;i&gt;&lt;/i&gt;</code>	Создает наклонный текст
<code>&lt;tt&gt;&lt;/tt&gt;</code>	Создает текст, имитирующий стиль печатной машинки
<code>&lt;cite&gt;&lt;/cite&gt;</code>	Используется для цитат, обычно наклонный текст
<code>&lt;em&gt;&lt;/em&gt;</code>	Используется для выделения из текста слова (наклонный или жирный текст)
<code>&lt;strong&gt;&lt;/strong&gt;</code>	Используется для выделения наиболее важных частей текста (наклонный или жирный текст)
<code>&lt;font size=?&gt;&lt;/font&gt;</code>	Устанавливает размер текста в пределах от 1 до 7
<code>&lt;font color=?&gt;&lt;/font&gt;</code>	Устанавливает цвет текста, используя значение цвета в виде RRGGBB
<code>&lt;marquee height="10" width="600" loop="2" bgcolor="#99CCFF"&gt;Текст&lt;/marquee&gt;</code>	Режим бегущей строки. Текст становится бегущей строкой. height, width – атрибуты текста, loop задает количество пробегов текста, bgcolor задает цвет фона
<code>&lt;p&gt;</code>	Создает новый параграф
<code>&lt;p align=?&gt;</code>	Выравнивает параграф относительно одной из сторон документа, значения:



	left, right, или center
 	Вставляет перевод строки
<blockquote> </blockquote>	Создает отступы с обеих сторон текста.
<dl></dl>	Создает список определений
<dt>	Определяет каждый из терминов списка
<dd>	Описывает каждое определение
<ol></ol>	Создает нумерованный список
<li>	Определяет каждый элемент списка и присваивает номер
<ul></ul>	Создает нумерованный список
<li>	Предваряет каждый элемент списка и добавляет кружок или квадратик
<div align=?>	Тег, используемый для форматирования больших блоков текста HTML документа, также используется в таблицах стилей

### Графические элементы

	Добавляет изображение в HTML-документ
	Выравнивает изображение к одной из сторон документа, ? принимает значения: left, right, center; bottom, top, middle
	Устанавливает толщину рамки вокруг изображения
<hr>	Добавляет в HTML-документ горизонтальную линию
<hr size=?>	Устанавливает высоту (толщину) линии

<code>&lt;hr width=?&gt;</code>	Устанавливает ширину линии, можно указать ширину в пикселах или процентах
<code>&lt;hr noshade&gt;</code>	Создает линию без тени.
<code>&lt;hr color=?&gt;</code>	Задаёт линии определенный цвет. Значение RRGGBB
<b>Таблицы</b>	
<code>&lt;table&gt;&lt;/table&gt;</code>	Создает таблицу
<code>&lt;tr&gt;&lt;/tr&gt;</code>	Определяет строку в таблице
<code>&lt;td&gt;&lt;/td&gt;</code>	Определяет отдельную ячейку в таблице
<code>&lt;th&gt;&lt;/th&gt;</code>	Определяет заголовок таблицы (нормальная ячейка с отцентрованным жирным текстом)
<code>&lt;table border=#&gt;</code>	Задаёт толщину рамки таблицы
<code>&lt;table cellspacing=#&gt;</code>	Задаёт расстояние между ячейками таблицы
<code>&lt;table cellpadding=#&gt;</code>	Задаёт расстояние между содержимым ячейки и ее рамкой
<code>&lt;table width=#&gt;</code>	Устанавливает ширину таблицы в пикселах или процентах от ширины документа
<code>&lt;tr align=?&gt;</code> или <code>&lt;td align=?&gt;</code>	Устанавливает выравнивание ячеек в таблице, принимает значения: left, center, или right.
<code>&lt;tr valign=?&gt;</code> или <code>&lt;td valign=?&gt;</code>	Устанавливает вертикальное выравнивание для ячеек таблицы, принимает значения: top, middle, или bottom.
<code>&lt;td colspan=#&gt;</code>	Указывает количество столбцов, которые объединены в одной ячейке (по умолчанию=1)

<code>&lt;td rowspan=#&gt;</code>	Указывает количество строк, которые объединены в одной ячейке (по умолчанию=1)
<code>&lt;td nowrap&gt;</code>	Не позволяет программе просмотра делать перевод строки в ячейке таблицы

### Гиперссылки

<code>&lt;a href="URL"&gt;&lt;/a&gt;</code>	Создает гиперссылку на другие документы или часть текущего документа
<code>&lt;a href="mailto:EMAIL"&gt; &lt;/a&gt;</code>	Создает гиперссылку вызова почтовой программы для написания письма автору документа
<code>&lt;a name="NAME"&gt;&lt;/a&gt;</code>	Отмечает часть текста как цель для гиперссылок в документе
<code>&lt;a href="#NAME"&gt;&lt;/a&gt;</code>	Создает гиперссылку на часть текущего документа

