

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ –
ВЫСШАЯ ШКОЛА ЭКОНОМИКИ
В САНКТ-ПЕТЕРБУРГЕ

Л. А. КЕРОВ

РАЗРАБОТКА WEB-САЙТОВ В MICROSOFT VISUAL STUDIO И ИХ РАЗМЕЩЕНИЕ В СЕТИ ИНТЕРНЕТ

Учебное пособие



Санкт-Петербург
2011

УДК 681.3.07
ББК 32.973.26-0.18.2.75

Рецензенты:

М. В. Дмитриева, доцент кафедры информатики СПбГУ

С. В. Афанасьева, доцент кафедры бизнес-информатики Национального исследовательского университета «Высшая школа экономики» в Санкт-Петербурге

Керов Л. А.

Разработка Web-сайтов в Microsoft Visual Studio и их размещение в сети Интернет. Учебное пособие. — Отдел оперативной полиграфии НИУ ВШЭ — Санкт-Петербург, 2011. — 92 с.

В учебном пособии рассматриваются практические приемы создания информационных Web-сайтов с использованием инструментальной системы Microsoft Visual Studio. Излагаются основные понятия компьютерных сетей, теги HTML-разметки Web-страниц, создание таблиц стилей с использованием визуального дизайнера, разработка визуального дизайна Web-сайта и хостинг Web-сайта с HTML-страницами. Приводятся схема функционирования Web-сайтов с HTML страницами и схема функционирования Web-сайтов с активными серверными страницами. Даются начальные сведения об использовании инструментальной системы Microsoft Visual Studio для создания Web-сайтов с активными серверными страницами, использующими C# в качестве языка программирования серверных скриптов.

Работа написана на основе материалов курсов, которые на протяжении ряда лет читался автором на факультете менеджмента в Санкт-Петербургском филиале государственного университета – Высшей Школы Экономики и на математико-механическом факультете Санкт-Петербургского государственного университета.

Для преподавателей, аспирантов, студентов и слушателей системы повышения квалификации и переподготовки кадров

Рекомендовано к печати Учебно-методическим советом НИУ ВШЭ в Санкт-Петербурге.

ОГЛАВЛЕНИЕ

ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ	5
1.1. Коммуникации и компьютерные сети	5
1.2. Виды компьютерных сетей	6
1.3. Каналы связи компьютерных сетей, сетевой адаптер, модем, коммуникационные устройства	8
1.4. Глобальная компьютерная сеть Internet, история создания и развития	9
1.5. Коммутация пакетов, протоколы передачи данных, адресация, виды подключений в сети Internet.	11
1.6. Сервисы, Web-сайты, браузеры и адресация информационных ресурсов в сети Internet	12
1.7. Доменное имя Web-сайта	13
1.8. Web-сайт как инструмент бизнеса	15
ГЛАВА 2. ЗНАКОМСТВО С WEB-САЙТАМИ И HTML-СТРАНИЦАМИ	17
2.1. Создание Web-сайта, добавление и разметка HTML- страницы	17
2.2. Оформление заголовков. Проверка функционирования HTML-страницы	20
2.3. Парные и одиночные теги. Атрибуты. Тег перехода на новую строку.	22
2.4. Оформление абзацев. Использование стилей	23
2.5. Оформление неупорядоченных (маркированных) списков.	24
2.6. Оформление упорядоченных (нумерованных) списков.	25
2.7. Оформление списков определений	26
2.8. Горизонтальные линии.	27
2.9. Комментарии	28
2.10. Специальные символы.	29
ГЛАВА 3. ГРАФИЧЕСКОЕ ОФОРМЛЕНИЕ HTML-СТРАНИЦЫ	30
3.1. Форматы графических файлов	30
3.2. Вставка изображения на страницу	31
3.3. Ссылка на внешнее изображение	32
3.4. Фоновое изображение.	34
3.5. Анимированное изображение	35

ГЛАВА 4. ГИПЕРССЫЛКИ 36

4.1. Понятие гипертекста. Создание гиперссылок 36

4.2. Создание закладок 37

4.3. Создание карт навигации 38

ГЛАВА 5. ТАБЛИЦЫ 40

5.1. Структура таблицы 40

5.2. Атрибуты тега <table> 41

5.3. Атрибуты тега <tr> и тега <td>. 42

ГЛАВА 6. СТИЛЕВОЕ ОФОРМЛЕНИЕ HTML-СТРАНИЦ 44

6.1. Понятие стиля 44

6.2. Внутренняя таблица стилей. Локальный стиль. 45

6.3. Внешняя таблица стилей 47

6.4. Индивидуальные стили для тегов 50

6.5. Стили гиперссылок 51

6.6. Пользовательские теги стилей 52

6.7. Использование стилей для форматирования текста 53

ГЛАВА 7. РАЗРАБОТКА ВИЗУАЛЬНОГО ДИЗАЙНА САЙТА 55

7.1. HTML-разметка домашней страницы 55

7.2. HTML-разметка страницы контента 61

ГЛАВА 8. ХОСТИНГ WEB-САЙТА С HTML-СТРАНИЦАМИ. 64

8.1. Регистрация доменного имени сайта 64

8.2. Загрузка файлов на сайт. 68

8.3. Управление файлами сайта. 69

ГЛАВА 9. ЗНАКОМСТВО С WEB-САЙТАМИ С ASP-СТРАНИЦАМИ 70

9.1. Схема функционирования Web-сайта 70

9.2. Создание шаблона Web-сайта с ASP-страницей 72

9.3. Проектирование ASP-страницы. 77

9.4. Программирование ASP-страницы. 82

Контрольные вопросы 87

Список литературы 91

ГЛАВА 1
ОСНОВНЫЕ ПОНЯТИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

1.1. Коммуникации и компьютерные сети

Коммуникация (лат. *communicatio* — сообщение, связь, передача) — процесс, путь и средства передачи какого-либо объекта или сообщения с одного места на другое. Коммуникации подразделяются на два вида (рис. 1.1):

- **материальные** коммуникации — передача физических объектов
- **информационные** коммуникации — передача информации

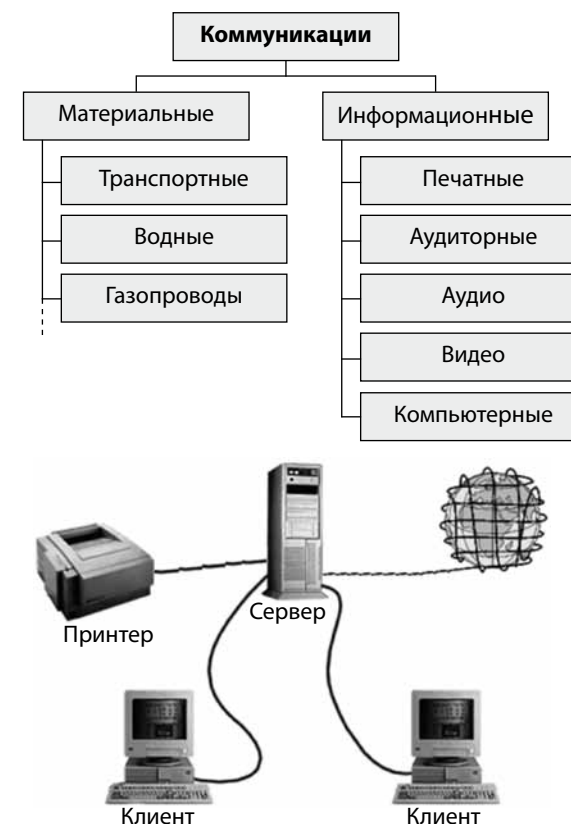


Рис. 1.1. Виды коммуникаций

Информационные коммуникации подразделяются на следующие виды:

- **печатные** коммуникации — книги, журналы, изобразительная продукция,...
- **аудиторные** коммуникации — лектории, театры, церкви, концертные залы,...
- **аудио** коммуникации — радио, телефон, магнитные записи,...
- **видео** коммуникации — видеофильмы, телевидение, кино,...
- **компьютерные** коммуникации — передача информации между компьютерами, которая рассматривается как информационный процесс и реализуется с помощью материальных носителей информации и компьютерных сетей

Компьютерная сеть (computer network) — это два или более компьютеров, соединенных вместе средствами связи для совместного использования ресурсов и обмена информацией. Компьютерная сеть позволяет:

- совместно использовать информацию, обмениваться ею, сохранять и защищать информацию,
- совместно использовать оборудование и программное обеспечение,
- выполнять централизованное администрирование и обслуживание компьютеров.

Сервер (server) компьютер, чьи ресурсы доступны сетевым пользователям (пользователям других компьютеров сети). **Клиент** (client) — компьютер, который осуществляют доступ к сетевым ресурсам, предоставляемым серверами. Сетевые **ресурсы** (network resources) — файлы, приложения и периферийные устройства (принтеры, сканеры, модемы, CD-ROM и др.), предоставляемые серверами и доступные пользователям сети. **Среда передачи** (media) — линии (каналы) связи компьютерной сети; среда передачи информации между компьютерами может быть реализована с использованием проводных и беспроводных технологий связи.

1.2. Виды компьютерных сетей

В зависимости от **размеров и территориального расположения** компьютерная сеть может быть отнесена к одному из следующих типов (рис. 1.2):

- **локальная вычислительная сеть** (Local Area Network, LAN) — компьютерная сеть, расположенная на ограниченной территории (в пределах одного помещения, здания)
- **глобальная вычислительная сеть** (Wide Area Network, WAN) — объединение двух или более компьютерных сетей, географически уда-

ленных друг от друга. Примером глобальной вычислительной сети может служить Internet.

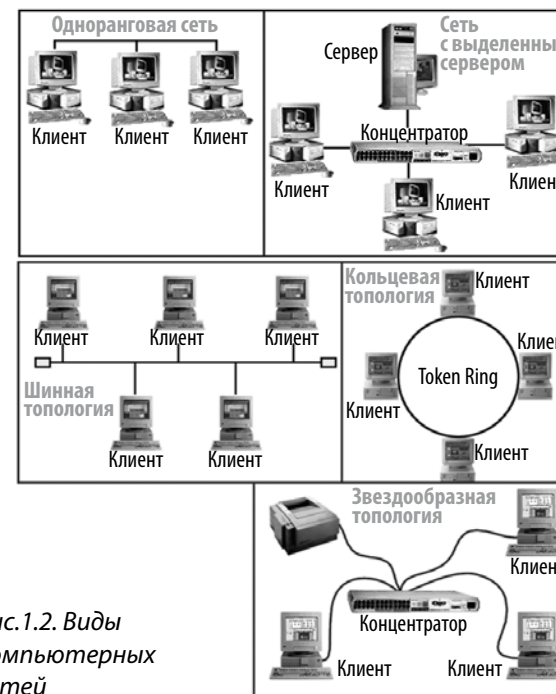


Рис. 1.2. Виды компьютерных сетей

В зависимости от **способа обмена информацией** компьютерная сеть может быть отнесена к одному из следующих типов:

- **одноранговая сеть** (peer-to-peer network) — сеть, в которой все компьютеры равноправны и в которой каждый компьютер функционирует и как клиент, и как сервер; пользователи сами решают, какие ресурсы предоставлять для общего доступа, т.е. сделать сетевыми
- **сеть с выделенным сервером** (client/server network) — сеть, в которой установлен один или несколько серверов, предоставляющих свои ресурсы в общее пользование и обеспечивающих централизованное управление сетью; остальные компьютеры сети являются клиентами и называются рабочими станциями (workstation)

Схема соединения узлов сети называется **топологией сети**. Основными видами топологий являются:

- **шинная** топология
- **кольцевая** топология
- **звездообразная** топология

1.3. Каналы связи компьютерных сетей, сетевой адаптер, модем, коммуникационные устройства

Взаимодействие компьютеров в сети осуществляется по проводным и беспроводным каналам связи. **Проводные каналы связи** реализуются с помощью следующих физических сред передачи сигналов:

- электрический кабель типа «**витая пара**»: 10–100 Мбит/с, низкая помехоустойчивость
- электрический **коаксиальный кабель**: до 10 Мбит/с, высокая помехоустойчивость
- **оптоволоконный кабель**: 10–100 Мбит/с; абсолютная помехоустойчивость
- **телефонная линия**: 1–2 Мбит/с, низкая помехоустойчивость

Беспроводные каналы связи реализуются с помощью следующих физических сред передачи сигналов: радиосвязь, сотовая связь, спутниковая связь, инфракрасная связь.

Для передачи информации по каналам связи необходимо преобразовывать компьютерные сигналы в сигналы физических сред, что выполняется с помощью специальных устройств. При передаче информации по кабельным сетям в качестве такого устройства используется **сетевой адаптер** (сетевой контроллер, сетевая плата) — устройство, с помощью которого компьютер подключается к сети. С одной стороны сетевой адаптер подключается к кабелю, с другой — вставляется в соответствующий разъем компьютера, подключаемого к сети, или размещается на материнской плате компьютера (рис. 1.3).

При передаче информации по телефонной линии в качестве устройства преобразования компьютерных сигналов в сигналы физической среды используется **модем**, который выполняет **модуляцию** (преобразование цифровых сигналов аналоговые сигналы) и **демодуляцию** (преобразование аналоговых сигналов в цифровые сигналы). С одной стороны сетевой адаптер подключается к телефонной линии, с другой — вставляется в соответствующий разъем компьютера, подключаемого к сети, или размещается на материнской плате компьютера.

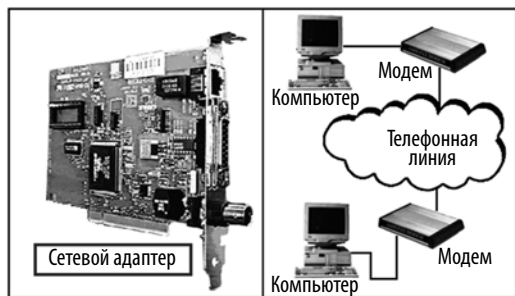


Рис.1.3. Сетевой адаптер и модем

В **беспроводных сетях** (WLAN — Wireless Local Area Network) в качестве среды передачи информации используются радиоволны или инфракрасные лучи и специализированные модемы, работающие по протоколу **Wi-Fi** (Wireless Fidelity — точность без проводов). Для беспроводного подключения периферийных устройств к компьютеру используются сетевые USB-адаптеры, работающие по протоколу **Bluetooth**. Технология Bluetooth (голубой зуб) названа в честь датского короля X века Гарольда II по прозвищу «Голубой Зуб», всемирно прославившегося собирательством датских земель.

Для соединения между собой компьютеров, для расширения сетей и для соединения между собой компьютерных сетей используются **коммуникационные устройства**:

- **концентратор (hub)** — устройство, которое «разветвляет» сеть на витой паре: любая информация, пришедшая на один из его портов, через небольшое время отсылается через все остальные порты. Все порты хаба — двунаправленные.
- **коммутатор (switch)** — как и концентратор, позволяет объединить несколько компьютеров, подключив их к одному серверу. В отличие от устаревших теперь концентраторов (hub), коммутатор позволяет пересылать информацию между несколькими сегментами сети, не загружая остальную сеть. Он анализирует адрес информации в ее заголовке и, сверившись с адресной таблицей, направляет эту информацию в соответствующий порт.
- **маршрутизатор (router)** — устройство, которое распознает адрес получателя и направляет информацию только туда, куда она адресована. Для этих целей может использоваться отдельный компьютер с несколькими сетевыми адаптерами; маршрутизатор можно применять для связи различных сетей; внутри одной сети применяются коммутаторы
- **мост (bridge)** — комбинация аппаратного и программного обеспечения, соединяющая две локальные сети и позволяющая осуществить коммуникации между их клиентами.

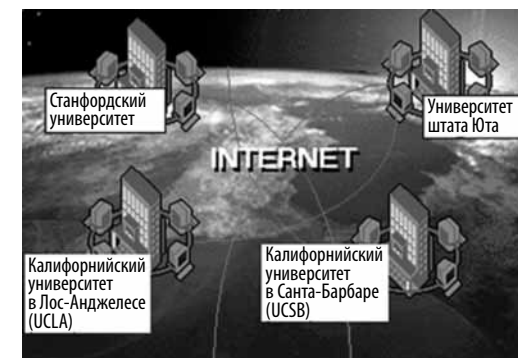
1.4. Глобальная компьютерная сеть Internet, история создания и развития

Internet — это система, объектами которой являются компьютерные сети (каждая из которых имеет свои принципы организации). Для соединения указанных компьютерных сетей используются специальные аппаратные средства (маршрутизаторы, мосты и др.) и общие правила передачи информации (протоколы).

Этапы создания и развития сети Internet:

- 1969 г.:** В Министерстве обороны США, в Агентстве передовых исследовательских проектов (Advanced Research Project Agency, ARPA) создана децентрализованная компьютерная сеть **ARPANET**, объединившая компьютеры четырех исследовательских центров (рис.1.4).
- 1983 г.:** Внедрен протокол передачи данных **TCP/IP**, на использование которого перешли все компьютеры ARPANET. Благодаря этому протоколу появилась возможность расширения ARPANET.
- 1986 г.:** Национальный научный фонд США (NSF) создал межрегиональную сеть NSFNET, которая стала основной опорной сетью для Internet. В июне 1990 г. **NSFNET** заменила ARPANET, которая прекратила свое существование.
- 1991 г.:** В Европейском центре исследований элементарных частиц (CERN, Швейцария) завершена работа над проектом **World Wide Web** («Всемирная паутина»). Система World Wide Web установлена в CERN для публикации статей, посвященных научным исследованиям и результатам эксперимент.
- 1992 г.:** Система World Wide Web и соответствующее программное обеспечение стали общедоступными. Сняты ограничения на коммерческое использование системы WWW, что открыло путь к созданию корпоративных порталов.
- 1993 г.:** В Национальном центре разработки приложений для суперкомпьютеров (NCSA, США) разработан первый графический браузер **Mozilla**.
- 1994 г.:** Компания Netscape выпустила одноименный Web-браузер, в который было внесено много усовершенствований по сравнению с Mozilla: поддержка нескольких одновременных TCP/IP-соединений, cookie-файлы на клиентских компьютерах, поддержка фреймов и JavaScript.
- 1995 г.:** Компания Microsoft выпустила Web-браузер Internet Explorer 1.0.
- 1996 г.:** Имея почти 40 млн пользователей в 150 странах мира, Internet стал необходимостью в работе компаний и важнейшим местом общения. Создание компанией Sun языка Java и создание компания Microsoft технологии Active Server Pages заложило фундамент объектно-ориентированной разработки Web-приложений класса «клиент-сервер». Microsoft выпустила Web-браузер Internet Explorer 3.0, в котором появилась поддержка внешних таблиц стилей (Cascading Style Sheet, CSS).
- 2002 г.:** Microsoft открыла новую эру в разработке Web-приложений, представив инструментальную среду .NET Framework и язык C#.

Рис. 1.4. Первый этап развития сети Internet



1.5. Коммутация пакетов, протоколы передачи данных, адресация, виды подключений в сети Internet

Коммуникационную инфраструктуру сети Internet образуют узлы и каналы связи. Узел сети Internet — это компьютер или коммуникационное устройство, подключенные к сети Internet. **Каналы связи** — это технические средства, необходимые для организации связи (соединения) между узлами. Каналы связи могут быть проводными и беспроводными.

Передача данных в сети Internet основана на принципе **коммутации пакетов**: поток данных, передаваемых от одного узла к другому, разбивается на пакеты; пакеты передаются через систему коммуникаций и маршрутизаторов независимо друг от друга и снова собираются вместе на приемной стороне.

Для передачи информации между узлами сети используются специальные правила (протоколы). В сети Internet стандартным протоколом передачи данных является протокол **TCP/IP** (Transmission Control Protocol / Internet Protocol), который состоит из двух протоколов:

- **TCP** — транспортный протокол, который отвечает за процесс формирования пакетов перед отправкой, за доставку данных, и за объединение пакетов при получении,
- **IP** — протокол обмена пакетами, который выполняет адресацию и выбор маршрута.

Каждый узел в сети Internet имеет свой адрес, который называется **IP-адресом** и состоит из 32 разрядов, разбитых на 4 поля по 8 разрядов. При записи адреса каждое из этих полей преобразуется в 10-ичное число из диапазона [0, 255]; например, **192.168.100.2**.

Вопросами распределения IP-адресов для сетей, присоединяемых к сети Internet, занимается международная некоммерческая орга-

низация ICANN (Internet Corporation for Assigned Names and Numbers; www.icann.org). В сети Internet применяются два типа IP-адресов: **общие IP-адреса** являются глобально уникальными в сети Internet; **частные IP-адреса** зарезервированы для использования в локальных сетях организаций. Компьютер в сети Internet, имеющий уникальный IP-адрес, называется **хостом** (host).

Услуги подключения к сети Internet оказывают организации, называемые **Internet-провайдерами** (Internet Service Provider, ISP). Провайдеры имеют постоянные высокоскоростные каналы соединения (связи) с Internet. Для подключения компьютера или компьютерной сети к Internet необходимо установить соединение с провайдером. В зависимости от длительности соединения с провайдером существуют **два вида подключения к Internet**:

- **сеансовое подключение**: для создания сеансового подключения необходимо каждый раз дозваниваться по определенным телефонным номерам провайдера; при этом подключение к сети Internet осуществляется на определенное ограниченное время (сеанс) телефонного соединения. К сеансовому подключению относятся следующие виды доступа: коммутируемый доступ (Dial-Up), для организации которого используется телефонная линия и модем; беспроводной доступ с использованием сотовой телефонной связи,
- **постоянное подключение**: подключение с Internet неограниченно по времени и может быть прекращено по желанию пользователя. К постоянному подключению относятся следующие виды доступа: доступ по выделенной линии: выделенные линии связи могут быть проводные, оптоволоконные, радио, спутниковые; доступ с использованием беспроводных сетей Wi-Fi (Wireless Fidelity): для организации доступа к Internet на компьютере необходимо установить Wi-Fi-адаптер.

1.6. Сервисы, web-сайты, браузеры и адресация информационных ресурсов в сети Internet

Сеть Internet предоставляет следующие услуги:

- **World Wide Web** (WWW, Web, «Всемирная паутина») — служба доступа к большому хранилищу информации, которая организована в виде Web-сайтов и хранится на серверах Internet-провайдеров, которые называются **Web-серверами**,
- **e-mail** (электронная почта) — служба обмена сообщениями,
- **FTP** (File Transfer Protocol) — служба доступа к библиотекам файлов,
- **IRC** (Internet Relay Chat) — служба для организации общения людей в режиме реального времени.

Web-сайт — это папка с файлами, среди которых могут быть графические файлы, файлы внешних таблиц стилей и др. но обязательно должны быть текстовые файлы, которые размечены с помощью языка HTML (HyperText Markup Language) и называются Web-страницами.

Программное обеспечение Web-сервера позволяет на одном компьютере организовать несколько хостов, называемых **виртуальными хостами**. Благодаря этому на одном Web-сервере может размещаться несколько Web-сайтов. Примером программного обеспечения для Web-сервера может служить **Microsoft Internet Information Services (IIS)**, которое функционирует под управлением операционных систем Windows 2000 Server, Windows 2003 Server, Windows XP Professional.

Для просмотра Web-страниц используются специальные программы, называемые **Web-браузерами**. Примером браузера может служить браузер **Microsoft Internet Explorer**.

Каждый информационный ресурс в сети Internet имеет уникальный адрес, который называется **URL** (Uniform Resource Locator) и состоит из трех частей:

- **схема доступа** к ресурсу (протокол),
- **имя компьютера** (хоста), на котором располагается ресурс,
- **имя ресурса**, указывается в виде пути и имени файла.

