

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»  
В САНКТ-ПЕТЕРБУРГЕ

---

Серия: «Управление образованием»

МАГИСТЕРСКАЯ ПРОГРАММА  
«УПРАВЛЕНИЕ ОБРАЗОВАНИЕМ»

Направление 081100.68  
«Государственное и муниципальное управление»

*Леонид Керов*

# **ИНФОРМАЦИОННО- КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНЧЕСКОЙ ПРАКТИКЕ**

## **Часть 1. Web-дизайн**

**Учебное пособие**



НИУ ВШЭ — СПб  
Санкт-Петербург  
2011

УДК 681.3.07  
ББК 32.973.26-0.18.2.75  
К36

Рецензенты:

**В. А. Костин**, доцент кафедры информатики СПбГУ;

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

К36 Керов Л. А. **Информационно-коммуникационные технологии в управленческой практике.** Часть 1. Web-дизайн. Учебное пособие. — СПб.: Отдел оперативной полиграфии НИУ ВШЭ — Санкт-Петербург, 2011. с.

Учебное пособие посвящено созданию и использованию в управленческой практике образовательных учреждений персональных и корпоративных Web-сайтов. При этом в качестве инструментальной системы для разработки сайтов применяется инструментальная система Microsoft Visual Studio версии 2008. Используется русифицированный вариант системы, что удобно для преподавания в вузах России.

В учебном пособии рассматриваются вопросы создания Web-сайтов с HTML-страницами в инструментальной среде Microsoft Visual Studio, излагаются технические детали публикации Web-сайтов в сети Интернет, детально рассматриваются вопросы стилового оформления HTML-страниц с использованием внутренних и внешних таблиц стилей, создаваемых с помощью визуального редактора стилей, входящего в состав Microsoft Visual Studio. Особое внимание уделяется программному управлению Web-сайтами с помощью языка JavaScript и работе с Web-формами. Приводятся примеры, иллюстрирующие создание образовательного контента и тестов, т.е. элементов систем класса e-Learning.

Изучение материала предполагает использование технологии продуктивного обучения, целью которой является не усвоение и воспроизведение некоторого набора знаний, а развитие компетенций, т.е. умения решать профессиональные задачи, основанного на полученных знаниях. Студенты проводят информационный поиск в Интернете по темам выбранных ими научно-исследовательских проектов в области информационно-коммуникационных технологий, создают собственные сайты, на которых публикуются результаты исследований и которые размещаются в Интернете, делают доклады о результатах исследований с использованием подготовленных ими электронных презентаций..

Материал учебного пособия апробирован автором в курсах, которые он преподавал студентам математико-механического факультета Санкт-Петербургского государственного университета и студентам Национального исследовательского университета – Высшей школы экономики.

Для студентов, обучающихся по направлению 081100.68 «Государственное и муниципальное управление» подготовки магистра, для магистерской программы «Управление образованием»

*Рекомендовано к печати*

*Учебно-методическим советом НИУ ВШЭ в Санкт-Петербурге*

© Керов Л., 2011

© Национальный исследовательский университет  
«Высшая школа экономики». Санкт-Петербург, 2011

## ГЛАВА 1

# СОЗДАНИЕ WEB-САЙТА В MICROSOFT VISUAL STUDIO

### 1.1. СОЗДАНИЕ САЙТА С HTML-СТРАНИЦАМИ

#### 1.1.1. Создание Web-сайта

Для создания Web-сайта нужно выполнить следующие действия (Рис. 1.1):

- 1) Запустить инструментальную систему Visual Studio 2008
- 2) Выбрать из меню **Файл** пункт **Создать** и команду **Веб-узел**
- 3) На экран будут выведено диалоговое окно **Новый веб-узел**, в верхнем списке которого нужно выбрать элемент **.NET Framework 3.5**
- 4) В списке **Шаблоны** следует выбрать элемент **Пустой веб-узел**
- 5) В списке **Расположение** следует выбрать элемент **Файловая система**, а в следующее за ним текстовое поле требуется ввести **полное имя папки**, в которой должны храниться страницы Web-сайта
- 6) В списке **Язык** нужно выбрать элемент **Visual C#**
- 7) В заключение следует щелкнуть кнопку **ОК**

В результате выполненных действий будет создан пустой Web-сайт. Содержимое Web-сайта в любое время работы над ним можно увидеть в окне **Обозреватель решений**.

В плане структурной организации Web-сайт – это папка с файлами. Среди этих файлов обязательно присутствуют файлы **Web-страниц**. Это текстовые файлы, содержащие особые команды, которые заключены в угловые скобки, называются **тегами** и используются для задания внешнего вида элементов Web-страницы в окне браузера или для указания назначения элементов Web-страницы.

Совокупность тегов и правил их использования называется **языком гипертекстовой разметки**. В Web-сайтах, разрабатываемых с помо-

щью Microsoft Visual Studio, используется самая новая версия этого языка – расширяемый язык гипертекстовой разметки (eXtensible HyperText Markup Language, **XHTML**).

Теги подразделяются на следующие виды:

Клиентские теги, которые «понимает» Web-браузер

Серверные теги, которые «понимает» только Web-сервер

Web-страница, которая содержит только клиентские теги, называется **HTML-страницей**. Примером Web-страницы с серверными тегами может служить ASP-страница (Active Server Page, активная серверная страница).

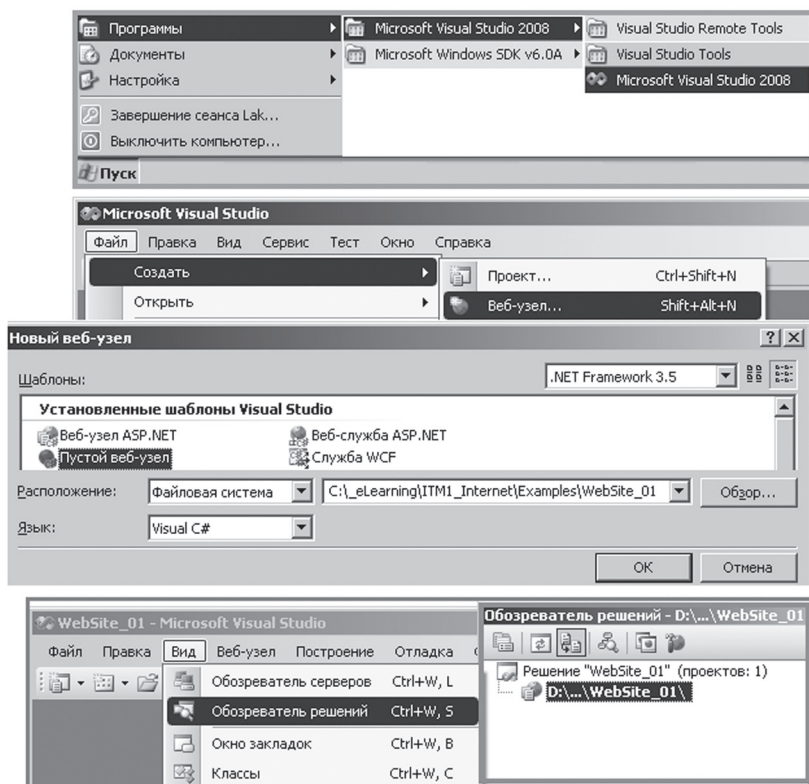


Рис. 1.1. Создание Web-сайта



### 1.1.2. Добавление HTML-страницы

Для добавления HTML-страницы в созданный сайт следует выбрать из меню **Веб-узел** команду **Добавить новый элемент**. На экран будет выведено одноименное диалоговое окно, в котором требуется выполнить следующие действия (Рис.1.2):

- В области **Шаблоны** выбрать шаблон **HTML-страница**
- В поле **Имя** отредактировать имя файла HTML-страницы
- Щелкнуть кнопку **Добавить**

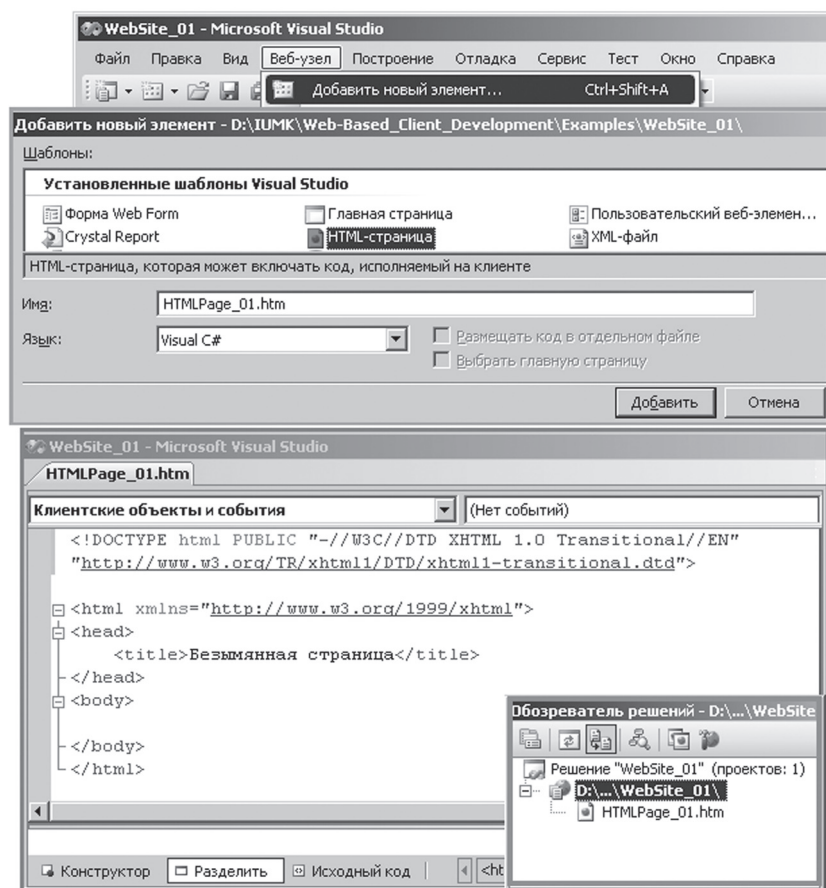


Рис. 1.2. Добавление HTML-страницы

В сайт будет добавлена страница, содержимое которой может быть отображено в одном из трех режимов:

- В режиме **Конструктор** страница показывается в таком виде, как она будет выглядеть в окне Web-браузера
- В режиме **Исходный код** показывается код разметки 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 страницы содержит следующие элементы:

- 1) Директиву **<!DOCTYPE>**, которая задает формат кода разметки (в данном случае – **XHTML 1.0 Transitional**)
- 2) Особые команды, которые заключены в угловые скобки, называются тегами и используются для задания внешнего вида элементов страницы в окне браузера или для указания назначения элементов страницы, в частности:

- **<html></html>** — теги начала и конца кода разметки страницы
- **<head></head>** — теги заголовка страницы
- **<title></title>** — теги названия страницы
- **<body></body>** — теги тела страницы

### 1.1.3. Разметка заголовков

Для выделения заголовков (header) в языке XHTML предусмотрены **теги заголовков**:

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

Пример разметки заголовков (рис. 1.3):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <title>Безымянная страница</title>  
  </head>  
  <body>  
    <h1>Санкт-Петербург</h1>  
    <h2>Санкт-Петербург</h2>  
    <h3>Санкт-Петербург</h3>  
    <h4>Санкт-Петербург</h4>  
    <h5>Санкт-Петербург</h5>  
    <h6>Санкт-Петербург</h6>  
  </body>  
</html>
```

Заметим, что большинство тегов в языке XHTML являются парными и состоят из **открывающего тега** (например, `<h1>`) и **закрывающего тега** (например, `</h1>`). Текст между открывающим и закрывающим тегами называется **содержанием тега**; именно к нему применяется действие тега.

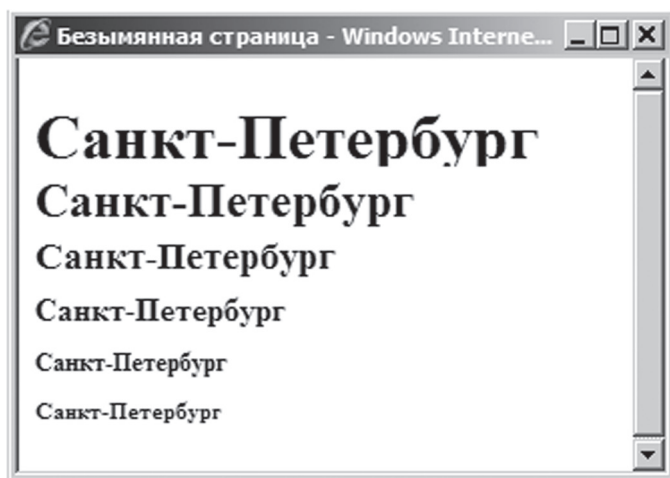


Рис. 1.3. Вид заголовков в окне браузера

### 1.1.4. Разметка абзацев

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

Пример разметки абзацев (рис.1.4):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Создание Санкт-Петербурга</title>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. Петром I в устье Невы была заложена
крепость, позднее получившая название Петропавловской. Спу-
стя год после ее основания на противоположной стороне Невы
была построена верфь – Адмиралтейство.
</p>
<p>На стрелке Васильевского острова затем обосновалась
торговая пристань.</p>
</body>
</html>
```

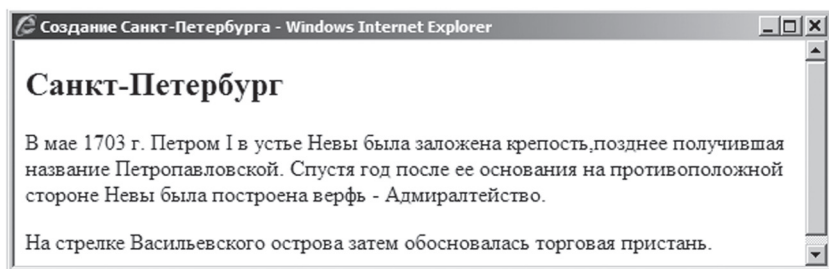


Рис.1.4. Вид абзацев в окне браузера

### 1.1.5. Вложенность тегов

В языке XHTML имеется **два типа тегов. Теги уровня блока** начинают текст с новой строки и могут содержать другие теги. К ним относятся:

- **<h1>/h1>, ..., <h6> /h6>**
- **<p>/p>**

**Встраиваемые теги** не начинают текст с новой строки и могут содержать только текст или другие встраиваемые теги. К ним относятся:

- **<b>/b>** — жирный шрифт (bold)
- **<u>/u>** — подчеркивание (underline)

Когда браузер встречает тег, вложенный в другой тег, он складывает эффект от применения этих тегов. При этом порядок следования закрывающих тегов должен быть обратным тому, в котором следуют открывающие теги. Если это правило нарушено, браузер может не отобразить страницу правильно (хотя Microsoft Internet Explorer славится своим умением исправлять мелкие ошибки).

Пример разметки страницы с вложенными тегами (рис. 1.5):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Создание Санкт-Петербурга</title>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b>Петром I</b> в устье Невы была заложена
крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.
</p>
<p>На стрелке Васильевского острова затем обосновалась
торговая пристань.</p>
</body>
</html>
```

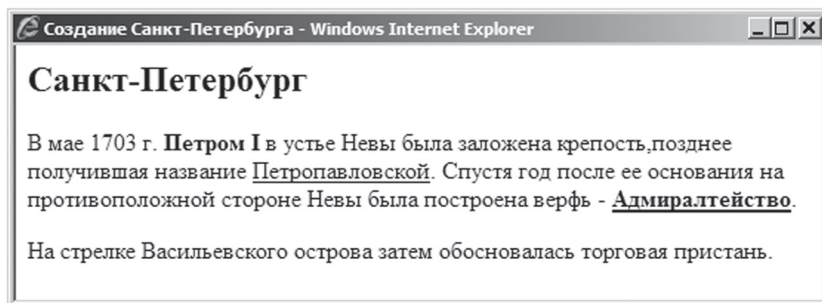


Рис. 1.5. Вид страницы с вложенными тегами на браузере

### 1.1.6. Просмотр страницы в браузере

Для проверки функционирования страницы в окне браузера нужно выполнить следующие действия:

- Щелкнуть правой кнопкой мыши в окне **Обозреватель решений** по имени страницы
- В контекстном меню выбрать команду **Просмотр в обозревателе**

В ответ на это действие инструментальная система Visual Studio выполнит следующие действия (рис. 1.6):

- Запустит встроенный в нее тестовый Web-сервер и отобразит на панели задач значок **ASP.NET Development Server**
- Разместит Web-сайт на тестовом Web-сервере
- Запустит **браузер Microsoft Internet Explorer** на клиентском компьютере и отобразит в нем страницу

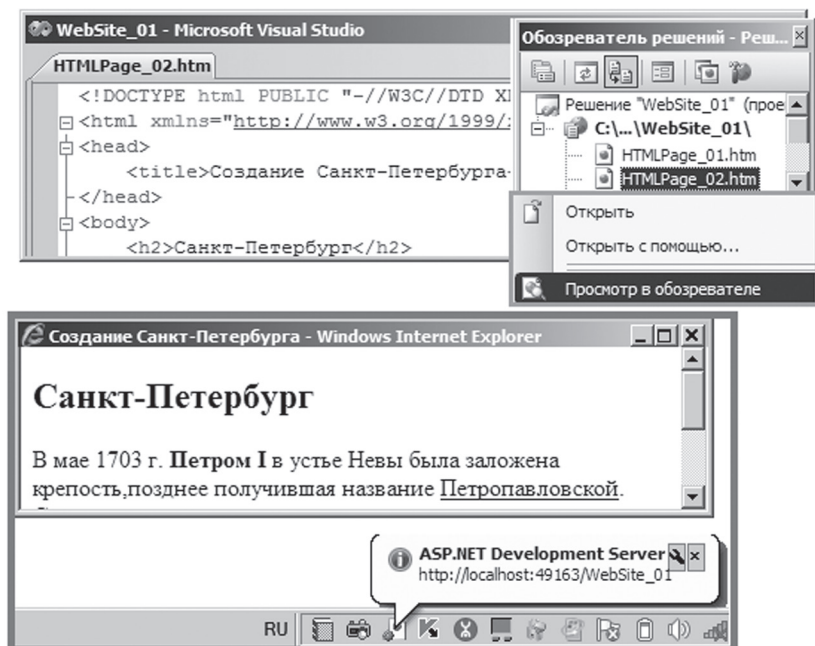


Рис. 1.6. Просмотр страницы на браузере

## 1.2. РАБОТА С ТЕКСТОМ HTML-СТРАНИЦЫ

### 1.2.1. Маркированные списки

**Маркированный список** (unnumbered list) заключается в теги `<ul></ul>`. **Элемент списка** (list item) заключается в теги `<li></li>`. В открывающем теге `<ul>` может быть указан атрибут **style**, параметр **list-style-type** которого может принимать значения:

- disc — элемент списка маркируется кружком (по умолчанию)
- circle — элемент списка маркируется окружностью
- square — элемент списка маркируется квадратом

## Пример разметки маркированных списков (рис.1.7):

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN»
    «http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Маркированные списки</title>
</head>
<body>
Добавление HTML-страницы:
<ul>
<li>В области Шаблоны выберем шаблон HTML-страница</li>
<li>В поле Имя отредактируем имя файла HTML-страницы</li>
<li>Щелкнем кнопку Добавить</li>
</ul>
Параметр <b>list-style-type</b> может принимать значения:
<ul style=»list-style-type:circle»>
<li><b>disc</b> - элемент списка маркируется кружком
(по умолчанию)</li>
<li><b>circle</b> - элемент списка маркируется окружностью</li>
<li><b>square</b> - элемент списка маркируется квадратом</li>
</ul>
</body>
</html>

```

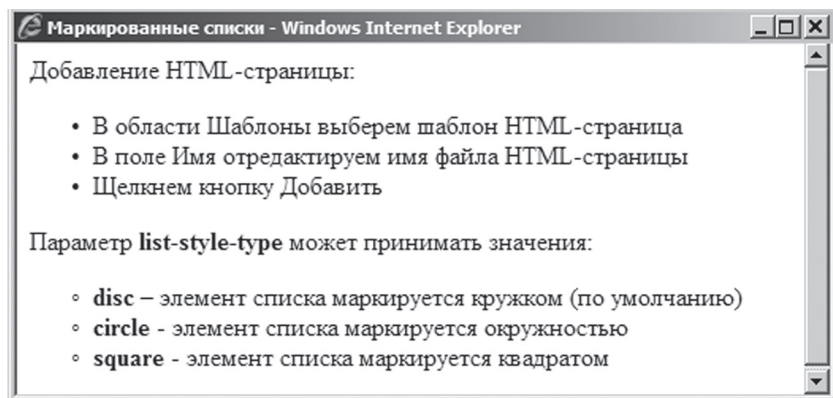


Рис. 1.7. Вид маркированных списков на браузере



### 1.2.2. Нумерованные списки

**Нумерованный список** (ordered list) заключается в теги `<ol></ol>`. **Элемент списка** (list item) заключается в теги `<li></li>`. В открывающем теге `<ol>` может быть указан атрибут **style**, параметр **list-style-type** которого может принимать значения:

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

В открывающем теге `<li>` может быть указан атрибут **value**, значение которого задает номер данного элемента. Последующие элементы будут нумероваться, начиная с заданного номера.

Пример разметки нумерованных списков (Рис.1.8):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Нумерованные списки</title>
</head>
<body>
Создание Web-сайта:
<ol>
<li>Запустим <b>Visual Studio 2008</b></li>
<li>Выберем команду Файл => Создать => Веб-узел</li>
</ol>
В диалоговом окне <u>Новый веб-узел</u>:,
<ol>
<li value="3">В списке <b>Шаблоны</b> выберем <b>Пустой веб-узел</b></li>
<li>В списке <b>Расположение</b> выберем <b>Файловая система</b></li>
</ol>
</body>
</html>
```

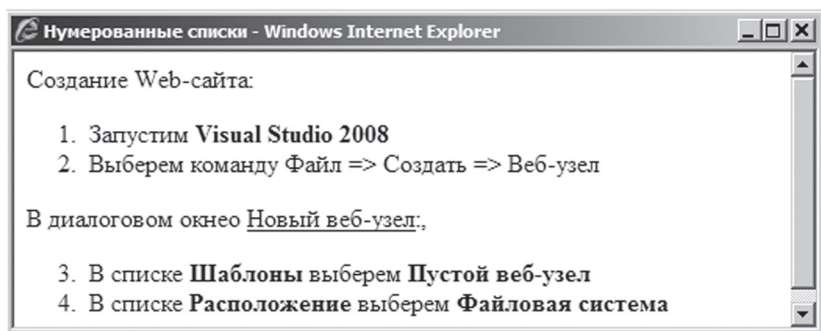


Рис. 1.8. Вид нумерованных списков на браузере

### 1.2.3. Списки определений

**Список определений** (definition list) состоит из термина и его определения и заключается в теги `<dl>` `</dl>`. **Термин** (definition term) заключается в теги `<dt>` `</dt>`. **Определение терминов** (definition description) заключается в теги `<dd>` `</dd>`.

Пример разметки списков определений (Рис.1.9):

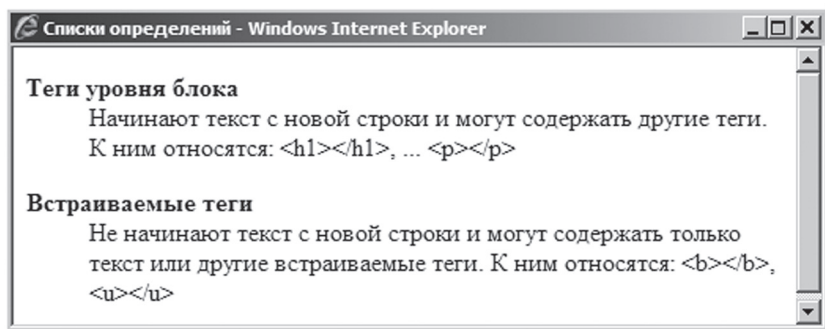


Рис.1.9. Вид списков определений на браузере

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transi-
tional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-tran-
sitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Списки определений</title>
</head>
<body>
```

```

<dl>
<dt><b>Теги уровня блока</b></dt>
<dd>Начинают текст с новой строки и могут содержать
другие теги. К ним относятся: <h1>;</h1>;,
<p>;</p>;</dd>
</dl>
<dl>
<dt><b>Встраиваемые теги</b></dt>
<dd>Не начинают текст с новой строки и могут содержать
только текст или другие встраиваемые теги. К ним относятся:
<b>;</b>;, <u>;</u>;</dd>
</dl>
</body>
</html>

```

### 1.2.4. Управление переносом строк

Браузер автоматически выполняет перенос слов в строке, если строка не помещается в окне браузера. Для принудительного разрыва текста в строке используется тег **<br />** (break). В точке расположения этого тега происходит разрыв текста, и следующий за ним текст начинается с новой строки. Этот тег является **одиночным**, т.е. не имеет парного тега. Перед закрывающей угловой скобкой он содержит символ слеш (/).