### Фреймы

<code>&lt;frameset&gt;&lt;/frameset&gt;</code>	Заменяет тег <code>&lt;body&gt;</code> в документе, определяющем структуру фреймов на веб-странице
<code>&lt;frameset rows="value,value"&gt;</code>	Определяет строки в таблице кадров, высота которых определена количеством пикселей или в процентном соотношении к высоте таблицы кадров
<code>&lt;frameset cols="value,value"&gt;</code>	Определяет столбцы в таблице кадров, ширина которых определена количеством пикселей или в процентном соотношении к ширине таблицы кадров
<code>&lt;frame&gt;</code>	Определяет единичный кадр или область в таблице кадров

<code>&lt;noframes&gt;&lt;/noframes&gt;</code>	Определяет, что будет показано в окне браузера, если он не поддерживает кадры
<code>&lt;frame src="URL"&gt;</code>	Определяет, какой из HTML-документов будет показан в кадре
<code>&lt;frame name="name"&gt;</code>	Указывает имя кадра или области, что позволяет перенаправлять информацию в этот кадр или область из других кадров
<code>&lt;frame marginwidth=#&gt;</code>	Определяет величину отступов по левому и правому краям кадра; должно быть равно или больше 1
<code>&lt;frame marginheight=#&gt;</code>	Определяет величину отступов по верхнему и нижнему краям кадра; должно быть равно или больше 1
<code>&lt;frame scrolling=VALUE&gt;</code>	Указывает будет ли выводиться линейка прокрутки в кадре; значение value может быть "yes," "no," или "auto". Значение по умолчанию для обычных документов – auto
<code>&lt;frame noresize&gt;</code>	Препятствует изменению размеров кадра

## Приложение 2. Цветовая палитра

Aliceblue F0F8FF	Antiquewhite FAEBD7	Aqua 00FFFF	Aquamarine 7FFFD4
Azure F0FFFF	Beige F5F5DC	Bisque FFE4C4	Black 000000
Blanchedalmond FFEBCD	Blue 0000FF	Blueviolet 8A2BE2	Brown A52A2A
Burlywood DEB887	Cadetblue 5F9EA0	Chartreuse 7FFF00	Chocolate D2691E
Coral FF7F50	Cornflowerblue 6495ED	Cornsilk FFF8DC	Crimson DC143C
Cyan 00FFFF	Darkblue 00008B	Darkcyan 008B8B	Darkgoldenrod B8860B
Darkgray A9A9A9	Darkgreen 006400	Darkkhaki BDB76B	Darkmagenta 8B008B
Darkolivegreen 556B2F	Darkorange FF8C00	Darkorchid 9932CC	Darkred 8B0000
Darksalmon E9967A	Darkseagreen 8FBC8F	Darkslateblue 483D8B	Darkslategray 2F4F4F
Darkturquoise 00CED1	Darkviolet 9400D3	deeppink FF1493	Deepskyblue 00BFFF
Dimgray 696969	Dodgerblue 1E90FF	Firebrick B22222	Floralwhite FFFAF0
Forestgreen 228B22	Fuchsia FF00FF	Gainsboro DCDCDC	Ghostwhite F8F8FF
Gold FFD700	Goldenrod DAA520	Gray 808080	Green 008000
Greenyellow ADFF2F	Honeydew F0FFF0	Hotpink FF69B4	Indianred CD5C5C
Indigo 4B0082	Ivory FFFFE0	Khaki F0E68C	Lavendar 9370DB

4B0082	FFFFF0	F0E68C	E6E6FA
Lavenderblush FFF0F5	Lawngreen 7CFC00	Lemonchiffon FFFACD	Lightblue ADD8E6
Lightcoral F08080	Lightcyan E0FFFF	Lightgoldenrodyellow FAFAD2	Lightgreen 90EE90
Lightgrey D3D3D3	Lightpink FFB6C1	Lightsalmon FFA07A	Lightseagreen 20B2AA
Lightskyblue 87CEFA	Lightslategray 778899	Lightsteelblue B0C4DE	Lightyellow FFFFE0
Lime 00FF00	Limegreen 32CD32	Linen FAF0E6	Magenta FF00FF
Maroon 800000	Mediumaquamarine 66CDAA	Mediumblue 0000CD	Mediumorchid BA55D3
Mediumpurple 9370D8	Mediumseagreen 3CB371	Mediumslateblue 7B68EE	Mediumspringgreen 00FA9A
Mediumturquoise 48D1CC	Mediumvioletred C71585	Midnightblue 191970	Mintcream F5FFFA
Mistyrose FFE4E1	Moccasin FFE4B5	Navajowhite FFDEAD	Navy 000080
Oldlace FDF5E6	Olive 808000	Olivedrab 688E23	Orange FFA500
Orangered FF4500	Orchid DA70D6	Palegoldenrod EEE8AA	Palegreen 98FB98
Paleturquoise AFEEEE	Palevioletred D87093	Papayawhip FFEFD5	Peachpuff FFDAB9
Peru CD853F	Pink FFC0CB	Plum DDA0DD	Powderblue B0E0E6