Если информационный ресурс находится во «Всемирной паутине», то для доступа к нему используется протокол **http** (HyperText Transfer Protocol). URL информационных ресурсов во «Всемирной паутине» имеет вид: **http://сервер/путь_к_файлу**.

Пример: **http://www.hse.ru/default.htm**.

Для почтовых сообщений используется протокола **mailto**, а URL имеет вид: **mailto:адрес_электронной_почты**.

Пример: **mailto:kerov@hse.spb.ru**

1.7. Доменное имя Web-сайта

Система адресации с сети Internet требует знания имени компьютера (хоста), на котором располагается ресурс. Каждый хост имеет IP-адрес, но эти адреса являются цифровыми и трудны для запоминания. В 1984 г. была разработана доменная система именования **DNS** (Domain Name System), которая позволяет присваивать IP-адресам символьные имена, называемые **доменными именами**. В сети Internet создана распределенная база данных, которая служит для установления соответствия между числовыми IP-адресами и символьными именами узлов сети.

Множество имен в DNS структурировано в виде дерева. Корень

этого дерева называется *корневым доменом* и не имеет имени. Корневой домен содержит множество *доменов верхнего уровня* (Top Level Domains, TLD), которые указывают *тип организации*, владеющей соответствующим доменным именем, или *код страны*, в которой зарегистрировано данное доменное имя. Примеры: **com**, **net**, **ru**. Регистрацией доменов верхнего уровня занимается международная некоммерческая организация **ICANN** (Internet Corporation for Assigned Names and Numbers; www.icann.org)

Каждый домен верхнего уровня содержит множество вложенных доменов второго уровня, которые идентифицируют конкретные организации и частные лица и регистрируются в доменах верхнего уровня организациями, которые аккредитованы в ICANN. Доменное имя второго уровня состоит из имени домена организации (или частного лица) и имени домена верхнего уровня, в котором зарегистрирован домен организации (или частного лица).

Домены второго уровня могут содержать произвольное множество вложенных доменов — *поддоменов*, которые регистрируются владельцами доменов второго уровня и идентифицируют, например, подразделения организации.

Каждый Web-сайт, размещенный в Internet, имеет уникальный адрес, который называется *доменным именем Web-сайта*. Оно состоит из префикса **www** и имени домена второго уровня, в котором зарегистрирован компьютер, содержащий Web-сайт. Примеры: **www.hse.ru**, **www.hse.spb.ru**.

Для размещения Web-сайта в Internet необходимо *зарегистрировать доменное имя сайта*. В России регистрацией доменных имен в доменах **com**, **org**, **net**, **biz**, **info**, **ru**, **su** занимаются Internet-провайдеры. Перед регистрацией доменного имени необходимо проверить, что оно свободно. Это можно сделать с помощью информационного сервиса **WHOIS** на сайтах Internet-провайдеров; например, на сайте **www.nic.ru**. Регистрация доменного имени в доменах верхнего уровня является платной.

После регистрации доменного имени необходимо *выбрать, где разместить Web-сайт*. Размещение Web-сайтов в Internet называется *хостингом*. Услуги хостинга сайтов предоставляют Internet-провайдеры, которые за определенную плату размещают Web-сайты заказчиков на своих Web-серверах. Существуют организации, которые в рекламных целях бесплатно регистрируют доменные имена сайтов в собственных доменах второго уровня и бесплатно осуществляют хостинг сайтов. Например, можно бесплатно зарегистрировать доменное имя и разместить персональный Web-сайт на сайте **http://www.narod.ru**.

1.8. Web-сайт как инструмент бизнеса

В XX веке считалось, что *Web-сайт — это рекламный буклет, брошюра или визитка, выложенная в Интернет*. Соответствующими были и требования к сайту — он должен был выглядеть красиво и не требовать особых затрат. Отдачи от него ожидали не больше, чем от любых других рекламных материалов.

В XXI веке корпоративный *Web-сайт стал инструментом маркетинга, продаж и коммуникаций с клиентами*. Он стал давать реальную отдачу в виде роста продаж, улучшения обслуживания потребителей, интерактивной работы с дилерами и дистрибьюторами, повышения эффективности найма персонала и формирования имиджа и т.д.

Web-сайт стал реально действующим инструментом бизнеса, отсутствие которого или его неэффективная работа могли нанести серьезный ущерб предприятию. Из категории рекламных материалов *Web-ресурсы переросли в разряд стратегических активов*, поддерживающих исполнение сотен бизнес-функций.

Перед созданием Web-сайта важно осознать и зафиксировать цели и задачи, которые вы перед ним ставите. Это позволит избежать пустой траты ресурсов и уберечь от разочарований. *Основными задачами*, на решение которых чаще всего ориентируются владельцы *Web-сайтов*, являются:

- предоставление информации о компании ее клиентам, партнерам по бизнесу, инвесторам, СМИ,
- продажа товаров или услуг,
- получение обратной связи от клиентов и партнеров,
- консультирование, обучение и прочие формы online-поддержки клиентов,
- сбор аналитической информации с помощью опросов и анкетирования,
- тестирование соискателей работы.

С точки зрения бизнеса *к ключевым характеристикам качественного Web-сайта* относятся:

- соответствие целям и задачам бизнеса, а не раздутым амбициям владельцев, дизайнерским предпочтениям или требованиям системных администраторов,
- простота и удобство использования для клиентов. Ваши клиенты — это единственная категория пользователей, требования которых надо учитывать полностью. Мелочей здесь практически нет. Например, неграмотное размещение фотографий или описание продуктов

на Web-странице будет иметь такие же последствия, как хаотичное разбрасывание товаров по полкам супермаркета.

Корпоративный Web-сайт должен быть:

- ориентированным на достижение определенных целей, т.е. способствовать решению ваших бизнес-задач,
- интегрированным, т. е. взаимодействовать с другими используемыми вами внешними и внутренними информационными системами, такими, как складские и бухгалтерские программы, CRM (client relations management — управление взаимодействием с клиентами), офисные пакеты,
- ориентированным на клиентов, т. е. ставить задачи клиентов превыше всего,
- лаконичным, легким для восприятия, лишенным всего, не имеющего значения для посетителей,
- обновляемым, открытым для изменений,
- надежным.

Не следует загромождать Web-сайт лишними текстами, графическими элементами или функциями. Нужно убрать с него все лишнее. Это позволит сделать из сайта реально работающий инструмент бизнеса. Все прочие требования являются вторичными. Например, важно, чтобы дизайн сайта соответствовал вашему фирменному стилю и максимально способствовал решению задач пользователей. Набор функциональностей (сервисов) тоже должен быть подобран, исходя из тех ожиданий, которые есть у посетителей.

ГЛАВА 2 ЗНАКОМСТВО С WEB-САЙТАМИ И HTML-СТРАНИЦАМИ

2.1. Создание Web-сайта, добавление и разметка HTML-страницы

Для создания Web-сайта выполним следующие действия (рис. 2.1):

- 1) запустим инструментальную систему Visual Studio 2008;
- 2) выберем из меню «**Файл**» пункт «**Создать**» и команду «**Веб-узел**»;
- 3) на экран будут выведено диалоговое окно «**Новый веб-узел**», в верхнем списке которого выберем «**.NET Framework 3.5**»;
- 4) в списке «**Шаблоны**» выберем «**Пустой веб-узел**»;
- 5) в списке «**Расположение**» выберем элемент «**Файловая система**», а в следующее за ним текстовое поле введем *полное имя папки*, в которой должны храниться страницы Web-сайта;
- 6) в поле «**Язык**» выберем «**Visual C#**»;
- 7) щелкнем кнопку «**ОК**».

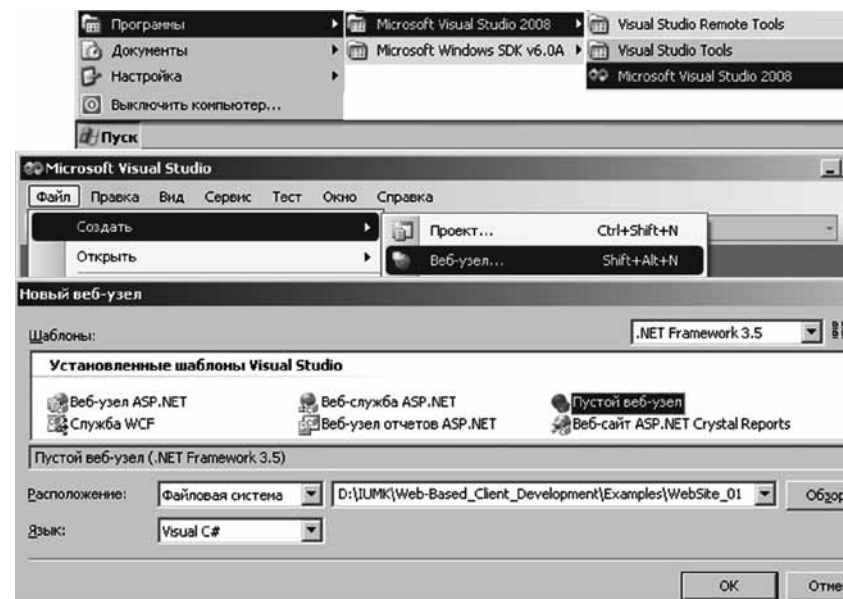


Рис. 2.1. Создание Web-сайта

В результате выполненных действий будет создан пустой Web-сайт, содержимое которого можно увидеть в окне «Обозреватель решений» (рис. 2.2).

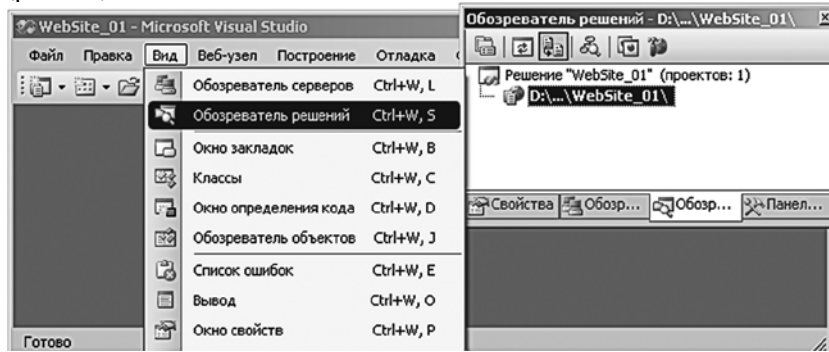


Рис. 2.2. Окно «Обозреватель решений»

Для добавления HTML-страницы в созданный сайт выберем из меню «Веб-узел» команду «Добавить новый элемент». На экран будет выведено одноименное диалоговое окно, в котором выполним следующие действия (рис. 2.3.):

- в области «Шаблоны» выберем шаблон «HTML-страница»,
- в поле «Имя» отредактируем (если это необходимо) имя файла HTML-страницы,
- щелкнем кнопку «Добавить».

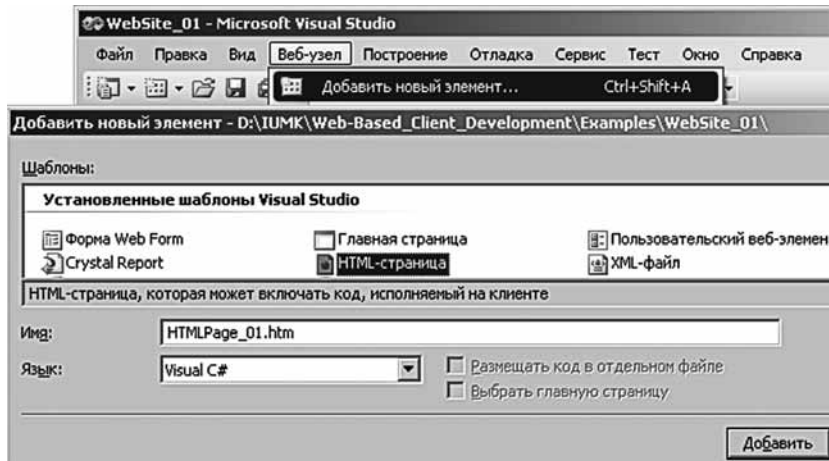


Рис. 2.3. Добавление HTML-страницы в Web-сайт

В результате выполненных действий в сайт будет добавлена HTML-страница, содержимое которой может быть отображено в одном из трех режимов:

- в режиме «Конструктор» HTML-страница показывается в таком виде, как она будет выглядеть в окне Web-браузера (например, Microsoft Internet Explorer),
- в режиме «Исходный код» показывается код разметки HTML-страницы,
- в режиме «Разделить» рабочая область Visual Studio разделяется на две части, одна из которых соответствует режиму «Исходный код», а вторая — режиму «Конструктор».

Переключение указанных выше режимов выполняется с помощью ярлычков в нижней части окна. Код разметки HTML-страницы содержит следующие элементы (рис. 2.4):

- 1) Директиву «!DOCTYPE», которая задает формат кода разметки (в данном случае — XHTML 1.0 Transitional),
- 2) Теги кода разметки, которые заключены в угловые скобки:
 - `<html>` `</html>` — теги начала и конца кода разметки HTML-страницы,
 - `<head>` `</head>` — теги заголовка HTML-страницы,
 - `<title>` `</title>` — теги названия HTML-страницы,
 - `<body>` `</body>` — теги тела HTML-страницы.

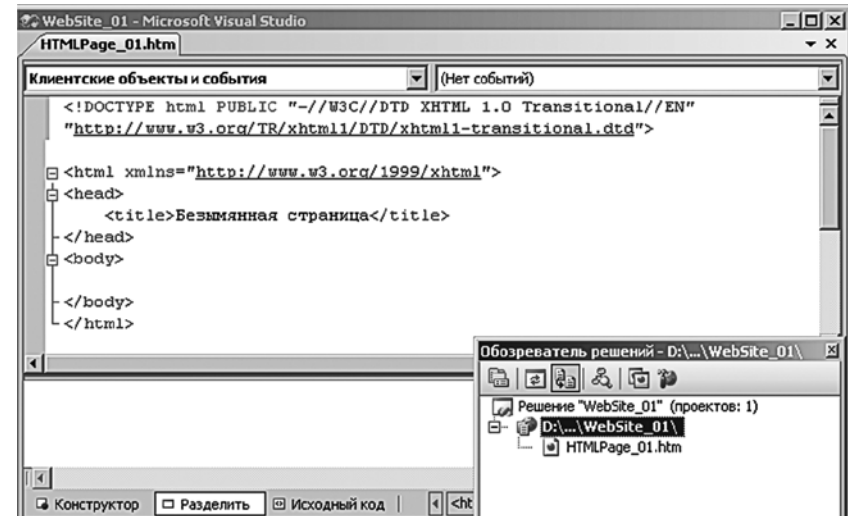


Рис. 2.4. Код разметки HTML-страницы

**2.2. Оформление заголовков.
Проверка функционирования HTML-страницы**

Если в тексте HTML-страницы нужно выделить заголовки разделов, то для этого используются *теги заголовков* (рис. 2.5):

- <h1> </h1>
- <h2> </h2>
- <h3> </h3>
- <h4> </h4>
- <h5> </h5>
- <h6> </h6>



Рис. 2.5. Оформление заголовков

Для проверки функционирования HTML-страницы выберем из меню «Отладка» команду «Запуск без отладки». В ответ на это инструментальная система Visual Studio выполнит следующие действия (рис. 2.6):

- запустит встроенный в нее тестовый Web-сервер, и на панели задач появится значок с комментарием «ASP.NET Development Server»,
- на тестовом Web-сервере будет размещен Web-сайт,
- на клиентском компьютере будет запущен браузер Microsoft Internet Explorer, и на нем будет отображена HTML-страница.

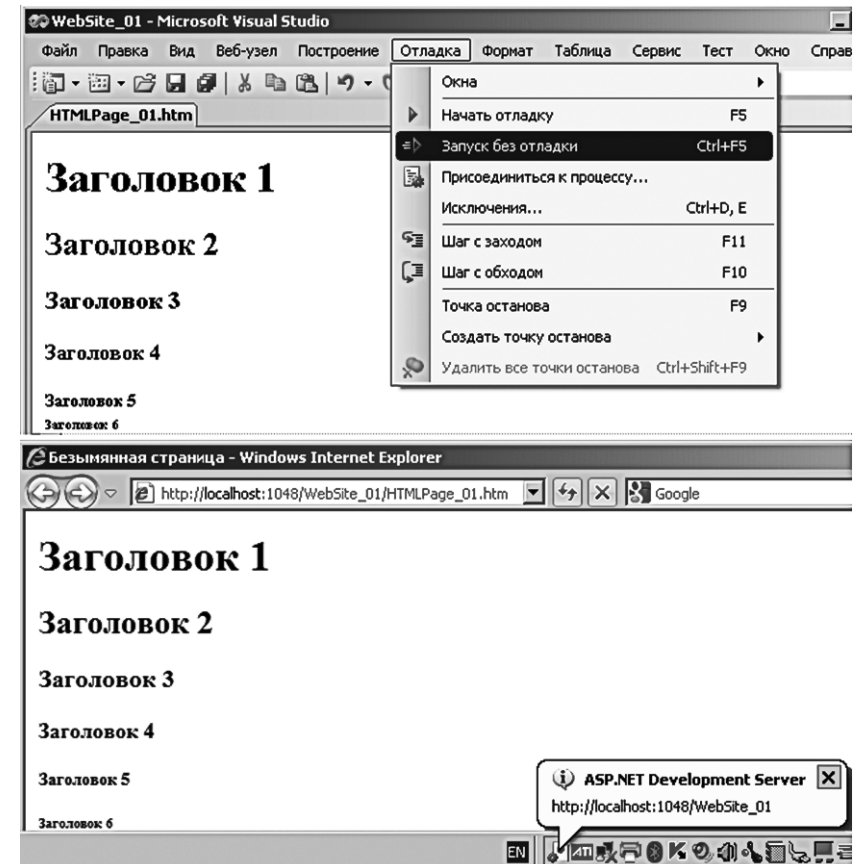


Рис.2.6. Проверка функционирования HTML-страницы

2.3. Парные и одиночные теги. Атрибуты. Тег перехода на новую строку

Большинство тегов в языке HTML являются парными, т. к. они воздействуют на часть содержимого HTML-страницы, заключенного между ними. **Парные теги** состоят из начального и конечного тега (рис. 2.7).

Начальный тег содержит имя тега, которое указывает браузеру способ отображения текста, заключенного между парными тегами. В начальном теге после имени тега могут быть указаны параметры, которые называются **атрибутами**, уточняют для браузера действие тега, отделяются один от другого пробелом и имеют вид:

название = «значение»

Если в начальном теге не указаны атрибуты, то считается, что действуют значения атрибутов, установленные «по умолчанию». **Конечный тег** состоит из имени тега, перед которым стоит символ слеш (/).

Кроме парных тегов бывают и **одиночные теги**. Примером может служить тег:

`
`

Этот тег указывает, что следующий за ним текст должен начинаться с новой строки. В одиночном теге перед закрывающей угловой скобкой стоит символ слеш (/).

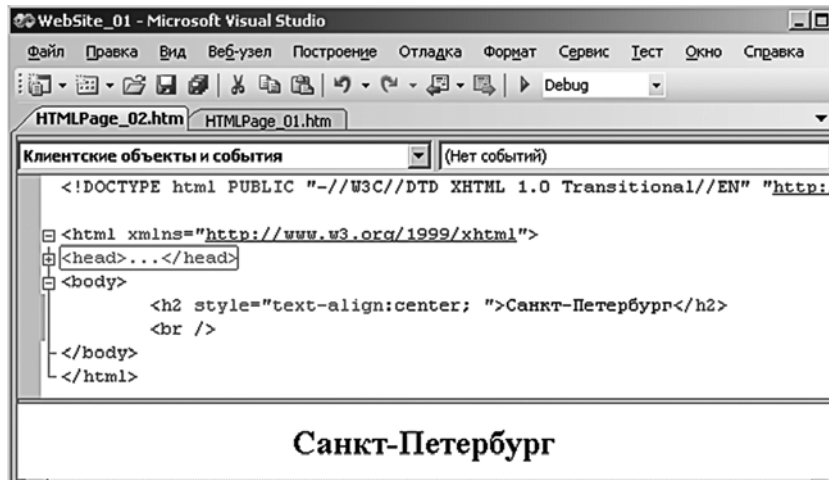


Рис. 2.7. Парные теги и одиночный тег

2.4. Оформление абзацев. Использование стилей

Если некоторый фрагмент текста HTML-страницы должен быть выделен в виде **абзаца**, то он заключается в парные теги `<p></p>`. Абзацы отделяются друг от друга пустой строкой.

Для оформления абзаца можно использовать атрибут «**style**», который позволяет задать вид шрифта, его размер, цвет и т. п. (рис. 2.8).

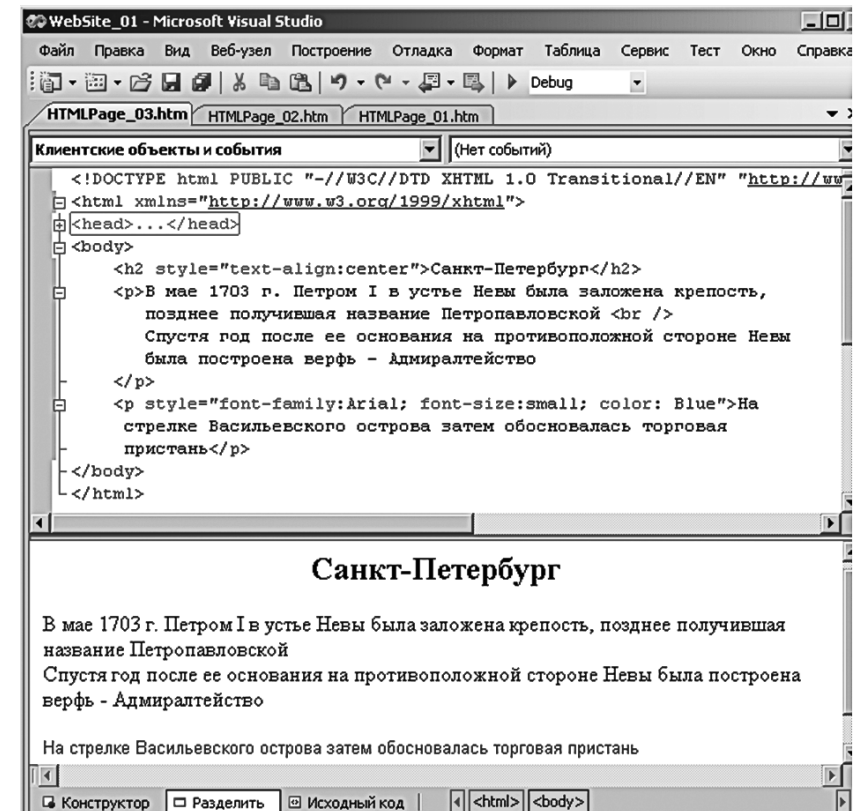


Рис. 2.8. Оформление абзацев и использование стиля

2.5. Оформление неупорядоченных (маркированных) списков

Неупорядоченные списки заключаются в теги . *Элементы списка* заключаются в теги . Параметр «list-style-type» может принимать значения (рис. 2.9):

- disc — маркируется кружком (по умолчанию),
- circle — маркируется окружностью,
- square — маркируется квадратом.