Пример разметки переноса строк (Рис.1.10):

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transi-
tional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-tran-
sitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Создание Санкт-Петербурга</title>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b>Петром I</b> в устье Невы была
заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.
<br /><br />
На стрелке Васильевского острова затем обосновалась тор-
говая пристань.</p>
</body>
</html>

```

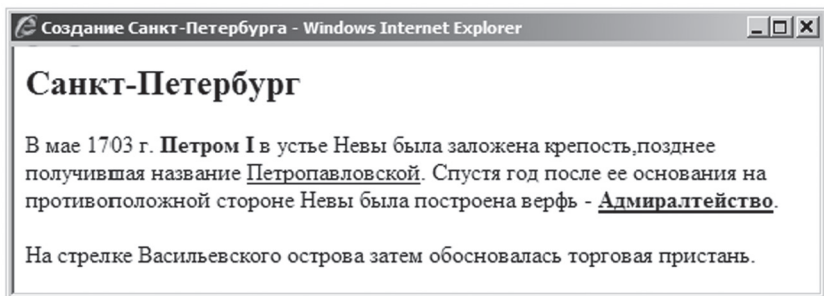


Рис.1.10. Вид переноса строк на браузере

### 1.2.5. Специальные символы

Есть еще один способ управления переносом строк: запрет переноса строки между определенными словами. Для этого достаточно между словами вставить не обычный пробел, а **неразрывный пробел**, для обозначения которого используется специальная кодировка: **&nbsp;**. Существуют и другие символы, для которых используется специальная кодировка:

Символ	Код	Примечание
<	&lt;	
>	&gt;	
неразрывный пробел	&nbsp;	
&	&amp;	
©	&copy;	
®	&reg;	
€	&euro;	
™	&trade;	
«	&quot;	Двойная кавычка
“	&#8220;	Левая кавычка
”	&#8221;	Правая кавычка
—	&#8222;	Длинное тире

Пример разметки специальных символов (рис. 1.11):

```
!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
```



- В открывающем теге **<a>** необходимо указать обязательный атрибут **href**. Его значением является URL-адрес целевой страницы, к которой нужно перейти по гиперссылке (hyper reference)
- В открывающем теге **<a>** может быть указан необязательный атрибут **target**. Если этот атрибут опущен или его значением является «**\_self**», то целевая страница открывается в том же окне, что и исходная. Если значением этого атрибута является «**\_blank**», то целевая страница открывается в новом окне.
- Между тегами **<a></a>** помещают некоторый текст, который называется **гиперссылкой**, выделяется визуально (обычно синим цветом и подчеркиванием) и реагирует на щелчок мыши

Пример создания гиперссылки (рис.1.12):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Создание гиперссылки</title>
</head>
<body>
Создание Web-сайта:
<ol>
<li>Запустим <b>Visual Studio 2008</b></li>
<li>Выберем команду Файл => Создать => Веб-узел</li>
</ol>
В диалоговом окнео <u>Новый веб-узел</u>:,
<ol>
<li value="3">В списке <b>Шаблоны</b> выберем <b>Пустой
веб-узел</b></li>
<li>В списке <b>Расположение</b> выберем <b>Файловая си-
стема</b></li>
</ol>
<a href=»HTMLPage_03.htm» target=»_blank»>Добавление
HTML-страницы</a>
</body>
</html>
```

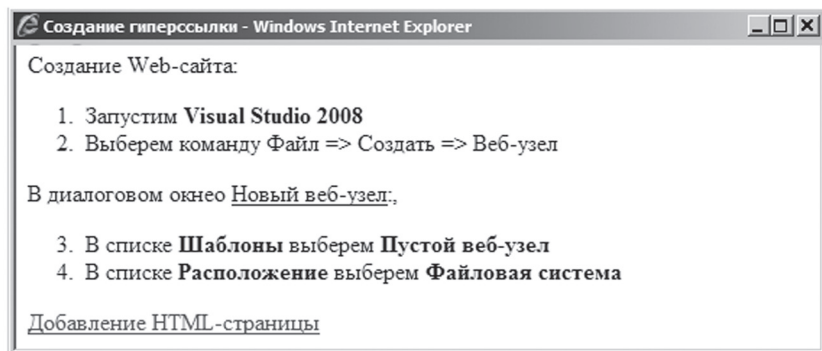


Рис. 1.12. Вид гиперссылки на браузере

### 1.3.2. Функционирование гиперссылки

Гиперссылки работают только при показе страницы в окне браузера (в Visual Studio, в режиме **Конструктор** они не работают). Если страница показана в окне браузера, то щелчок по гиперссылке приводит к переходу по URL-адресу, указанному в атрибуте **href**. Вследствие этого страница, соответствующая указанному URL-адресу, отображается в окне браузера

В данном примере у атрибута **target** было указано значение «**\_blank**», поэтому целевая страница открылась в новом окне (рис. 1.13).

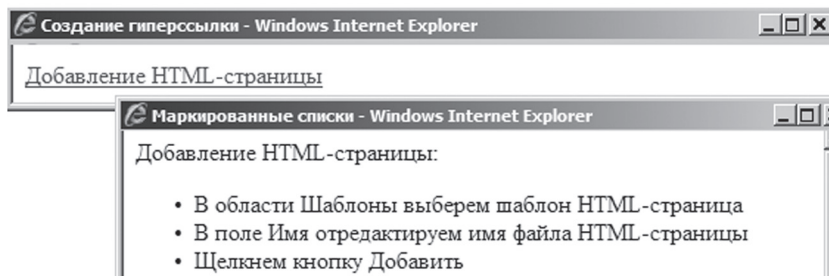


Рис. 1.13. Пример перехода по гиперссылке

### 1.3.3. Адресация в World Wide Web

Адрес Интернет-ресурса (URL — Uniform Resource Locator), указанный в качестве значения атрибута **href**, может быть задан несколькими способами:

- Полный адрес Web-страницы с указанием доменного имени Web-сервера, например: `<a href = "http://www.kerov10.narod.ru/index.htm">`
- Краткий адрес Web-страницы в пределах данного сайта, например: `<a href = "about.htm">`
- Адрес архивного файла, например: `<a href = "News/2010-11_1Year_M3-4.zip">`. При щелчке в окне браузера по этой гиперссылке будет предложено открыть или сохранить архивный файл
- Адрес видео файла, например: `<a href = "Images/Monitor.AVI">`. При щелчке в окне браузера по этой гиперссылке будет запущена программа видеопроигрывателя, установленная на компьютере
- Адрес электронной почты, например: `<a href = mailto:kerov@hse.spb.ru>`. При щелчке в окне браузера по этой гиперссылке будет запущена программа почтового клиента, установленная на компьютере

Пример использования адресов Интернет-ресурсов:

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml«>
<head>
<title>Адресация в World Wide Web</title>
</head>
<body>
<a href=»http://www.kerov10.narod.ru/index.htm«>
www.kerov10.narod.ru</a><br /><br />
<a href=»about.htm«>Об автопе</a><br /><br />
<a href=»News/2010-11_1Year_M3-4.zip«>
Рабочая ведомость преподавателя</a><br /><br />
<a href=»Images/Monitor.AVI«>
Принцип работы монитора</a><br /><br />
<a href=»mailto:kerov@hse.spb.ru«>kerov@hse.spb.ru</a>
</body>
</html>
```



### 1.3.4. Создание закладок

Гиперссылки можно использовать для навигации внутри страницы:

- В тексте страницы создаются **закладки**. Для этого используются теги `<a></a>`, причем в начальном теге `<a>` указывается атрибут **name**, значением которого является имя закладки
- **Гиперссылки на закладки** обычно указываются в начале страницы и создаются при помощи тегов `<a></a>`. При этом в начальном теге `<a>` в качестве значения атрибута **href** указывается имя закладки с префиксом «#»

Пример создания закладок (рис.1.14):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Создание закладок</title>
</head>
<body>
<p> <a href="#1">1. Создание Web-сайта</a> <br />
<a href="#2">2. Добавление HTML-страницы</a>
</p>
<a name="1"></a><h2>1. Создание Web-сайта</h2>
<ol>
<li>Запустим <b>Visual Studio 2008</b></li>
<li>Выберем команду Файл => Создать =>
Веб-узел</li>
</ol>
<a name="2"></a><h2>2. Добавление HTML-страниц</h2>
<ul>
<li>В области <b>Шаблоны</b> выберем шаблон HTML-
страница</li>
<li>В поле <b>Имя</b> отредактируем имя файла HTML-
страницы</li>
<li>Щелкнем кнопку <b>Добавить</b></li>
</ul>
</body>
</html>
```

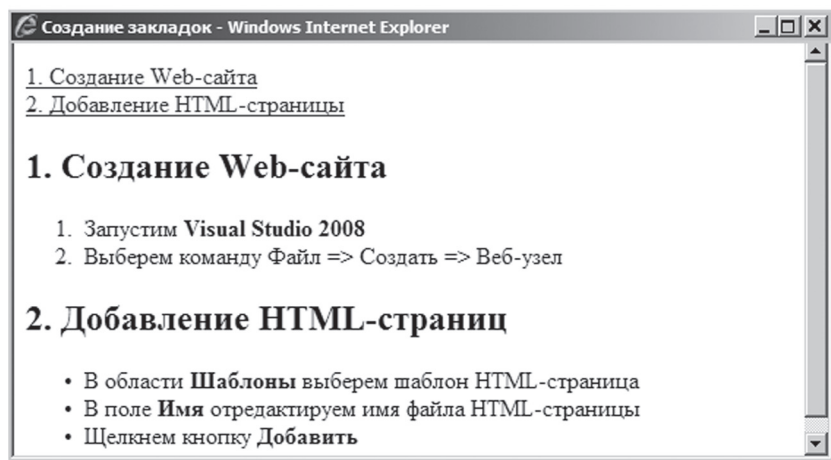


Рис. 1.14. Вид закладок на браузере

## 1.4. РАБОТА С ГРАФИКОЙ HTML-СТРАНИЦЫ

### 1.4.1. Графические файлы

Графические иллюстрации, которые должны присутствовать на странице, хранятся отдельно от текста страницы в виде графических файлов следующих форматов:

- **GIF** — Graphics Interchange Format: этот формат сжимает изображение не очень сильно, но зато обеспечивает лучшее качество и поэтому применяется для хранения элементов оформления страниц (линейки, маркеры списков и т.п.) и штриховых изображений (карты, схемы, рисунки). Данный формат позволяет в одном файле хранить последовательность графических изображений, обеспечивающих эффект анимации.
- **JPG** — Joint Photographic Experts Group: сжимает изображения сильнее, но с некоторой потерей качества; поэтому применяется для полутоновых изображений с большим количеством цветов (фотографии, картины и т.п.), на которых такие потери не очень заметны.

Для хранения графических файлов в папке сайта обычно создают вложенную папку с именем **Images**. После создания папки **Images** в нее копируют заранее подготовленные графические файлы. Чтобы папка **Images** и ее содержимое появились в окне **Обозревателя решений**, следует щелкнуть на панели инструментов этого окна кнопку **Обновить** (рис. 1.15).

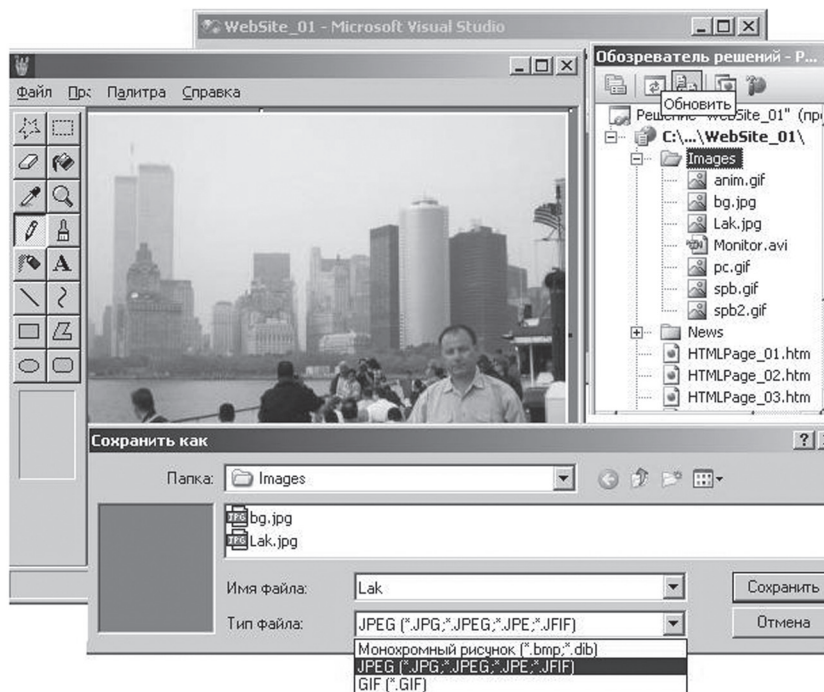


Рис. 1.15. Графические файлы в сайте

### 1.4.2. Вставка изображения

Чтобы поместить на страницу графическое изображение (image), используют одиночный тег **<img />**, имеющий следующие атрибуты:

- **src** — имя графического файла (source);
- **alt** — надпись, которая появляется при наведении указателя мыши на изображение при показе его на браузере (alternate);

- **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 (horizontal space);
- **vspace** — расстояние (в пикселях) между рисунком и текстом по вертикали (vertical space).

Пример вставки изображения (рис. 1.16):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Вставка изображения на страницу</title>
</head>
<body>
<h2>Санкт-Петербург</h2>
<img src=»Images/spb2.gif» alt=»Петропавловская
крпость» align=»left» hspace=»10» vspace=»3» bor-
der=»2»/>
<p>В мае 1703 г. <b>Петром I</b> в устье Невы была
заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.
<br /><br />
На стрелке Васильевского острова затем обосновалась тор-
говая пристань.</p>
</body>
</html>
```

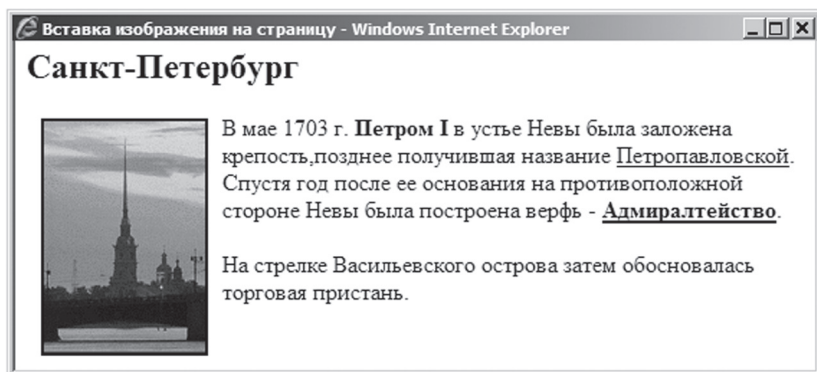


Рис. 1.16. Вид изображения на браузере

### 1.4.3. Изображение-гиперссылка

Между тегами `<a>`/`</a>` обычно указывают некоторый текст, который называется **гиперссылкой**, выделяется визуально и реагирует на щелчок мыши. Вместо текста между тегами `<a>`/`</a>` можно вставить графическое изображение (с помощью тега `<img/>`), которое будет функционировать как **гиперссылка** и будет выделено визуально (обычно синей рамкой).

Изображение-гиперссылка может использоваться для ссылки на внешнее изображение:

Чтобы экономить место на странице, на нее обычно помещают графические иллюстрации небольшого размера.

Если необходимо показать большую иллюстрацию, то на странице создают гиперссылку на соответствующий графический файл.

Пример вставки изображения-гиперссылки (Рис.1.17):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Изображение-гиперссылка</title>
```

```

</head>
<body>
<h2>Санкт-Петербург</h2>
<a href=»Images/spb.gif»> <img src=»Images/spb2.gif»
alt=»Петропавловская крпкость» align=»left» hspace=»10»
vspace=»3» border=»2»/>
</a>
<p>В мае 1703 г. <b>Петром I</b> в устье Невы была
заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.
<br /><br />
На стрелке Васильевского острова затем обосновалась тор-
говая пристань.</p>
</body>
</html>

```

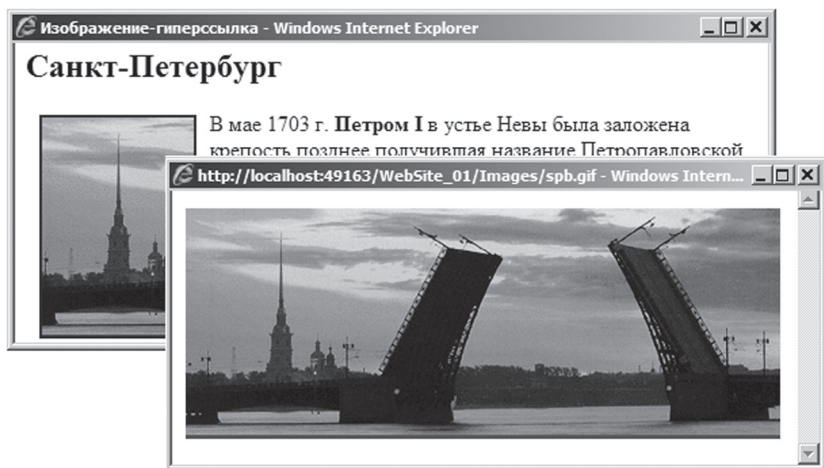


Рис. 1.17. Пример функционирования гиперссылки-изображения на браузере

#### 1.4.4. Создание карт навигации

**Карта навигации** — это обычное изображение на странице, разбитое на части, каждая из которых является гиперссылкой, указывающей на свой Интернет-адрес. Можно считать, что карта навигации — это набор изображений-гиперссылок, объединенных в одном изображении.

Карта навигации состоит из двух частей:

1) **Изображение** — размещается на странице с помощью тега `<img />`, в качестве значения атрибута **usemap** которого указывается имя карты с префиксом «#». Перед префиксом может быть указан Интернет-адрес страницы, на которой определена карта.

2) **Карта** — определяется внутри парных тегов `<map></map>` и задает форму, координаты и размеры изображений-гиперссылок, являющихся частями исходного изображения. В открывающем теге `<map>` карты с помощью атрибута **name** ей присваивается имя, на которое и ссылается исходное изображение с помощью атрибута **usemap**.

Внутри тегов `<map></map>` карты с помощью одиночных тегов `<area />` определяются **изображения-гиперссылки**, являющиеся частями исходного изображения. Для этой цели в теге `<area />` используются следующие атрибуты:

- **href** — его значением является Интернет-адрес, по которому происходит переход после щелчка по данному изображению-гиперссылке;
- **shape** — его значением является идентификатор геометрической формы изображения-гиперссылки: **rect** — прямоугольник, **circ** — круг, **poly** — многоугольник;
- **coords** — его значением являются: для **rect** — координаты левого верхнего и правого нижнего углов прямоугольника; для **circ** — координаты центра и радиус круга; для **poly** — набор координат вершин многоугольника. Координаты задаются в пикселях относительно левого верхнего угла исходного графического изображения карты навигации;
- **target** — если его значением является «**\_self**», то целевая страница открывается в том же окне, что и исходная; если значением этого атрибута является «**\_blank**», то целевая страница открывается в новом окне;
- **alt** — текстовое описание логической области.

Вместо атрибута **href** может быть указан атрибут **nohref**. Он означает, что определяется область, не являющаяся изображением-гиперссылкой. Несколько областей **могут накладываться** друг на друга. При этом на щелчок реагирует та область, которая описана раньше внутри тегов `<map></map>`.

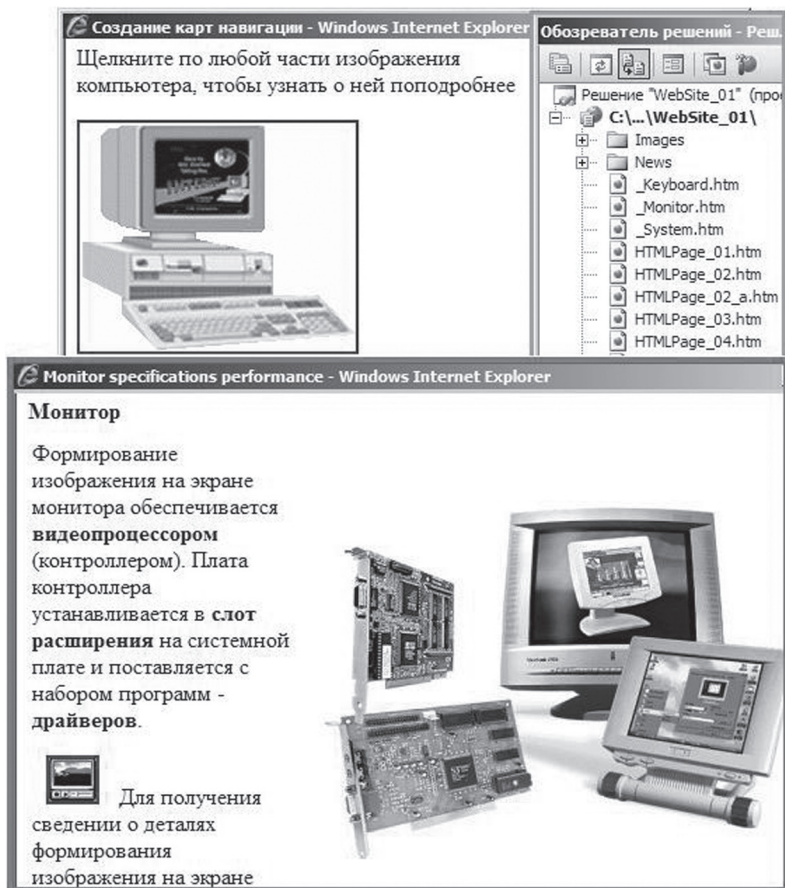


Рис. 1.18. Пример функционирования карты навигации на браузере

Пример разметки для карты навигации (Рис.1.18):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Создание карт навигации</title>
</head>
<body>
```



```

<p>Щелкните по любой части изображения <br />
компьютера, чтобы узнать о ней поподробнее</p>
<img src=»Images/pc.gif» usemap=»#PC» alt=»» />
<map name=»PC» id=»»>
  <area href=»_Monitor.htm» shape=»rect» coords=»17, 4,
132, 76» alt=»Monitor»/>
  <area href=»_System.htm» shape=»rect» coords=»5, 87,
136, 121» alt=»System Block»/>
  <area href=»_Keyboard.htm» shape=»rect» coords=»33,
130, 196, 160» alt=»Keyboard»/>
</map>
</body>
</html>

```

### Пример разметки для страницы **\_Monitor.htm**:

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
  <html xmlns=»http://www.w3.org/1999/xhtml»>
  <head>
    <title>Monitor specifications performance</title>
  </head>
  <body>
    <h4>Монитор</h4>
    <table>
      <tr>
        <td valign=»top»>
          <p>Формирование изображения на экране монитора обеспе-
чивается <b>видеопроцессором</b> (контроллером). Плата
контроллера устанавливается в <b>слот расширения</b> на
системной плате и поставляется с набором программ -
<b>драйверов</b>.</p>
          <a href=»Images/Monitor.AVI»><img src=»Images/mplay.
gif» hspace=»10» alt=»» /></a>
          Для получения сведений о деталях формирования изображе-
ния на экране монитора щелкните изображение видеоплеера.
        </td>
        <td>
          <img src=»Images/Monitor.jpg» hspace=»10» alt=»» />
        </td>
      </tr>
    </table>
  </body>
</html>

```

## 1.5. РАБОТА С ТАБЛИЦАМИ HTML-СТРАНИЦЫ

### 1.5.1. Создание таблицы

**Таблица** имеет следующую разметку:

```
<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>
```

Содержимым ячейки таблицы могут быть:

- Текст, в том числе разделенный на абзацы
- Гиперссылки
- Графические изображения
- Другие таблицы

Пример разметки таблицы:

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Создание таблицы</title>
</head>
<body>
<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>
</body>
</html>
```

### 1.5.2. Атрибуты тега <table>

Тег **<table>** описывает таблицу в целом и имеет следующие атрибуты:

- **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, hside — граница по верхнему и нижнему краю таблицы, lhs — граница по левому краю таблицы, rhs — граница по правому краю таблицы, vside — граница по левому и правому краю таблицы, void — нет границ).

Пример использования атрибутов тега <table> (Рис.1.19):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" <http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Атрибуты тега <table></title>
</head>
<body>
<table border="2" width="60%" align = "left">
<caption>2009 год</caption>
```

```

<tr>
<th align=»center» bgcolor=»yellow»>Месяц</th>
<th align=»center» bgcolor=»silver»>К-во дней</th>
</tr>
<tr><td>Январь</td><td>31</td></tr>
<tr><td>Февраль</td><td>28</td></tr>
</table>
</body>
</html>

```

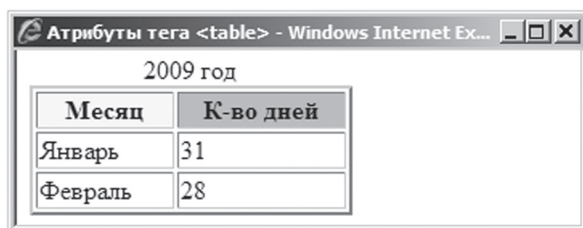


Рис. 1.19. Отображение атрибутов тега <table> на браузере

### 1.5.3. Атрибуты тегов <tr> и <td>

Тег <tr> описывает строку таблицы и имеет атрибуты:

- **align** — выравнивание ячеек строки по горизонтали (left, right, center (middle));
- **valign** — выравнивание ячеек строки по вертикали (top, bottom, center (middle));
- **bgcolor** — цвет фона строки;

Тег <td> описывает ячейку таблицы и имеет атрибуты:

- **align** — выравнивание содержимого ячейки по горизонтали (left, right, center (middle));
- **valign** — выравнивание содержимого ячейки по вертикали (top, bottom, center (middle));
- **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.