Purple 800080	Red FF0000	Rosybrown BC8F8F	Royalblue 4169E1
Saddlebrown 8B4513	Salmon FA8072	Sandybrown F4A460	Seagreen 2E8B57
Seashell FFF5EE	Sienna A0522D	Silver C0C0C0	Skyblue 87CEEB
Slateblue 6A5ACD	Slategray 708090	Snow FFFAFA	Springgreen 00FF7F
Steelblue 4682B4	Tan D2B48C	Teal 008080	Thistle D8BFD8
Tomato FF6347	Turquoise 40E0D0	Violet EE82EE	Wheat F5DEB3
White FFFFFF	Whitesmoke F5F5F5	Yellow FFFF00	YellowGreen 9ACD32

### Приложение 3. Темы для Web-сайтов

- 1) Российские государственные библиотечные информационные ресурсы: структура, формирование, распространение.
- 2) Российские государственные научно-технические информационные ресурсы: структура, формирование, распространение.
- 3) Российские государственные архивные информационные ресурсы: структура, формирование, распространение.
- 4) Российские государственные правовые информационные ресурсы: структура, формирование, распространение.
- 5) Российские государственные статистические информационные ресурсы: структура, формирование, распространение.
- 6) Геоинформационные системы: структура, ведущие производители, сферы применения.
- 7) Электронные информационные ресурсы органов власти Томской области.
- 8) Государственная регистрация информационных ресурсов: принципы, правила, организация.
- 9) Обработка и распространение биржевой и финансовой информации: ведущие российские и зарубежные компании.
- 10) Медиаимперии: характеристика ведущих мировых распространителей новостной и развлекательной информации.
- 11) Поисковые системы Интернет: история, услуги, принципы работы.
- 12) Информационная безопасность: методы и способы защиты информационных ресурсов.
- 13) Электронные библиотеки в Интернет: классификация, ресурсы, услуги.
- 14) Wiki-технология: история, принципы, перспективы развития.



15) Семантический Web (WWW второго поколения): концепция, составляющие, перспективы развития.

16) Международное сотрудничество в информационной сфере: правовые и экономические аспекты.

17) Глобальные информационные сети: классификация, структура, доступ.

18) Информационные продукты: классификация, ведущие производители и поставщики.

19) Информационные услуги: классификация, ведущие поставщики.

20) Государственная информационная политика: российская и зарубежная практика.

21) Управление Интернет-сообществом: принципы, структура, проблемы.

22) Проблемы сохранения цифровых информационных ресурсов.

23) Мировой рынок информации: сегменты, характеристика, тенденции развития.

24) Модели и стандарты представления информации и метаинформации.

25) Data Mining: методы и средства поиска, обнаружения, извлечения и анализа текстовых, аудио- и других данных.

**Приложение 4. Программное обеспечение, необходимое  
для выполнения лабораторных работ по дисциплине  
«Мировые информационные ресурсы»**

№	Программа	Назначение	Дистрибутив	Объем, Мб	Лицензия
1	Notepad++ v.4.5	Текстовый редактор. Удобен для написания html-кода	npp.4.5.Installer.exe	1,7	freeware
2	GeoHTML v.2.1	Программа создания изображения–навигационной карты	geohtm21.exe	1,2	freeware
3	WinHTTrack Website Copier 3.42-2	Оффлайн-браузер	httrack-3.42-2.exe	3,33	freeware
4	Download Master	Менеджер загрузок	dmaster-5.6.1.1187-setup.exe	7,25	freeware
5	SaveChm 1.2	Аддон (программное дополнение) к браузеру, позволяющий сохранять web-страницы	savechm-1.2.exe	0,49	freeware
6	Quintura Search, или Search + v.1.0.0.0	Метапоисковик	Quintura_Search_RU.exe  srchplus.exe	5,52  1,11	freeware
7	Amaya 11.4.4	WYSIWYG-редактор HTML	amaya-WinXP-11.4.4.exe	13,88	open source