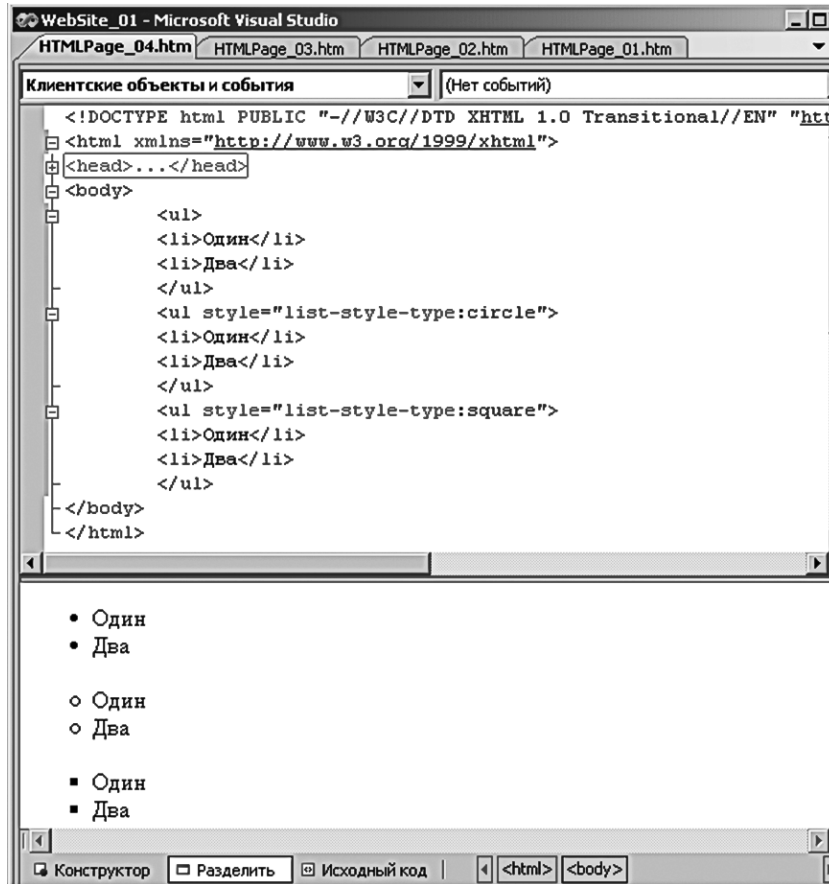


Рис. 2.9. Оформление неупорядоченных списков

2.6. Оформление упорядоченных (нумерованных) списков

Упорядоченные списки заключаются в теги . *Элементы списка* заключаются в теги . Параметр «list-style-type» может принимать значения (рис. 2.10):

- decimal — маркируется 1, 2, 3,.. (по умолчанию)
- upper-alpha — маркируется A, B, C,..
- upper-roman — маркируется I, II, III, IV,..

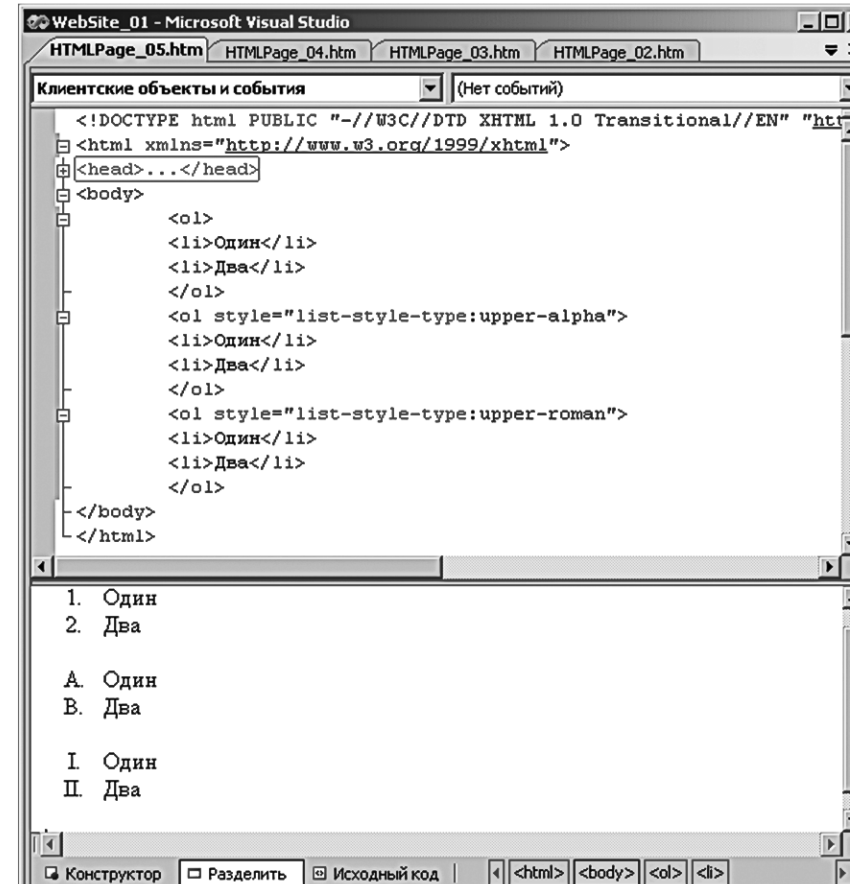


Рис. 2.10. Оформление упорядоченных списков

2.7. Оформление списков определений

Списки определений состоят из терминов и их определений и заключаются в теги `<dl>` `</dl>`. *Термины* заключаются в теги `<dt>` `</dt>`. *Определения терминов* заключаются в теги `<dd>` `</dd>` (рис. 2.11).

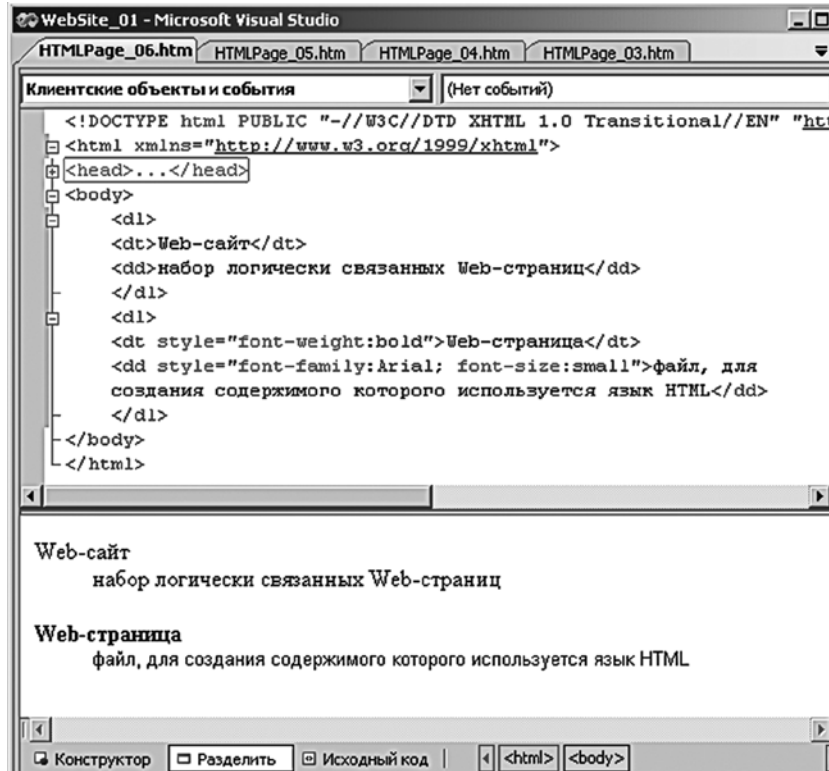


Рис. 2.11. Оформление списков определений

2.8. Горизонтальные линии

Горизонтальные линии предназначены для разграничения каких-либо фрагментов HTML-страницы и задаются с помощью тега `<hr/>` (рис. 2.12).

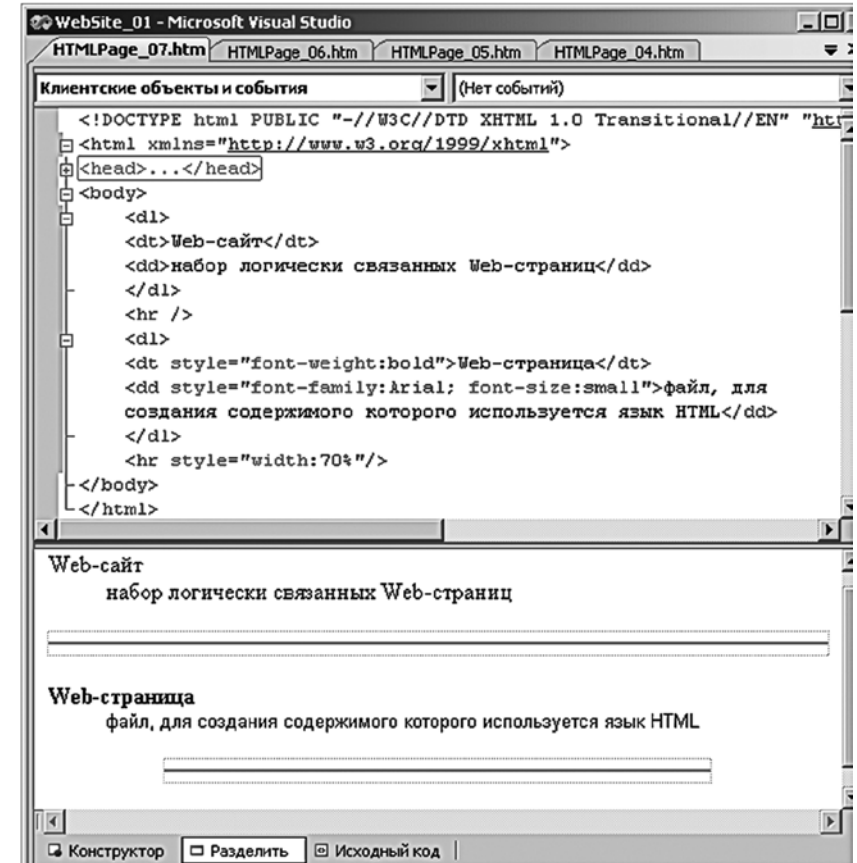


Рис. 2.12. Оформление горизонтальных линий

2.9. Комментарии

Комментарии заключаются в теги `<!-- -->` и содержат информацию, которая не отображается браузером (рис. 2.13).

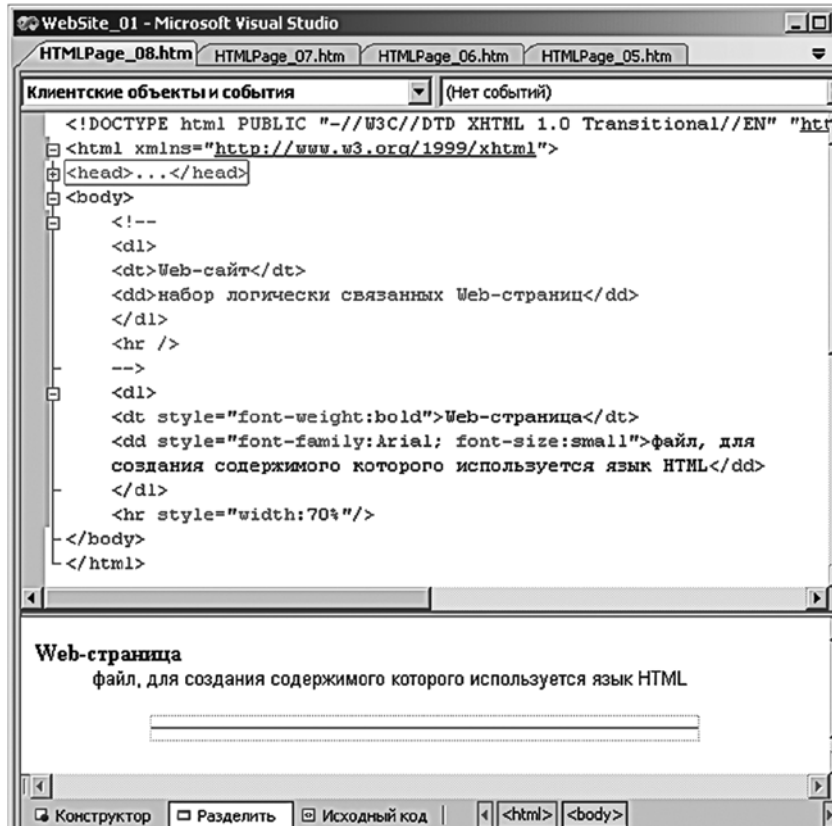


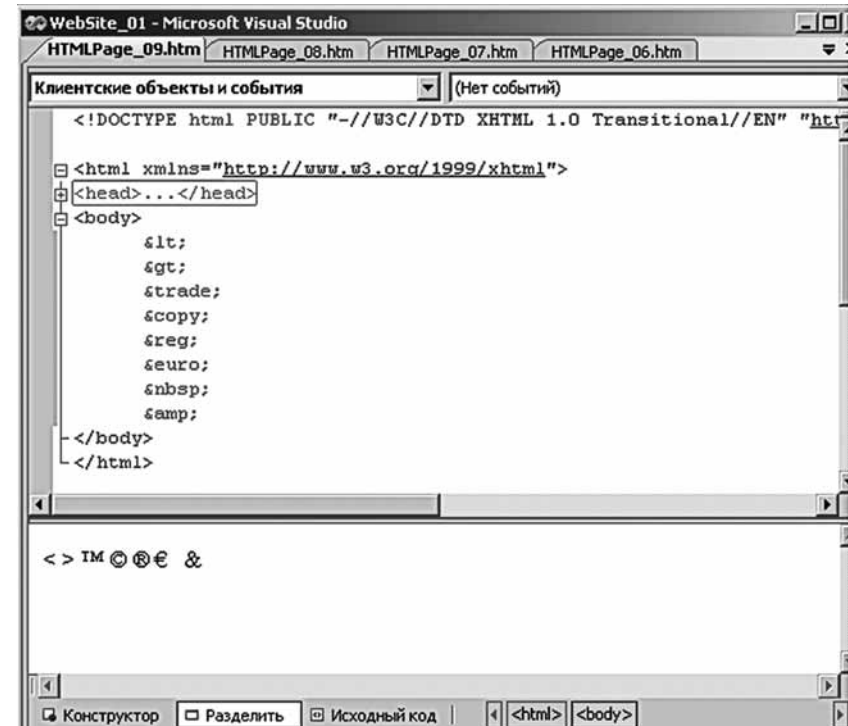
Рис. 2.13. Оформление комментария

2.10. Специальные символы

В языке HTML имеются символы, которые зарезервированы; например, «знак больше» (`>`). Для таких символов используется **специальная кодировка** (рис. 2.14):

Символ	Код	Имя
<	<	<
>	>	>
™	™	™
©	©	©
®	®	®
€	€	€
неразрывный пробел	 	
&	&	&

Рис. 2.14. Оформление специальных символов



**ГЛАВА 3
ГРАФИЧЕСКОЕ ОФОРМЛЕНИЕ HTML-СТРАНИЦЫ**

3.1. Форматы графических файлов

Любое изображение выводится на экран компьютера как множество прямоугольных пикселей. Способ, при помощи которого изображение сохраняется в файле, называется *графическим форматом*. В сети Internet используются и поддерживаются большинством Web-браузеров два графических формата:

- **jpeg** — формат для фотографических изображений (Joint Photographic Experts Group, объединенная группа экспертов в области фотографии),
- **gif** — используется для хранения рисунков и диаграмм (Graphics Interchange Format, формат графического обмена).

Добавим в Web-сайт папку с графическими файлами (рис. 3.1):

- с помощью программы «Проводник» создадим в папке Web-сайта вложенную папку с именем «Images» и скопируем в созданную папку заранее подготовленные графические файлы,
- для обновления информации в окне «Обозреватель решений» щелкнем на панели инструментов этого окна кнопку «Обновить».

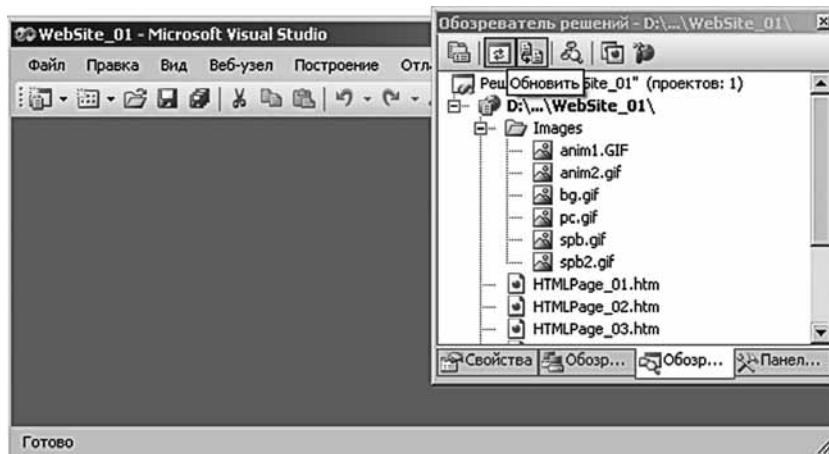


Рис. 3.1. Добавление в Web-сайт папки с графическими файлами

3.2. Вставка изображения на страницу

Для *вставки изображения* на HTML-страницу предназначен тег `` с атрибутами (рис. 3.2):

- **src** — имя графического файла,
- **alt** — надпись, которая появляется при наведении указателя мыши на изображение (при показе его на браузере),
- **align** — способ выравнивания рисунка относительно текста (**left** — рисунок выравнивается по левому краю, а текст располагается справа; **right** — рисунок выравнивается по правому краю, а текст располагается слева),
- **border** — ширина (в пикселях) рамки вокруг рисунка (0 — рамки нет)
- **hspace** — расстояние (в пикселях) между рисунком и текстом по горизонтали,
- **vspace** — расстояние (в пикселях) между рисунком и текстом по вертикали.

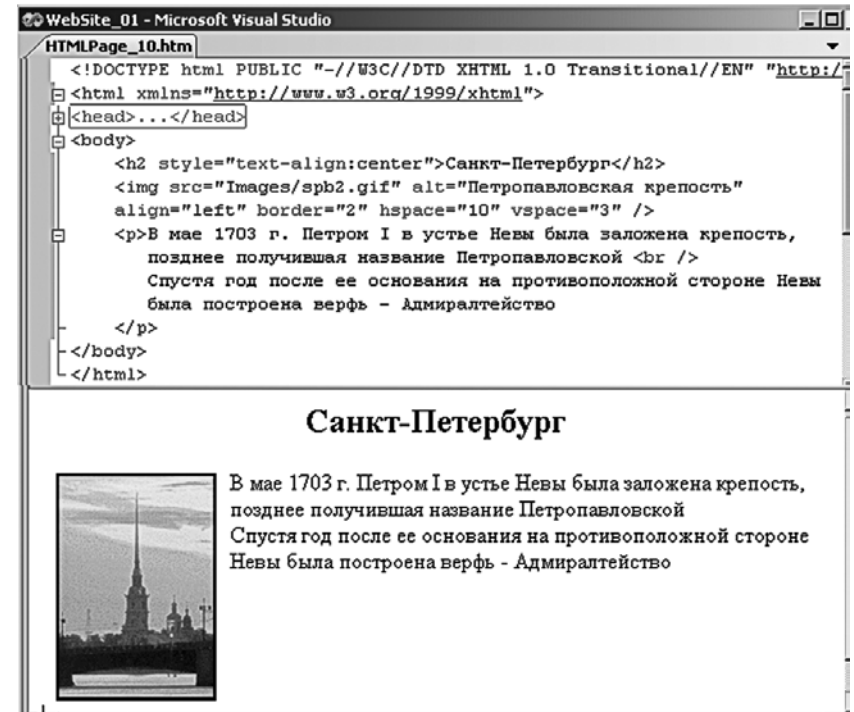


Рис. 3.2. Вставка изображения на страницу

3.3. Ссылка на внешнее изображение

Внешние изображения не отображаются на странице автоматически. Чтобы увидеть внешнее изображение, необходимо щелкнуть мышкой по **гиперссылке**.

Гиперссылка заключается между тегами <a> и может иметь вид (рис. 3.3):

- графического изображения,
- последовательности символов.

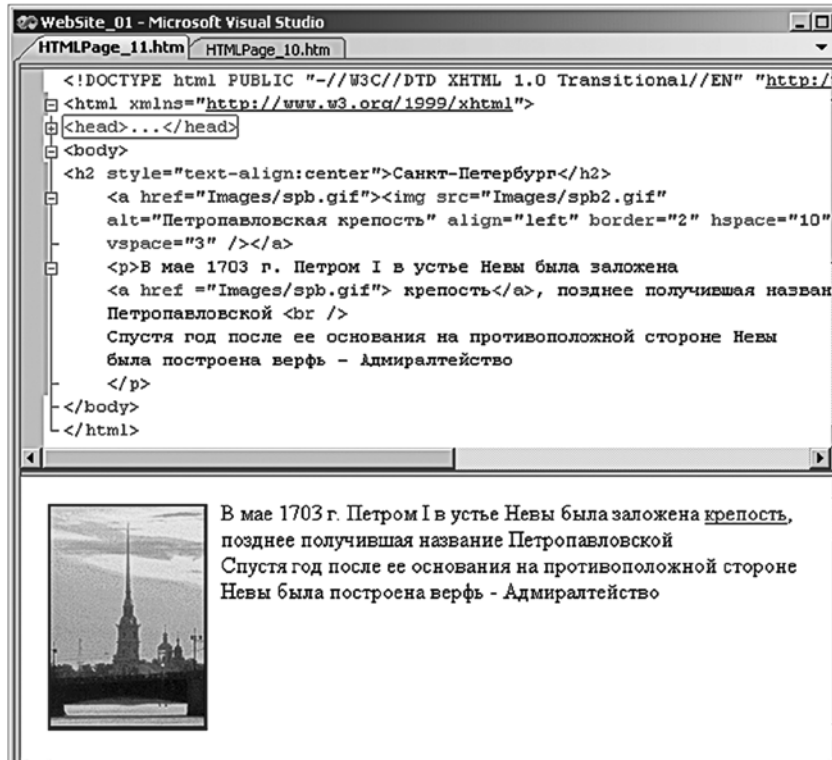


Рис. 3.3. Гиперссылки на внешнее изображение

Гиперссылки работают только при показе HTML-страницы на Web-браузере (рис. 3.4).

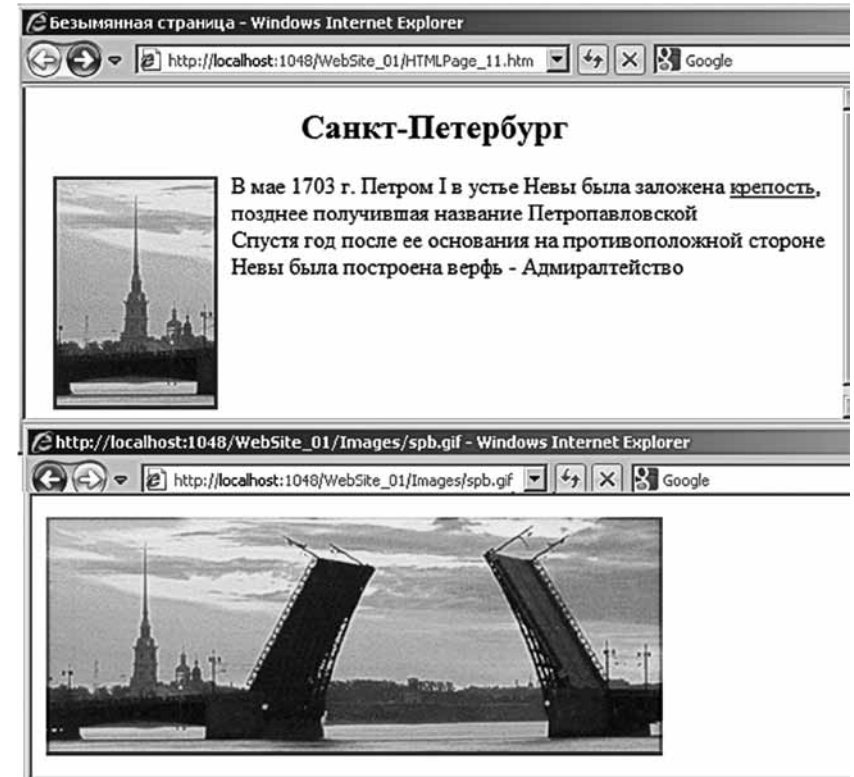


Рис. 3.4. Функционирование гиперссылки

3.4. Фоновое изображение

Фооновое изображение на HTML-странице задается с помощью атрибута «style» тега <body> (рис. 3.5).