Пример использования атрибутов тега `<tr>` и `<td>` (Рис.1.20):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtmll»>
<head>
<title>Атрибуты тегов <tr> и <td>;</title>
</head>
<body>
<table border=»2» width=»60%» align = «left»>
<caption>2009 год</caption>
<tr>
<th align=»center» bgcolor=»yellow»>Месяц</th>
<th align=»center» bgcolor=»silver»>К-во дней</th>
</tr>
<tr>
<td>Январь</td><td>31</td>
</tr>
<tr align=»center» valign=»middle» bgcolor=»aqua»>
<td>Февраль</td><td>28</td>
</tr>
<tr>
<td colspan=»2» align=»center»>невисокосный <br/>год</td>
</tr>
</table>
</body>
</html>
```

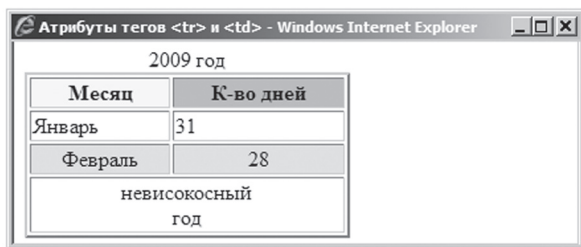


Рис.1.20. Отображение атрибутов тегов `<tr>` и `<td>` на браузере

### 1.5.4. Объединение ячеек в строке

Для объединения нескольких ячеек по горизонтали в одну нужно выполнить следующие действия:

- Найти тег **<td>** (**<th>**), соответствующий первой из объединяемых ячеек
- Добавить в него атрибут **colspan** и указать в качестве его значения количество объединяемых ячеек
- Удалить в данной строке теги **<td>**/**<td>** (**<th>**/**<th>**), остальных объединяемых ячеек

Пример разметки для объединения ячеек в строке (Рис.1.21):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Объединение ячеек в строке</title>
</head>
<body>
<table border=»1»>
<tr><td> 11 </td><td> 12 </td><td> 13 </td><td> 14 </td></tr>
<tr><td> 21 </td><td> 22 </td><td> 23 </td><td> 24 </td></tr>
<tr><td> 31 </td><td> 32 </td><td> 33 </td><td> 34 </td></tr>
<tr><td> 41 </td><td> 42 </td><td> 43 </td><td> 44 </td></tr>
</table><br />
<table border=»1»>
<tr>
<td> 11 </td><td bgcolor=»#ffccff» colspan=»3»> 12 + 13 + 14 </td>
</tr>
<tr><td> 21 </td><td> 22 </td><td> 23 </td><td> 24 </td>
</tr>
<tr>
<td> 31 </td><td> 32 </td><td bgcolor=»#ffccff» colspan=»2»> 33 + 34 </td>
</tr>
```

```

<tr>
<td> 41 </td><td> 42 </td><td> 43 </td><td> 44 </td>
</tr>
</table>
</body>
</html>

```

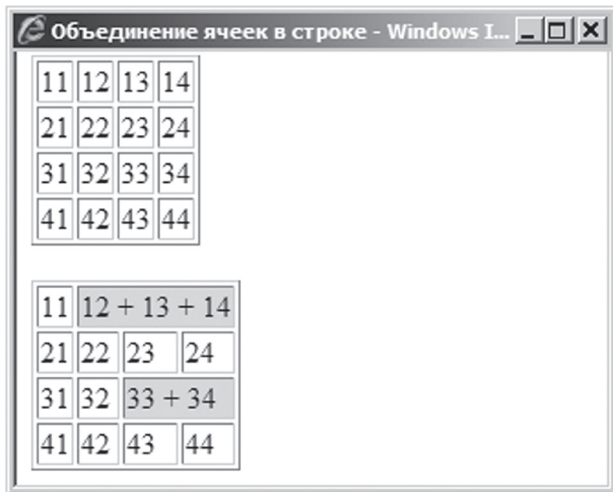


Рис. 1.21. Отображение результата объединения ячеек в строке на браузере

### 1.5.5. Объединение ячеек в столбце

Для объединения нескольких ячеек по вертикали в одну нужно выполнить следующие действия:

- Найти тег **<td>**, соответствующий первой из объединяемых ячеек
- Добавить в него атрибут **rowspan** и указать в качестве его значения количество объединяемых ячеек
- Удалить в данном столбце теги **<td></td>**, остальных объединяемых ячеек

## Пример разметки для объединения ячеек в столбце (рис. 1.22):

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transition-
al//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.
dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Объединение ячеек в столбце</title>
</head>
<body>
<table border=»1» title=»Это исходная таблица»>
<tr><td> 11 </td><td> 12 </td><td> 13 </td><td> 14 </
td></tr>
<tr><td> 21 </td><td> 22 </td><td> 23 </td><td> 24 </
td></tr>
<tr><td> 31 </td><td> 32 </td><td> 33 </td><td> 34 </
td></tr>
<tr><td> > 41 </td><td> 42 </td><td> 43 </td><td> 44 </
td></tr>
</table>
<hr style="width:80%" title="Это горизонтальная линия"/>
<table border=»1»>
<tr>
<td> 11 </td><td bgcolor=»#ffccff» colspan=»3»> 12 + 13
+ 14 </td>
</tr>
<tr>
<td rowspan=»2» bgcolor=»#ffff99» title=»Объединенная
ячейка»> 21+31</td>
<td>22</td><td>23</td><td>24</td>
</tr>
<tr>
<td> 32 </td><td bgcolor="ffccff" colspan="2"> 33 + 34
</td> </tr>
<tr>
<td> 41 </td><td> 42 </td><td> 43 </td><td> 44 </td>
</tr>
</table>
</body>
</html>

```



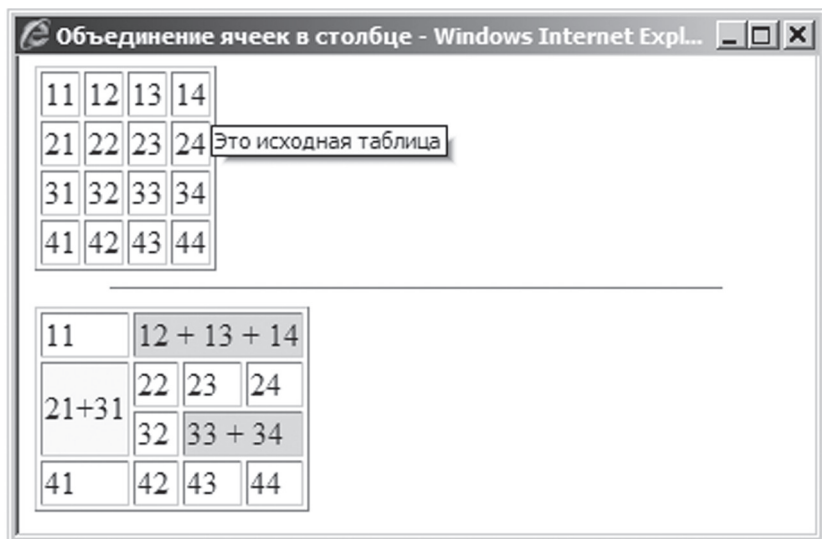


Рис. 1.22. Отображение результата объединения ячеек в столбце на браузере

## 1.6. Всплывающие подсказки

Теги языка XHTML поддерживают необязательный атрибут **title**, значением которого является некоторый текст, появляющийся в виде всплывающей подсказки при наведении указателя мыши на соответствующий тегу визуальный элемент (рис. 1.22).

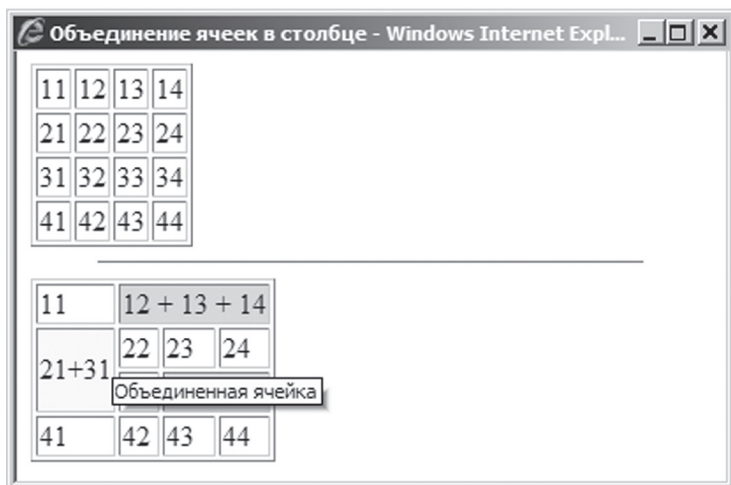


Рис. 1.22. Отображение всплывающей подсказки на браузере  
(предыдущая разметка)

## 1.7. Горизонтальные линии

**Горизонтальные линии** предназначены для разграничения каких-либо фрагментов страницы и задаются с помощью тега **<hr/>** (рис. 1.23).

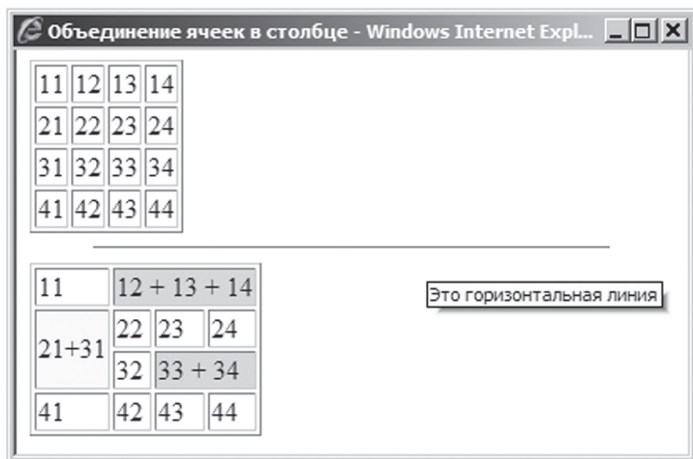


Рис. 1.23. Отображение горизонтальной линии на браузере  
(предыдущая разметка)

## 1.8. КОММЕНТАРИИ

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

Пример разметки комментария (рис. 1.24):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional-
al//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Объединение ячеек в столбце</title>
</head>
<body>
<!--table border="1" title="Это исходная таблица">
<tr><td> 11 </td><td> 12 </td><td> 13 </td><td> 14 </
td></tr>
<tr><td> 21 </td><td> 22 </td><td> 23 </td><td> 24 </
td></tr>
<tr><td> 31 </td><td> 32 </td><td> 33 </td><td> 34 </
td></tr>
<tr><td> 41 </td><td> 42 </td><td> 43 </td><td> 44 </
td></tr>
</table>
<hr style="width:80%" title="Это горизонтальная ли-
ния"/-->
<table border="1">
<tr>
<td> 11 </td><td bgcolor="#ffccff" colspan="3"> 12 + 13
+ 14 </td>
</tr>
<tr>
<td rowspan="2" bgcolor="#ffff99" title="Объединенная
ячейка"> 21+31</td>
<td>22</td><td>23</td><td>24</td>
</tr>
<tr>
<td> 32 </td><td bgcolor="#ffccff" colspan="2"> 33 + 34
</td> </tr>
<tr>
<td> 41 </td><td> 42 </td><td> 43 </td><td> 44 </td>
</tr>
</table>
</body>
</html>
```

11	12 + 13 + 14
21+31	22 23 24
41	32 33 + 34
41	42 43 44

Рис. 1.24. Влияние комментария на отображение на браузере

## 1.9. Пролог HTML-страницы

**Пролог Web-страницы** предшествует любому XHTML-коду, задается с помощью одиночного тега `<!DOCTYPE>` и используется для указания версии языка XHTML, например (рис. 1.25):

- **Transitional** — позволяет использовать все возможности XHTML, кроме фреймов
- **Frameset** — добавляет к Transitional возможность использования фреймов

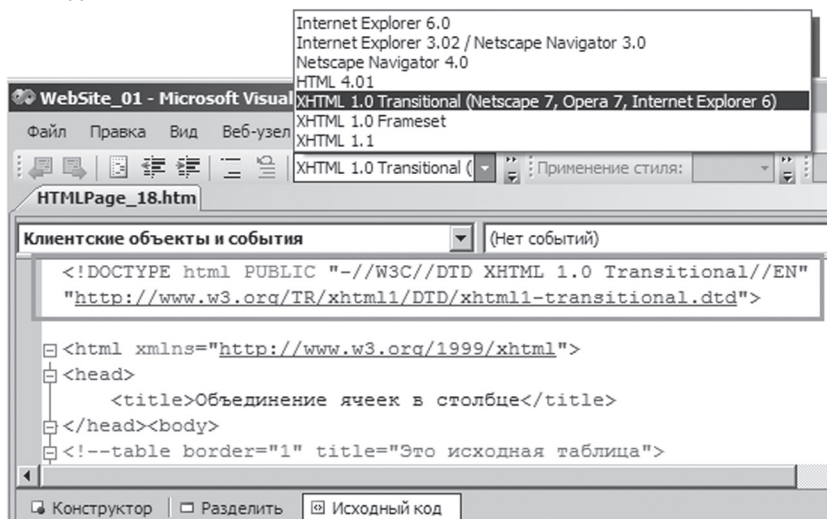


Рис. 1.25. Пролог HTML-страницы

## 1.10. РАБОТА С ФРЕЙМАМИ HTML-СТРАНИЦЫ

### 1.10.1. Разделение окна браузера

Окно браузера можно разделить на прямоугольные области, называемые **фреймами**. В каждый фрейм можно загрузить ту или иную страницу. Например, можно разделить окно браузера на три вертикальных области, вторую из которых разделить на две горизонтальных области. Для создания фреймов необходимо использовать версию **Frameset** языка XHTML (она расширяет версию XHTML Transitional возможностью использования фреймов).

Фреймовая структура окна браузера задается страницей, у которой вместо раздела `<body></body>` указывается раздел `<frameset></frameset>`. Для указания обозревателю вертикального расположения фреймов и их количества используется атрибут **rows** тега `<frameset>`. Его значением является набор размеров фреймов, разделенным запятыми; при этом:

- Число задает размер фрейма в пикселях
- Число с символом процента задает размер фрейма в процентах от размера страницы
- Символ «звездочка» (\*) указывает, что для соответствующего ей фрейма нужно отвести все свободное пространство, не занятое другими фреймами

Для создания самого фрейма внутри тегов `<frameset></frameset>` используется одиночный тег `<frame/>`. Если некоторый фрейм нужно разбить на несколько фреймов, то для него вместо тега `<frame/>` указываются теги `<frameset></frameset>` (сложенными тегами `<frame/>`). Для указания обозревателю горизонтального расположения фреймов и их количества используется атрибут **cols** тега `<frameset>`.

Пример разметки для разделения окна браузера на фреймы (рис.1.26):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
<http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Фреймовая структура</title>
</head>
```

```

<frameset rows=»70, *, 50»>
<frame/>
<frameset cols=»25%,75%»>
<frame/>
<frame/>
</frameset>
</frameset>
</html>

```

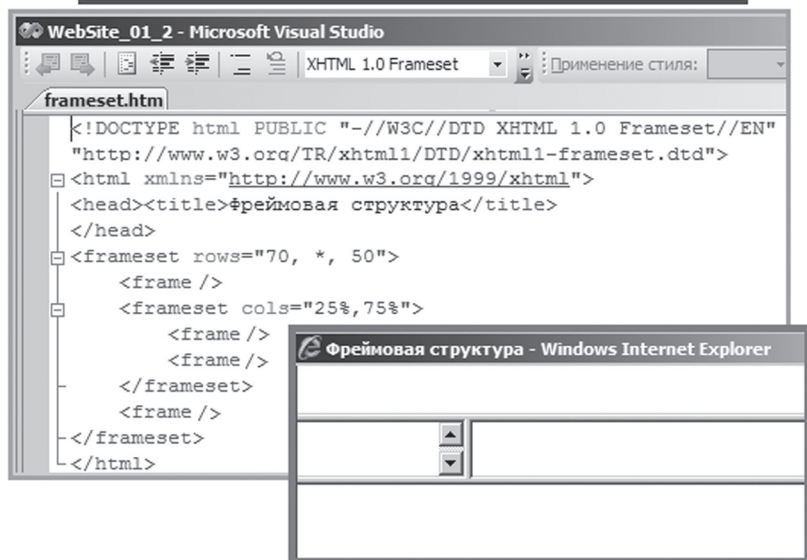


Рис. 1.26. Разделение окна браузера на фреймы

### 1.10.2. Задание параметров фреймов

У каждого фрейма должен быть указан обязательный атрибут **src**, значением которого является адрес страницы, которая будет загружена в этот фрейм изначально. У фрейма также может быть указан атрибут **name**, задающий имя фрейма.

Пример разметки с указанием параметров фреймов:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd»>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Фреймовая структура</title>
</head>
<frameset rows="70, *, 50">
<frame src="header.htm"/>
<frameset cols="25%,75%">
<frame src="menu.htm"/>
<frame src="empty.htm" name="content"/>
</frameset>
<frame src="footer.htm"/>
</frameset>
</html>
```

### 1.10.3. Структура загружаемых страниц

Страница, загружаемая во фрейм верхнего колонтитула, страница, загружаемая во фрейм нижнего колонтитула, и страница, загружаемая во фрейм содержимого Web-страниц,- это обычные Web-страницы. Загрузка их во фреймы никак не отражается на их оформлении.

Пример разметки страницы header/htm:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<h1 align="center">Языки программирования</h1>
</body>
</html>
```

**Пример разметки страницы footer.htm:**

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title></title>
</head>
<body>
&copy; Leonid A/ KEROV, 2011
</body>
</html>
```

**Пример разметки страницы empty.htm:**

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title></title>
</head>
<body>
</body>
</html>
```

**Пример разметки страницы lang1.htm:**

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.
dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Язык C </title>
</head>
<body>
Язык C разработан в 1970 году в компании Bell Lab; ав-
тор – Dennis Ritchie
</body>
</html>
```



### Пример разметки страницы lang2.htm:

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml«>
<head>
<title>Язык C++</title>
</head>
<body>
Язык C++ разработан в 1979 году в компании Bell Lab;
автор - Bjarne Stroustrup
</body>
</html>

```

### Пример разметки страницы lang3.htm:

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Frameset//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml«>
<head>
<title>Язык C#</title>
</head>
<body>
Язык C# разработан в 2000 году в компании Microsoft; ав-
тор - Andres Hejlsberg
</body>
</html>

```

#### 1.10.4. Страница с гиперссылками

Страница, загружаемая во фрейм гипертекстового меню, имеет следующую особенность оформления. Теги **<a>** содержат атрибут **target**, в качестве значения которого указывается имя (значение атрибута **name**) фрейма, в который должна быть загружена страница, указанная в атрибуте **href**. В качестве значения атрибута **target** можно указать следующие стандартные имена для фреймов:

- **\_blank** — страница загружается в новое окно;
- **\_self** — страница загружается в текущий фрейм (то же самое происходит, если атрибут опущен **target**);
- **\_top** — страница загружается во все окно браузера;

- **\_paren** — страница загружается в область, занимаемую фреймом — родителем.

Пример разметки страницы с гиперссылками - menu.htm (рис. 1.27):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»»
<html xmlns=»http://www.w3.org/1999/xhtml1«»
<head>
<title>Меню</title>
</head>
<body>
<a href=»lang1.htm» target=»content»>Язык C</a><br />
<a href=»lang2.htm» target=»content»>Язык C++</a><br />
<a href=»lang3.htm» target=»content»>Язык C#</a><br />
</body>
</html>
```

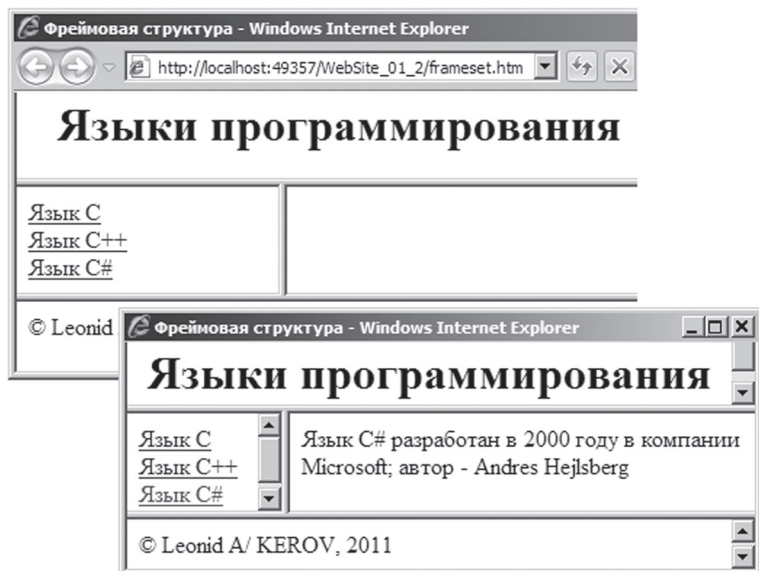


Рис.1.27. Первоначальная загрузка страницы *empty.htm*, затем — *lang3.htm*

### 1.10.5. Навигация внутри страницы

Для навигации внутри Web-страницы используется следующая фреймовая структура. Экран браузера разделяется на два фрейма (рис.1.28):

- Один из фреймов в окне браузера содержит «меню» в виде перечня гиперссылок, каждая из которых содержит ссылку на закладку внутри Web-страницы (**href**) и имя фрейма для загрузки (**target**);
- Другой фрейм содержит саму Web-страницу с закладками.

Пример разметки страницы frameset.htm:

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Frameset//
EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Навигация внутри страницы</title>
</head>
<frameset rows=»70, *, 50»>
<frame src=»header.htm»/>
<frameset cols=»25%,75%»>
<frame src=»menu.htm» />
<frame src=»languages.htm» name=»content» />
</frameset>
<frame src=»footer.htm»/>
</frameset>
</html>
```

Пример разметки страницы menu.htm:

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Меню</title>
</head>
<body>
<a href=»languages.htm#lang1» target=»content»>Язык C/>
a><br />
<a href=»languages.htm#lang2» target=»content»>Язык
C++</a><br />
```

```
<a href=»languages.htm#lang3» target=»content»>Язык
C#</a><br />
</body>
</html>
```

### Пример разметки страницы languages.htm:

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Content</title>
</head>
<body>
<p> [ <a href=»#lang1»>Язык C</a> ]
    [ <a href=»#lang2»>Язык C++</a> ]
    [ <a href=»#lang3»>Язык C#</a> ] </p>
<p><a name=»lang1»>Язык C</a> разработан в 1970 году<br/>
в компании Bell Lab; автор - Dennis Ritchie</p>
<p><a name=»lang2»>Язык C++</a> разработан в 1979 году<br/>
в компании Bell Lab; автор - Bjarne Stroustrup </p>
<p><a name=»lang3»>Язык C#</a> разработан в 2000 году<br/>
в компании Microsoft; автор - Andres Hejlsberg </p>
</body>
</html>
```

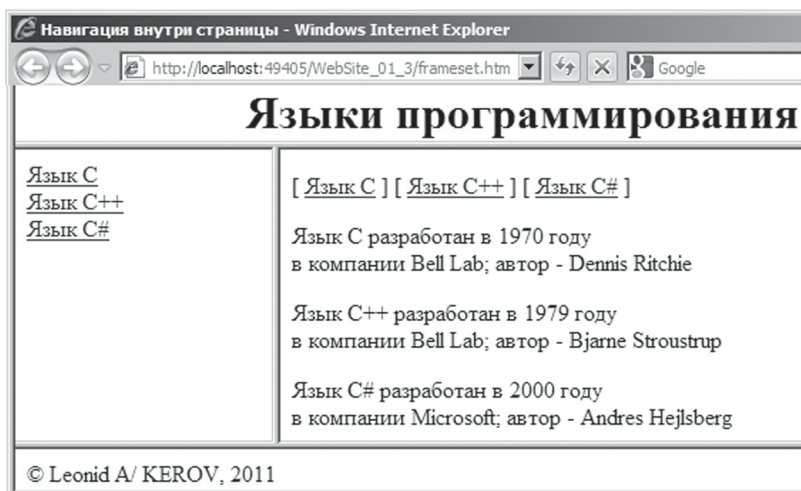


Рис. 1.28. Навигация внутри страницы

### Задания для практических занятий и самостоятельной работы

Для изучения материала первой главы в тематическом плане дисциплины предусмотрено 28 часов:

Название тем	Всего часов	Аудиторные часы		Самостоятельная работа
		Лекции	Практические занятия	
Раздел 1. Создание и использование в управленческой практике персональных и корпоративных Web-сайтов				
Тема 1.1. Создание Web-сайта в Microsoft Visual Studio. Размещение сайта в Интернете	28	4	4	20

**Задание № 1.1.1.** Выбор темы научно-исследовательского проекта, проведение информационного поиска по этой теме в Интернете и оформление результатов поиска в виде конспекта, выполняющего роль контента для создания Web-сайта.

Конспект оформляется в редакторе Microsoft Office Word 2007, наличие формулировки темы, оглавления и графических иллюстраций обязательно. Объем конспекта не регламентируется; он может составлять, например, от **30 000** до **70 000** знаков, включая пробелы. В конце конспекта должен быть приведен список использованных источников.

### Примерная тематика научно-исследовательских проектов:

Тема научно-исследовательского проекта	Базовый информационный источник
Исследование технологии <sup>1</sup> облачных вычислений (Cloud computing)	[1]
Исследование технологии коллективного интеллекта (Collective intelligence)	[2]
Исследование технологии электронного государства (eGovernment)	[3]
Исследование технологии электронного обучения (eLearning)	[4]
Исследование технологии JavaScript	[5]
Исследование технологии Library 2.0	[6]
Исследование технологии офисных пакетов (Office suite)	[7]
Исследование технологии Science 2.0	[8]
Исследование технологии Web 2.0	[9]

Тема научно-исследовательского проекта	Базовый информационный источник
Анализ рынка образовательных услуг Web-обучения в секторе K-12 США и источники его эффективности (школьное образование: primary school + secondary school)	[10]
Анализ рынка образовательных услуг Web-обучения в секторе P-2 США и источники его эффективности (высшее образование: post-secondary school)	[10]
Анализ рынка образовательных услуг Web-обучения в секторе CT США и источники его эффективности (повышение квалификации: corporate training)	[10]
Анализ рынка образовательных услуг Web-обучения в секторе LL США и источники его эффективности (непрерывное образование: lifelong learning)	[10]

### Базовые информационные источники:

1. Cloud computing. Интернет-публикация. — [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing).
2. Collective intelligence: Интернет-публикация. — [http://en.wikipedia.org/wiki/Collective\\_intelligence](http://en.wikipedia.org/wiki/Collective_intelligence).
3. e-Government: Интернет-публикация. — <http://en.wikipedia.org/wiki/E-Government>.
4. e-Learning: Интернет-публикация. — <http://en.wikipedia.org/wiki/E-learning>.
5. JavaScript: Интернет-публикация. — <http://en.wikipedia.org/wiki/JavaScript>.
6. Library 2.0: Интернет-публикация. — [http://en.wikipedia.org/wiki/Library\\_2.0](http://en.wikipedia.org/wiki/Library_2.0).
7. Office suite: Интернет-публикация. — [http://en.wikipedia.org/wiki/Office\\_suite](http://en.wikipedia.org/wiki/Office_suite).
8. Science 2.0: Интернет-публикация. — [http://en.wikipedia.org/wiki/Science\\_2.0](http://en.wikipedia.org/wiki/Science_2.0).
9. Web 2.0: Интернет-публикация. — [http://en.wikipedia.org/wiki/Web\\_2.0ru](http://en.wikipedia.org/wiki/Web_2.0ru).
10. Close R.C., Humphreys R., Rutenbug B.W. E-Learning and Knowledge Technology. Technology & The Internet Are Changing the Way We Learn. — Sun Trust Equitable Securities, March 2000. — 190 pp.

### Порядок передачи результатов выполнения задания №1.1.1 для их оценивания:

- Следует послать письмо преподавателю по электронной почте.
- В качестве темы письма нужно указать номер подгруппы, фамилию и инициалы студента; например: 221В, Иванов А.С.
- К письму следует прикрепить zip-архив файла конспекта.
- Максимальная оценка выполнения задания №1.1.1 составляет 1 балл.

**Задание № 1.1.2.** Используя инструментальную систему Microsoft Visual Studio 2008, создайте пустой Web-сайт. Затем выполните следующие действия:

- Добавьте в сайт несколько HTML-страниц, рассмотренных в данной теме (может быть, все).
- Добавьте в сайт HTML-страницу с именем **index.htm** и разместите на ней гиперссылки на остальные HTML-страницы.
- Разместите созданный сайт в Интернете на сервере **www.narod.ru**. Регистрацию доменного имени сайта рекомендуется выполнять не из компьютерного класса. После регистрации сайта с ним можно работать с любого компьютера, в том числе из компьютерного класса.

**Порядок передачи результатов выполнения задания №1.1.2 для их оценивания:**

- Следует послать письмо преподавателю по электронной почте.
- В качестве темы письма нужно указать номер подгруппы, фамилию и инициалы студента; например: 221В, Иванов А.С.
- В тексте электронного письма следует указать доменное имя, разработанного студентом сайта; например: [www.ivanov.narod.ru](http://www.ivanov.narod.ru).
- Максимальная оценка выполнения задания №1.1.2 составляет 1 балл.

## ГЛАВА 2. СТИЛЕВОЕ ОФОРМЛЕНИЕ HTML-СТРАНИЦ

### 2.1. Понятие и виды стилей

#### 2.1.1. Встроенный стиль

При создании Web-страниц можно использовать различные параметры форматирования, которые определяют внешний вид этих страниц. Теги языка XHTML поддерживают необязательный атрибут **style**, который позволяет применить параметры форматирования непосредственно к этому тегу, например:

`<p style=»font-family: Arial; font-size: small; color: Green»>`

Такой стиль называется **встроенным**. Его значением является множество пар вида **параметр : значение**, которые заключаются в двойные кавычки и отделяются друг от друга точкой с запятой.

Пример применения встроенного стиля для второго абзаца (Рис.2.1):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title></title>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b>Петром I</b> в устье Невы была
заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.</p>
<p style="font-family:Arial; font-size:small;
```



```
color:Navy">На стрелке Васильевского острова затем обо-
сновалась торговая пристань.</p>
</body>
</html>
```

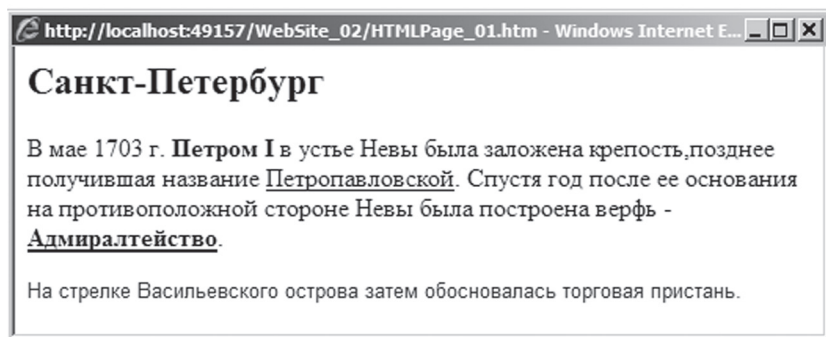


Рис. 2.1. Результат применения встроенного стиля ко второму абзацу

### 2.1.2. Внутренняя таблица стилей

Если параметры форматирования нужно применить ко всем тегам страницы, то их определение помещается во **внутреннюю таблицу стилей**, которая определяется с помощью парных тегов `<style></style>` и размещается в заголовке страницы. В такой таблице можно **переопределить стиль** для того или иного тега.

Переопределение стиля:

- Начинается с **указания тега**, для которого стиль переопределяется.
- Состоит из пар вида **параметр : значение**, которые заключаются в фигурные скобки и отделяются друг от друга точкой с запятой.

Пример переопределения стилей для абзацев и заголовка (Рис.2.2):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title></title>
<style type=»text/css»>
p {font-family:Arial; font-size:small; color:Navy}
```

```

h2 { background-color:Navy; color:White}
</style>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b>Петром I</b> в устье Невы была
заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.</p>
<p>На стрелке Васильевского острова затем обосновалась
торговая пристань.</p>
</body>
</html>

```

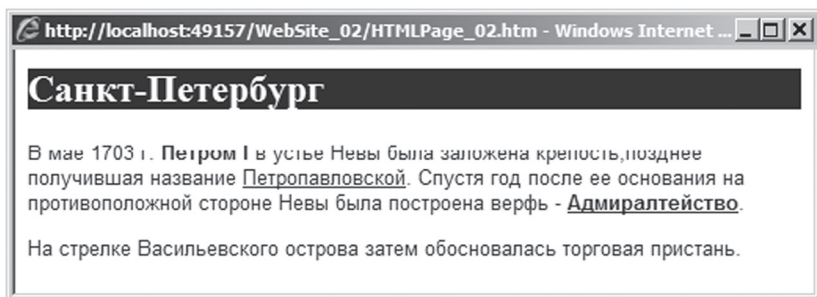


Рис. 2.2. Результат переопределения стилей для абзацев и заголовка

### 2.1.3. Стилиевой класс

Предположим, что во втором абзаце необходимо заменить шрифт на Times New Roman. Для этой цели можно в таблице стилей определить стилевой класс, который:

- Начинается с символа «точка» (.), после которого указывается имя класса (в данном случае — `tnrText`)
- После имени стилевого класса, через пробел, в фигурных скобках указываются пары вида **параметр : значение**, которые отделяются друг от друга точкой с запятой

Стилиевой класс может быть применен к любому тегу. Для этого после имени тега указывается атрибут **class**, значением которого является имя стилевого класса.

Пример определения и применения стилового класса (рис.2.3):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title></title>
<style type=»text/css»>
p {font-family:Arial; font-size:small; color:Navy}
h2 { background-color:Navy; color:White}
.tnrText { font-family:Times New Roman}
</style>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b>Петром I</b> в устье Невы была
заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.</p>
<p class=»tnrText»На стрелке Васильевского острова за-
тем обосновалась торговая пристань.</p>
</body>
</html>
```

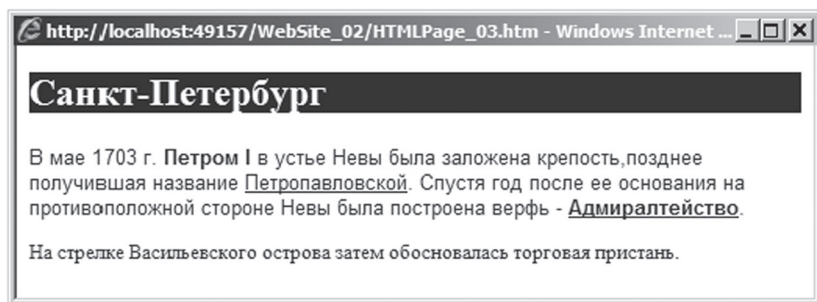


Рис. 2.3. Результат применения стилового класса

### 2.1.4. Стилиевой селектор

Определение стилового селектора отличается от определения стилового класса только тем, что начинается с символа «решетка» (#), а не с символа «точка» (.). Стилиевой селектор может быть применен к любому тегу. Для этого после имени тега указывается атрибут **id**, значением которого является имя стилового селектора.

Заметим, что значение атрибута **id** должно быть уникальным в пределах Web-страницы. Стилиевые селекторы обычно применяются, если какой-то стиль нужно применить к одному тегу на странице.

Пример определения и применения стилового селектора (Рис.2.4):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd»>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<style type="text/css">
p {font-family:Arial; font-size:small; color:Navy}
h2 { background-color:Navy; color:White}
.tnrText { font-family:Times New Roman}
#redText { color:Red; font-size:larger}
</style>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b id="redText">Петром I</b> в устье
Невы была заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.</p>
<p class="tnrText">На стрелке Васильевского острова за-
тем обосновалась торговая пристань.</p>
</body>
</html>
```

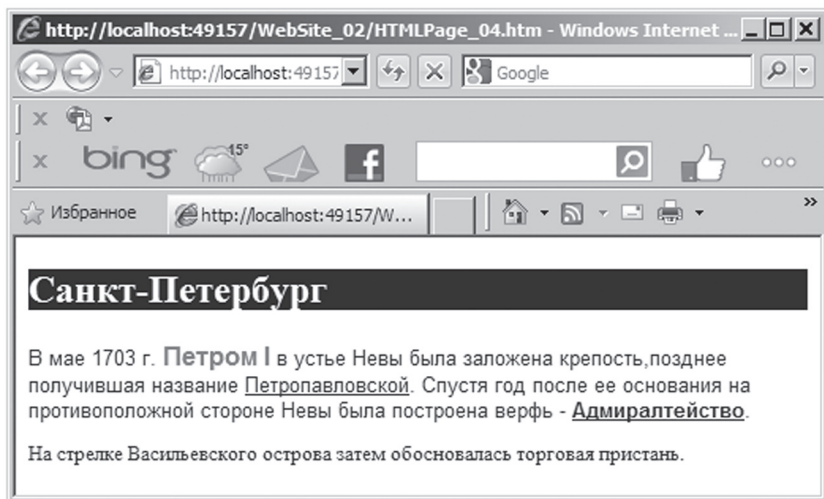


Рис. 2.4. Результат применения стилового селектора

### 2.1.5. Стилъ для нескольких тегов

В таблице стилей можно переопределить стиль сразу для нескольких тегов. Для этого перед определением стиля **через запятую** перечисляются имена тегов, для которых стиль переопределяется.

Пример определения и применения стиля для нескольких тегов (Рис.2.5):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title></title>
<style type=»text/css»>
p {font-family:Arial; font-size:small; color:Navy}
h2 { background-color:Navy; color:White}
.tnrText { font-family:Times New Roman}
#redText { color:Red; font-size:larger}
b, u { font-family:Courier New}
</style>
</head>
```

```

<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b id="redText">Петром I</b> в устье
Невы была заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.</p>
<p class="tnrText">На стрелке Васильевского острова
затем обосновалась торговая пристань.</p>
</body>
</html>

```

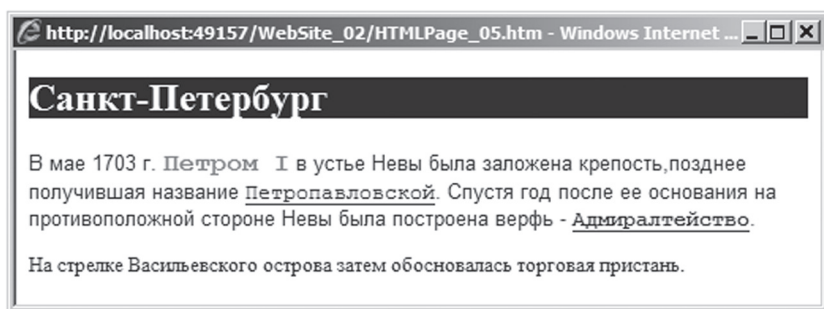


Рис. 2.5. Результат применения стиля для нескольких тегов

### 2.1.6. Комбинированный стиль

Перед определением стиля можно **через пробел** указать комбинацию тегов, для которой стиль переопределяется. Например, стиль вида

```
b u { color: Blue}
```

будет применен только к тексту, находящемуся в теге **<u>**, вложенном в тег **<b>**.

Пример определения и применения стиля для нескольких тегов (рис. 2.5):

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd»>
<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head>
<title></title>
<style type="text/css">
p {font-family:Arial; font-size:small; color:Navy}
h2 { background-color:Navy; color:White}
.tnrText { font-family:Times New Roman}
#redText { color:Red; font-size:larger}
u, b { font-family:Courier New}
b u { color:Blue}
</style>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b id="redText">Петром I</b> в устье
Невы была заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.</p>
<p class="tnrText">На стрелке Васильевского острова за-
тем обосновалась торговая пристань.</p>
</body>
</html>

```

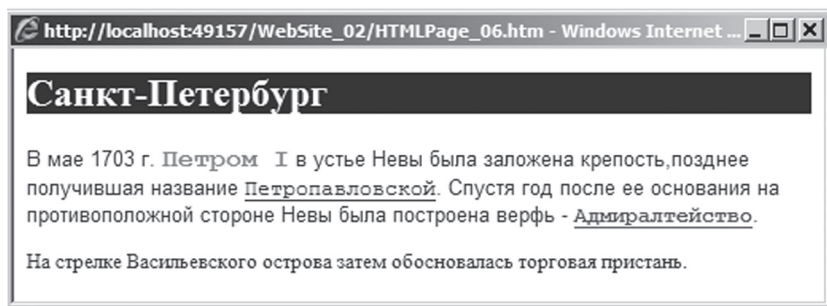


Рис. 2.6. Результат применения комбинированного стиля

### 2.1.7. Внешняя таблица стилей

Если параметры форматирования требуется применить ко всем страницам сайта, то определение этих параметров помещается во **внешнюю таблицу стилей**, которая добавляется в сайт следующим образом (рис. 2.7):

- Из меню **Веб-узел** нужно выбрать команду **Добавить новый элемент**.
- На экран будет выведено одноименное диалоговое окно, в котором следует выбрать шаблон **Таблица стилей**
- В поле **Имя** можно отредактировать имя файла таблицы стилей
- Щелкнуть кнопку **Добавить**

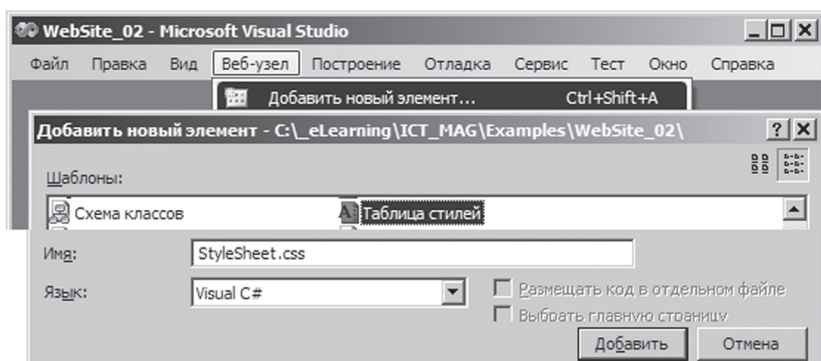


Рис. 2.7. Добавление в сайт внешней таблицы стилей

Определение стиля во внешней таблице ничем не отличается от определения стиля во внутренней таблице (за исключением отсутствия тегов `<style></style>`).

Пример содержимого файла StyleSheet.css:

```
p {font-family:Arial; font-size:small; color:Navy}
h2 { background-color:Navy; color:White}
.tnrText { font-family:Times New Roman}
#redText { color:Red; font-size:larger}
u, b { font-family:Courier New}
b u { color:Blue}
```

### 2.1.8. Связь HTML-страницы с таблицей

Для связывания страницы с файлом внешней таблицей стилей используется тег `<link>`, который размещается в заголовке страницы и содержит следующие атрибуты:

- **href** — указывает имя файла, связанного со страницей.
- **type** — указывает тип содержимого этого файла.
- **rel** — указывает тип связи страницы с файлом.



Пример связывания страницы с внешней таблицей стилей (Рис.2.8):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml«>
<head>
<title></title>
<link href=»StyleSheet.css» type=»text/css»
rel=»Stylesheet» />
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b id="redText">Петром I</b> в устье
Невы была заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.</p>
<p class="tnrText">На стрелке Васильевского острова за-
тем обосновалась торговая пристань.</p>
</body>
</html>
```

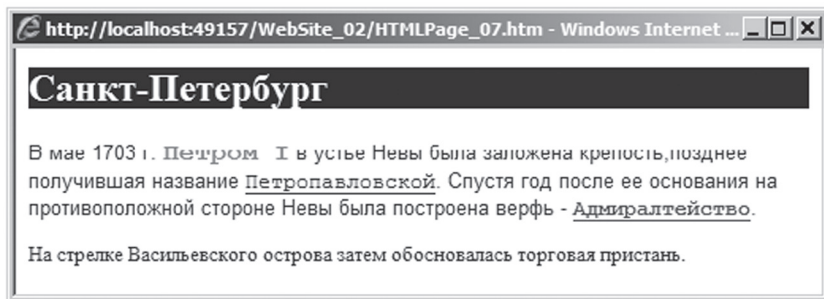


Рис. 2.8. Результат связывания страницы с внешней таблицей стилей

### 2.1.9. Каскадное определение стиля

Если стиль определен в нескольких таблицах стилей, браузер поступает следующим образом:

- Сначала загружается и сохраняется в памяти внешняя таблица стилей. При этом определения в стилевом классе имеют приоритет перед переопределением стиля
- Затем обрабатывается внутренняя таблица стилей, и ее содержимое добавляется к определениям внешней таблицы стилей

Для стиля второго абзаца получим:

```
font-family:Arial;
font-size:small;
color:Navy
font-family:Times New Roman;
font-size:large
```

Если в процессе такого каскадного определения стиля какой-то его параметр переопределяется, то предыдущее определение этого параметра аннулируется. В данном случае это параметры **font-family** и **font-size**. Поскольку определен и встроенный стиль, его определения добавляются к уже имеющимся стилям, а конфликтующее предыдущее определение параметра **font-family** аннулируется:

```
font-family:Arial;
font-size:small;
color:Navy
font-family:Times New Roman;
font-size:large
font-weight:bold;
font-family:Courier New
```

Пример каскадного определения стилей (Рис.2.9):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" <http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
```

```

<link href="StyleSheet.css" type="text/css"
rel="stylesheet" />
<style type="text/css">
.tnrText { font-size:large}
</style>
</head>
<body>
<h2>Санкт-Петербург</h2>
<p>В мае 1703 г. <b id="redText">Петром I</b> в устье
Невы была заложена крепость, позднее получившая название
<u>Петропавловской</u>. Спустя год после ее основания на
противоположной стороне Невы была построена верфь -
<b><u>Адмиралтейство</u></b>.</p>
<p class="tnrText" style="font-weight:bold; font-
family:Courier New"> На стрелке Васильевского острова
затем обосновалась торговая пристань.</p>
</body>
</html>

```

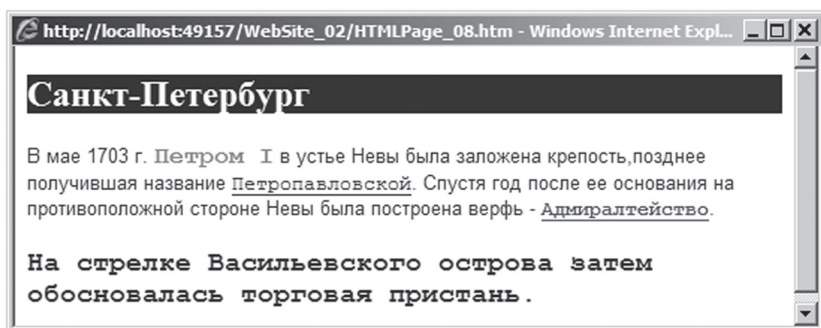


Рис. 2.9. Результат каскадного определения стилей

## 2.2. РЕДАКТОР И АТРИБУТЫ СТИЛЕЙ

### 2.2.1. Запуск редактора стилей

Если щелкнуть правой кнопкой мыши внутри определения стиля во внешней таблице стилей, то на экран будут выведено контекстное меню с командой **Построить стиль**. Выбор этой команды выводит на экран диалоговое окно редактора стилей, позволяющего в диалоговом режиме задать все необходимые параметры стиля (рис. 2.10).