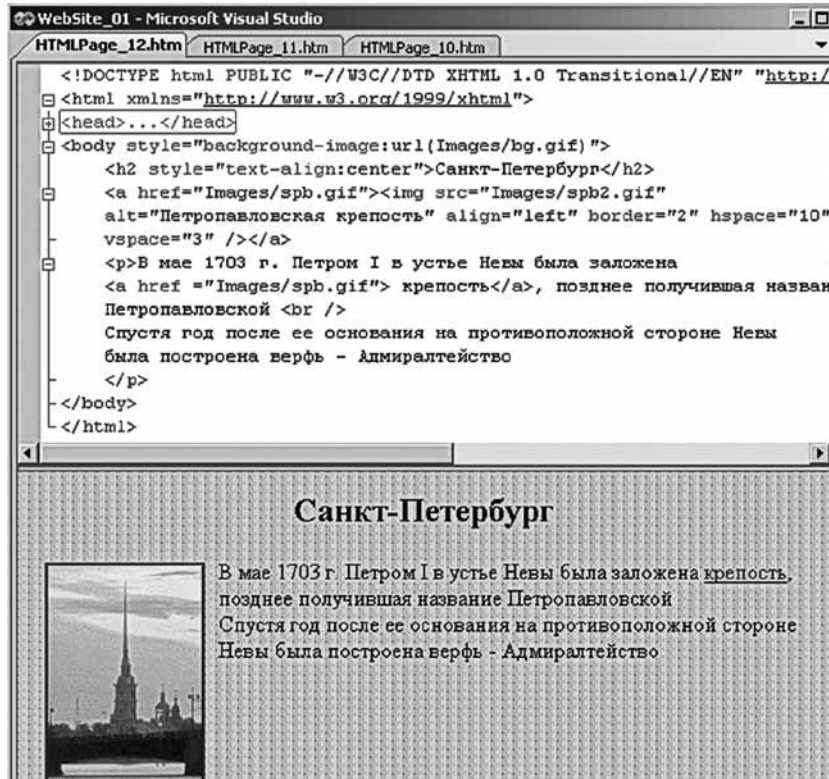


Рис. 3.5. Фооновое изображение на странице

3.5. Анимированное изображение

Некоторые изображения формата GIF являются анимированными. Их показ ничем не отличается от показа статических изображений (рис. 3.6).

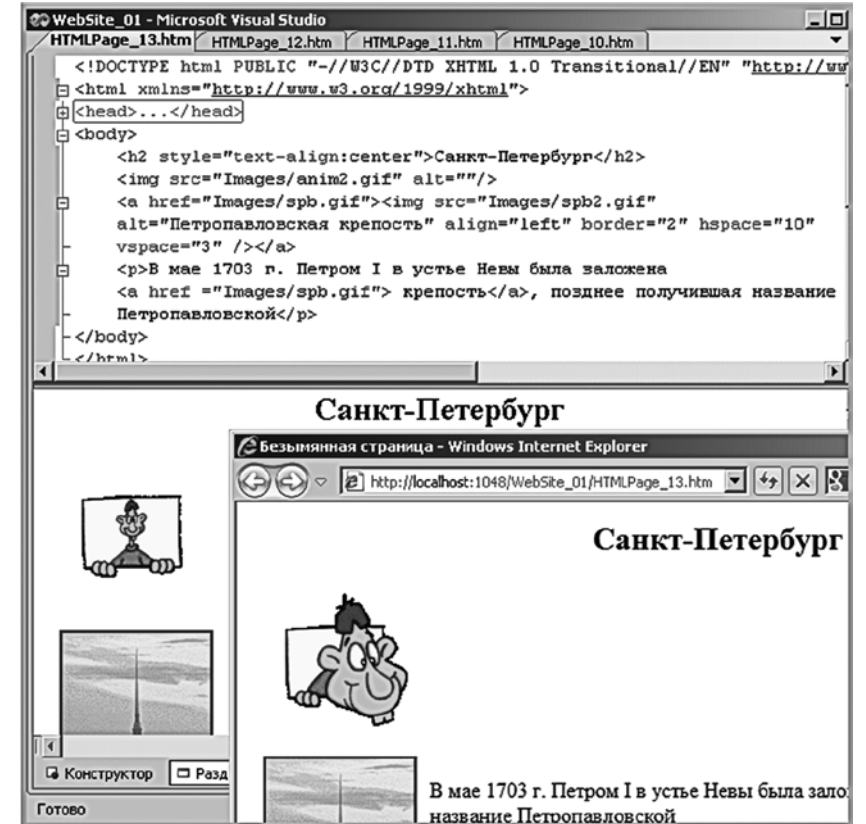


Рис. 3.6. Анимированное изображение на странице

**ГЛАВА 4
ГИПЕРССЫЛКИ**

**4.1. Понятие гипертекста.
Создание гиперссылок**

Гипертекст — это форма организации текста, которая позволяет создавать связи между различными частями одного текстового документа или частями других текстовых документов, позволяя читать документ нелинейным образом. Для создания связей между фрагментами текстовых документов используются *гиперссылки*. Гиперссылки работают только при показе HTML-страницы на браузере.

Гиперссылка создается с помощью парных тегов `<a>` `` (рис. 4.1):

- между тегами `<a>` `` указывается последовательность символов или ссылка на изображение, которые визуально выделяются в тексте,
- в начальном теге указывается атрибут `href`, значением которого является **адрес ресурса (URL)**, к которому нужно перейти по гиперссылке.

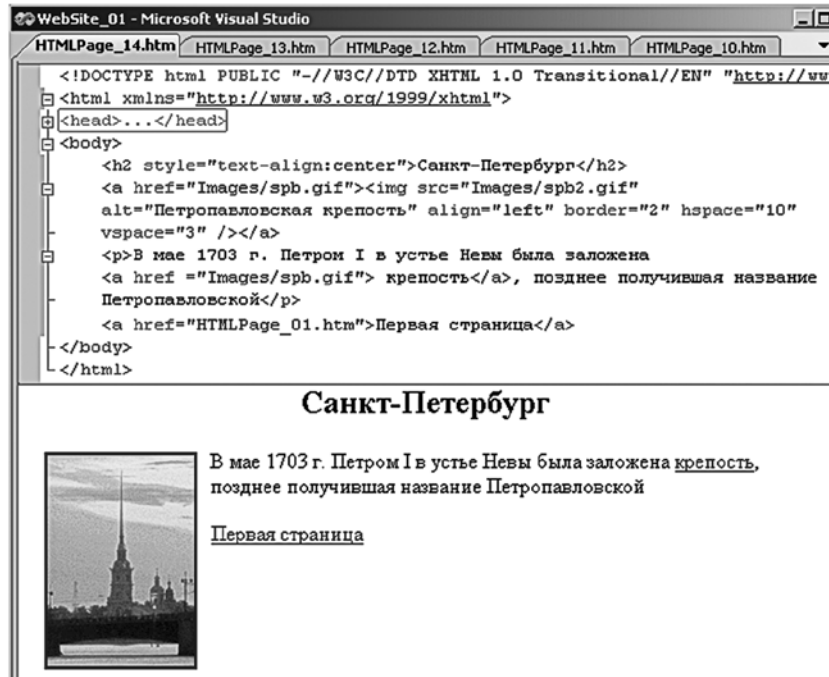


Рис. 4.1. Создание гиперссылок

4.2. Создание закладок

Теги `<a>` `` можно использовать для создания **закладок** в пределах данной HTML-страницы (рис. 4.2):

- *гиперссылка на закладку* создается при помощи тега `<a>` и атрибута `href`, в качестве значения которого указывается имя закладки с префиксом `«#»`,
- сама **закладка** создается в тексте HTML-страницы при помощи тега `<a>` и атрибута `name`.

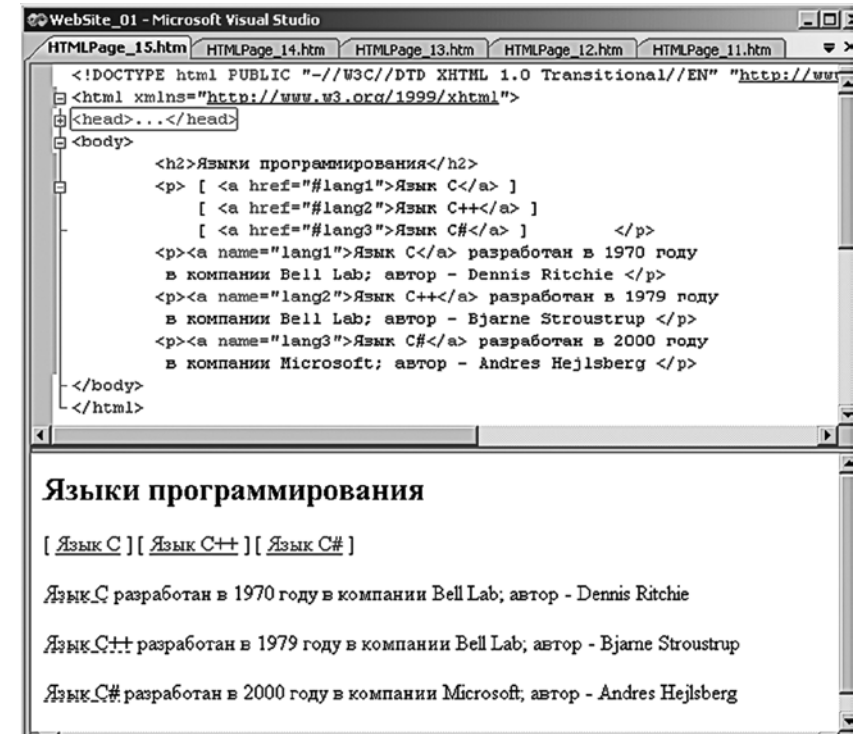


Рис. 4.2. Создание закладок

4.3. Создание карт навигации

Карта навигации — это рисунок, разделенный на логические области, каждая из которых является ссылкой на Web-ресурс (рис. 4.3).

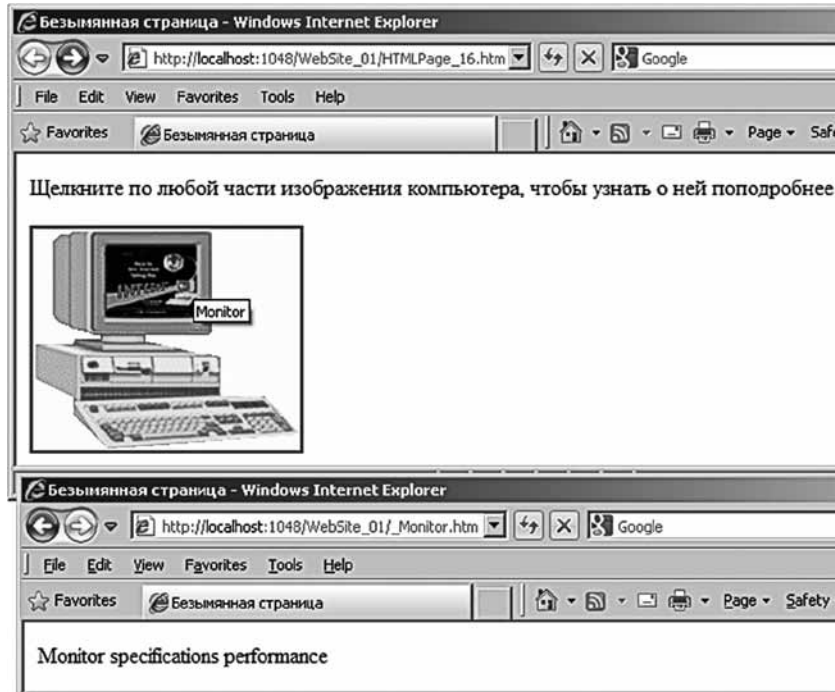


Рис. 4.3. Пример карты навигации

Карта навигации создается с помощью тегов `<map>` `</map>`. Начальный тег имеет атрибут `name` для идентификации карты. **Области карты** размечаются между тегами `<map>` `</map>` с помощью тега `<area>`, имеющего атрибуты:

- **href** — URL ресурса. Обычно это HTML-страница, соединенная с областью
- **shape** — геометрическая форма рисунка: **rect** — прямоугольник, который задается координатами левого верхнего и правого нижнего углов; **circ** — круг, который задается координатами центра и радиусом; **poly** — многоугольник, который задается набором координат вершин

- **coords** — координаты, описывающие расположение и размер области (координаты задаются в пикселях относительно левого верхнего угла изображения карты)
- **alt** — текстовое описание области
- **nohref** — определяет область, не имеющую гиперссылки

Связь рисунка с картой выполняется при помощи тега ``, имеющего атрибут `usemap`. Несколько областей **могут накладываться** друг на друга. При этом на щелчок реагирует та область, которая описана раньше (рис. 4.4).

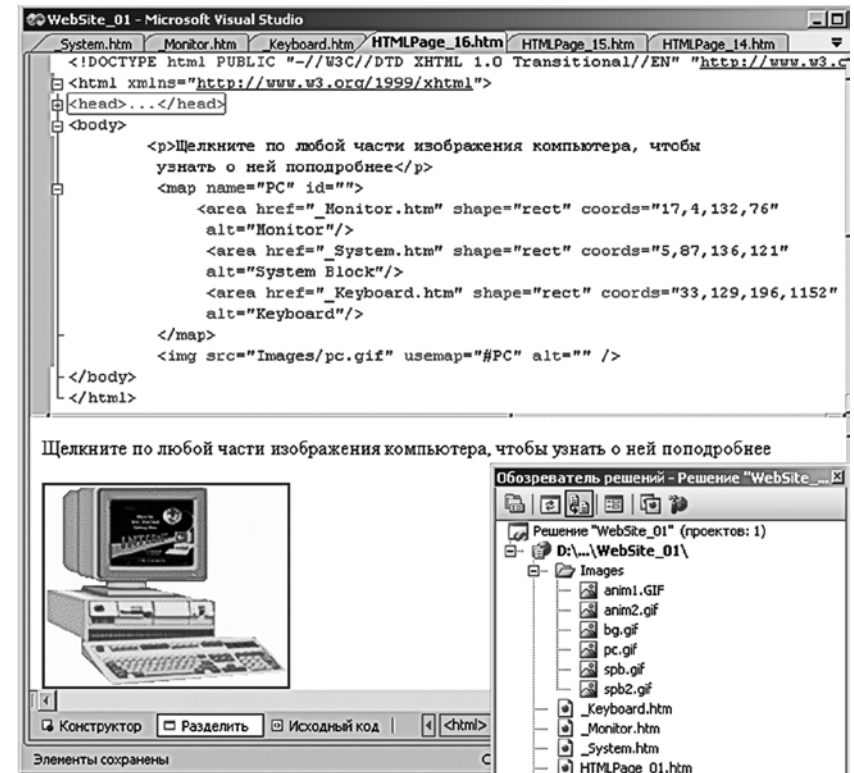


Рис. 4.4. Создание карты навигации

ГЛАВА 5 ТАБЛИЦЫ

5.1. Структура таблицы

Таблица имеет следующую разметку (рис. 5.1):

```
<table>
  <caption> Название таблицы </caption>
  <tr> <th> Заголовок 1 </th> <th> Заголовок 2 </th> </tr>
  <tr> <td> ячейка 11 </td><td> ячейка 12 </td></tr>
  <tr> <td> ячейка 21 </td><td> ячейка 22 </td></tr>
</table>
```

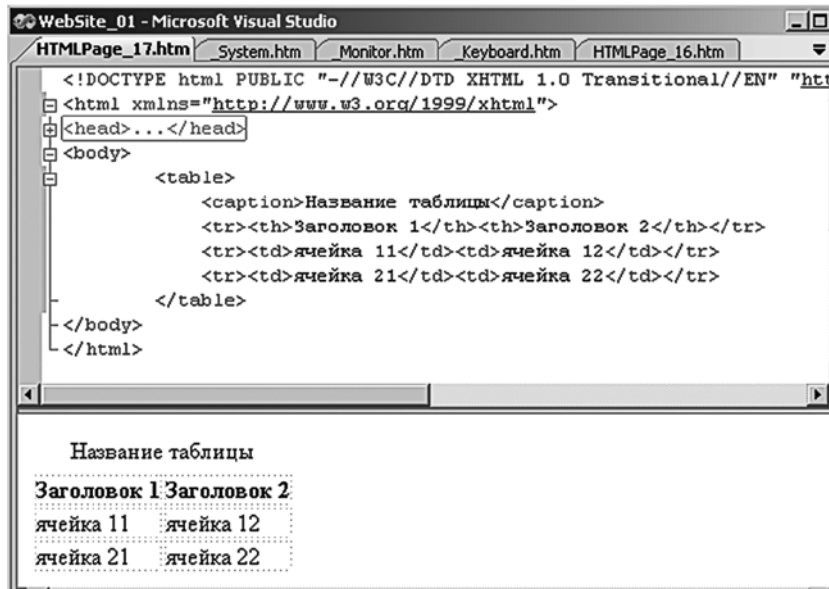


Рис. 5.1. Разметка таблицы

5.2. Атрибуты тега <table>

Тег **<table>** описывает таблицу в целом и имеет следующие атрибуты (рис. 5.2):

- **align** — выравнивание таблицы на странице относительно окружающего текста (**left**, **right** — таблица размещается по левой (правой) границе текста, который обтекает страницу, **center** — текст располагается сверху и снизу страницы),
- **background** — значением является URL фонового изображения,
- **bgcolor** — цвет фона таблицы,
- **border** — ширина внешних линий рамки таблицы в пикселях (по умолчанию 0, внешние и внутренние границы таблицы отсутствуют; значение атрибута не влияет на толщину внутренних линий),
- **bordercolor** — цвет линий рамки,
- **cellpadding** — отступ от границ ячейки таблицы до данных, размещенных в ячейке (в пикселях),
- **cellspacing** — ширина расстояния между соседними ячейками таблицы (в пикселях),
- **height** — высота таблицы в процентах (указывается «%») или в пикселях,
- **width** — ширина таблицы в процентах (указывается «%») или в пикселях,
- **frame** — вид границы (**border**, **box** — границы по всем краям таблицы, **above** — граница по верхнему краю таблицы, **below** — граница по нижнему краю таблицы, **hsides** — граница по верхнему и нижнему краю таблицы, **lhs** — граница по левому краю таблицы, **rhs** — граница по правому краю таблицы, **vsides** — граница по левому и правому краю таблицы, **void** — нет границ).

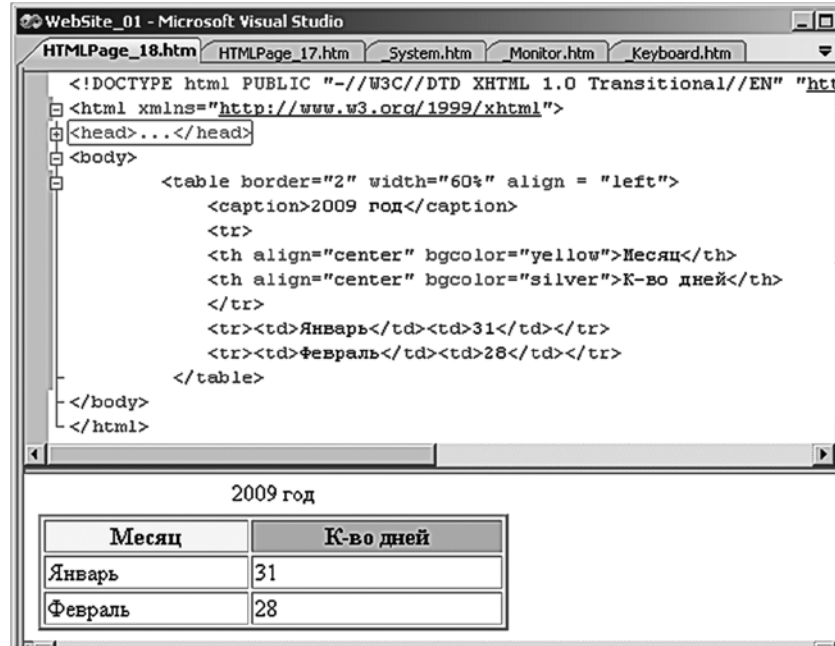


Рис. 5.2. Атрибуты тега <table>

5.3. Атрибуты тега <tr> и тега <td>

Тег <tr> описывает строку таблицы и имеет атрибуты (рис. 5.3):

- **align** — выравнивание ячеек строки по,
- **valign** — выравнивание ячеек строки по вертикали,
- **bgcolor** — цвет фона строки.

Тег <td> описывает ячейку таблицы и имеет атрибуты:

- **align** — выравнивание содержимого ячейки по горизонтали,
- **valign** — выравнивание содержимого ячейки по вертикали,
- **bgcolor** — цвет ячейки,
- **width** — ширина ячейки в процентах (указывается «%») или в пикселях,
- **height** — высота ячейки в процентах (указывается «%») или в пикселях,
- **colspan** — объединение ячеек в строке,
- **rowspan** — объединение ячеек в столбце.

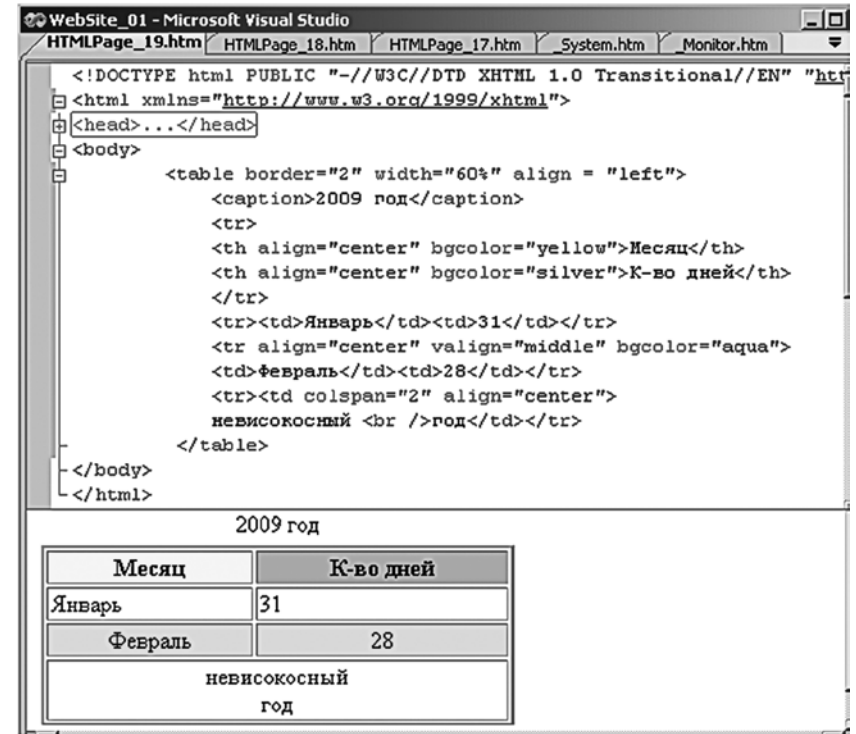


Рис. 5.3. Атрибуты тегов <tr> и <td>

**ГЛАВА 6
СТИЛОВОЕ ОФОРМЛЕНИЕ HTML-СТРАНИЦ**

6.1. Понятие стиля

При создании HTML-страниц можно использовать различные параметры форматирования, которые определяют внешний вид HTML-страниц. Например, при обсуждении некоторой проблемы можно использовать различные цвета для вопросов и ответов. Если список вопросов и ответов нужно будет дополнить, то придется вставлять тег `` перед каждым новым вопросом. Если в дальнейшем потребуется изменить цвет вопросов, то нужно будет внести изменения в каждый тег `` (рис. 6.1).