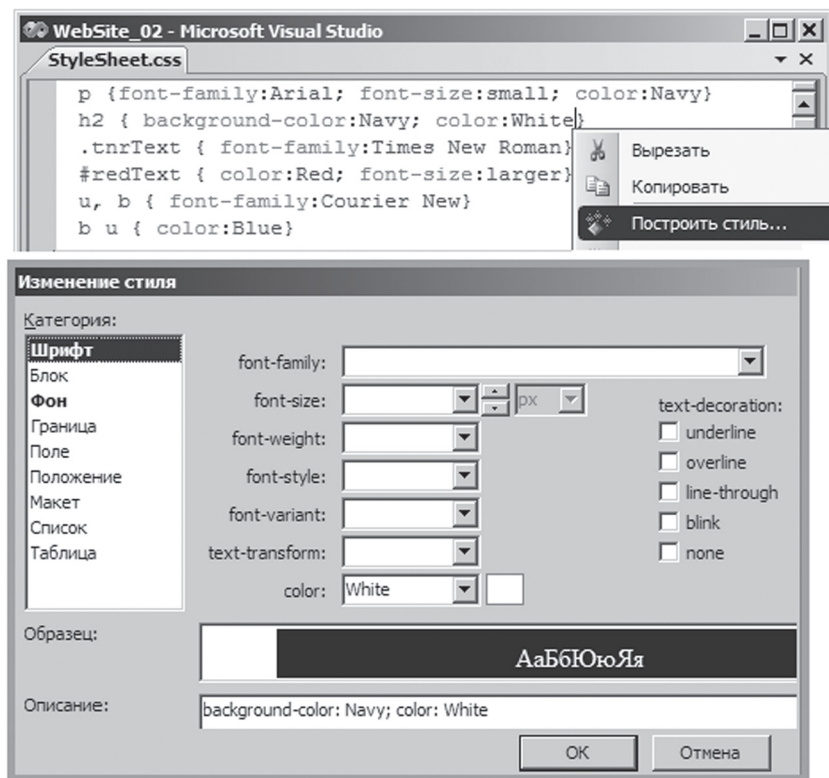


Рис. 2.10. Запуск редактора стилей

### 2.2.2. Атрибуты шрифта

Категория **Шрифт** задает параметры шрифта (рис. 2.11):

- **font-family** — шрифт текста
- **font-size** — размер шрифта
- **font-weight** — «жирность» шрифта
- **font-style** — начертание шрифта
- **font-variant** — вид букв шрифта
- **text-transform** — регистр символов шрифта
- **color** — цвет текста
- **text-decoration** — «украшение» шрифта

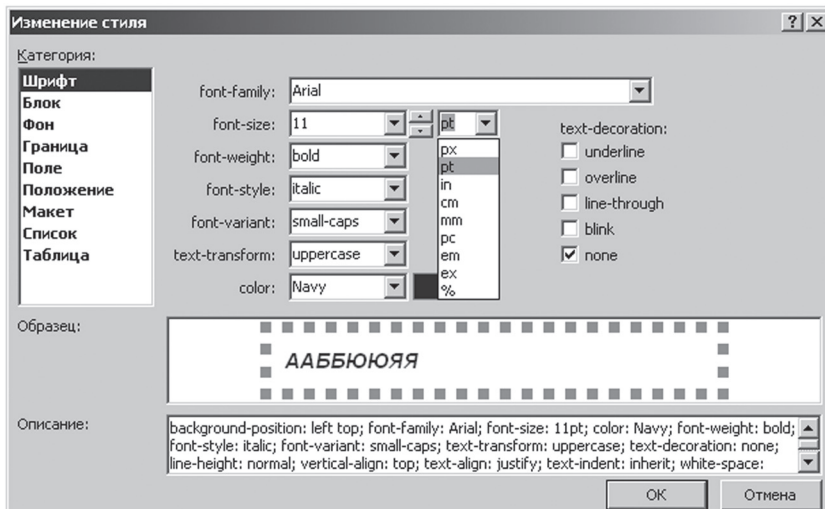


Рис. 2.11. Атрибуты шрифта

### 2.2.3. Атрибуты блока

Категория **Блок** задает параметры блочного элемента страницы `<div></div>`, `<p></p>`, `<table></table>` (рис. 2.12):

- **line-height** — вертикальное расстояние между двумя строками текста
- **vertical-align** — вертикальное выравнивание текста
- **text-align** — горизонтальное выравнивание текста
- **text-indent** — отступ «красной» строки
- **white-space** — управление переносом строк
- **word-spacing** — промежуток между словами
- **letter-spacing** — промежуток между символами текста

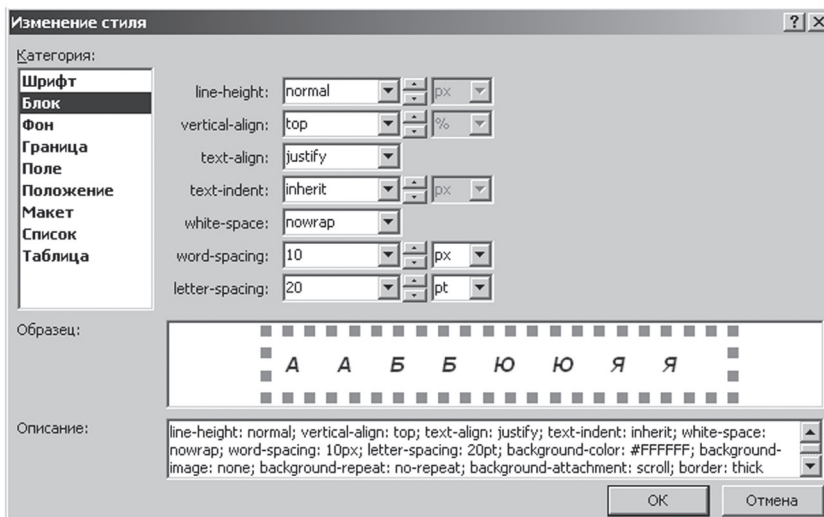


Рис. 2.12. Атрибуты блока

#### 2.2.4. Атрибуты фона

Категория **Фон** задает параметры фона (рис. 2.13):

- **background-color** — задание цвета фона
- **background-image** — задание фонового изображения
- **background-repeat** — параметры повторения фонового изображения
- **background-attachment** — запрет/разрешение прокрутки графического фона
- **(x)background-position** — горизонтальная позиция фонового изображения
- **(y)background-position** — вертикальная позиция фонового изображения

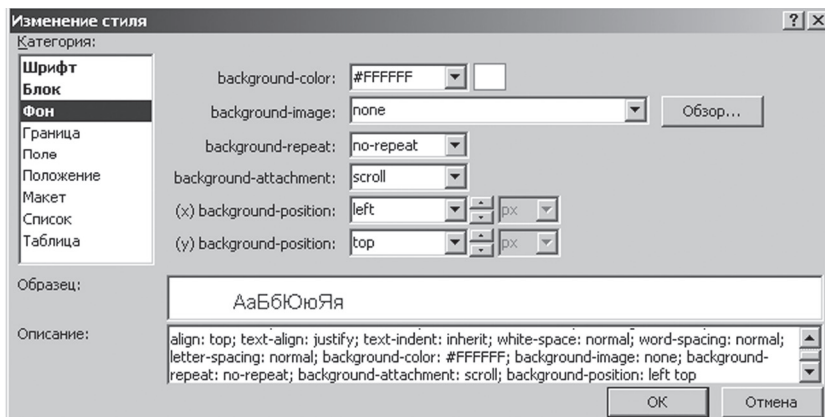


Рис. 2.13. Атрибуты фона

## 2.2.5. Атрибуты границы

Категория **Граница** задает параметры рамки по воображаемой границе элемента страницы (рис. 2.14):

- **border-style** — стиль рамки
- **border-width** — ширина рамки
- **border-color** — цвет рамки

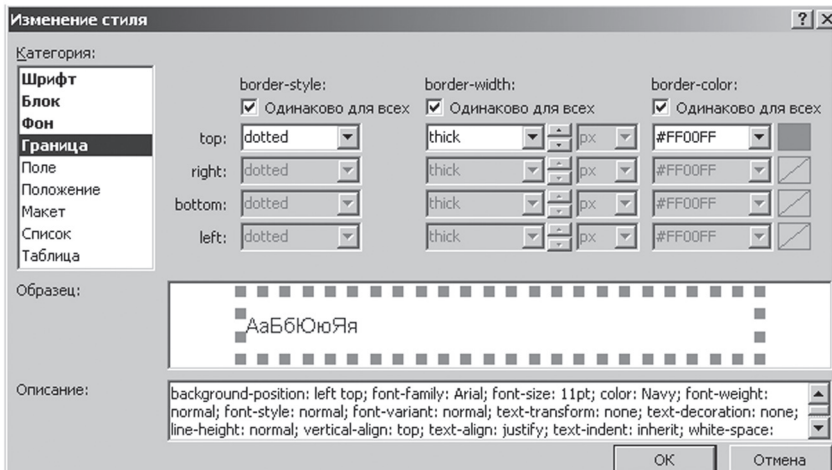


Рис. 2.14. Атрибуты границы

### 2.2.6. Атрибуты поля

Категория **Поля** задает параметры отступов между содержимым блочного элемента (абзац, изображение, таблица и т.п.) и его воображаемой границей, а также между его границей и границами соседних элементов (рис. 2.15):

- **padding** — отступы между содержимым блочного элемента и его воображаемой границей
- **margin** — отступы между границей блочного элемента и границами соседних элементов

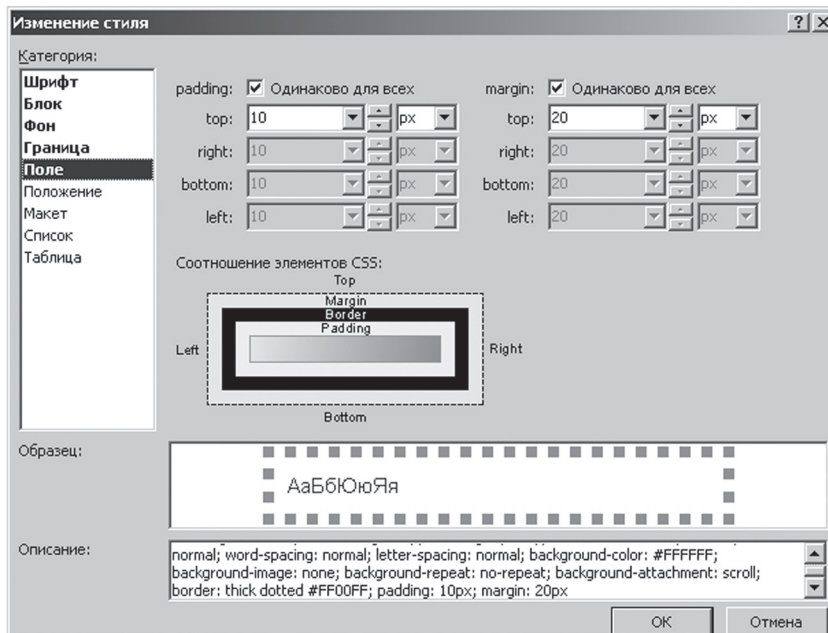


Рис. 2.15. Атрибуты поля

### 2.2.7. Атрибуты положения

Категория **Положение** задает параметры размера и положения элемента страницы (рис. 2.16):



- **position** — устанавливает режим, определяющий доступные параметры позиционирования
- **z-index** — задает z-порядок элемента; z-порядок определяет отображение перекрывающихся элементов; элементы с более высоким значением z-порядка отображаются перед элементами с меньшим z-порядком
- **width** — ширина элемента страницы
- **height** — высота элемента страницы
- **top** — устанавливает положение верхнего края элемента страницы
- **right** — устанавливает положение правого края элемента страницы
- **bottom** — устанавливает положение нижнего края элемента страницы
- **left** — устанавливает положение левого края элемента страницы

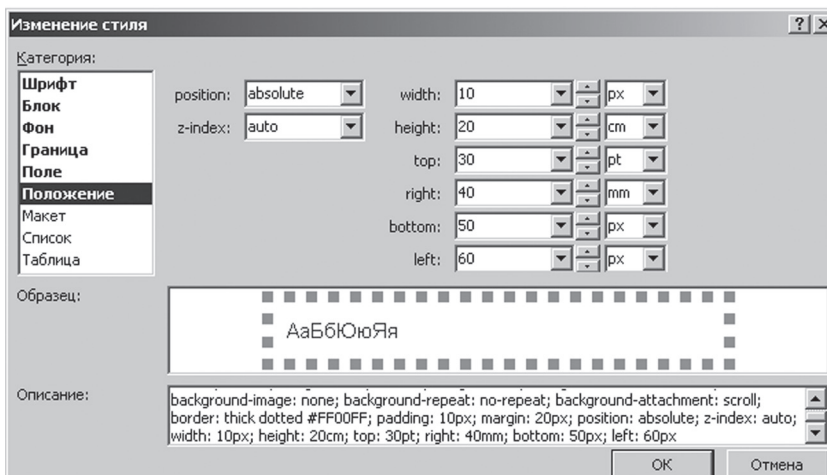


Рис. 2.16. Атрибуты положения

### 2.2.8. Атрибуты макета

Категория **Макет** задает параметры внешнего вида страницы (рис. 2.17):

- **visibility** — позволяет либо скрыть (сделать невидимым) элемент страницы, либо сделать его видимым.
- **display** — можно указать, будет ли элемент отображаться, и каким будет вид отображаемого элемента: в виде элемента блока или внутри линейного потока

- **float** — задает атрибуты, определяющие, каким образом текст обтекает элемент
- **clear** — устанавливает атрибуты, определяющие, может ли объект свободно перемещаться по странице
- **cursor** — форма курсора мыши
- **overflow** — устанавливает атрибуты, контролирующие поведение элементов, содержимое которых выходит за пределы, задаваемые высотой и шириной элемента
- **clip** — устанавливает атрибуты, которые управляют отображением прямоугольной области элемента; например, можно отобразить только правый верхний угол элемента: введите значения в одном или нескольких полях (Вверху, Внизу, Слева, или Справа) и выберите параметр единицы измерения: px (по умолчанию), pt, pc, mm, cm, in, em, ex или %.

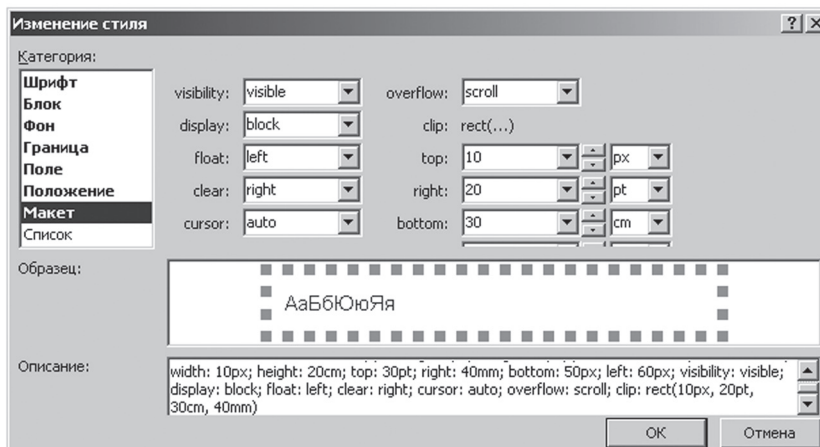


Рис. 2.17. Атрибуты макета

### 2.2.9. Атрибуты списка

Категория **Список** задает параметры списка (Рис.2.18):

- **list-style-type** — вид маркера списка
- **list-style-image** — задание графического изображения в качестве маркеров списка
- **list-style-position** — местоположение маркера или нумерации элементов списка

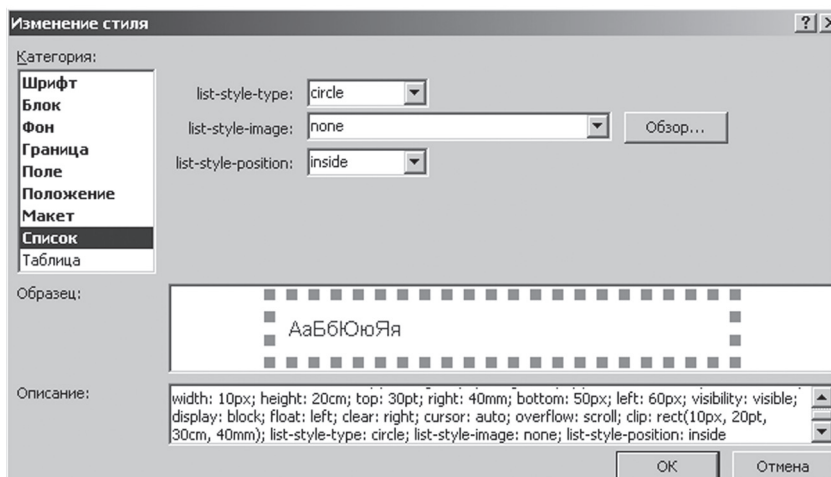


Рис. 2.18. Атрибуты списка

### 2.2.10. Атрибуты таблицы

Категория **Таблица** задает параметры таблицы (Рис.2.19):

- **table-layout** – устанавливает атрибуты для макета таблицы в данном стиле; атрибуты устанавливают либо фиксированный размер ячейки, либо разрешают автоматическую регулировку размера ячеек в соответствии с содержимым
- **border-collapse** – устанавливает атрибуты, определяющие стиль границ таблицы.
- **border-spacing** – размер границы между ячейками
- **empty-cells** – параметр показа пустых ячеек
- **caption-side** – размещение заголовка таблицы

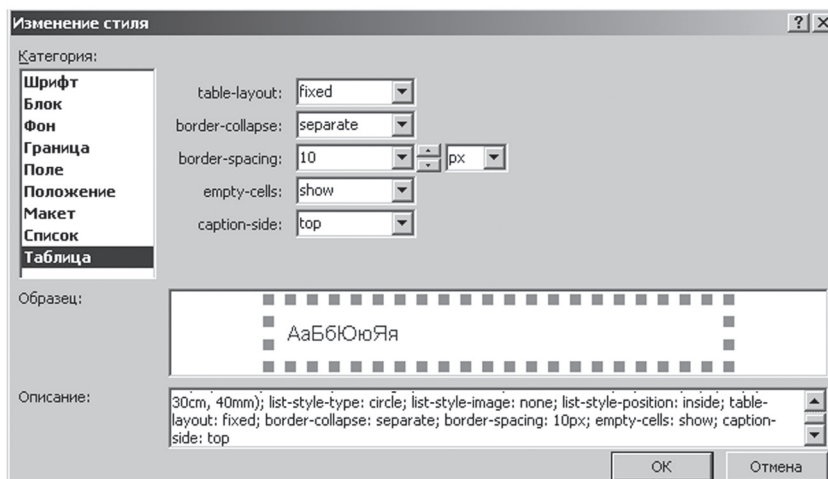


Рис. 2.19. Атрибуты таблицы

## 2.3. Стили гиперссылок

**Стили гиперссылок** определяют параметры форматирования для текстовых гиперссылок:

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

Пример определения стилей гиперссылок во внешней таблице стилей:

```
p {font-family:Arial; font-size:small; color:Navy}
h2{background-color: Navy; color: White;}
.tnrText { font-family:Times New Roman}
#redText { color:Red; font-size:larger}
u, b { font-family:Courier New}
b u { color:Blue}
a:link {color:Blue}
```

```
a:visited {color:Purple}
a:active {color:Red}
a:hover {background-color:White}
```

## 2.4. КОНТЕЙНЕРЫ HTML-СТРАНИЦЫ

**Контейнер** – это парные теги, служащие «упаковкой» для других тегов (например, тегов абзаца). Содержимое контейнера ведет себя как единое целое, поэтому можно, например, применить к нему стилевой класс. Контейнеры бывают двух видов: блочные и встроенные.

**Блочный контейнер** (или просто **блок**) создается с помощью тегов `<div></div>` и ведет себя как абзац (начинается с новой строки и отделяется пустой строкой от других элементов страницы).

**Встроенный контейнер** создается с помощью тегов `<span></span>` и ведет себя как часть абзаца.

Пример определения контейнеров на странице (Рис.2.20):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" «http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd»»
<html xmlns=»http://www.w3.org/1999/xhtml»»
<head>
<title></title>
<link href=»StyleSheet.css» type=»text/css»
rel=»Stylesheet» />
</head>
<body>
<h2>Санкт-Петербург</h2>
<div class="tnrText">
<p>В мае 1703 г. <span id="redText">Петром I </span>в
устье Невы была заложена крепость, позднее получившая на-
звание <u>Петропавловской</u>. Спустя год после ее основа-
ния на противоположной стороне Невы была построена верфь –
<b><u>Адмиралтейство</u></b>.</p>
<p>На стрелке Васильевского острова затем обосновалась
торговая пристань.</p>
</div>
</body>
</html>
```

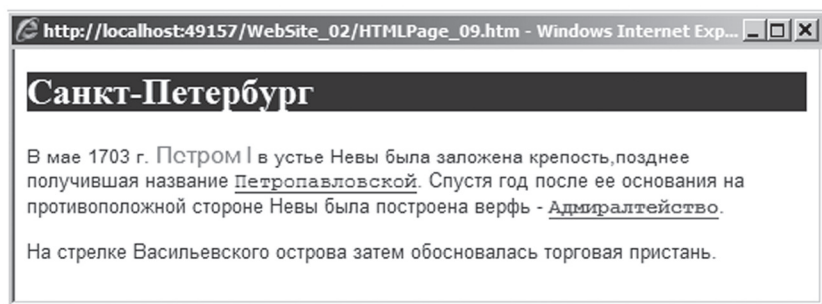


Рис. 2.20. Результат использования контейнеров на странице

## 2.5. ПРИМЕР WEB-ДИЗАЙНА САЙТА

### 2.5.1. Графика для домашней страницы

Для графического оформления домашней страницы сайта добавим в папку **Images** следующие графические файлы (Рис.2.21):

- **Lak\_2.jpg** — фотография автора сайта
- **Lak\_Header.GIF** — файл с логотипом сайта. Это картинка в формате GIF (или JPG), которую можно получить с помощью механизма захвата экрана. Необходимо знать ее размеры. Для этого графический файл следует 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** — это изображение получается следующим образом: у изображения **Lak\_Header.GIF** нужно отрезать справа полосу шириной в один пиксель. Данный файл нужен для того, чтобы можно было создать эффект «растягивания по ширине» изображения с логотипом сайта.
- **pc.gif** — некоторое изображение, которое является иллюстрацией содержимого страницы контента сайта



Рис. 2.21. Графика для домашней страницы

### 2.5.2. Стилизовое оформление страниц

Для стилизового оформления домашней страницы добавим в сайт еще одну внешнюю таблицу стилей с именем `StyleSheet2.css`. Для заполнения внешней таблицы стилей воспользуемся визуальным дизайнером стилей:

```
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.content {font-family:Times New Roman Cyr; text-
align:justify;
```

```

        color:Navy;}
div.header {font-family: Arial; font-size: 11pt; font-
weight: bold;}
span.term {font-family: Courier New; font-weight: bold;
color: #000066; background-color: #FFFFCC;
font-size: 11pt;}
span.term2 {font-family: Courier New; font-weight:
normal;
color: #000066; font-size: 11pt; background-color:
#F0FFFF;}

```

### 2.5.3. Разметка домашней страницы

Добавим в Web-сайт страницу **index.htm** и выполним ее разметку ее тела с помощью тегов таблицы: 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 таблицы в первой ячейке содержит фотографию автора курса, а во второй ячейке – указание на владельца сайта и ссылку на его авторскую страницу.
- В третьей строке таблицы две ячейки объединены в одну. Эта строка содержит название раздела контента сайта, выровненное по центру страницы.
- В первой ячейке четвертой строки таблицы содержится графическая иллюстрация раздела контента сайта, а во второй ячейке содержатся ссылки на страницы первого раздела контента.

Для стиливого оформления домашней страницы свяжем ее со второй таблицей стилей (рис. 2.22):

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```



```

        <title></title>
        <link rel=»Stylesheet» type=»text/css»
href=»StyleSheet2.css»/>
    </head>
    <body>
        <table width=»100%» cellpadding=»0» cellspacing=»0»>
        <tr>
        <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>
        <td width=»465px»><img src=»Images/Lak_2.jpg» alt=»» />
        </td>
        <td align=»center»>
        <h2>Web-сайт Леонида А. КЕРОВА</h2>
        <div class=»header»><a href=»about.htm»>Об авторе</a></
div>
        </td>
        </tr>
        <tr>
        <td colspan = "2" align="center"><h3>Устройство персо-
нального компьютера</h3>
        </td>
        </tr>
        <tr>
        <td width=»465px»><img src=»Images/pc.gif» alt=»» /><br />
        </td>
        <td >
        <div class=»header»><a href=»PC_System.htm»>Системный
блок ПК</a></div><br />
        <div class=»header»><a href=»PC_Monitor.htm»>Монитор
ПК</a></div><br />
        <div class=»header»><a href=»PC_Keyboard.
htm»>Клавиатура ПК</a></div>
        </td>
        </tr>
        </table>
    </body>
</html>

```



Рис. 2.22. Вид домашней страницы в окне браузера

#### 2.5.4. Разметка страницы контента

Добавим в сайт указанные в разметке **index.htm** страницы контента и необходимую для них графику. Выполним разметку каждой страницы контента:

- Раздел **body** начинается с указания тематики страницы контента
- Содержание страницы контента структурировано с помощью таблицы, состоящей из одной строки и двух ячеек.
- В левой ячейке размещается текстовое содержание контента раздела, а в правой – графическая иллюстрация к этому разделу

Пример разметки страницы PC\_Monitor.htm (рис. 2.23):



Рис. 2.23. Вид страницы PC\_Monitor.htm в окне браузера

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
  <title>Monitor specifications performance</title>
  <link rel=»Stylesheet» type=»text/css»
href=»StyleSheet2.css» />
</head>
<body>
<h4>Монитор</h4>
<table>
<tr>
<td valign=»top»><div class=»content»>
  <p>Формирование изображения на экране монитора обеспе-
чивается <b>видеопроцессором</b> (контроллером). Плата
контроллера устанавливается в <b>слот расширения</b> на
системной плате и поставляется с набором программ -
<b>драйверов</b>.</p>
  <a href=»Images/Monitor.AVI»><img src=»Images/mplay.
gif» hspace=»10» alt=»» /></a>
```

```

Для получения сведения о деталях формирования изображения
на экране монитора щелкните изображение видеоплеера. </div>
</td>
<td><img src=>Images/Monitor.jpg> hspace=>10> alt=>>>/>
</td>
</tr>
</table>
</body>
</html>

```

### 2.5.5. Проверка функционирования сайта

Для проверки функционирования сайта следует открыть в браузере его домашнюю страницу. Затем щелчком по соответствующей гиперссылке открыть ту или иную страницу контента (рис. 2.24).



Рис.2.24. Проверка функционирования сайта

### Задания для практических занятий и самостоятельной работы

Для изучения материала второй главы в тематическом плане дисциплины предусмотрено 28 часов:

Название тем	Всего часов	Аудиторные часы		Самостоятельная работа
		Лекции	Практические занятия	
Раздел 1. Создание и использование в управленческой практике персональных и корпоративных Web-сайтов				
Тема 1.2. Стилизовое оформление HTML-страниц	28	4	4	20

**Задание №1.2.1.** Используя приведенный выше пример Web-дизайна сайта как образец, выполните следующие действия:

- Добавьте в сайт собственную графику для оформления новой домашней страницы
- Используя редактор стилей, разработайте внешнюю таблицу стилей
- Создайте новую домашнюю страницу с рекомендуемой разметкой, и замените домашнюю страницу на сервере **www.narod.ru** на новую домашнюю страницу

**Порядок передачи результатов выполнения задания №1.2.1 для их оценивания:**

- Следует послать письмо преподавателю по электронной почте
- В качестве темы письма нужно указать номер подгруппы, фамилию и инициалы студента; например: 221В, Иванов А.С.
- В тексте электронного письма следует указать доменное имя, разработанного студентом сайта; например: [www.ivanov.narod.ru](http://www.ivanov.narod.ru)
- Максимальная оценка выполнения задания №1.2.1 составляет 1 балл.

**Задание № 1.2.2.** Используя приведенный выше пример Web-дизайна сайта как образец, выполните следующие действия:

- Добавьте в сайт собственную графику, которая используется в разработанном ранее конспекте по теме научно-исследовательского проекта

- Добавьте в сайт новую HTML-страницу (страницу контента) и разместите на ней материал конспекта
- Выполните стилевое оформление страницы контента, используя внешнюю таблицу стилей
- Добавьте страницу контента на сайт на сервере **www.narod.ru**
- На домашней странице сайта предусмотрите гиперссылку на страницу контента

**Порядок передачи результатов выполнения задания № 1.2.2 для их оценивания:**

- Следует послать письмо преподавателю по электронной почте
- В качестве темы письма нужно указать номер подгруппы, фамилию и инициалы студента; например: 221В, Иванов А.С.
- В тексте электронного письма следует указать доменное имя, разработанного студентом сайта; например: [www.ivanov.narod.ru](http://www.ivanov.narod.ru)
- Максимальная оценка выполнения задания №1.2.2 составляет 1 балл

## ГЛАВА 3. ПРОГРАММНОЕ УПРАВЛЕНИЕ HTML-СТРАНИЦАМИ

### 3.1. СЦЕНАРИИ

#### 3.1.1. Понятие сценария

**Сценарий** – это последовательность операторов языка программирования сценариев (наиболее распространенным из таких языков является JavaScript), которые заключены между парными тегами `<script></script>`:

```
<script type=»text/javascript»>  
операторы  
</script>
```

В открывающем теге `<script>` указывается атрибут **type**, значением которого является указание на язык программирования сценариев. Правила написания операторов сценария:

- Если несколько операторов располагаются в одной строке, то между ними ставится знак «точка с запятой» (;).
- Если каждый оператор размещается в отдельной строке, точку с запятой можно не указывать.
- Оператор может располагаться в нескольких строках.

Если на Web-странице имеется сценарий, браузер тотчас же выполняет его. Однако если в сценарии имеется определение функции, то функция в момент анализа не выполняется, а запоминается для дальнейшего использования. Для выполнения функции ее нужно вызвать явным образом.

Некоторые функции могут быть связаны с объектами на Web-странице; например, с кнопками или текстовыми полями в формах. Они вызываются тогда, когда происходит некоторое событие, связанное с кнопкой или полем.

Сценарии можно помещать в любом месте HTML-страницы. Однако, определения функций обычно располагают в заголовке HTML-страницы, между тегами **<head></head>**. Это удобное место, т.к. заголовок находится в начале страницы, и определения функций анализируются раньше, чем загрузится вся страница.

Пример сценария (рис. 3.1):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<p>
<script type="text/javascript">
var d = new Date()
document.write(d.toLocaleString())
</script>
</p>
</body>
</html>
```



Рис. 3.1. Результат выполнения сценария в «теле» страницы

В приведенном примере сценарий создает экземпляр **d** класса **Date** с помощью конструктора без параметров. **Конструктор класса** – это функция, которая определена внутри класса, имеет имя, совпадающее с именем класса, вызывается с помощью оператора **new** и предназначена для создания в оперативной памяти компьютера экземпляра класса. Экземпляр **d** класса **Date** содержит текущую дату и время.

В следующей строке сценария вызывается метод **write** экземпляра **document** класса **HTMLDocument**: сначала указывается экземпляр класса, затем, через точку, метод, который применяется к экземпляру. **Метод** — это функция, которая определена внутри класса и предназначена для работы с экземплярами класса. Методу **write** передается (с помощью метода **toLocaleString** класса **Date**) в качестве аргумента строковое пред-



ставление даты и времени с учетом региональных настроек операционной системы **Windows**. Класс **HTMLDocument** описывает структуру Web-страницы. Экземпляр **document** создается браузером во время загрузки страницы. Метод **write** экземпляра **document** класса **HTMLDocument** выводит переданную ему в качестве аргумента строку в то место страницы, где встретилось вызвавшее этот метод выражение.

### 3.1.2. Литералы и операции

**Литерал** – простейшие данные, с которыми оперирует сценарий:

- Целые числа, например: 2010, +62, -2
- Дробные числа, например: 3.14, -2.73
- Строки символов, например: «Hello», 'Hello'
- Логические значения: true, false
- Неопределенное значение: null

Для чисел определены **арифметические операции**:

Обозначение операции	Выполняемое действие
+	сложение
–	вычитание
*	умножение
/	деление
%	остаток от деления целых чисел
++	инкремент (увеличение на 1)
--	декремент (уменьшение на 1)

Операция «плюс» (+) определена также для символьных строк: ее результатом является строка, полученная в результате сцепления строк — слагаемых. Для чисел и символьных строк определены **операции сравнения** (для строк используется лексикографическое сравнение):

Обозначение операции	Выполняемое действие
<	меньше
>	больше
==	равно
!=	не равно
<=	меньше или равно
>=	больше или равно

Для логических значений определены **логические операции**:

Обозначение операции	Выполняемое действие
!	логическое НЕ
&&	логическое И
	логическое ИЛИ

Пример использования в сценарии литералов и операций (Рис.3.2):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=http://www.w3.org/1999/xhtml>
<head>
<title></title>
</head>
<body>
<p>
<script type=»text/javascript»>
document.write(2+2)
document.write(«<br />» + «to»+ « « + «be»)
</script>
</p>
</body>
</html>
```



Рис. 3.2. Результат выполнения сценария с литералами и операциями

### 3.1.3. Понятие переменной

**Переменная** – именованная ячейка в памяти компьютера, содержащая некоторое значение (оно называется **значением переменной**). **Имя переменной** должна начинаться с латинской буквы, после которой могут быть указаны латинские буквы, цифры и знаки подчеркивания. Переменная **v** определяется с помощью оператора **var**:

**var v**

При определении переменной **v** ее можно инициализировать значением выражения **e**:

**var v = e**

Переменным, которые не инициализированы явным образом, по умолчанию присваивается значение **undefined**. Переменные в HTML-документе можно определять как в разделе **<head></head>**, так и в разделе **<body></body>**. Переменная, определенная в одном из присутствующих на странице сценариев, является доступной для всех остальных сценариев.

Пример определения и использования переменной (рис. 3.3):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title></title>
<script type=»text/javascript»>
var d = new Date()
</script>
</head>
<body>
<p>
<script type=»text/javascript»>
document.write(d.toLocaleString())
</script>
</p>
</body>
</html>
```



Рис. 3.3. Результат выполнения двух сценариев  
(в заголовке и в «теле» страницы)

### 3.1.4. Оператор присваивания

**Оператор присваивания** предназначен для изменения содержимого некоторой ячейки памяти компьютера (в частности, для изменения значения переменной) и имеет вид

$$V = E$$

где: **V** — обозначение некоторой определенной ячейки памяти компьютера (в частности, имя переменной); **E** — выражение.

**Выражение** — это правило вычисления значения, имеющее вид литерала, переменной или конструкции, построенной из литералов, переменных, знаков операций и скобок. Оператор присваивания **выполняется справа налево**: сначала вычисляется значение выражения **E**, затем оно заносится в ячейку **V**.

**Оператор инкремента** имеет вид **V++** и является краткой формой записи оператора присваивания вида: **V = V + 1**.

**Оператор декремента** имеет вид **V--** и является краткой формой записи оператора присваивания вида: **V = V - 1**.

Существуют **сокращенные формы оператора присваивания**:

Обозначение операции	Выполняемое действие
$V += E$	$V = V + E$
$V -= E$	$V = V - E$
$V *= E$	$V = V * E$
$V /= E$	$V = V / E$
$V \% E$	$V = V \% E$

Пример использования оператора присваивания для изменения шрифта (рис. 3.4):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" <http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<p id="sample">Текст</p>
<script type="text/javascript">
```

```
sample.style.fontFamily = «Arial»
sample.style.fontSize = «Large»
</script>
</body>
</html>
```



Рис. 3.4. Результат выполнения сценария, изменяющего шрифт

### 3.1.5. Файлы сценариев

**Файл сценария** – это текстовый файл, содержащий операторы сценария. Для добавления файла сценария в сайт следует выбрать из меню **Веб-узел** команду **Добавить новый элемент**. На экран будет выведено одноименное диалоговое окно, в котором требуется выполнить следующие действия (Рис.3.5):

- В области **Шаблоны** выбрать шаблон **Файл JScript**
- В поле **Имя** отредактировать имя файла сценария
- Щелкнуть кнопку **Добавить**

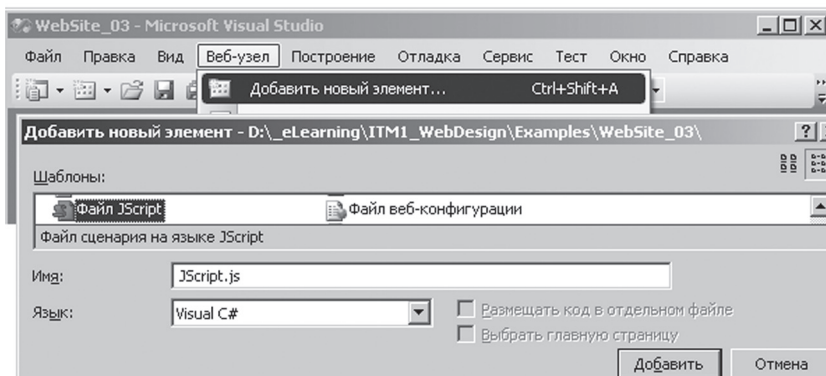


Рис. 3.5. Добавление файла сценария в сайт

### Пример содержимого файла JScript.js

```
var d = new Date()
document.write(d.toLocaleString())
```

Для включения файла сценария в страницу используются парные теги **<script></script>** с пустым содержимым. В открывающем теге **<script>** указывается атрибут **src**, значением которого является Интернет-адрес (URL) файла сценария, например (рис. 3.6).

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title></title>
</head>
<body>
<p>
<script type=»text/javascript» src=»JScript.js»>
</script>
</p>
</body>
</html>
```



Рис. 3.6. Результат выполнения файла сценария

## 3.2. ОБРАБОТКА СОБЫТИЙ СТРАНИЦЫ

### 3.2.1. События и их обработчики

Когда пользователь выполняет какие-то действия со страницей (щелкает кнопкой мыши, прокручивает содержимое окна, перемещает окно и т.п.), Web-браузер генерирует соответствующее событие.

Можно сделать так, чтобы браузер отреагировал на событие. Для этого нужно определить функцию, обрабатывающее данное событие, и обеспечить вызов этой функции при возникновении события.

**Определение функции** имеет вид:

```
function name(parameters){body},
```

где:

- **function** — служебное слово языка JavaScript,
- **name** — имя функции,
- **parameters** — список переменных (перечисленных через запятую и называемых **параметрами функции**; он может быть пустым),
- **body** — список операторов (перечисленных через точку с запятой, определяющих алгоритм функции и называемый **телом функции**).

Пример функции, обрабатывающей щелчок мыши по абзацу и переключающей шрифт текста абзаца с Times New Roman на Arial и обратно (рис. 3.7):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml«>
<head>
<title></title>
<script type=»text/javascript«>
var flag = false
function sampleClick()
{ var sampleObj = document.getElementById(«sample»)
if (flag)
sampleObj.style.fontFamily = «Times New Roman»
else
sampleObj.style.fontFamily = «Arial»
flag = !flag
}
</script>
</head>
<body>
<p id =»sample» onclick=»sampleClick()«>Text</p>
</body>
</html>
```

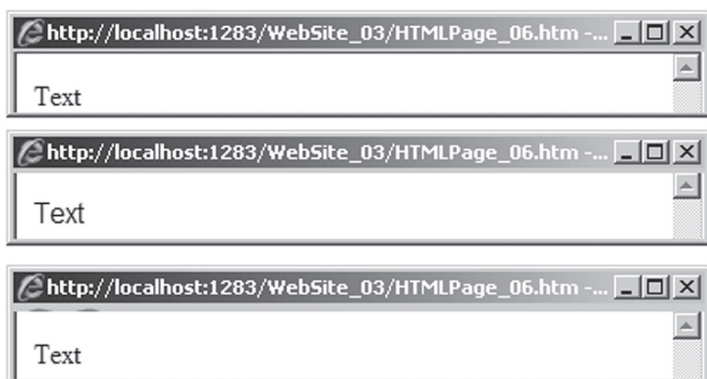


Рис. 3.7. Результат работы функции, обрабатывающей щелчок мыши

В приведенном примере в определении функции используется метод **getElementById** класса **HTMLElement**, который возвращает ссылку на элемент страницы с заданным идентификатором (страница представлена экземпляром **document**). В данном случае элементом страницы является абзац, определенный с помощью тегов **<p></p>**.

В качестве второго оператора в теле функции указан **условный оператор**, который имеет вид:

```
if (B)
    S1
else
    S2
```

где: **B** — выражение, значением которого являются литералы **true** или **false**; **S<sub>1</sub>** и **S<sub>2</sub>** — оператор или последовательность операторов, заключенная в фигурные скобки.

Выполнение условного оператора происходит следующим образом (рис. 3.8):

- Сначала вычисляется значение выражения **B**.
- Если оно равно **true**, то выполняется **S<sub>1</sub>**.
- В противном случае выполняется **S<sub>2</sub>**.



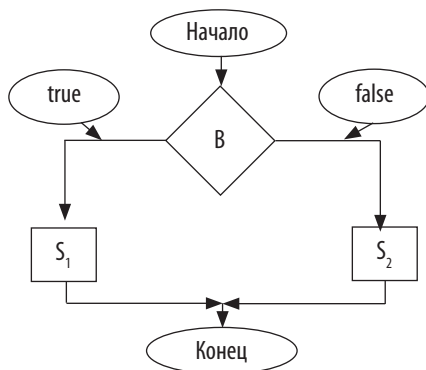


Рис. 3.8. Блок-схема выполнения условного оператора

Частным случаем условного оператора является его сокращенная форма: **if(B)S<sub>1</sub>**, т.е. **S<sub>1</sub>** выполняется только в том случае, когда **B** истинно.

Для вызова функции, которая является обработчиком события, в теге **<p>** используется атрибут с префиксом **on**. После префикса следует имя события, которое должно быть обработано; в данном случае используется атрибут **onclick**, который вызывает функцию **sampleClick**.

### 3.2.2. Обработчик с параметром

Параметры функции внутри ее тела используются как обычные переменные. Начальные значения этим переменным присваиваются оператором вызова функции. Если определение функции имеет вид

**function f(v<sub>1</sub>, v<sub>2</sub>, ..., v<sub>n</sub>) {S},**

то оператор вызова функции должен иметь вид

**f(e<sub>1</sub>, e<sub>2</sub>, ..., e<sub>n</sub>)**

где **e<sub>1</sub>, e<sub>2</sub>, ..., e<sub>n</sub>** – выражения, значения которых являются аргументами функции.

Пример определения и вызова функции с параметром, значением которого является идентификатор абзаца, в котором следует изменять шрифт текста:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" <http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd>>
```

```
<html xmlns=>http://www.w3.org/1999/xhtml>>
<head>
<title></title>
<script type=>text/javascript>>
var flag = false
function sampleClick(x)
{   var sampleObj = document.getElementById(x)
if (flag)
sampleObj.style.fontFamily = <'Times New Roman'>
else
sampleObj.style.fontFamily = <'Arial'>
flag = !flag
}
</script>
</head>
<body>
<p id =>sample> onclick=>sampleClick('sample')>>Text</p>
<p id =>s2> onclick=>sampleClick('s2')>>Text2</p>
</body>
</html>
```

Значения аргументов функции присваиваются ее параметрам, после чего выполняется тело **S** функции, в котором используются параметры функции.

### ***Задание для практических занятий и самостоятельной работы***

Для изучения материала третьей главы в тематическом плане дисциплины предусмотрено 12 часов:

Название тем	Всего часов	Аудиторные часы		Самостоятельная работа
		Лекции	Практические занятия	
Раздел 1. Создание и использование в управленческой практике персональных и корпоративных Web-сайтов				
Тема 1.3. Программное управление HTML-страницами	12	2	2	8

**Задание №1.3.1.** Используя инструментальную систему Microsoft Visual Studio 2008, добавьте в сайт несколько HTML-страниц, рассмотренных в данной теме (может быть, все), затем выполните следующие действия:

- Убедитесь в корректном функционировании добавленных страниц на Web-браузере
- Разместите созданные страницы на вашем сайте в Интернете (на сервере [www.narod.ru](http://www.narod.ru))
- Убедитесь в корректном функционировании добавленных страниц в Интернете

**Порядок передачи результатов выполнения задания №1.3.1 для их оценивания:**

- Следует послать письмо преподавателю по электронной почте
- В качестве темы письма нужно указать номер подгруппы, фамилию и инициалы студента; например: 221В, Иванов А.С.
- В тексте электронного письма следует указать доменное имя, разработанного студентом сайта; например: [www.ivanov.narod.ru](http://www.ivanov.narod.ru)
- Максимальная оценка выполнения задания №1.3.1 составляет 1 балл.

## ГЛАВА 4. РАБОТА С WEB-ФОРМАМИ

### 4.1. ПОНЯТИЕ И СОЗДАНИЕ WEB-ФОРМЫ

#### 4.1.1. Понятие Web-формы

**Web-форма** — это элемент Web-страницы, содержащий текстовые поля, переключатели, флажки, кнопки и другие элементы управления и предназначенный для ввода, подготовки и отправки данных другой странице или Web-серверу. Для создания формы используются парные теги **<form>****</form>**, с обязательными атрибутами:

- **action** – интернет-адрес (URL) страницы, которой будут переданы данные с формы
- **name** – имя формы, необходимое для получения к ней доступа из сценария

Фрагмент разметки страницы, «тело» которой содержит Web-форму (рис. 4.1):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transi-
tional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-tran-
sitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml«>
<head>
<title>Расстояние от точки (x,y) до начала координат</
title>
</head>
<body>
<form action=»HTMLPage_01.htm» name=»form1» >
x = <input id=»x» type=»text» />
y = <input id=»y» type=»text» /> <br />
<input id=»Button1» type=»button» value=»Вычислить
расстояние»>
```

```

onclick=>document.form1.h.value=hypotenuse()» /> <br />
h = <input id=>h» type=>text» /> <br />
<input id=>Reset1» type=>reset» value=>Очистить поля
формы» />
</form>
</body>
</html>

```

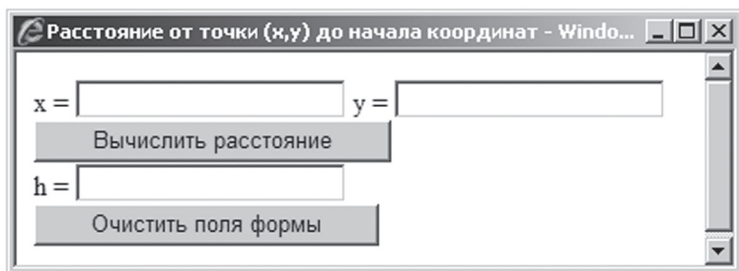


Рис. 4.1. Вид Web-формы в окне браузера

#### 4.1.2. Текстовые поля и кнопки

Для ввода текстовых данных на форму и вывода текстовых данных из формы используется текстовое поле, создаваемое с помощью тега **<input type="text"/>**. Атрибут **value** задает значение, которое содержится в текстовом поле изначально или выбирается из него.

На форме с помощью тега **<input type="button"/>** можно создать кнопку, реакция на щелчок по которой может быть указана в качестве значения атрибута **onclick**. Атрибут **value** задает надпись на кнопке.

С помощью тега **<input type="reset"/>** на форме можно создать кнопку, щелчок по которой очищает все текстовые поля формы. Атрибут **value** задает надпись на кнопке.

Разметку для элементов Web-формы (в частности, текстовые поля и кнопки) можно генерировать автоматически, перемещая в режиме **Конструктор** соответствующие элементы из панели элементов на форму. Для вывода **Панели элементов** на экран следует из меню **Вид** выбрать команду **Панель элементов** (рис. 4.2):

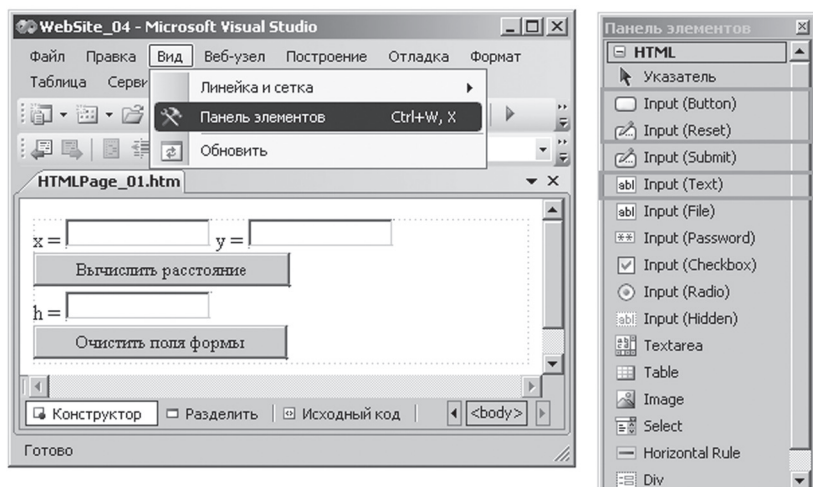


Рис. 4.2. Создание формы с помощью Панели элементов

Приведем полную разметку страницы, на которой с помощью теоремы Пифагора  $h = \sqrt{x^2 + y^2}$  решается задача нахождения расстояния **h** от точки **(x, y)** на плоскости до начала координат (рис. 4.3):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Расстояние от точки (x,y) до начала координат</title>
<script type="text/javascript">
function hypotenuse() //вычисление гипотенузы по двум
катетам
{
var x = Number(document.form1.x.value)
var y = Number(document.form1.y.value)
var h = Math.sqrt(x*x+y*y); return h
}
</script>
</head>
<body>
```

```

<form action=»HTMLPage_01.htm» name=»form1» >
x = <input id=»x» type=»text» />
y = <input id=»y» type=»text» /> <br />
<input id=»Button1» type=»button» value=»Вычислить рас-
стояние» onclick=»document.form1.h.value=hypotenuse()»/>
<br/>
h = <input id=»h» type=»text» /> <br />
<input id=»Reset1» type=»reset» value=»Очистить поля
формы» />
</form>
</body>
</html>

```

В заголовок HTML-страницы добавлен сценарий, в котором определена функция **hypotenuse** с пустым списком параметров. В теле функции сначала определяется переменная **x**, которой присваивается значение стандартной функции **Number**, преобразующей строковое значение выражения **document.form1.x.value** в число. Значением указанного выражения являются данные, введенные в текстовое поле с id-идентификатором **x** формы с именем **form1**.

Затем аналогичным образом определяется переменная **y**. После этого определяется переменная **h**, которой присваивается значение функции **sqrt** класса **Math**, извлекающей квадратный корень из значения своего аргумента. Оператор **return** возвращает значение переменной **h** в качестве значения функции **hypotenuse**.

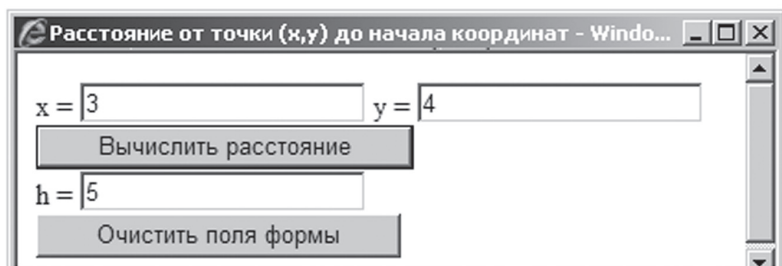


Рис. 4.3. Пример решения задачи в окне браузера

#### 4.1.3. Возврат нескольких значений на формулу

Продemonстрируем использование функции для решения сразу двух задач (рис. 4.4):

- Нахождения расстояния  $h$  от точки  $(x,y)$  на плоскости до начала координат
- Вычисление площади прямоугольного треугольника с катетами  $x,y$  и гипотенузой  $h$

[illegible]



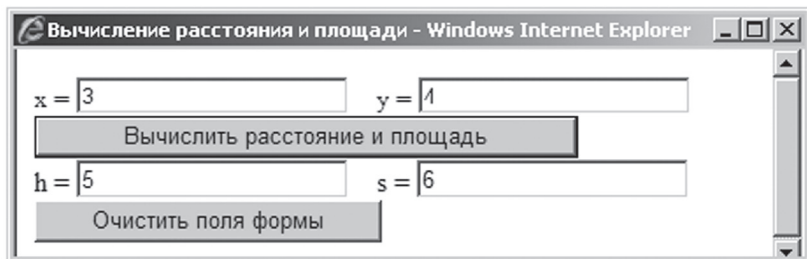


Рис. 4.4. Пример решения в окне браузера сразу двух задач

В теле функции **hs** сначала определим переменную **d**, которой присвоим ссылку на форму документа. Это позволит более кратко записать аргумент стандартной функции **Number**.

Вместо оператора **return** напишем два оператора присваивания:

- Первый оператор записывает в текстовое поле с id-идентификатором **h** вычисленное значение расстояния
- Второй оператор записывает в текстовое поле с id-идентификатором **s** вычисленное значение площади

В теле HTML-страницы определена форма с именем **form2**, в которой содержатся текстовые поля с id-идентификаторами **x**, **y**, **h**, **s** и кнопка с id-идентификатором **Button1**. После щелчка кнопка вызывает функцию **hs**. Кнопка с id-идентификатором **Reset1** очищает все текстовые поля формы.

#### 4.1.4. Возврат значений на форму несколькими функциями

Приведем пример страницы, в сценарии которой определены две функции:

- Функция **circle\_length(r)** с параметром **r** вычисляет длину окружности радиуса **r**. В этой функции используется константа **PI**, которая определена в классе **Math**
- Функция **circle\_square(r)** с параметром **r** вычисляет площадь круга радиуса **r**. В этой функции используются константа **PI** и функция **pow**, которые определены в классе **Math**

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transi-
tional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-tran-
sitional.dtd»>
<html xmlns=http://www.w3.org/1999/xhtml>
<head>
<title>Вычисление длины окружности и площади круга</
title>
<script type=»text/javascript»>
function circle_length(r) //вычисление длины окружности
{   return 2*Math.PI*r
}
function circle_square(r) //вычисление площади круга
{return Math.PI*Math.pow(r,2)    //Math.pow(r,2) = r*r
}
</script>
</head>
<body>
<form action=»HTMLPage_03.htm» name=»form3» >
r = <input id=»r» type=»text» /> <br />
<input id=»Button1» type=»button» value=»Длина окружно-
сти» onclick=»document.form3.len.value=circle_
length(document.form3.r.value)» />
<input id=»len» type=»text» /> <br />
<input id=»Button2» type=»button» value=»Площадь круга»
onclick=»document.form3.squ.value=circle_square(document.
form3.r.value)» />
<input id=»squ» type=»text» /> <br />
<input id=»Reset1» type=»reset» value=»Очистить поля
формы» />
</form>
</body>
</html>

```

В теле страницы определена форма с именем **form3**, содержащая (рис. 4.5):

- Текст «**r** = » и следующее за ним текстовое поле с id-идентификаторами **r**, в которое вводится величина радиуса
- Кнопка с id-идентификатором **Button1** и следующее за ней текстовое поле с id-идентификаторами **len**. Щелчок по этой кнопке приводит к тому, что выбирается величина радиуса, к нему применяется функция **circle\_length**, а ее значение записывается в указанное после кнопки текстовое поле

- Кнопка с id-идентификатором **Button2** и следующее за ней текстовое поле с id-идентификаторами **squ**. Щелчок по этой кнопке приводит к тому, что выбирается величина радиуса, к нему применяется функция **circle\_square**, а ее значение записывается в указанное после кнопки текстовое поле
- Кнопка с id-идентификатором **Reset1** очищает все текстовые поля формы.