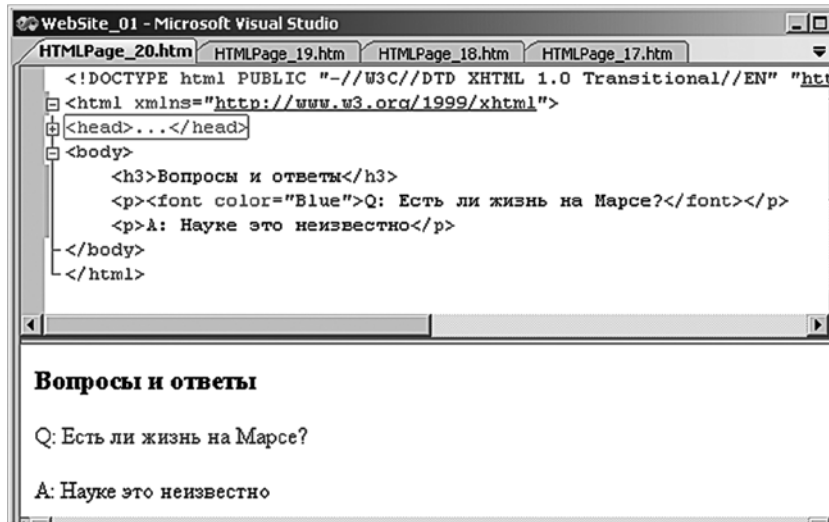


Рис. 6.1. Использование тега `` для форматирования текста

Альтернативным способом установления связи между контентом HTML-страницы и его отображением является определение стиля с помощью тега `<style>`. На данной странице стиль определен для тега `<p>`. Каждый раз при появлении в теле HTML-страницы тега `<p>` с атрибутом `«class»`, равным `«question»`, текст абзаца будет отображаться синим цветом (рис. 6.2).

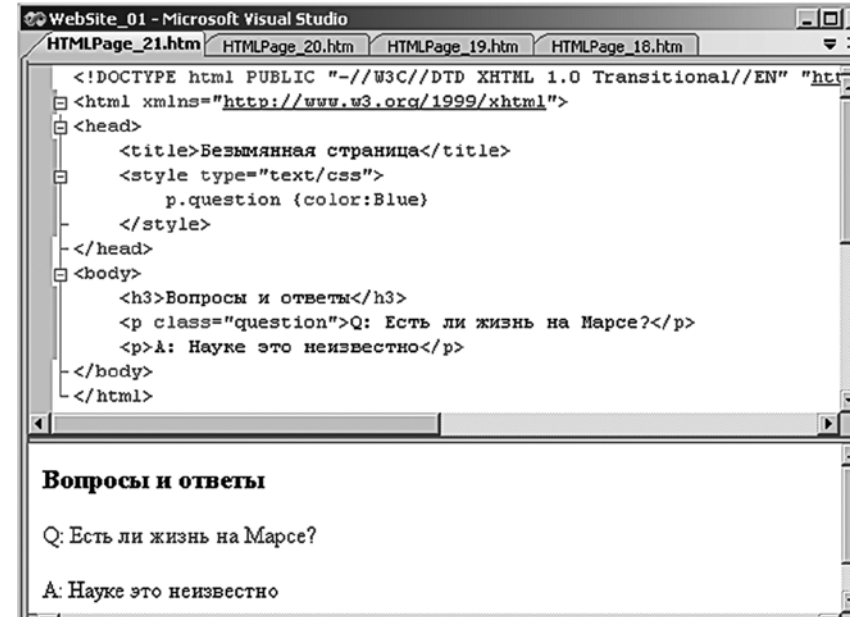


Рис. 6.2. Определение стиля с помощью тега `<style>`

**6.2. Внутренняя таблица стилей.
Локальный стиль**

Набор стилей, которые определяются в заголовке HTML-страницы с помощью парных тегов `<style>`, называется *внутренней таблицей стилей*. Областью действия внутренней таблицы стилей является вся HTML-страница.

На странице HTMLPage_22.htm (рис. 6.3):

- для тега `<p>` определен класс *стиля* с именем `«question»`. Каждый раз при появлении тега `<p>` с атрибутом `«class»`, равным `«question»`, текст абзаца будет отображаться синим цветом,
- стиль, определенный для тега `<h3>`, действует на *все* экземпляры тега.

Определение стиля:

- начинается с *указания тега*, к которому он применяется (возможно, с указанием класса стиля),
- содержит перечень *свойств стиля* в виде имен `color`, `font-family`, `background`, `font-weight` и т. п.,
- для свойств стиля указываются их *значения*, отделенные от свойств двоеточием (:),

- пары «*свойство : значение*» отделяются друг от друга точкой с запятой,
 - и заключаются в фигурные скобки {}.
- Можно применять стиль непосредственно к тегу, например:
<h1 style=»font-weight:bold; font-size:14pt; color:orange»>.
- Такой стиль называется *локальным*:
- в его определении пары «*свойство : значение*» вместо фигурных скобок заключаются в двойные кавычки,
 - он воздействует непосредственно на экземпляр тега и отменяет действие стиля, определенного в таблице стилей.

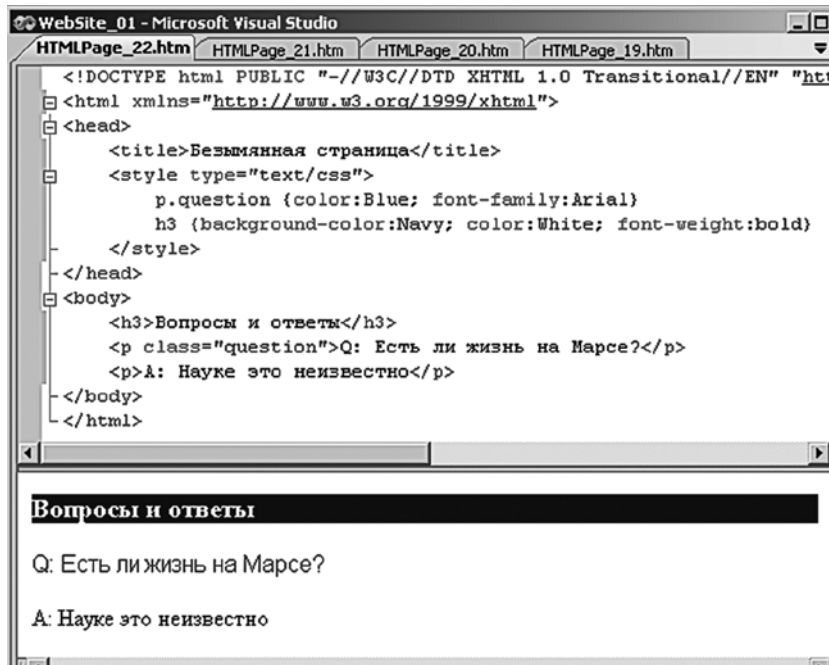


Рис. 6.3. Определение класса стиля и стиля для всех тегов

6.3. Внешняя таблица стилей

Добавим в Web-сайт *внешнюю таблицу стилей* (в виде файла с расширением *css*) с таким же содержимым, как у приведенной выше внутренней таблицы стилей (рис. 6.4).

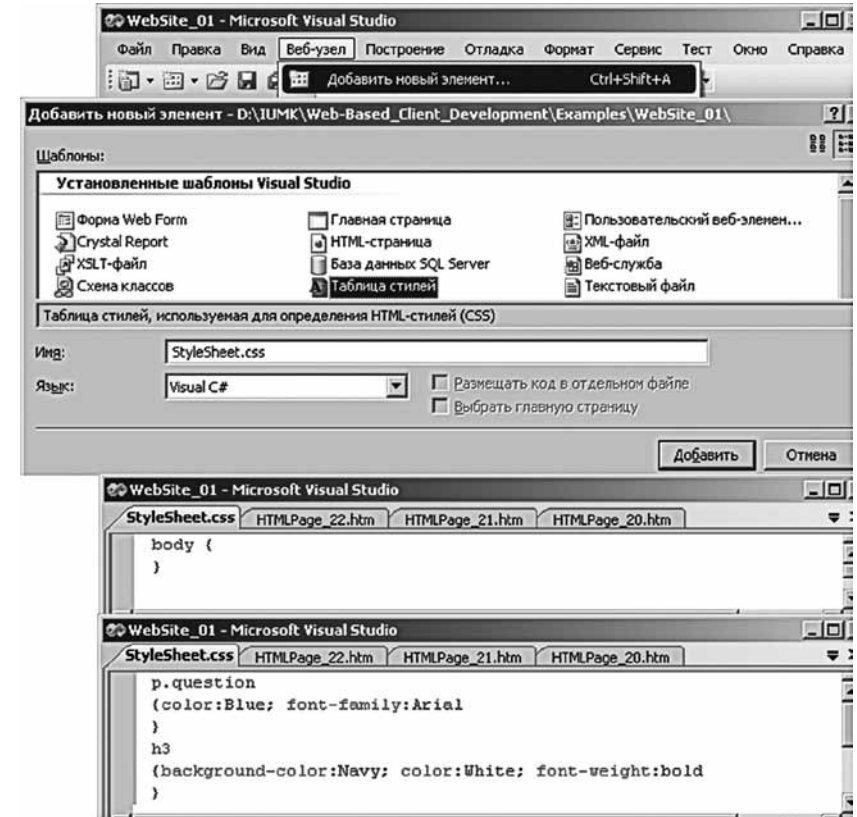


Рис. 6.4. Добавление в Web-сайт внешней таблицы стилей

Если щелкнуть правой кнопкой мыши внутри определения стиля, то на экран будут выведено контекстное меню с командой «Построить стиль». Выбор этой команды выводит на экран диалоговое окно визуального дизайнера стиля, позволяющего в диалоговом режиме задать все необходимые параметры стиля (рис. 6.5).

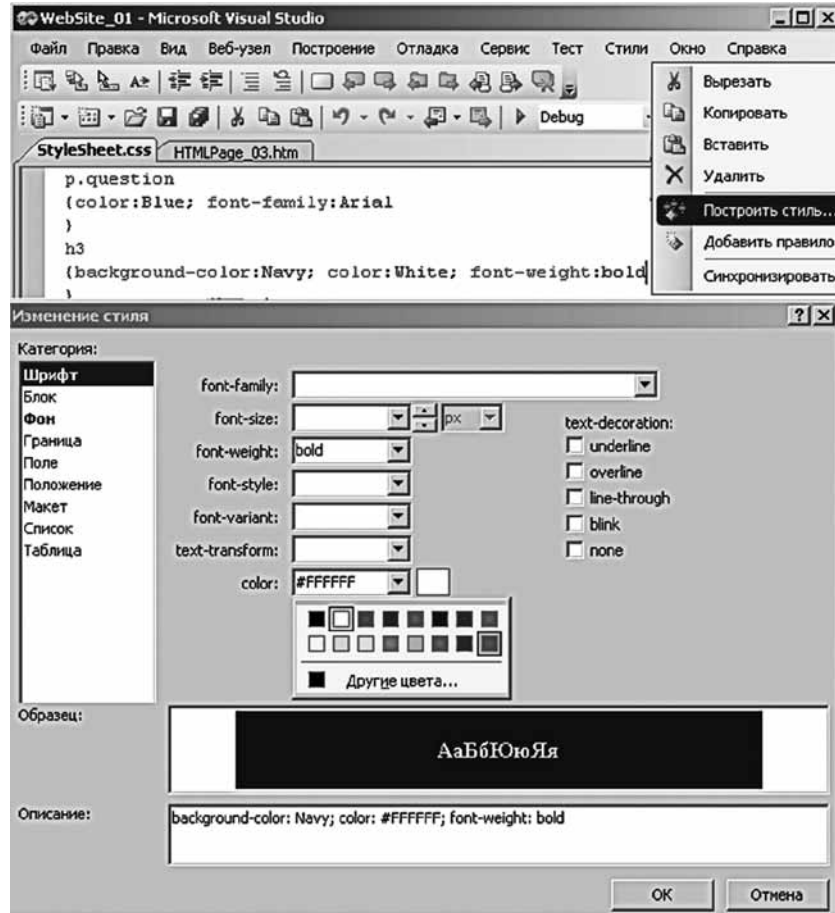


Рис. 6.5. Вызов визуального дизайнера стиля

Внешние таблицы стилей можно использовать для множества HTML-страниц, которые должны быть оформлены одинаково. Для связывания HTML-страницы с файлом внешней таблицы стилей используется тег `<link>`, который размещается в заголовке HTML-страницы и содержит следующие атрибуты (рис. 6.6):

- **rel** — указывает на то что к HTML-странице должна быть применена таблица стилей,
- **type** — определяет тип содержимого таблицы стилей,
- **href** — указывает имя файла, содержащего таблицу стилей.

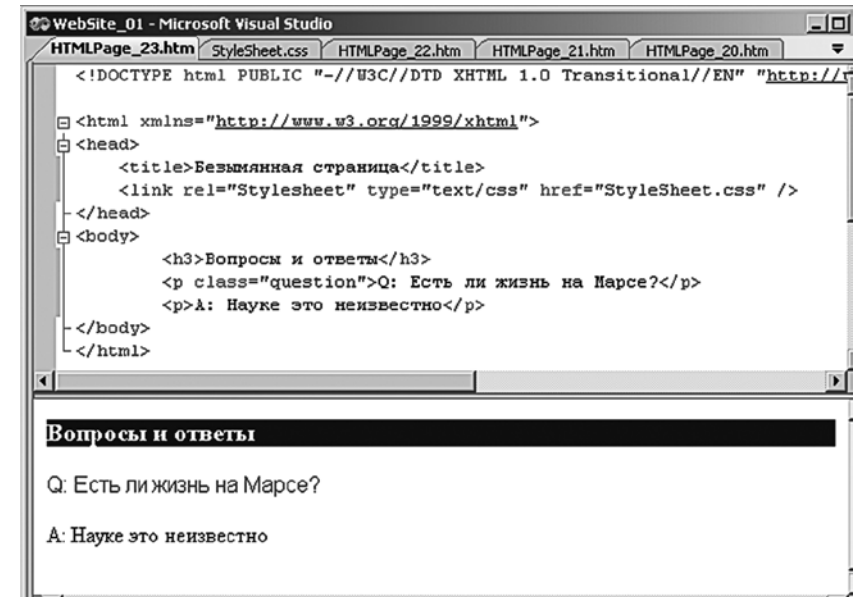


Рис. 6.6. Связывание страницы с внешней таблицей стилей

6.4. Индивидуальные стили для тегов

Индивидуальный стиль — это стиль, применяемый к конкретному экземпляру тега HTML-страницы. Чтобы создать индивидуальный стиль для тега (рис. 6.7):

- в таблице стилей нужно указать **имя тега** (например, **p**) и его **идентификатор** (например, **copyright**), отделенные друг от друга символом «#»,
- чтобы стиль подействовал на экземпляр тега, в теле HTML-страницы нужно воспользоваться атрибутом «**id**».

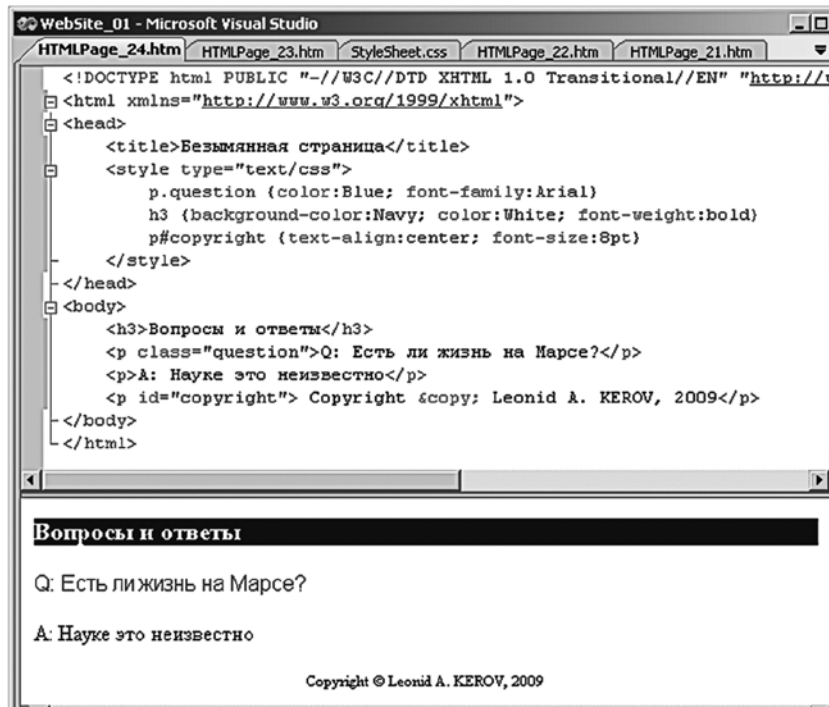


Рис. 6.7. Создание индивидуального стиля для тега

6.5. Стили гиперссылок

Стили гиперссылок определяют параметры форматирования для текстовых гиперссылок (рис. 6.8):

- **link** — еще не просмотренная гиперссылка,
- **visited** — просмотренная гиперссылка,
- **active** — активная гиперссылка,
- **hover** — гиперссылка, на которую указывает курсор мыши.

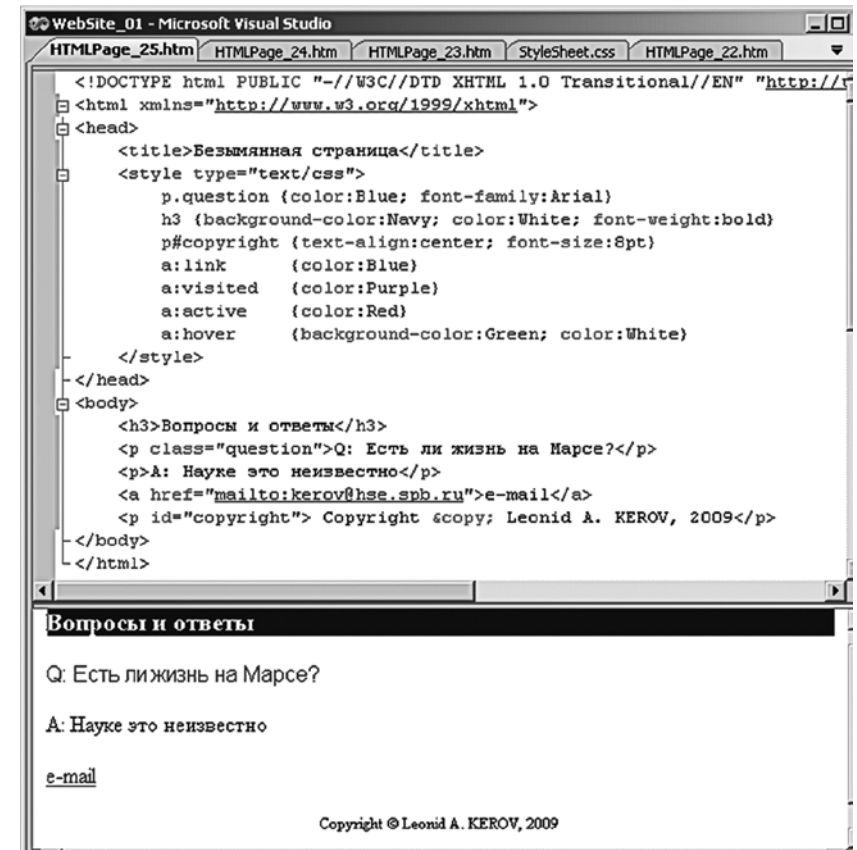


Рис. 6.8. Стили гиперссылок

6.6. Пользовательские теги стилей

В языке HTML имеются *два типа тегов*:

- **теги уровня блока** (<p>, <body>, <table>, <h1>, ..., <h6>) начинают текст с новой строки и могут содержать другие теги,
- **встраиваемые теги** (, <i>, ,) не начинают текст с новой строки и могут содержать только текст или другие встраиваемые теги.

Пользовательские теги уровня блока формируются с помощью тега <div>:

- имя стиля включает в себя имя тега «div»,
- символ точки (.),
- имя пользовательского тега (например, **question**).

Пользовательские встраиваемые теги формируются с помощью тега :

- имя стиля включает в себя имя тега «span»,
- символ точки (.),
- имя пользовательского тега (например, **qa**).

Пользовательские теги стилей фактически являются *классами стилей*, связанных с тегами «div» и «span». Они применяются так же, как классы стилей: имя класса указывается в качестве значения атрибута «class» соответствующего тега, «div» или «span» (рис. 6.9).

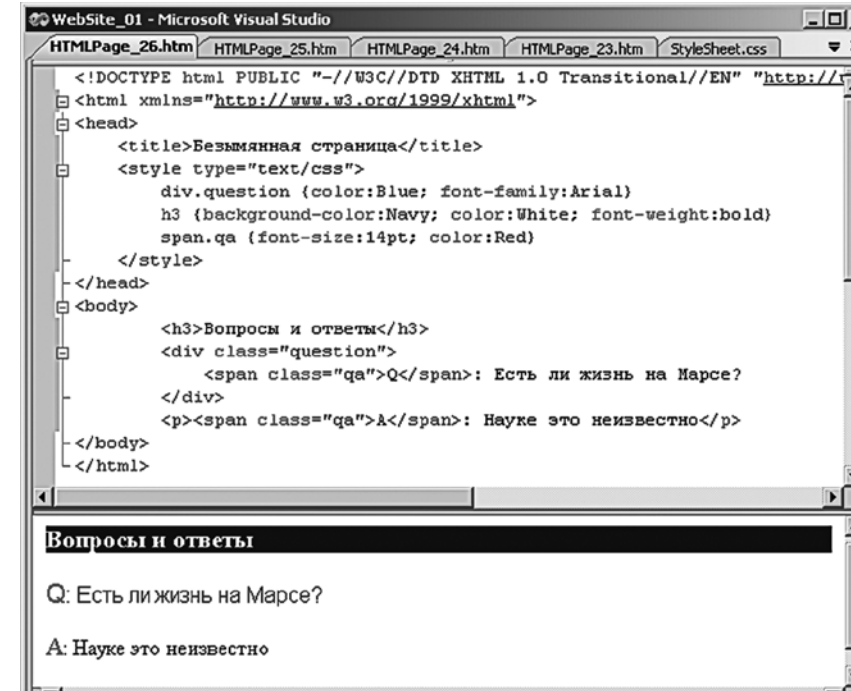


Рис. 6.9. Пользовательские теги стилей

6.7. Использование стилей для форматирования текста

Главное назначение таблиц стилей — форматирование текста. Можно выделить несколько категорий текстовых стилей (рис. 6.10).

Стили шрифтов:

- **font-style** — стиль шрифта: **normal** (по умолчанию), **italic**,
- **font-weight** — толщина шрифта: **normal** (по умолчанию), **light**, **bold**, **bolder**,
- **font-size** — размер шрифта задается одной из единиц: пункт (**pt**), дюйм (**in**), сантиметр (**cm**), пиксель (**px**); например, **font-size:12pt**,
- **font-family** — семейство шрифта: **Times New Roman**, **Arial**, **Courier** и т. д.,
- **font-variant** — вариант шрифта **normal** (по умолчанию), **small-caps** (отображает строчные буквы как заглавные меньшего размера).