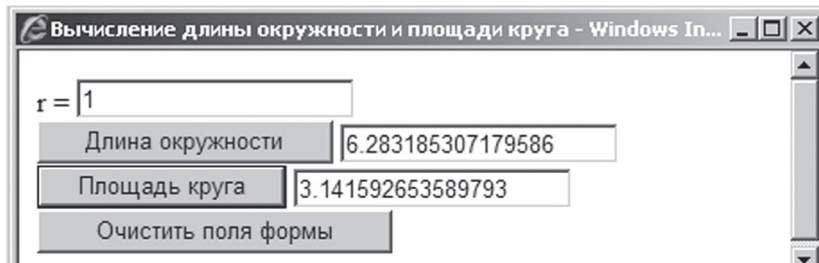


Рис. 4.5. Пример вычисления длины окружности и площади круга в окне браузера

## 4.2. ИСПОЛЬЗОВАНИЕ ВЕТВЛЕНИЙ И ЦИКЛОВ

### 4.2.1. Решение квадратного уравнения

В качестве еще одного примера использования условного оператора рассмотрим решение квадратного уравнения. Для вывода результата решения квадратного уравнения используется многострочное текстовое поле, создаваемое с помощью парных тегов **<textarea>** **</textarea>** и имеющее следующие основные атрибуты (рис. 4.6):

- **cols** — задает размер поля по горизонтали (число символов)
- **rows** задает размер поля по вертикали (число строк; если текст занимает большее число строк, то появляется полоса прокрутки)

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Условный оператор</title>
<script type=»text/javascript»>
```

```

function quadratic_equation() //решение квадратного
уравнения
{ var f = document.form4
  var a = Number(f.Text1.value); var b = Number(f.Text2.
value)
  var c = Number(f.Text3.value); var d = b*b-4*a*c
  if (d<0)
  f.TextArea1.value = "Уравнение решений не имеет"
  else if (d==0)
  f.TextArea1.value = «x = « + (-b/2/a)
  else
  f.TextArea1.value = «x1 = « + ((-b+Math.sqrt(d))/2/a) +
«\n\r» + «x2 = « + ((-b-Math.sqrt(d))/2/a)
  }
</script>
</head>
<body>
<form action=»HTMLPage_01.htm» name=»form4» >
a = <input id=»Text1» type=»text» />&nbsp;
b = <input id=»Text2» type=»text» />&nbsp;
c = <input id=»Text3» type=»text» /> <br />
<input id=»Button1» type=»button» value=»Решить ква-
дратное уравнение»
onclick=»quadratic_equation()» /> <br />
<textarea id=»TextArea1» cols=»50» rows=»2»></textarea>
<input id=»Reset1» type=»reset» value=»Сброс» />
</form>
</body>
</html>

```

Заметим, что если в тексте, который присваивается в качестве значения этому полю, содержится строка «\n\r», то она играет роль управляющего символа перехода на новую строку в элементе **textarea**.

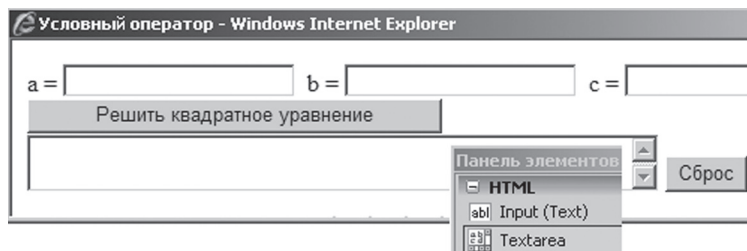


Рис. 4.6. Вид формы с многострочным текстовым полем в окне браузера

#### 4.2.2. Тестирование решения уравнения

Приведем примеры выполнения указанного выше сценария для трех вариантов решения квадратного уравнения (рис .4.7).

The figure consists of four screenshots of a web form titled "Условный оператор - Windows Internet Explorer". Each screenshot shows a different state of the form after clicking the "Решить квадратное уравнение" button.

- First screenshot:** Inputs are  $a=1$ ,  $b=1$ ,  $c=1$ . The result displayed is "Уравнение решений не имеет".
- Second screenshot:** Inputs are  $a=1$ ,  $b=2$ ,  $c=1$ . The result displayed is  $x = -1$ .
- Third screenshot:** Inputs are  $a=1$ ,  $b=3$ ,  $c=1$ . The results displayed are  $x1 = -0.3819660112501051$  and  $x2 = -2.618033988749895$ .
- Fourth screenshot:** Inputs are  $a=abc$ ,  $b=3$ ,  $c=1$ . The results displayed are  $x1 = NaN$  and  $x2 = NaN$ .

Each screenshot includes a "Сброс" (Reset) button.

Рис. 4.7. Пример тестирования сценария решения квадратного уравнения

В четвертом случае для первого параметра квадратного уравнения было указано не числовое значение. Если в сценарии делается попытка выполнить арифметические операции или сравнения над значением, которое не является числом, то результатом будет специальное значение **NaN** (Not a Number)

### 4.2.3. Оператор цикла «for»

При разработке алгоритмов возникает необходимость повторить выполнение отдельного оператора или последовательности операторов определенное число раз. Для этой цели используется **оператор цикла «for»**, который имеет вид (рис. 4.8):

`for (A; B; C) S`

где: **S** — **тело** оператора цикла: отдельный оператор или последовательность операторов, заключенная в фигурные скобки; **A** — оператор определения **переменной цикла** и присвоения ей начального значения; **B** — условие достижения переменной цикла конечного значения, **C** — оператор присваивания переменной цикла нового значения после выполнения тела цикла.

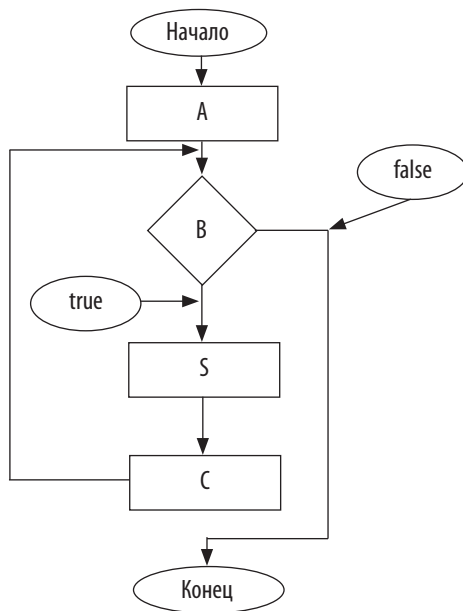


Рис. 4.8. Блок-схема выполнения оператора цикла «for»

Выполнение оператора цикла происходит следующим образом:

- Сначала выполняется оператор A, т.е. определяется переменная цикла и ей присваивается начальное значение.

- Затем вычисляется значение выражения В. Если оно имеет значение false, то выполнение оператора цикла заканчивается.
- Если выражение В имеет значение true, то выполняется S — тело цикла.
- После этого переменной цикла присваивается новое значение с помощью оператора С.
- Затем снова вычисляется значение выражения В. Если оно имеет значение false, то выполнение оператора цикла заканчивается.
- Если выражение В имеет значение true, то выполняется S — тело цикла.
- После этого переменной цикла присваивается новое значение с помощью оператора С и т.д.

#### 4.2.4. Расчет таблицы факториалов

В качестве примера использования оператора цикла рассмотрим решение задачи расчета и вывода на экран таблицы факториалов (рис. 4.9):

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0
Transitional//EN» «http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd»>
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
<title>Таблица факториалов</title>
<script type=»text/javascript»>
function factorials_table()
{
var f = 1; var n = Number(document.form5.Text1.value)
for (var i = 0; i <= n; i++)
{
if (i == 0)
var s = «0!\t=\t1\n\r»
else
{ f *= i; s += i + «!\t=\t» + f + «\n\r» }
}
document.form5.TextArea1.value = s
}
</script>
</head>
<body>
<form action=»HTMLPage_02.htm» name=»form5» >
n = <input id=»Text1» type=»text» /> <br />
```

```

<input id=»Button1» type=»button» value=»Рассчитать
таблицу факториалов»
onclick=»factorials_table()» /> <br />
<textarea id=»TextAreал» cols=»50» rows=»7»></textarea>
<input id=»Reset1» type=»reset» value=»Сброс» />
</form>
</body>
</html>

```

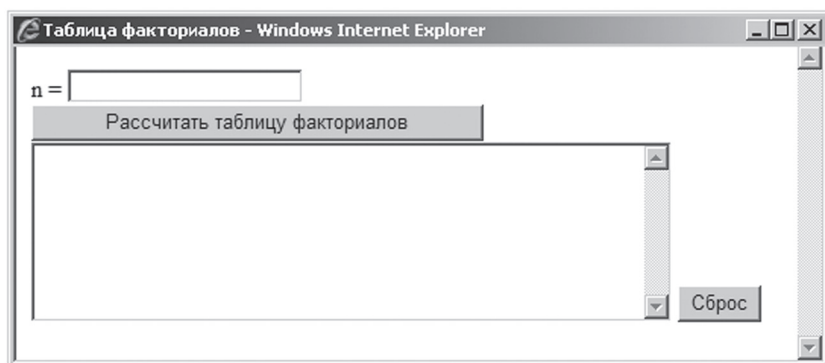


Рис. 4.9. Вид формы для расчета таблицы факториалов в окне браузера

#### 4.2.5. Тестирование расчета таблицы

Приведем примеры выполнения указанного выше сценария (рис. 4.10). В третьем случае для аргумента функции «факториал» было указано не числовое значение. Это привело к тому, что переменной **n** было присвоено значение **NaN**, а результатом проверки условие **i <= n** в заголовке цикла явилось значение **false**. В результате переменной **s** не было присвоено никакого значения. При попытке использовать такую переменную выдается специальное значение **undefined**.



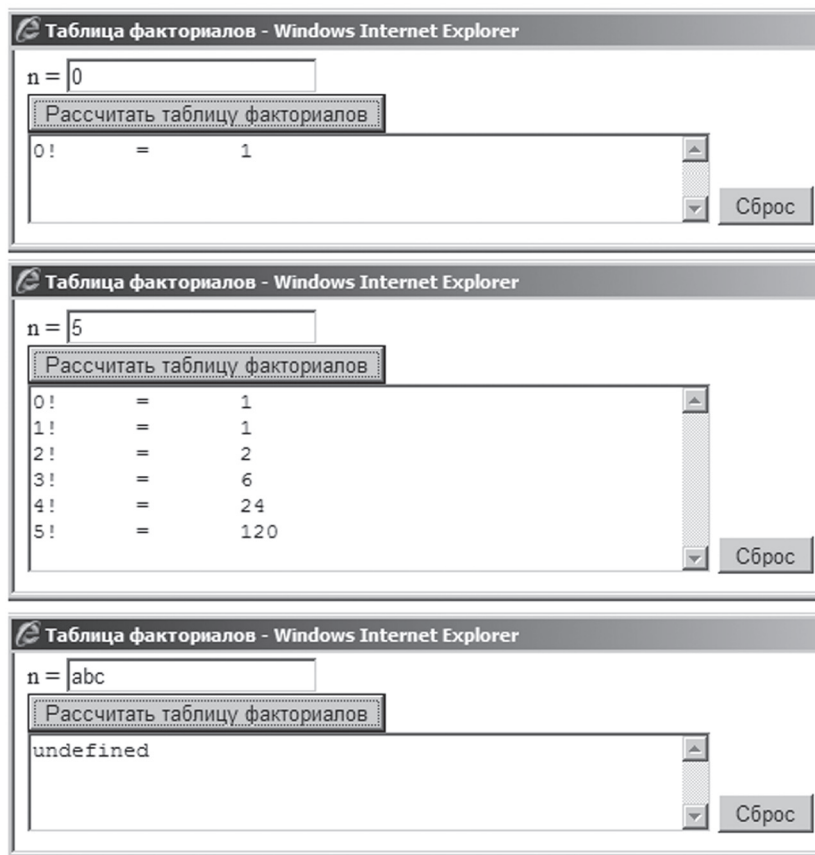


Рис. 4.10. Пример тестирования сценария расчета таблицы факториалов

#### 4.2.6. Оператор цикла «while»

**Оператор цикла «while»** используется в том случае, когда число повторений зависит от некоторого условия:

```
while (B) S
```

где: **S** — **тело** оператора цикла: отдельный оператор или последовательность операторов, заключенная в фигурные скобки; **B** — условие выполнения тела цикла.

Выполнение оператора цикла происходит следующим образом (рис. 4.11):

- Вычисляется значение выражения **B**. Если оно имеет значение **false**, то выполнение оператора цикла заканчивается.
- Если выражение **B** имеет значение **true**, то выполняется **S** — тело цикла.
- Затем снова вычисляется значение выражения **B**. Если оно имеет значение **false**, то выполнение оператора цикла заканчивается.
- Если выражение **B** имеет значение **true**, то выполняется **S** — тело цикла и т.д.

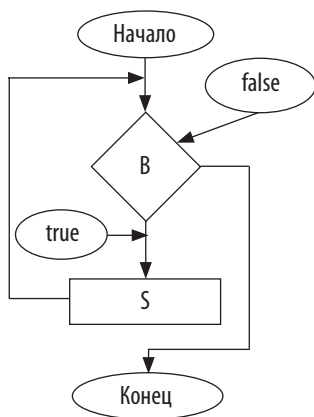


Рис. 4.11. Блок-схема выполнения оператора цикла «while»

#### 4.2.7. Нахождение общего делителя

Рассмотрим решение задачи нахождения наибольшего общего делителя двух чисел, которое использует алгоритм Евклида (Рис.4.12):

$$\text{NOD}(n, 0) = n$$

$$\text{NOD}(n, m) = \text{NOD}(m, n \% m)$$

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Нахождение наибольшего общего делителя двух
чисел</title>
<script type="text/javascript">
function nod(obj)
{
var n = obj.num1.value
var m = obj.num2.value
var p = n % m;
while (p != 0)
{
n = m
m = p
p = n % m
}
obj.num3.value = m
}
</script>
</head>
<body>
<form action="HTMLPage_06.htm" name="form6"> >
Введите число n: <input id="num1" type="text"
size="8"/> <br />
Введите число m: <input id="num2" type="text"
size="8"/> <br />
<input type="button" value="НОД(n,m)= «
onclick="nod(form6)"/>
<input id="num3" type="text" size="8"/> <br />
</form>
</body>
</html>

```

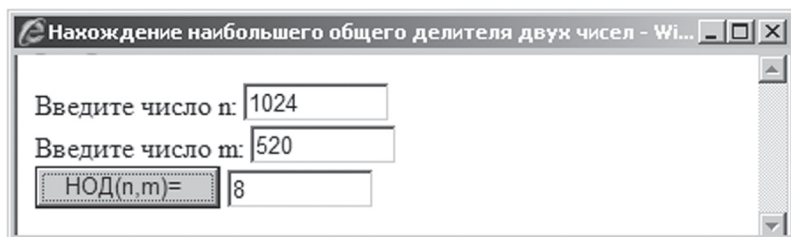


Рис. 4.12. Пример решения задачи нахождения наибольшего общего делителя двух чисел

### 4.3. ИСПОЛЬЗОВАНИЕ ПЕРЕКЛЮЧАТЕЛЕЙ, ФЛАЖКОВ И СПИСКОВ

#### 4.3.1. Использование переключателей на форме

Переключатель, т.е. это элемент управления **Input(Radio)**, : предназначен для выбора одного варианта из некоторого набора вариантов. В качестве примера приведем форму, в которой переключатели используются для выбора формулы для вычисления суммы ряда (рис. 4.13).

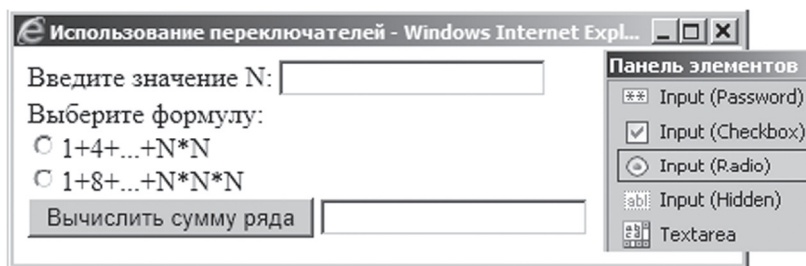


Рис. 4.13. Использование переключателей для выбора формулы

Переключатель функционирует в составе группы переключателей, все элементы которой должны иметь одинаковое значение атрибута **name**. У выбранного переключателя значение свойства **checked** равно **true**, у остальных — **false**.

Приведем пример разметки страницы, решающей указанную выше задачу:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Использование переключателей</title>
<script type="text/javascript">
function sum()
{
var d = document.form7; var N = d.Text1.value
if ( N == "" )
window.alert("Не введено значение N")
else
```

```

{
var s = 0
if (d.Radio1.checked)
{
for(var i = 1; i <= N; i++)
s += Math.pow(i,2);
d.Text2.value = s
}
else if (d.Radio2.checked)
{
for(var i = 1; i <= N; i++)
s += Math.pow(i,3);
d.Text2.value = s
}
else
window.alert("Не выбрана формула")
}
}
</script>
</head>
<body>
<form action=»HTMLPage_07.htm» name=»form7» >
Введите значение N: <input id=»Text1» type=»text» /><br
/> Выберите формулу:<br />
<input id=»Radio1» type=»radio» name =»g»
/>1+4+...+N*N<br />
<input id=»Radio2» type=»radio» name =»g»
/>1+8+...+N*N<br />
<input id=»Button1» type=»button» value=»Вычислить
сумму ряда» onclick=»sum()» />
<input id=»Text2» type=»text» /><br />
</form>
</body>
</html>

```

В данном сценарии используется функция **windows.alert**, которая выводит на экран диалоговое окно с сообщением для пользователя.

### 4.3.2. Тестирование переключателей

Приведем примеры выполнения указанного выше сценария (рис. 4.14).

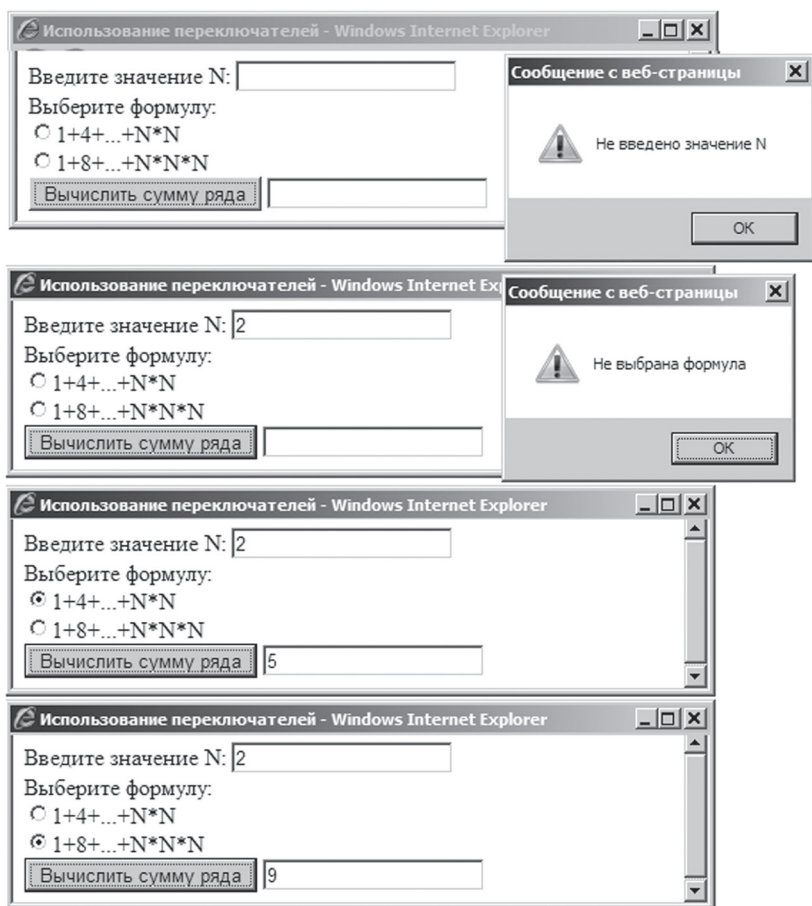


Рис. 4.14. Тестирование выполнения задачи расчета выбранной формулы

### 4.3.3. Использование флажков на форме

Флажок, т.е. элемент управления `Input(Checkbox)`, предназначен используется в том случае, когда из предложенных вариантов можно выбрать как один, так и несколько. У выбранного флажка значения свойства `checked` равно `true`

В качестве примера приведем форму, в которой флажки используются для ответа на вопрос теста (рис. 4.15).