Стили оформления шрифтов:

- **color** — цвет текста; например: **color:yellow**
- **background-color** — цвет фона для текста; например: **background-color:gray**,
- **background-image** — изображение для фона текста; например: **background-image:url(spb.gif)**,
- **background-repeat** — расположение фонового рисунка; например: **background-repeat:repeat-x**, **background-repeat:repeat-y**, **background-repeat:no-repeat**.

Стили табуляции:

- **letter-spacing** — расстояние между символами задается одной из единиц: пункт (pt), дюйм (in), сантиметр (cm), пиксель (px); например, **letter-spacing:4px**,
- **text-indent** — размер отступа для первой строки строки абзаца задается одной из единиц: pt, in, cm, px; например, **text-indent:2cm**.

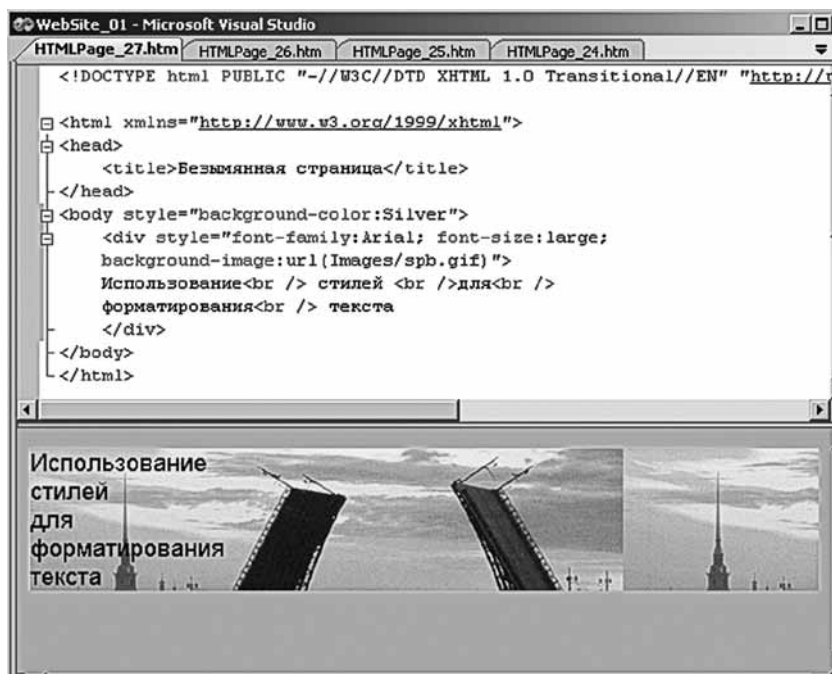


Рис. 6.10. Использование стилей для форматирования текста

ГЛАВА 7 РАЗРАБОТКА ВИЗУАЛЬНОГО ДИЗАЙНА САЙТА

7.1. HTML-разметка домашней страницы

Создадим в Web-сайте вложенную папку Images и добавим в нее следующие графические файлы:

- e_Office.jpg — некоторое изображение, которое является иллюстрацией контента в целом
- kerov_02.jpg — фотографиями автора сайта
- Lak_Header.GIF — это файл с логотипом сайта. Это может быть любая картинка в формате GIF или JPG, которую можно нарисовать в графическом редакторе или получить с помощью механизма захвата экрана. Необходимо знать ее размеры. Для этого графический файл можно открыть, например, в графическом редакторе Paint, и из меню «Рисунок» выбрать команду «Атрибуты». В данном примере размеры картинки составили 465 пикселей в длину и 184 пикселей в высоту.
- Lak_HeaderSlice.GIF — эта картинка получается следующим образом: у картинки Lak_Header.GIF нужно отрезать справа полоску шириной в один пиксель. Эта картинка нужна для того, чтобы можно было создать эффект «растягивания» по ширине изображения с логотипом сайта.

Добавим в Web-сайт страницу *index.htm* и выполним ее разметку ее «тела» с помощью тегов таблицы (рис. 7.1):

```

<body>
<table width=»100%» cellpadding=»0» cellspacing=»0»>
<tr> <!-- Верхний колонтитул -->
    <td width=»465px» height=»184px»>
        <img src=»Images/Lak_Header.GIF» alt=»» />
    </td>
    <td style=»background-image:url (Images/Lak_
HeaderSlice.GIF)»></td>
</tr>
<tr> <!-- Фотография автора и название сайта -->
    <td width=»465px»><img src=»Images/Lak_2.jpg» alt=»»
/></td>
    <td ><h2>Web-сайт Леонида А. КЕРОВА</h2>
        <div class=»header»><a href=»about.htm»>Об ав-
торе</a></div>
    </td>
</tr>
<tr> <!-- Название курса -->
    <td colspan = «2» align=»center»>
        <h3>Учебные материалы по курсу
            «Компьютерные технологии офисной деятель-
ности»
        </h3>
    </td>
</tr>
<tr> <!--Граф.иллюстрация курса и ссылки на страницы с кон-
тентом -->
    <td width=»465px»><img src=»Images/e_Office.jpg»
alt=»» />
    </td>
    <td >
        <div class=»header»><a href=»CompTech/01.htm»>
Тема 1. Основные понятия комп.технологий</a></div><br />
        <div class=»header»><a href=»CompTech/02.htm»>
Тема 2.Создание док.в редакторе Word 2007</a></div><br />
    </td>
</tr>
</table>
</body>

```



Рис. 7.1. HTML-разметка домашней страницы

Таблица состоит из четырех строк. Первая строка служит для оформления графического колонтитула страницы. В первой ячейке этой строки содержится логотип сайта, а во второй ячейке — полоска шириной в один пиксель. Эффект «растягивания» логотипа получается благодаря указанию параметра «*width*» для внешней таблицы; нулевые значения параметров «*cellspacing*» и «*cellpadding*» позволяют устранить зазор между картинками

Вторая строка в первой ячейке содержит фотографию автора курса, а во второй строке — указание на владельца сайта и ссылку на его авторскую страницу. В третьей строке две ячейки объединены в одну. Эта строка содержит название контента сайта, выровненное по центру страницы. В первой ячейке четвертой строки содержится графическая

иллюстрация контента сайта, а во второй ячейке содержатся ссылки на страницы с разделами контента.

Для визуального оформления страницы *index.htm* добавим в сайт внешнюю таблицу стилей и свяжем ее с домашней страницей:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Везьянная страница</title>
  <link rel="stylesheet" type="text/css"
href="StyleSheet.css" />
</head>
<body>
...
</body>
</html>
```

Для заполнения внешней таблицы стилей воспользуемся визуальным дизайнером стиля (Рис.7.2):

```
body {font-family:Times New Roman Cyr; text-align:justify;
color:Navy;}
h2 {font-family: Arial; color: #999999;}
h3 {font-family: Arial; background-color: #000099;
color: White;}
h4 {font-family: Arial; background-color: #808080;
color: White;
font-size: larger;font-weight: bold;}
h5 {font-family: Arial; background-color: #EAEAEA;
color: #000099;
font-size: 12pt; font-weight: bold;}
div.bookmark {position:relative; top:0px; left:0px; margin-
left:50px;
font-family:Arial; }
div.content {font-family:Times New Roman Cyr; text-
align:justify;
color:Navy;}
div.comment {font-family: Arial; font-size: small; color:
#FF0000;}
div.program {font-family:'Courier New';font-
size:small;color:#800000;}
div.header {font-family: Arial;font-size: 11pt;font-
weight: bold;}
div#header {padding: 0px; margin: 0px; width: 100%; height:
184px;
background-image:url (Images/Lak_HeaderSlice.
gif);}
div#header2 {padding: 0px; margin: 0px; width: 465px;
height: 184px;
```

```
background-image: url (Images/Lak_Header.gif);}
div#header3 {position: relative;top: 153px;left:
245px;width: 100%;
padding: 2px 2px 2px 2px;font-family: 'Courier New';
font-size: smaller;color: #333333;}
span.term {font-family: Courier New;font-weight: bold;
color: #000066;
background-color: #FFFFCC; font-size: 11pt;}
a:link { color: Blue; }
a:visited { color:Purple; }
a:hover{color: #000099; background: #FFFFCC;}
```

Внешний вид домашней страницы сайта, оформленной с использованием внешней таблицы стилей, показан на рис. 7.3.

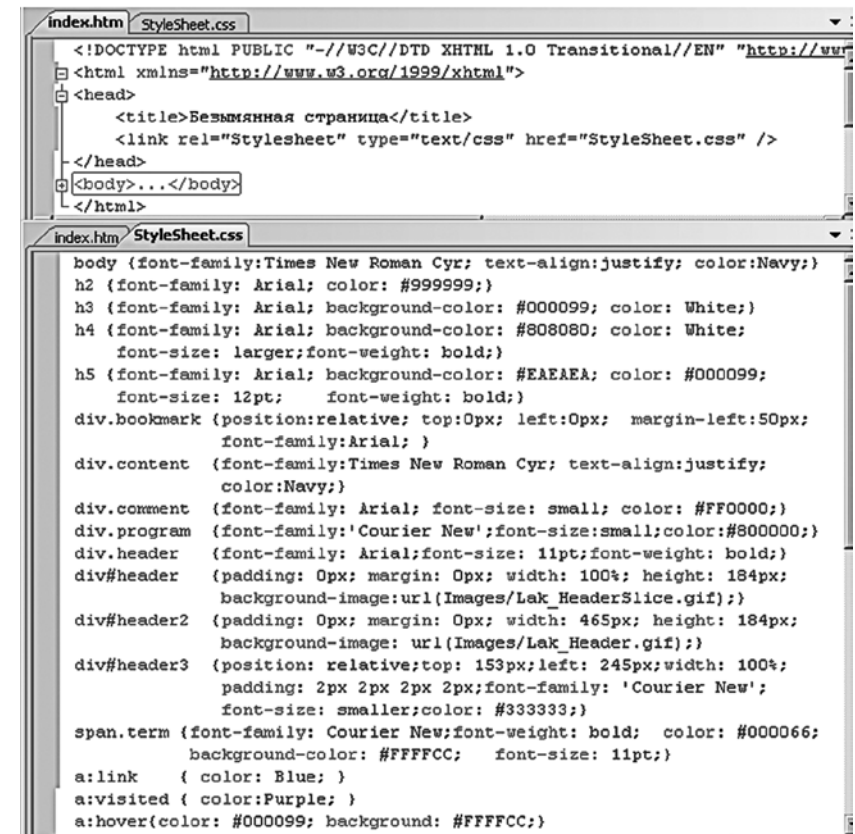


Рис. 7.2. Внешняя таблица стилей сайта



Рис. 7.3. Внешний вид домашней страницы сайта

7.2. HTML-разметка страницы контента

Создадим в папке сайта вложенную папку **CompTech** для размещения страниц контента. Внутри этой папки создадим вложенную папку **Images** для размещения графических иллюстраций контента (рис. 7.4).

Добавим в папку **CompTech** страницу контента **01.htm** подключим к ней внешнюю таблицу стилей и выполним разметку страницы контента (рис. 7.5):

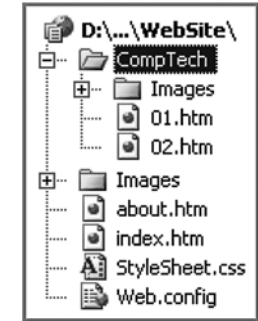


Рис. 7.4. Папка для страниц контента

```
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
  <title>Copyright (C) Leonid A. KEROV, 2010</title>
  <link rel=»stylesheet» type=»text/css»
  href=»StyleSheet.css» />
</head>
<body>
<h2><u>Тема</u>: Основные понятия компьютерных технологий</h2>

<div class=»bookmark»>
  <a href=»#1»>1. Объект и процесс</a> <br />
  <a href=»#2»>2. Данные, информация, знания, компетент-
  ность</a> <br />
</div>

<h3><a name=»1»>1. Объект и процесс</a></h3>
<table>
<tr>
<td valign=»top»>
<div class=»content»>
Содержание контента
</div>
</td>
<td valign=»top»><img src=»Images/0101.jpg» hspace=»10»
alt=»»= /></td>
</tr>
</table>

</body>
</html>
```

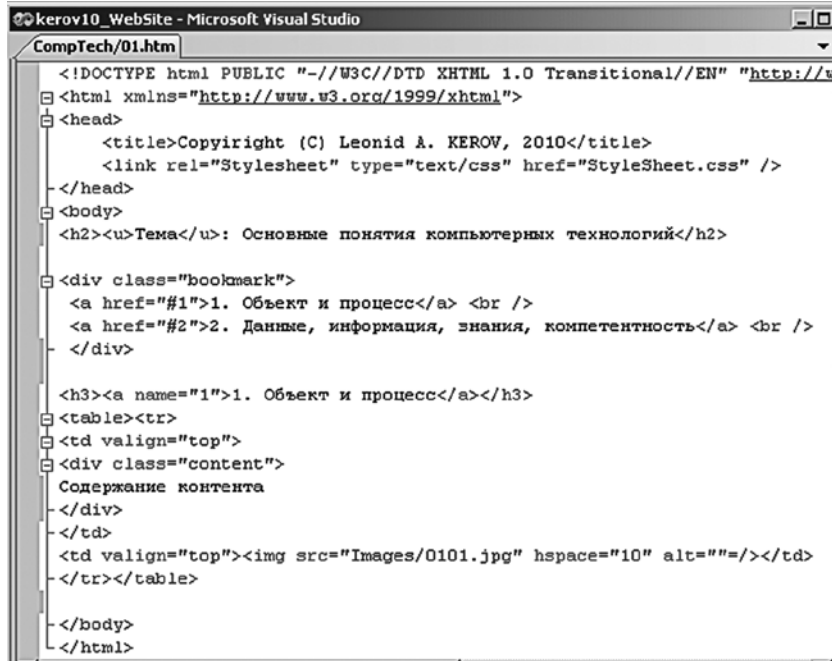


Рис. 7.5. Разметка страницы контента

Раздел <body> страницы контента начинается с указания тематики этой страницы:

```
<h2><u>Тема</u>: Основные понятия компьютерных технологий</h2>
```

Далее следует оглавление разделов страницы, оформленной в виде ярлыков для закладок:

```
<div class="bookmark">
<a href=#1>>1. Объект и процесс</a> <br />
<a href=#2>>2. Данные, информация, знания, компетентность</a> <br />
</div>
```

Содержание каждого раздела начинается с заголовка раздела, оформленного в виде закладки:

```
<h3><a name="1">>1. Объект и процесс</a></h3>
```

Содержание раздела оформлено в виде таблицы, состоящей из одной строки и двух ячеек. В левой ячейке размещается текстовое содержание

контента раздела, а в правой — графическая иллюстрация к этому разделу.

```
<table>
<tr>
<td valign="top">>
<div class="content">>
Содержание контента
</div>
</td>
<td valign="top">>/></td>
</tr>
</table>
```

Внешний вид страницы контента, оформленной с использованием внешней таблицы стилей, показан на рис. 7.6.



Рис. 7.6. Внешний вид страницы контента

**ГЛАВА 8
ХОСТИНГ WEB-САЙТА С HTML-СТРАНИЦАМИ**

8.1. Регистрация доменного имени сайта

Для регистрации доменного имени сайта следует зайти на Web-сайт www.narod.ru и щелкнуть ссылку «Создайте свой сайт» (рис. 8.1).

Рис.8.1.
Ссылка
«Создайте свой сайт»

В результате выполненных действий происходит переход на страницу входа, где следует щелкнуть гиперссылку «Зарегистрироваться» (рис. 8.2).

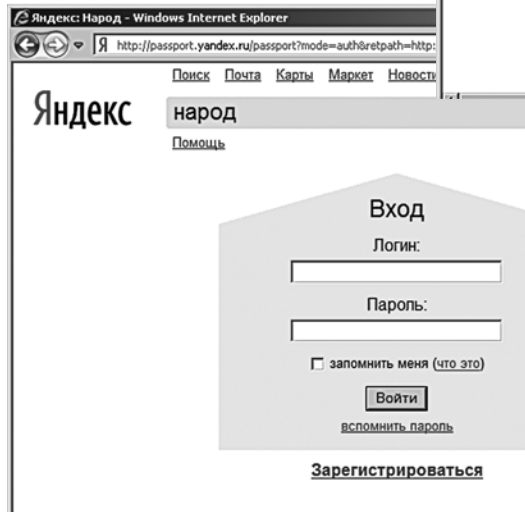


Рис. 8.2. Страница входа со ссылкой «Зарегистрироваться»

Щелчок по гиперссылке «Зарегистрироваться» приводит к переходу на первую страницу регистрации, где следует указать ваши имя и фамилию, а также придумать и ввести ваш логин (регистрационное имя владельца сайта).

Если придуманный вами логин кем-то уже занят, то справа от текстового поля будет выведено соответствующее сообщение красным цветом. Нужно придумать и ввести другой логин. Ниже текстового поля для ввода логина приводится список вариантов свободных логинов.

Если введен свободный логин, то справа от текстового поля появится соответствующее сообщение зеленым цветом. После этого нужно щелкнуть кнопку «Дальше» (рис. 8.3).

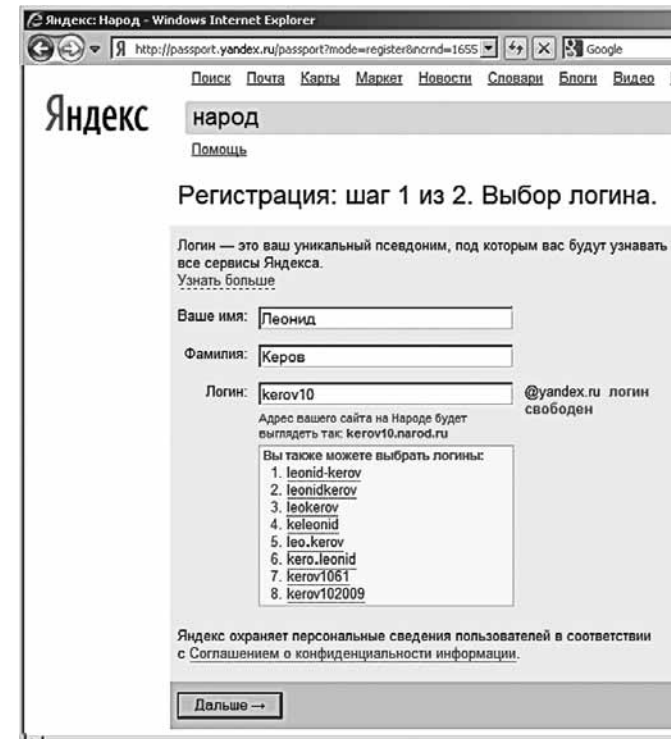


Рис. 8.3. Первая страница регистрации

Щелчок по кнопке «Дальше» приводит к переходу на вторую страницу регистрации, где следует указать пароль для владельца регистрируемого сайта и другие параметры регистрации. В заключение нужно щелкнуть кнопку «Зарегистрировать» (рис. 8.4).

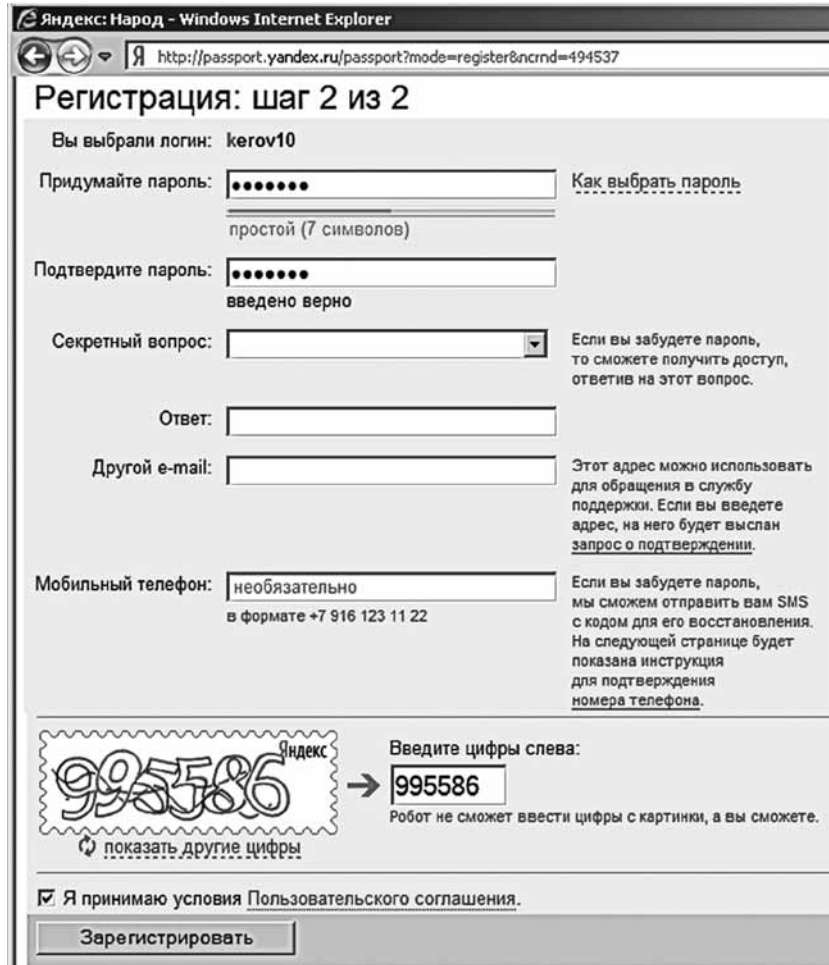


Рис. 8.4. Вторая страница регистрации

Щелчок по кнопке «**Зарегистрировать**» приводит к завершению регистрации доменного имени вашего сайта и переход на страницу «**Управление сайтом**». На этой странице слева от зеленой кнопки с надписью «Создать» указано доменное имя вашего сайта. Для выполнения операций по заполнению сайта следует щелкнуть гиперссылку «**Или перейдите в мастерскую**» (рис. 8.5).

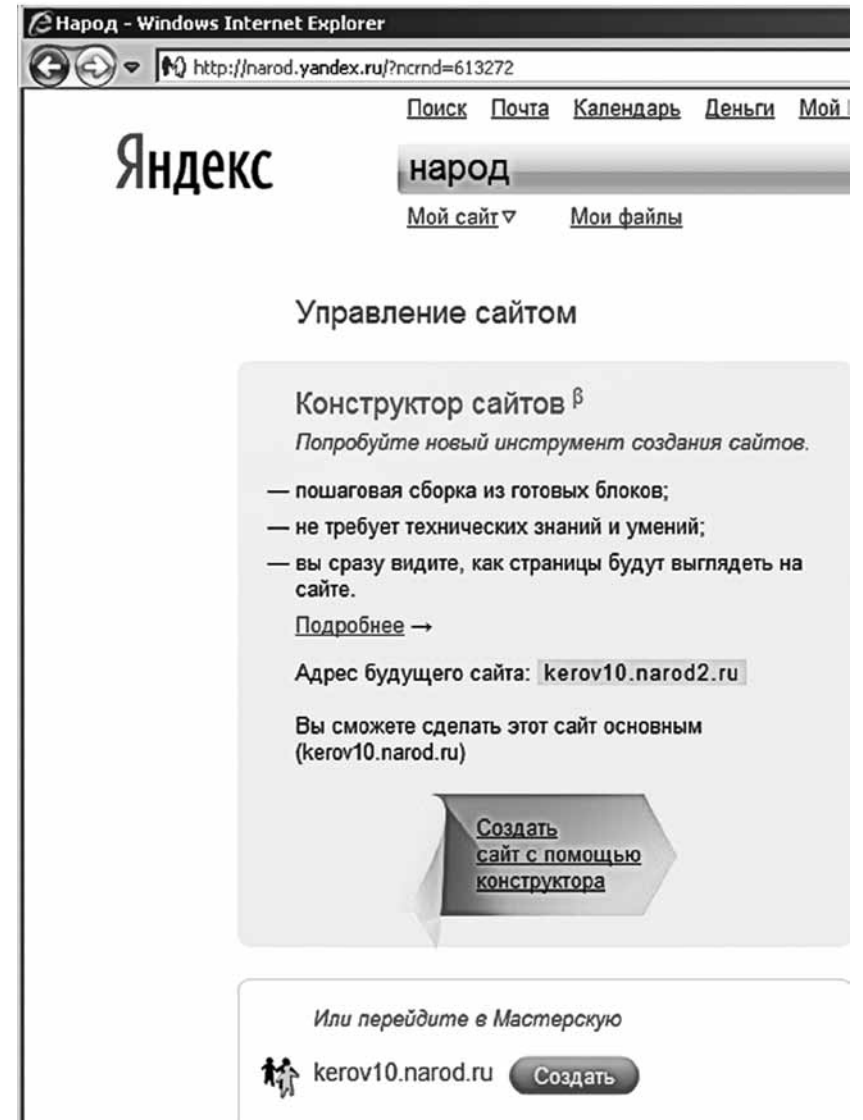


Рис. 8.5. Завершение регистрации доменного имени сайта

8.2. Загрузка файлов на сайт

Для загрузки ранее созданных файлов на сайт следует на странице «Мастерская» щелкнуть гиперссылку «*Загрузка файлов на сайт*». Для выполнения операций по управлению файлами Web-сайта щелкнуть гиперссылку «*Управление файлами и HTML-редактор*» (рис. 8.6).

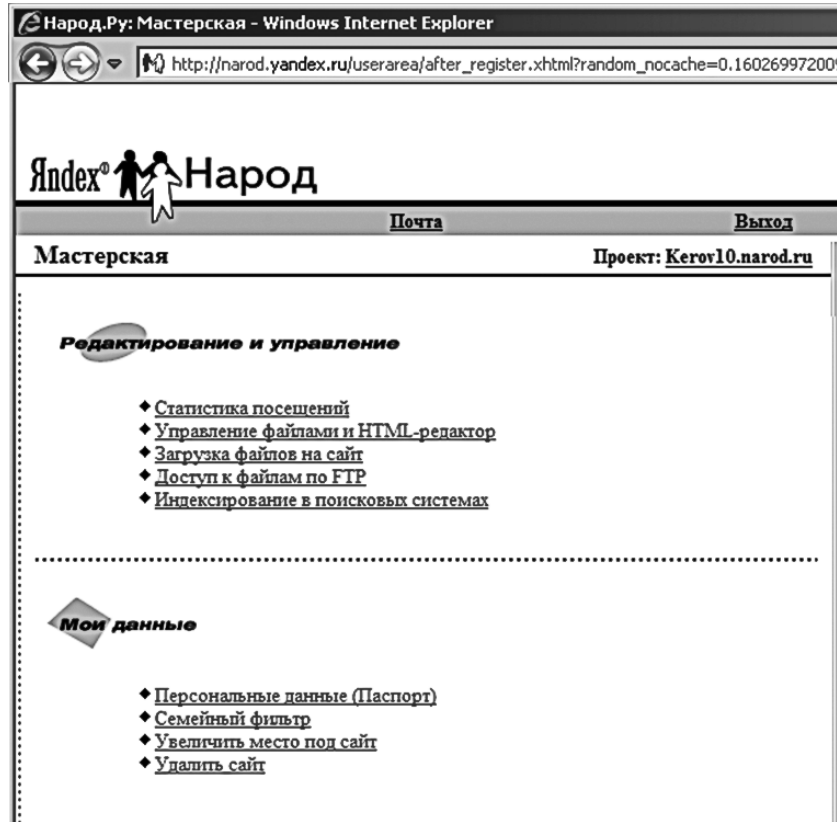


Рис. 8.6. Страница «Мастерская»

8.3. Управление файлами сайта

На странице управления файлами сайта следует выбрать из меню необходимую команду (рис. 8.7):

- «Создать папку» — эта команда позволяет создать в сайте вложенную папку,
- «Загрузить файлы» — эта команда позволяет загрузить разработанные файлы на сайт.

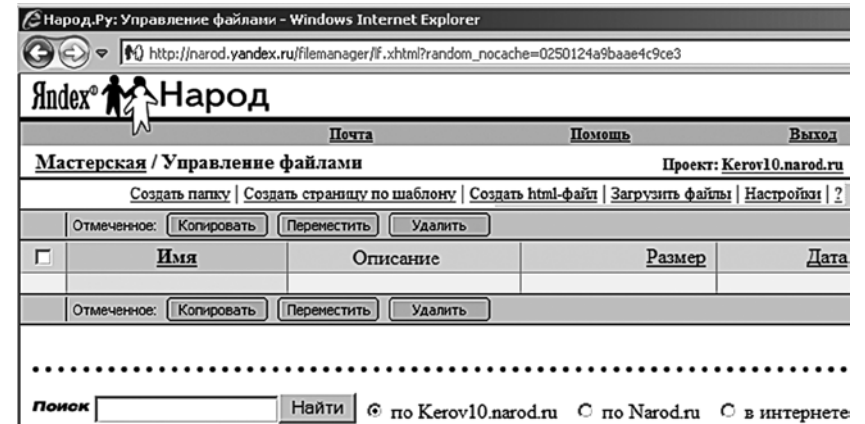


Рис. 8.7. Страница управления файлами

ГЛАВА 9
ЗНАКОМСТВО С WEB-САЙТАМИ С ASP-СТРАНИЦАМИ

9.1. Схема функционирования Web-сайта

Web-браузер — это независимое от платформы средство отображения Web-страниц, написанных на языке HTML. **Web-сервер** — это компьютер, на котором хранятся файлы Web-страниц. Обращение Web-браузера к Web-серверу называется **запросом**. Данные, отправляемые Web-сервером Web-браузеру, называются **откликом** (рис. 9.1).



Рис. 9.1. Схемы функционирования Web-сайтов

В течение первых лет функционирования системы World Wide Web серверы предназначались для хранения Web-сайтов с простыми статическими HTML страницами. В ответ на запрос браузера Web-сервер **откликался отсылкой запрошенной HTML страницы**. Затем Web-сервер закрывал соединение и освобождал все ресурсы, вовлеченные в обработку запроса. Это было необходимо, поскольку Web-серверу приходилось обрабатывать тысячи запросов в минуту.

Современные Web-серверы предоставляют существенно более широкий ассортимент сервисов. В дополнение к хранению файлов статических HTML-страниц, они могут хранить файлы **активных серверных страниц** (Active Server Pages, ASP). Это Web-страницы, которые содержат программный код, выполняемый на сервере. Результаты выполнения этого кода и отправляются браузеру в качестве отклика.

Современные Web-серверы также поддерживают сохранение указанных в запросах данных, что позволяет создавать Web-приложения для решения различных задач бизнеса. Приложение, развернутое на Web-сервере, может содержать много Web-страниц. Представление, что Web-сервер закрывает соединение после отправки браузеру единственной Web-страницы, считается устаревшим. Теперь Web-сервер может поддерживать постоянное соединение с Web-браузером, непрерывно ожидая дополнительные запросы.

9.2. Создание шаблона Web-сайта с ASP-страницей

Создадим шаблон Web-сайта с ASP-страницей, выполнив следующие действия (рис. 9.2):

- выберем из меню **Файл** пункт **Создать** и команду **Веб-узел**,
- в диалоговом окне **Новый веб-узел** выберем шаблон **Веб-узел ASP.NET**,
- в списке **Расположение** выберем элемент **Файловая система**, и в следующее за ним текстовое поле введем **полное имя папки**, в которой должны храниться страницы Web-сайта,
- в списке **Язык** выберем элемент **Visual C#** и щелкнем кнопку **OK**.

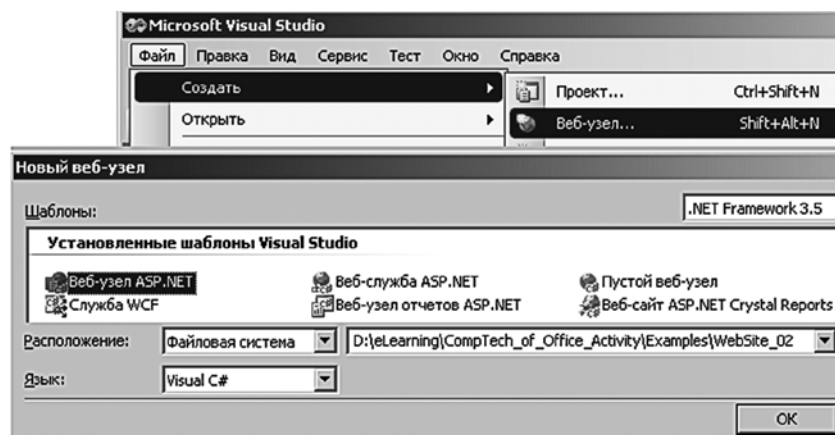


Рис. 9.2. Создание шаблона Web-сайта с ASP-страницей

В результате выполненных действий будет создан шаблон Web-сайта с ASP-страницей. В окне **«Обозреватель решений»** можно увидеть перечень файлов Web-сайта с ASP-страницей (рис. 9.3):

- файл ASP-страницы с именем **Default.aspx**,
- связанный с этой ASP-страницей файл программного кода с именем **Default.aspx.cs**,
- файл конфигурации Web-сайта кода с именем **web.config**.

Если окна **Обозреватель решений** нет на экране, то из меню **Вид** нужно выбрать команду **Обозреватель решений**.

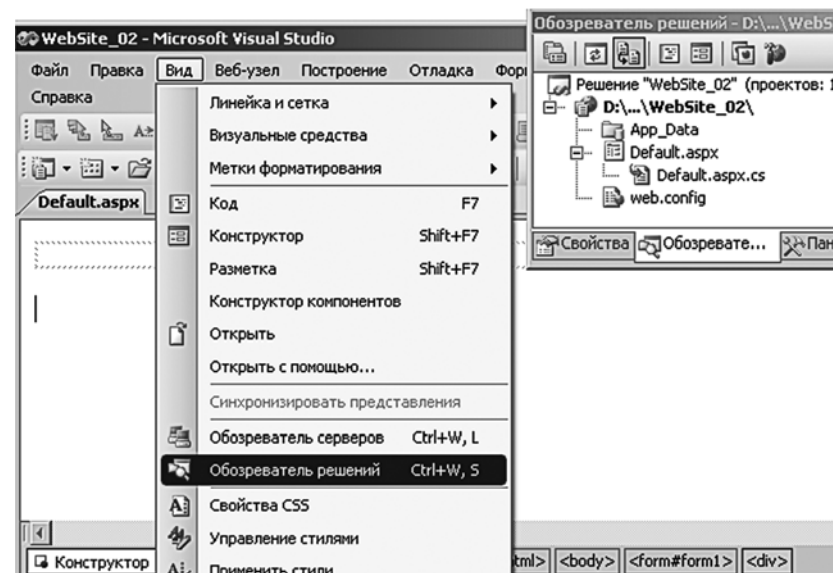


Рис. 9.3. Перечень файлов Web-сайта с ASP-страницей

Содержимое ASP-страницы может быть отображено в одном из трех режимов, переключение которых выполняется с помощью ярлыков в нижней части окна (рис. 9.4):

- в режиме «**Конструктор**» ASP-страница показывается в таком виде, как она будет выглядеть в окне Web-браузера (например, Microsoft Internet Explorer),
- в режиме «**Исходный код**» показывается код разметки ASP-страницы,
- в режиме «**Разделить**» рабочая область Visual Studio разделяется на две части, одна из которых соответствует режиму «**Исходный код**», а вторая — режиму «**Конструктор**».

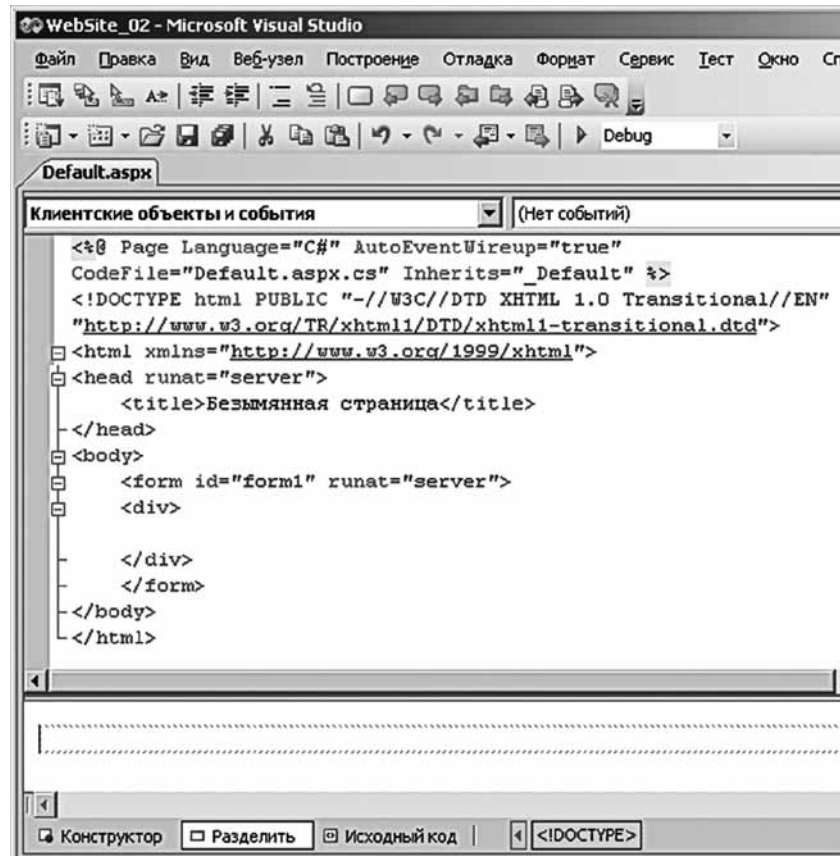


Рис. 9.4. Содержимое шаблона ASP-страницы

Код разметки ASP-страницы начинается с директивы «**@Page**», которая задает общие атрибуты страницы:

- **Language** — язык программирования ASP-страницы,
- **AutoEventWireup** — режим вызова обработчиков событий, воздействующих на страницу (true — автоматический режим вызова обработчиков событий),
- **CodeFile** — имя файла программного кода, связанного со страницей,
- **Inherits** — имя класса, определяющего функциональность программного кода, связанного со страницей.

Ниже располагается директива «**!DOCTYPE**», которая задает формат кода разметки (в данном случае — XHTML 1.0 Transitional). Затем следуют теги HTML-разметки активной серверной страницы:

- **<html> </html>** — теги начала и конца HTML-разметки,
- **<head> </head>** — теги заголовка страницы,
- **<title> </title>** — теги названия страницы,
- **<body> </body>** — теги тела страницы,
- **<form> </form>** — теги Web-формы,
- **<div> </div>** — теги раздела.

Файл программного кода, связанного с ASP-страницей, создается с использованием того языка программирования, который был указан при создании шаблона Web-сайта (в данном примере — язык C#). Файл начинается с директив **«using»**, которые указывают, какие пространства имен используются в файле программного кода (упрощенно говоря, это что-то вроде указания на то, какие системные библиотеки можно использовать в файле программного кода; рис. 9.5).

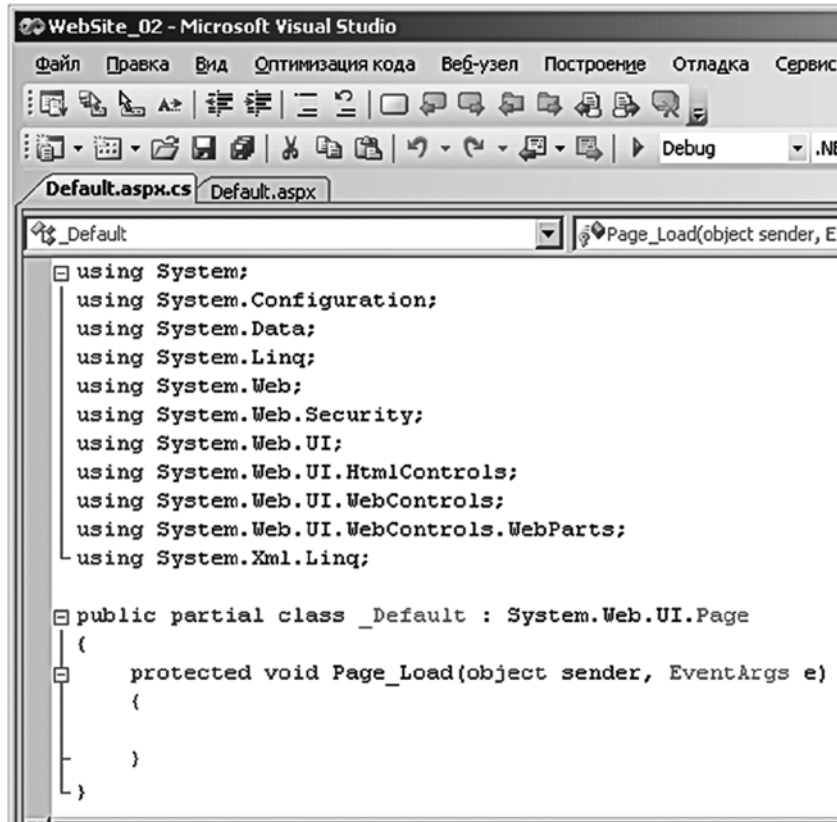


Рис. 9.5. Файл программного кода, связанного с ASP-страницей

После директив «using» следует **определение класса**, определяющего функциональность программного кода, связанного со страницей:

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
    
```

Первая строка **этого** кода является **заголовком класса**; после него в фигурных скобках следует **тело класса**. В теле класса можно определить:

- переменные, которые называются **полями класса**,
- функции, которые называются **методами класса**.

В процессе создания шаблона Web-сайта с ASP-страницей в тело класса добавляется определение **метода Page_Load** с пустым телом. Этот метод вызывается Web-сервером автоматически после создания объекта указанного класса и обычно используется для инициализации этого объекта.

9.3. Проектирование ASP-страницы

Проектирование ASP-страницы можно выполнять в режиме **«Конструктор»** с использованием панели элементов проектирования (рис. 9.6). Если панели элементов нет на экране, то из меню **«Вид»** нужно выбрать команду **«Панель элементов»**.

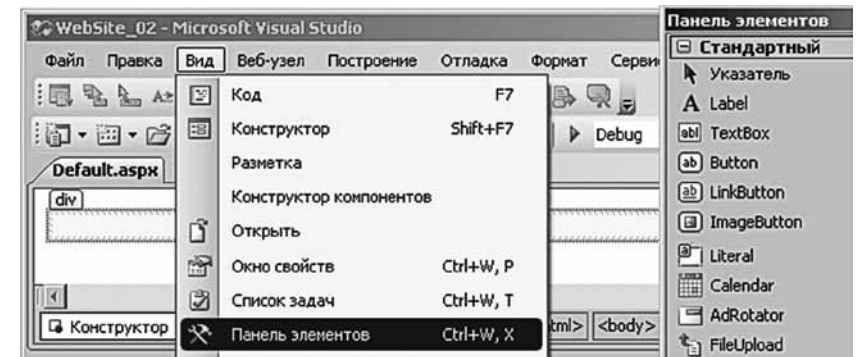


Рис. 9.6. Панель элементов

Переместим *надпись* (элемент «*Label*») из панели элементов на ASP-страницу. Щелкнем правой кнопкой мыши по надписи, и из контекстного меню выберем команду «*Свойства*». На экран будет выведено диалоговое окно свойств, в котором можно уточнить или изменить значения свойств надписи; в частности (рис. 9.7):

- значением свойства (*ID*) является имя элемента «*Label*», которое используется в программном коде для обращения к этому элементу; в данном случае надписи «по умолчанию» присвоено имя «*Label1*»,
- значением свойства «*Text*» является текст надписи. Значение этого свойства «по умолчанию» является пустым текстом. Скорректируем значение свойства «*Text*», введя справа от него текст «Введите ваш рост в см.»:

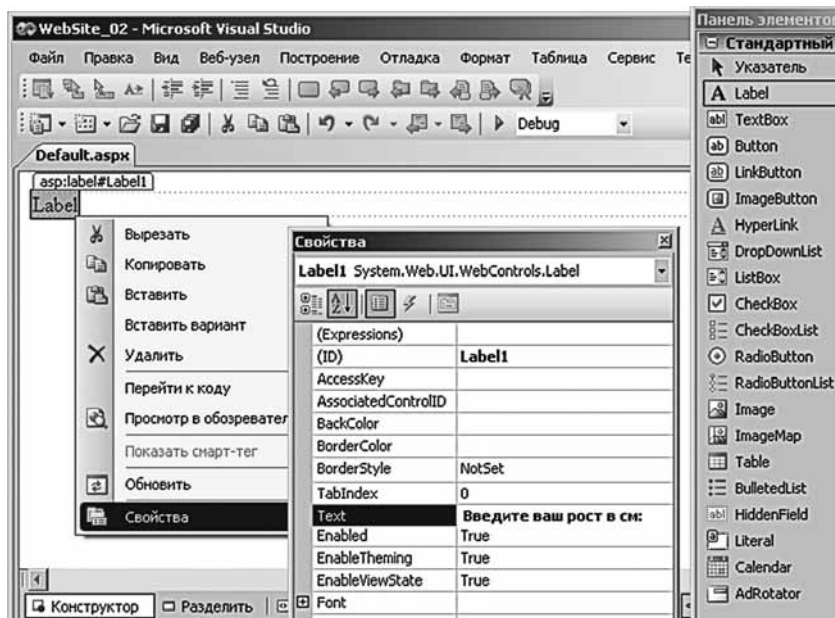


Рис. 9.7. Создание надписи на странице

Переместим *текстовое поле* (элемент «*TextBox*») из панели элементов на ASP-страницу. Щелкнем правой кнопкой мыши по текстовому полю, и из контекстного меню выберем команду «*Свойства*». На экран будет выведено диалоговое окно свойств, в котором можно уточнить или изменить значения свойств текстового поля; в частности (рис. 9.8):

- значением свойства (*ID*) является имя элемента «*TextBox*», которое используется в программном коде для обращения к этому элементу;

ту; в данном случае текстовому полю «по умолчанию» присвоено имя «*TextBox1*»,

- значением свойства «*Text*» является текст в текстовом поле. Значение этого свойства «по умолчанию» является пустым текстом,
- значением свойства «*ReadOnly*» является логическое значение. Значение «*False*» означает возможность редактирования текста в текстовом поле. Значение «*True*» означает, что текст можно только читать, но нельзя изменить,
- свойство «*TextMode*» может принимать следующие значения: «*SingleLine*» — текстовое поле может содержать только одну строку текста; «*MultiLine*» — текстовое поле может содержать несколько строк текста; «*Password*» — эхо-отображением вводимых в текстовое поле символов являются звездочки.

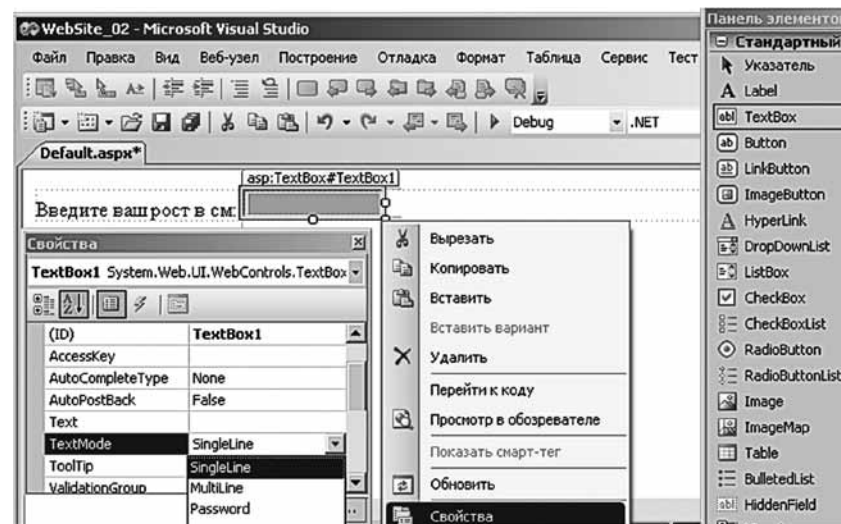


Рис. 9.8. Создание текстового поля на странице

Переместим *кнопку* (элемент «*Button*») из панели элементов на ASP-страницу. Щелчком правой кнопкой мыши по кнопке, и из контекстного меню выберем команду «*Свойства*». На экран будет выведено диалоговое окно свойств, в котором можно уточнить или изменить значения свойств кнопки; в частности (рис. 9.9):

- значением свойства (*ID*) является имя элемента «*Button*», которое используется в программном коде для обращения к этому элементу; в данном случае кнопке «по умолчанию» присвоено имя «*Button1*»,
- значением свойства «*Text*» является текст надписи на кнопке. Значение этого свойства «по умолчанию» совпадает с именем кнопки. Скорректируем значение свойства «*Text*», введя справа от него текст «*Рассчитать рост в дюймах*».

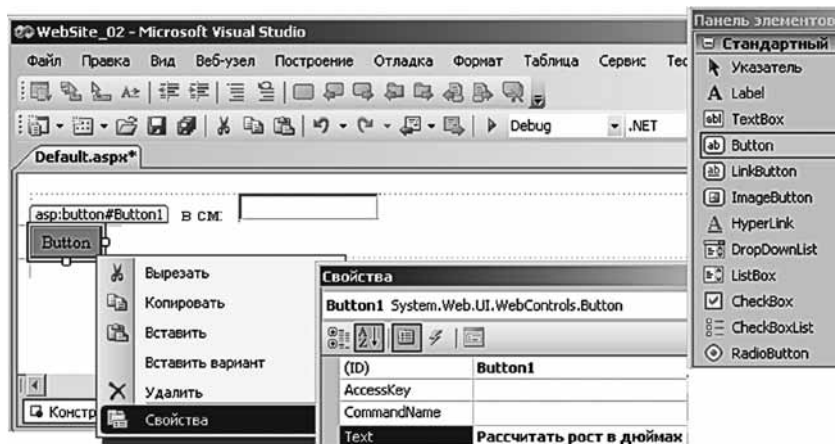


Рис. 9.9. Создание кнопки на странице

Переместим из палитры элементов на ASP-страницу еще одну надпись («*Ваш рост в дюймах*») и еще одно текстовое поле (*TextBox2*). Мы проектировали пользовательский интерфейс ASP-страницы, работая в режиме «*Конструктор*» и пользуясь панелью элементов. При этом результат каждого нашего действия преобразовывался системой Visual Studio в соответствующий HTML-код. Переключившись в режим «*Исходный код*», можно увидеть этот HTML-код (рис. 9.10).

Имеется альтернативный подход к проектированию пользовательского интерфейса ASP-страницы: можно писать HTML-код страницы «вручную» в режиме «*Исходный код*». Можно комбинировать эти два подхода, выполняя часть работы в режиме «*Конструктор*», а часть работы — в режиме «*Исходный код*». Для этой цели удобно использовать режим «*Разделить*».

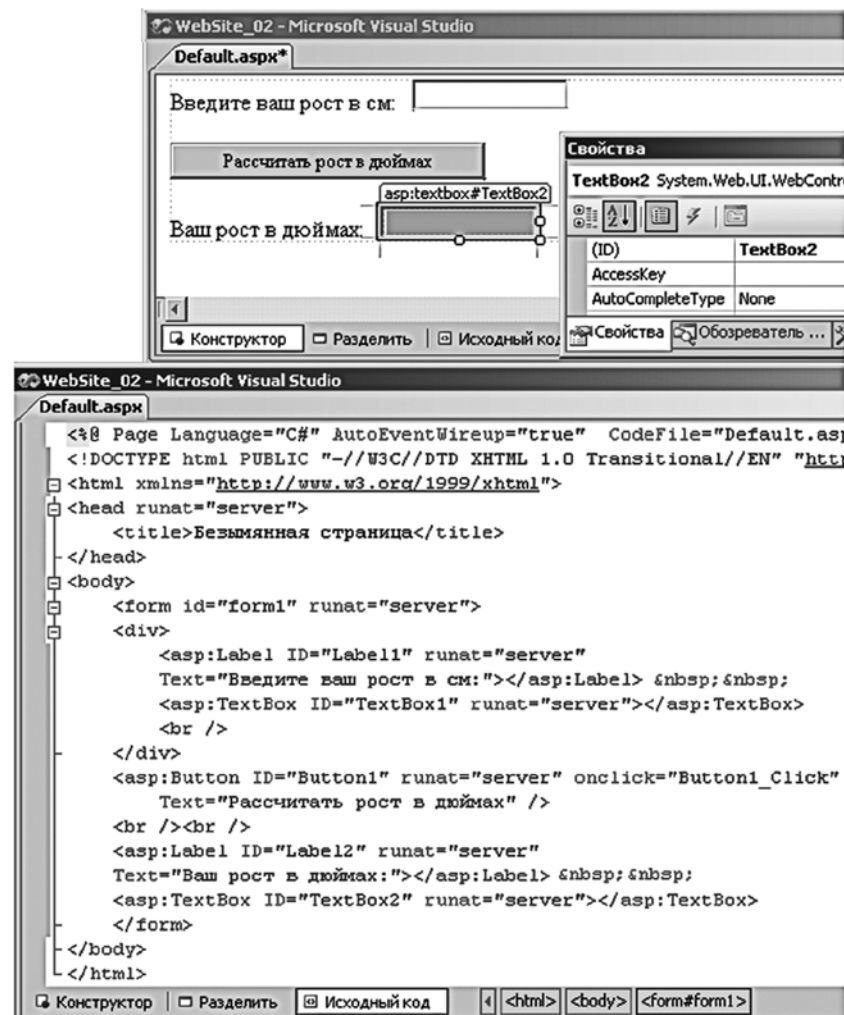


Рис. 9.10. HTML-код активной серверной страницы

9.4. Программирование ASP-страницы

Выполним *двойной щелчок* мышкой по кнопке с надписью «Рассчитать рост в дюймах». В ответ на это действие система Visual Studio перейдет к редактированию файла программного кода «Default.aspx.cs», связанного с ASP – страницей, и создаст в этом файле каркас обработчика события «Click» с именем используя свойство (ID) кнопки и имя события (рис. 9.11).

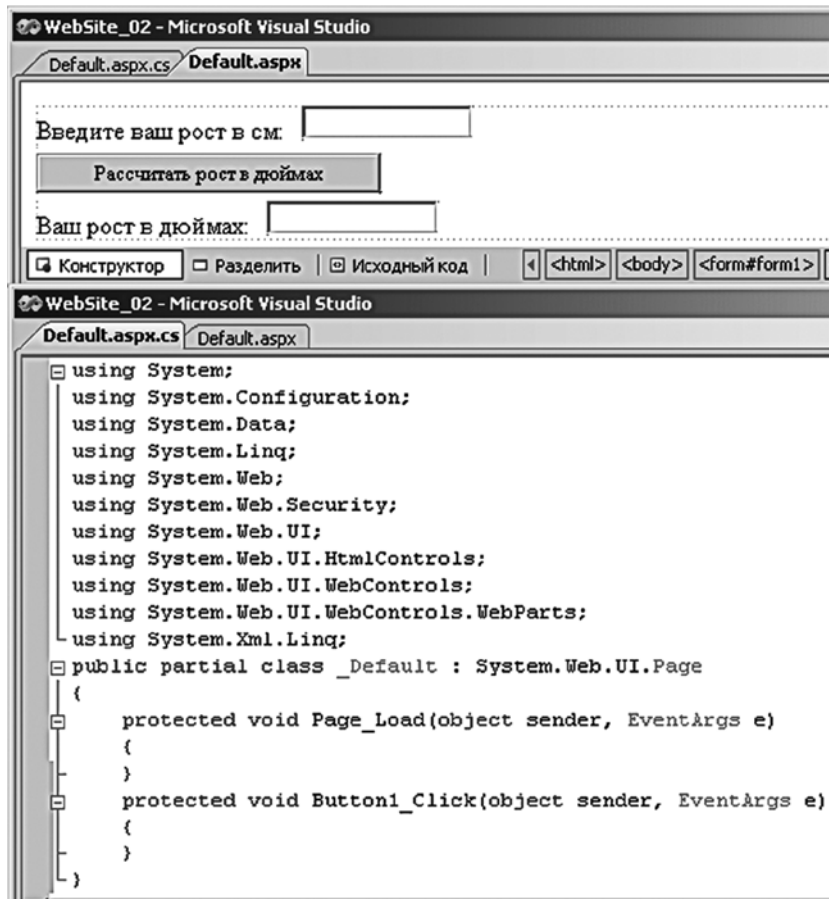


Рис. 9.11. Каркас обработчика события «Click» для кнопки

Определим функциональность метода «*Button1_Click*» посредством ввода необходимых операторов языка С# в тело этого метода (рис. 9.12). Тело метода «*Button1_Click*» начинается с определения переменной, значением которой будет введенный пользователем его рост в сантиметрах:

```
double height;
```

Определение переменной начинается с указания типа переменной (**double**), за которым следует имя переменной (**height**) и точка с запятой. *Тип* «**double**» указывает на то, что значением переменной должно быть дробное число. Примеры других типов: «**int**» — *целое* число; «**string**» — строка символов. *Имя переменной* должно начинаться с латинской буквы, может содержать латинские буквы, цифры или знаки подчеркивания и не должно совпадать с ключевым словом языка С#. Ключевое слово в тексте программы изображается синим цветом.

Физический смысл определения переменной состоит в том, что в области оперативной памяти *компьютера* (которая называется «стек») выделяется ячейка для хранения значения переменной. Размер этой ячейки определяется типом переменной (для **double** — 64 бита, т. е. 8 байт). Переменные, значения которых хранятся в стеке, называются переменными *значимого типа* (value type).

Присвоение переменной «**height**» некоторого значения выполняется посредством *оператора присваивания*:

```
height = double.Parse(TextBox1.Text);
```

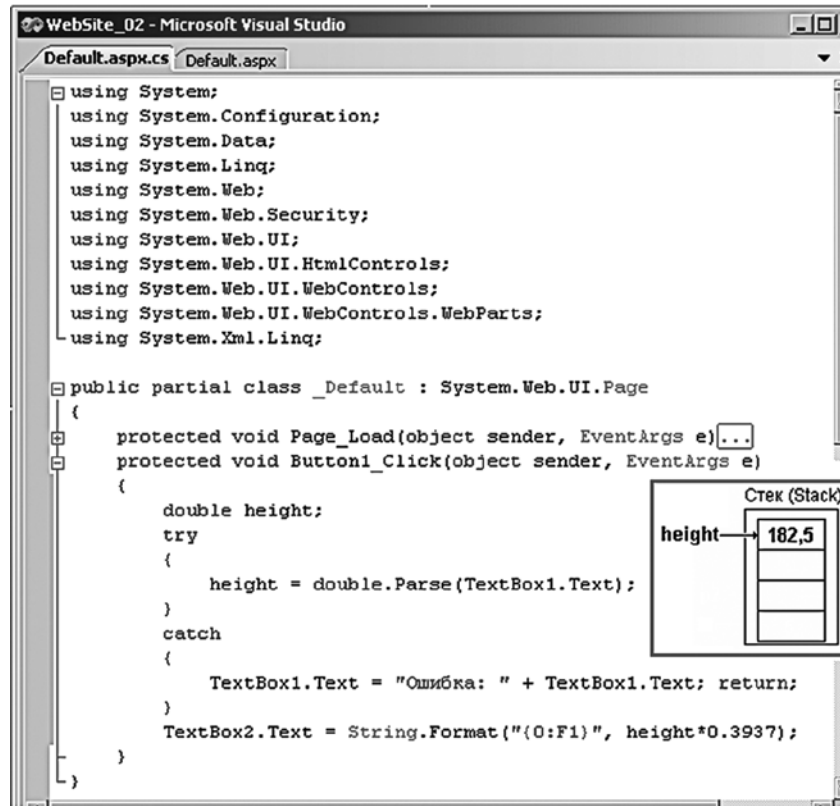


Рис. 9.12. Определение функциональности метода «Button1_Click»

Оператор присваивания изображается знаком равенства. Слева от него указывается имя переменной, а справа — выражение. **Выражение** — это правило вычисления некоторого значения; для построения выражения можно использовать константы, переменные, знаки операций и вызовы функций. Значением выражения `TextBox1.Text` является рост в сантиметрах, введенный пользователем в текстовом поле «`TextBox1`».

Заметим, что пользователь вводит с клавиатуры последовательность символов, т.е. некоторый текст. Программа должна проверить, является ли этот текст изображением дробного числа и, если это так, пересылает соответствующее дробное число в ячейку переменной, указанной слева от знака равенства. Проверку выполняет статический метод «`Parse`» класса «`double`». Если проверка закончилась неудачно, то генерируется сообщение об **исключении** (Exception). Если сообщение

об исключении получает операционная система компьютера, то она аварийно завершает выполнение программы.

Можно предотвратить аварийное завершение программы, перехватив сообщение об исключении. Программируется это следующим образом. Операторы, которые могут вызывать исключение, помещаются в блок «`try`», сразу за которым указывается блок «`catch`». Если в блоке «`try`» возникает исключение, то сообщение об этом перехватывает блок «`catch`». В данном примере блок «`catch`» выводит в текстовое поле «`TextBox1`» символы «Ошибка: », затем — текст, введенный пользователем. После этого посредством оператора «`return`» завершается выполнение метода «`Button1_Click`».

Если исключительной ситуации не возникло, то выполняется оператор, указанный **после** блока «`catch`». Он выводит в текстовое поле «`TextBox2`» текст, который является изображением величины роста в дюймах. При этом величина роста в дюймах вычисляется по формуле:

$$\text{height} * 0.3937$$

В формулах можно использовать следующие арифметические операции:

Операция	Семантика
+	сложение
-	вычитание
*	умножение
/	деление
%	остаток от деления
++	инкремент (увеличение на 1)
--	декремент (уменьшение на 1)

Преобразование дробного числа в текст выполняет статический **метод** «`Format`» класса «`String`». Первый аргумент этого метода называется **строкой форматирования**. Строка форматирования вида «`{0:F1}`» означает, что значением функции «`String.Format`» должна быть последовательность символов, которая соответствует значению первого выражения, следующего за строкой форматирования. Нумерация начинается с нуля, поэтому `{0}` — это номер первого выражения. Значение выражения берется с одним знаком после запятой, это указано в виде «`F1`» после номера выражения.

Для тестирования ASP-страницы нужно в окне обозревателя реше-

ний щелкнуть по ее имени правой кнопкой мыши, и выбрать команду «*Просмотр в обозревателе*». Затем в текстовое поле с надписью «Введите ваш рост в см» следует ввести дробное число и щелкнуть кнопку с надписью «Рассчитать рост в дюймах». Можно ввести и целое число; оно будет рассматриваться программой как дробное число в нулевой дробной частью. Если при вводе дробного числа допущена ошибка (например, введены символы «abc»), то сообщение об ошибке выводится в это же текстовое поле. Нужно повторить ввод заново, и снова щелкнуть кнопку с надписью «Рассчитать рост в дюймах». Если введенные данные были правильными, то программа рассчитывает соответствующий рост в дюймах и выводит его в текстовое поле с соответствующей надписью (рис. 9.13).

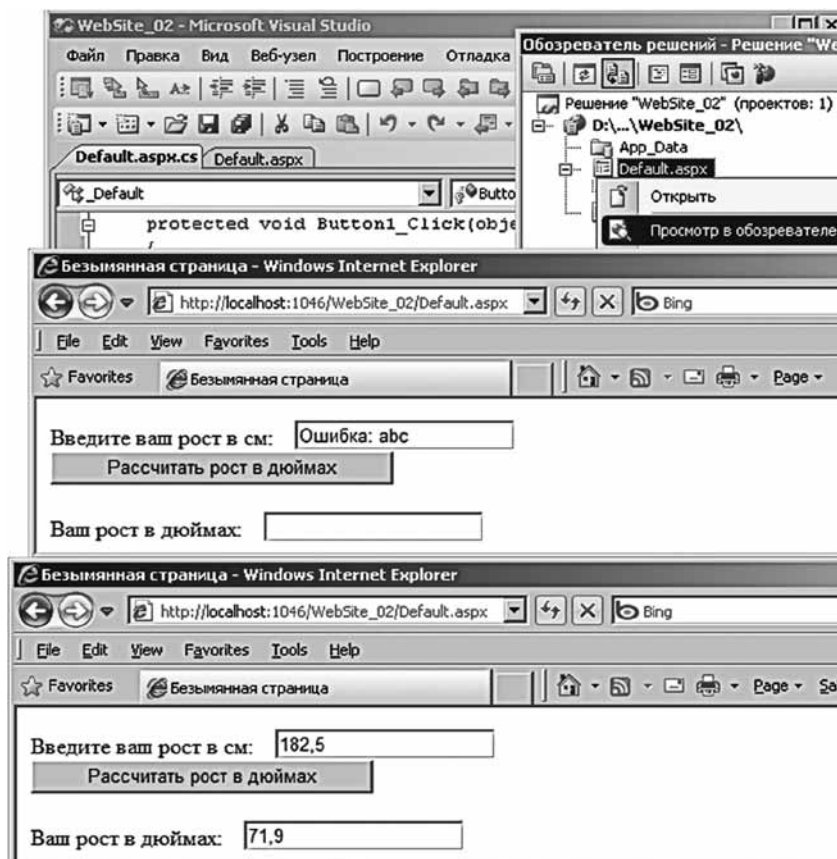


Рис. 9.13. Тестирование ASP-страницы

КОНТРОЛЬНЫЕ ВОПРОСЫ

Вопрос № 1. Для создания в Visual Studio 2008 шаблона информационного Web-сайта с HTML-страницами следует выбрать шаблон...

Варианты ответов:

- веб-узел ASP.NET
- веб-узел отчетов ASP.NET
- пустой веб-узел
- веб-сайт ASP.NET Crystal Reports

Вопрос № 2. Для просмотра в Visual Studio 2008 содержимого Web-сайта с HTML-страницами следует вывести на экран окно...

Варианты ответов:

- обозревателя серверов
- обозревателя решений
- обозревателя объектов
- определения кода

Вопрос № 3. Укажите правильный тег перехода на новую строку

Варианты ответов:

- < br >
- </ br >
- < br />
- < newline >

Вопрос № 4. Неупорядоченные списки заключаются в теги...

Варианты ответов:

-
-
- <dl> </dl>
-

Вопрос № 5. Упорядоченные списки заключаются в теги...

Варианты ответов:

-
-
- <dl> </dl>
-

Вопрос № 6. Списки определений заключаются в теги...

Варианты ответов:

- ` `
- ` `
- `<dl> </dl>`
- ` `

Вопрос № 7. Элементы списка заключаются в теги...

Варианты ответов:

- ` `
- ` `
- `<dl> </dl>`
- ` `

Вопрос № 8. Горизонтальные линии задаются с помощью тега...

Варианты ответов:

- `<hr >`
- `<hr />`
- `</hr>`
- `<line >`

Вопрос № 9. Комментарии заключаются в теги...

Варианты ответов:

- `<!-- -->`
- `<-- -->`
- `<-- --!>`
- `<!-- --!>`

Вопрос № 10. Неразрывный пробел в HTML-коде задается комбинацией...

Варианты ответов:

- `<nbsp>`
- `&space;`
- ` `
- `<space>`

Вопрос № 11. Для вставки изображения на HTML-страницу из файла предназначен тег...

Варианты ответов:

- `<pic />`
- ``
- `<insert />`
- `<link />`

Вопрос № 12. Гиперссылка заключается между тегами...

Варианты ответов:

- `<a> `
- ` `
- `<c> </c>`
- `<d> </d>`

Вопрос № 13. Фоновое изображение на HTML-странице задается с помощью тега...

Варианты ответов:

- `<html>`
- ``
- `<link />`
- `<body>`

Вопрос № 14. Для создания закладок в пределах данной HTML-страницы используются теги...

Варианты ответов:

- `<a> `
- `<p> </p>`
- `<div> </div>`
- `<href> </href>`

Вопрос № 15. Связь рисунка с картой выполняется при помощи тега...

Варианты ответов:

- ``
- `<map />`
- `<href />`
- `<usemap />`

Вопрос № 16. Для объединения ячеек в строке таблицы используется параметр...

Варианты ответов:

- colspan
- rowspan
- span
- join

Вопрос № 17. Для объединения ячеек в столбце таблицы используется параметр...

Варианты ответов:

- colspan
- rowspan
- span
- join

Вопрос № 18. Связывание HTML-страницы с файлом внешней таблицы стилей используется тег...

Варианты ответов:

- <html>
- <head>
- <link />
- <body>

Вопрос № 19. К встраиваемым тегам относится тег...

Варианты ответов:

- <p>
- <h1>
- <table>
-

Вопрос № 20. К тегам уровня блока относится тег...

Варианты ответов:

- <p>
-
- <i>
-

СПИСОК ЛИТЕРАТУРЫ

1. Беллиньясо М. Разработка Web-приложений в среде ASP.NET 2.0: задача-проект-решение: Пер. с англ. — М.: ИД «Вильямс», 2007. — 640 с.
2. Гасанов Э. В. Практикум по Web-дизайну: Практический курс разработки Web-сайтов/ ГУ-ВШЭ. — М.: ТЕИС, 2006. — 158 с.
3. Гасанов Э. В. Сборник заданий для практических занятий по Web-дизайну и методические указания по их выполнению/ ГУ-ВШЭ. — М.: ТЕИС, 2006. — 158 с.
4. Мак-Дональд М., Шпушта М. Microsoft ASP.NET 3.5 с примерами на C# 2008 для профессионалов. 2-е изд.- М.: ООО «ИД Вильямс», 2008. — 1424 с.
5. Моррисон М. HTML и XML. Быстро и эффективно. — СПб.: Питер, 2005. — 303 с.
6. Нейгел К., Ивьян Б., Глинн Дж., Уотсон К. C# 2005 для профессионалов: Пер. с англ. — М.: ИД «Вильямс», 2007. — 1376 с.
7. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. — СПб.: Питер, 2001. — 672 с.
8. Пайк М. Internet в подлиннике: Пер. с англ. — СПб.: BHV — Санкт-Петербург, 1996. — 640 с.
9. Сергеев А. П. HTML и XML. Профессиональная работа. — М.: Издательский дом «Вильямс», 2004. — 880 с.
10. Шапошников И. В. Самоучитель HTML 4. — СПб.: БХВ — Петербург, 2003. — 288 с.

Подготовлено к печати и отпечатано
Отделом оперативной полиграфии
НИУ ВШЭ — Санкт-Петербург
198099, Санкт-Петербург, ул. Промышленная, д. 17а
Тел./факс (812) 786-58-92