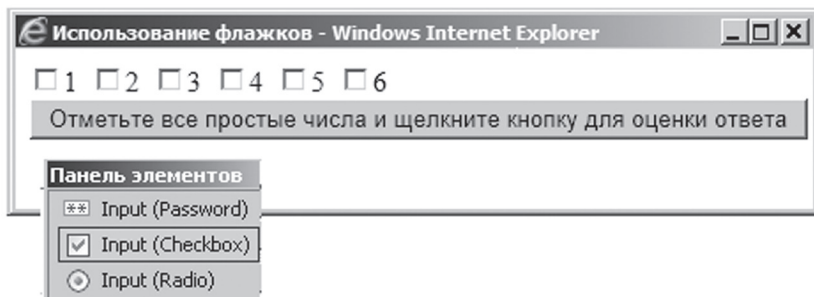


Рис. 4.15. Использование флажков для выбора ответа на вопрос теста

Приведем пример разметки страницы, решающей указанную выше задачу:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" <http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Использование флажков</title>
<script type="text/javascript">
function test()
{
    var d = document.form8; var correct = 0; incorrect = 0;
message = «»; var prime_number =
    ", т.к. делится только на самого себя и 1\n\r"
    if ( d.Ch1.checked)
        correct += 1
    else
    {
        incorrect += 1;
        message += «1 - простое число» + prime_number
    }
    if ( d.Ch2.checked)
        correct += 1
    else
    {
        incorrect += 1;
        message += "2 - простое число» + prime_number
    }
    if ( d.Ch3.checked)
        correct += 1
    else
```

```

{
incorrect += 1;
message += «3 - простое число» + prime_number
}
if ( d.Ch4.checked)
{
incorrect += 1;
message +=
"4 - не простое число, т.к. делится на 2\n\r" }
if ( d.Ch5.checked)
correct += 1
else
{
incorrect += 1;
message += «5 - простое число» + prime_number
}
if ( d.Ch6.checked)
{
incorrect += 1;
message +=
"6 - не простое число, т.к. делится на 2 и 3\n\r"
}
var s =
«Правильных ответов : « + correct + «\n\r» Ошибок : « +
incorrect
if (correct < 4 || incorrect > 0)
s += «\n\r» Запомните: \n\r\n\r» + message
window.alert(s)
}
</script>
</head>
<body>
<form action=»HTMLPage_08.htm» name=»form8» >
<input id=»Ch1» type=»checkbox» />1&nbsp; <input
id=»Ch2» type=»checkbox» />2&nbsp; <input
<input id=»Ch3» type=»checkbox» />3&nbsp; <input
id=»Ch4» type=»checkbox» />4&nbsp; <input
<input id=»Ch5» type=»checkbox» />5&nbsp; <input
id=»Ch6» type=»checkbox» />6 <br />
<input id=»Button1» type=»button» onclick=»test()»
value="Отметьте все простые числа" +
" и щелкните кнопку для оценки ответа" />
</form>
</body>
</html>

```



#### 4.3.4. Тестирование флажков

Приведем примеры выполнения указанного выше сценария (рис. 4.16).

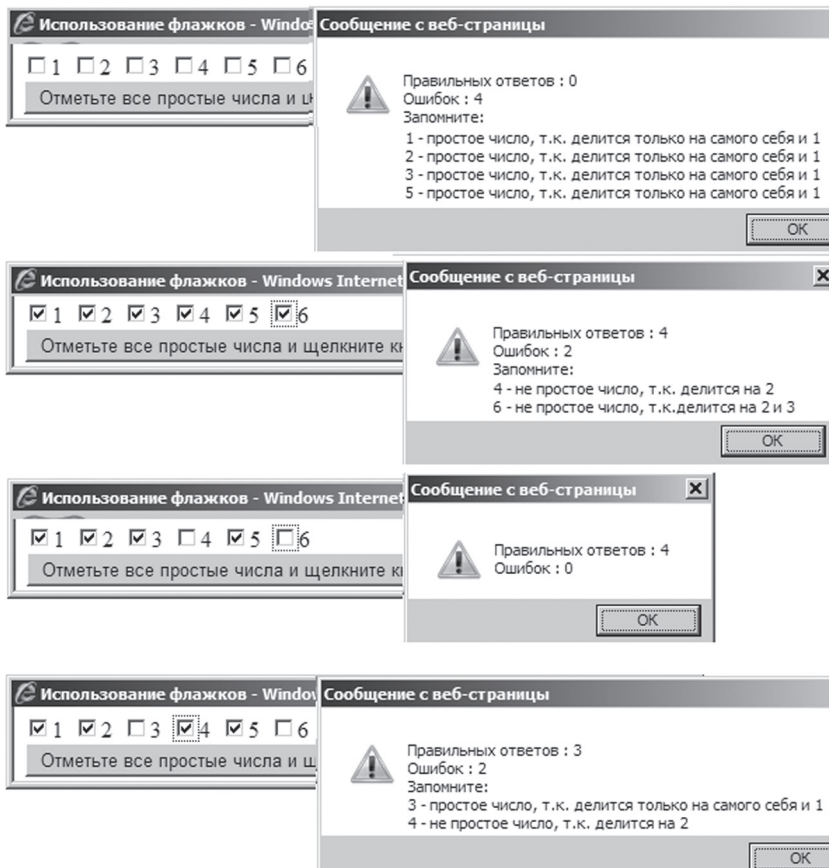


Рис. 4.16. Тестирование выполнения задачи выбора ответа на вопрос теста

#### 4.3.5. Использование списков на форме

Если элементов, из которых нужно сделать выбор, много, то представление их с помощью переключателей или флажков увеличивает размер формы. В этом случае варианты выбора могут быть представлены в окне

Web-браузера более компактно, если используется список, т.е. элемент управления **Select**. В качестве примера приведем форму, в которой список используется для решения той же задачи: выбора ответа на вопрос теста (рис. 4.17).

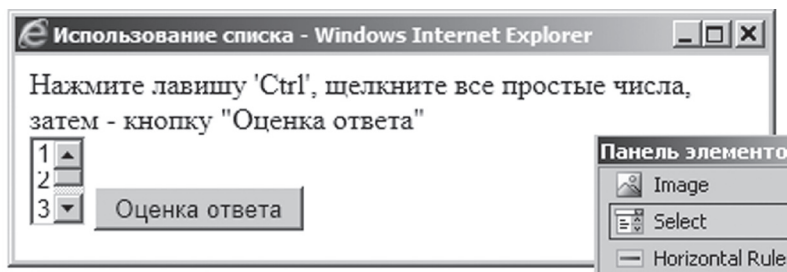


Рис. 4.17. Использование списка для выбора ответа на вопрос теста

Приведем пример разметки страницы, решающей указанную выше задачу:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Использование списка</title>
<script type="text/javascript">
function test()
{
var d = document.form3; var correct = 0; incorrect = 0;
message = «»; var prime_number =
", т.к. делится только на самого себя и 1\n\r"
if ((( d.Select1)[0]).selected)
correct += 1
else
{
incorrect += 1;
message += «1 - простое число» + prime_number
}
if ((( d.Select1)[1]).selected)
correct += 1
else
{
incorrect += 1;
```

```

message += «2 - простое число» + prime_number
}
if ((( d.Select1)[2]).selected)
correct += 1
else
{
incorrect += 1;
message += «3 - простое число» + prime_number
}
if ((( d.Select1)[3]).selected)
{
incorrect += 1;
message +=
"4 - не простое число, т.к. делится на 2\n\r" }
if ( ((d.Select1)[4]).selected)
correct += 1
else
{
incorrect += 1;
message += «5 - простое число» + prime_number
}
if ( ((d.Select1)[5]).selected)
{
incorrect += 1;
message +=
"6 - не простое число, т.к. делится на 2 и 3\n\r"
}
var s = "Правильных ответов : " + correct +
«\n\rОшибок : « + incorrect
if (correct < 4 || incorrect > 0)
s += «\n\rЗапомните: \n\r\n\r» + message
window.alert(s)
}
</script>
</head>
<body>
<form action=»HTMLPage_03.htm» name=»form3» >
Нажмите лавишу 'Ctrl', щелкните все простые числа,
затем - кнопку "Оценка ответа"<br />
<select id=»Select1» size=»3» multiple=»multiple»>
<option>1</option>
<option>2</option>
<option>3</option>
<option>4</option>
<option>5</option>

```

```
<option>6</option>
</select>
<input id=»Button1« type=»button« onclick=»test()«
value=»Оценка ответа« />
</form>
</body>
</html>
```

Видимое на форме число элементов списка задается атрибутом **size**. Если у тега **<select>** указан атрибут **multiple**, то список позволяет сделать множественный выбор элементов при нажатой клавише **Ctrl** (т.е. функционирует как группа флажков). Если этот атрибут отсутствует, то список позволяет сделать только один выбор (т.е. функционирует как группа переключателей).

Для обращения к конкретному элементу списка следует указать его порядковый номер, заключенный в квадратные скобки. Нумерация начинается с нуля. У выбранного элемента списка значение свойства **selected** равно **true**

#### 4.3.6. Тестирование списков

Если в списке сделаны такие же выборы, как в предыдущем примере, то список демонстрирует такую же реакцию, как группа флажков (рис. 4.18).

#### Задания для практических занятий и самостоятельной работы

Для изучения материала четвертой главы в тематическом плане дисциплины предусмотрено 16 часов:

Название тем	Всего часов	Аудиторные часы		Самостоятельная работа
		Лекции	Практические занятия	
Раздел 1. Создание и использование в управленческой практике персональных и корпоративных Web-сайтов				
Тема 1.4. Работа с Web-формами	16	2	6	8

**Задание №1.4.1.** Используя инструментальную систему Microsoft Visual Studio 2008, добавьте в сайт несколько HTML-страниц, рассмотренных в данной теме (может быть, все), затем выполните следующие действия:

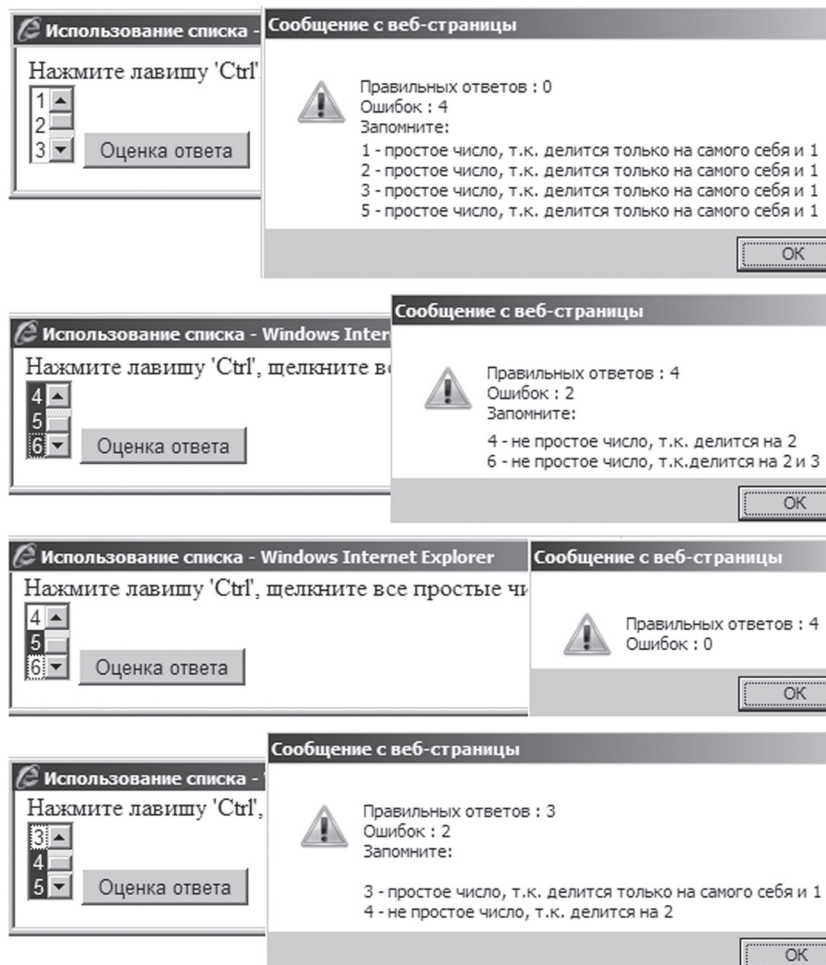


Рис. 4.18. Тестирование выполнения задачи выбора из списка ответа на вопрос теста

- Убедитесь в корректном функционировании добавленных страниц на Web-браузере
- Разместите созданные страницы на вашем сайте в Интернете (на сервере [www.narod.ru](http://www.narod.ru))
- Убедитесь в корректном функционировании добавленных страниц в Интернете

**Порядок передачи результатов выполнения задания №1.4.1 для их оценивания:**

- 
- Следует послать письмо преподавателю по электронной почте
- В качестве темы письма нужно указать номер подгруппы, фамилию и инициалы студента; например: 221В, Иванов А.С.
- В тексте электронного письма следует указать доменное имя, разработанного студентом сайта; например: [www.ivanov.narod.ru](http://www.ivanov.narod.ru)
- Максимальная оценка выполнения задания №1.4.1 составляет 1 балл.

**Задание №1.4.2.** В рамках данной темы 4 часа практических занятий посвящается подготовке к контрольной работе №1:

- Оформление в редакторе Microsoft Office Word 2007 научной статьи по теме научно-исследовательского проекта
- Подготовка на основе статьи доклада в форме электронной презентации в Microsoft Office PowerPoint 2007 (Время на доклад – в пределах 5 минут)

Статья должна состоять из трех частей:

- Введение (обоснование актуальности темы)
- Основная часть (разбитый на отдельные главы обзор исследований, существующих по рассматриваемой теме; информация о сайте в Интернете, на котором опубликованы результаты исследования)
- Заключение (выводы автора о возможном использовании результатов исследования)

Объем статьи не регламентируется; например, она может содержать от 30 000 до 70 000 знаков, включая пробелы. Шрифт Times New Roman, размер 11 пунктов.

В конце работы должна быть приведена библиография, оформленная в соответствии со следующими образцами:

*Для книг:*

*Зомбарт В.* Буржуа. Этюды по истории духовного развития современного экономического человека. М.: Наука, 1994.

*Шаститко А.Е.* Новая теория фирмы. М.: ТЕИС, 1996.

*Для многотомных изданий:*

*Реале Д., Антисерри Дж.* Западная философия от истоков до наших дней: Новое время. Кн. 3. М.: Пневма, 2001.

Для периодических изданий:

Заславская Т.И. Бизнес-слой российского общества: сущность, структура, статус // Социологические исследования. 1995. № 3. С. 3–12.

Для нормативных актов:

Федеральный Закон «О несостоятельности (банкротстве)»// СЗ РФ. 1996. № 11. Ст. 1445.

Для книг на иностранном языке:

Abolafia M. Making Markets, Opportunism and Restraints on Wall Street. Cambridge: Harvard University Press, 1996.

**Порядок передачи результатов выполнения задания №1.4.2 для их оценивания:**

- Следует послать письмо преподавателю по электронной почте
- В качестве темы письма нужно указать номер подгруппы, фамилию и инициалы студента; например: 221В, Иванов А.С.
- К письму следует прикрепить zip-архив, содержащий файл статьи и файл презентации
- Максимальная оценка выполнения задания № 1.4.2 составляет 2 балла.

## ПРИЛОЖЕНИЕ

## РАЗМЕЩЕНИЕ WEB-САЙТА В ИНТЕРНЕТЕ

## П1. Регистрация доменного имени сайта

## П1.1. Запрос на размещение сайта на сервере

Для запроса на размещение сайта на сервере **www.narod.ru** нужно выполнить следующие действия:

- Запустить Web-браузер
- Ввести в поле адреса строку **www.narod.ru**, нажать клавишу Enter
- На странице сервиса «Народ» щелкнуть гиперссылку «Создайте свой сайт» (рис. П1)

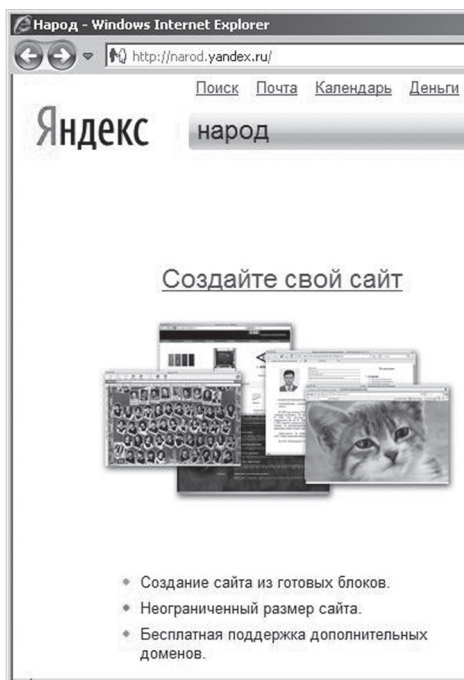


Рис. П1. Страница запроса на размещение сайта на сервере



### П1.2. Заявка на регистрацию доменного имени сайта

В результате выполненных действий происходит переход на страницу входа на сервер (Рис.П2). Если предварительно уже выполнена регистрация доменного имени сайта на этом сервере, то следует ввести логин и пароль, затем щелкнуть кнопку **«Войти»**. В противном случае следует щелкнуть гиперссылку **«Зарегистрироваться»**.

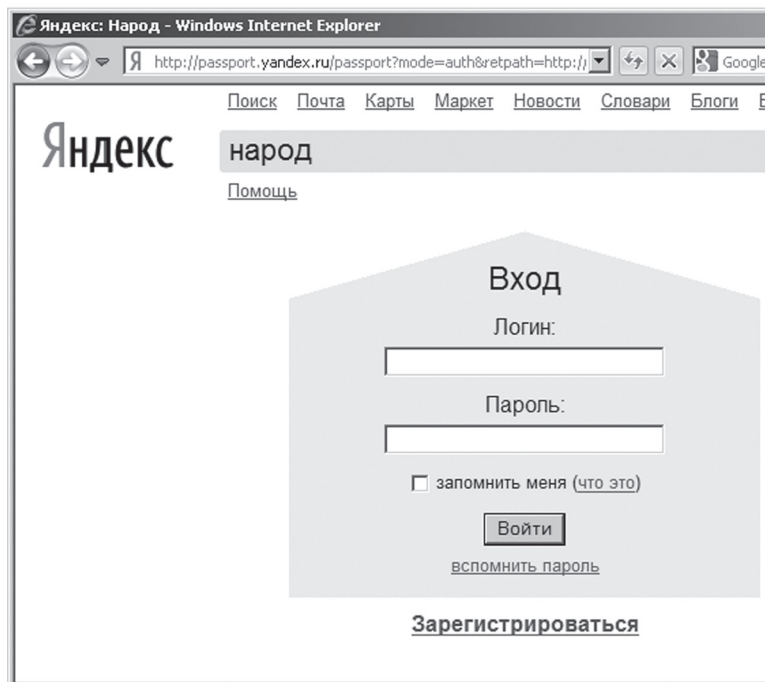


Рис. П2. Страница входа на сервер

### П1.3. Шаг №1 регистрации доменного имени сайта

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, т.е. (регистрационное имя владельца сайта (рис. П3)).

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».

Яндекс: Народ - Windows Internet Explorer

Я http://passport.yandex.ru/passport?mode=register&ncrnd=1655 Google

Поиск Почта Карты Маркет Новости Словари Блоги Видео К

Яндекс народ

Помощь

Регистрация: шаг 1 из 2. Выбор логина.

Логин — это ваш уникальный псевдоним, под которым вас будут узнавать все сервисы Яндекса.  
[Узнать больше](#)

Ваше имя:

Фамилия:

Логин:  @yandex.ru логин свободен

Адрес вашего сайта на Народе будет выглядеть так: kerov10.narod.ru

Вы также можете выбрать логины:

1. leonid-kerov
2. leonidkerov
3. leokerov
4. keleonid
5. leo.kerov
6. kero.leonid
7. kerov1061
8. kerov102009

Яндекс охраняет персональные сведения пользователей в соответствии с [Соглашением о конфиденциальности информации](#).

Дальше →

Рис. ПЗ. Страница для выбора логина

#### П1.4. Шаг №2 регистрации доменного имени сайта

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. П4).

Яндекс: Народ - Windows Internet Explorer

http://passport.yandex.ru/passport?mode=register&ncrnd=494537

### Регистрация: шаг 2 из 2

Вы выбрали логин: **kerov10**

Придумайте пароль:  [Как выбрать пароль](#)  
простой (7 символов)


Подтвердите пароль:   
введено верно

Секретный вопрос:  Если вы забудете пароль, то сможете получить доступ, ответив на этот вопрос.

Ответ:

Другой e-mail:  Этот адрес можно использовать для обращения в службу поддержки. Если вы введете адрес, на него будет выслан запрос о подтверждении.

Мобильный телефон:  необязательно  
в формате +7 916 123 11 22  
Если вы забудете пароль, мы сможем отправить вам SMS с кодом для его восстановления. На следующей странице будет показана инструкция для подтверждения номера телефона.



Введите цифры слева:

Робот не сможет ввести цифры с картинки, а вы сможете.

☒ Я принимаю условия Пользовательского соглашения.

Рис. П4. Страница для выбора пароля

## П2. Публикация сайта

### П2.1. Переход на страницу управления сайтом

Щелчок по кнопке «**Зарегистрировать**» приводит к завершению регистрации доменного имени вашего сайта и переход на страницу «**Управление сайтом**» (Рис.П5).

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**».

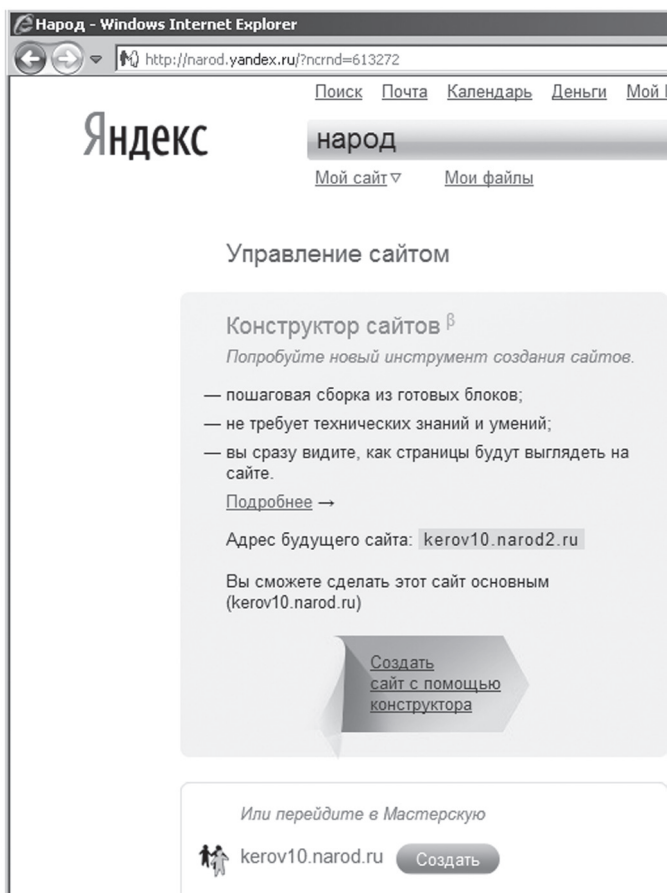


Рис. П5. Страница управления сайтом

## П2.2. Страница «Мастерская»

Страница «**Мастерская**» предназначена для выбора команд управления сайтом, основными из которых являются (Рис.П6):

- Загрузка файлов на сайт
- Управление файлами и HTML-редактор

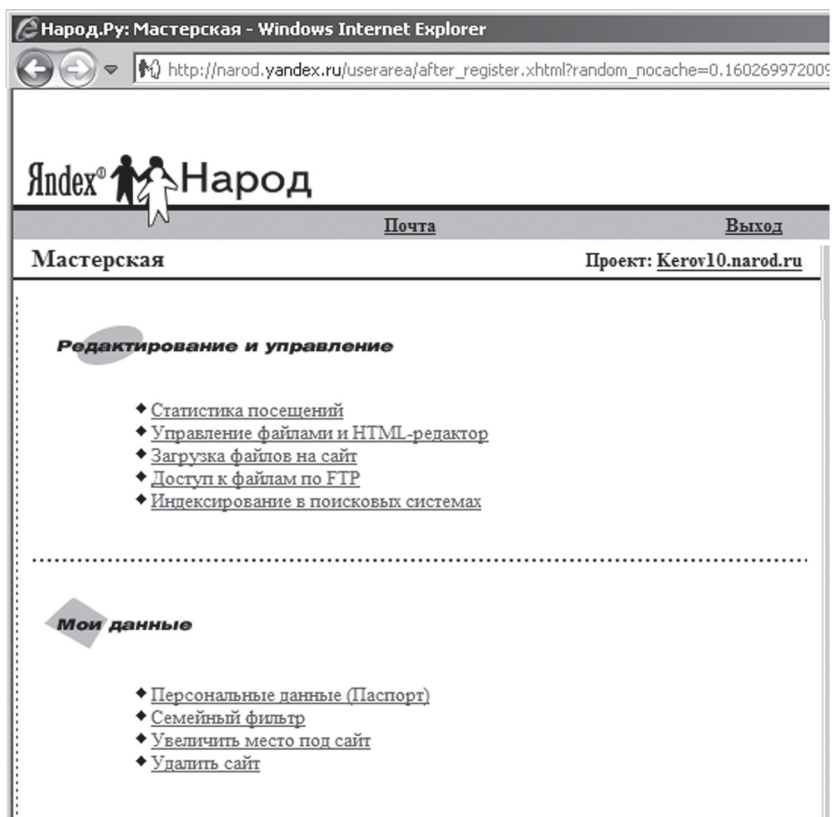


Рис. П6. Страница «Мастерская»

### П2.3. Выбор папки для загрузки файлов


Выбор команды **«Загрузка файлов на сайт»** приводит к переходу на страницу загрузки файлов. Если файлы нужно загружать не в корневую папку сайта, то следует щелкнуть по гиперссылке **«Выбрать папку»**.

В результате происходит переход на страницу выбора папки для загрузки, где следует выбрать нужную папку и щелкнуть кнопку **«Выбрать»** (рис. П7).

Мастерская / Управление файлами / Загрузка файлов

Загрузка в папку «Корневая папка» [Выбрать папку](#)

<input type="text"/>	<input data-bbox="342 539 445 560" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="681 539 784 560" type="button" value="Browse..."/>
<input type="text"/>	<input data-bbox="342 571 445 592" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="681 571 784 592" type="button" value="Browse..."/>
<input type="text"/>	<input data-bbox="342 603 445 624" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="681 603 784 624" type="button" value="Browse..."/>
<input type="text"/>	<input data-bbox="342 635 445 655" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="681 635 784 655" type="button" value="Browse..."/>
<input type="text"/>	<input data-bbox="342 667 445 687" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="681 667 784 687" type="button" value="Browse..."/>

Яндекс  Народ

[Почта](#)
[Помощь](#)

Мастерская / Управление файлами / Загрузка файлов / [Выбор папки для загрузки](#)

☒ корневая папка  
☐ Images  
☐ News

Рис. П7. Выбор папки для загрузки файлов

#### П2.4. Загрузка файлов на сайт

Для загрузки ранее созданных файлов на сайт следует щелкнуть кнопку «**Browse**», затем в диалоговом окне выбрать требуемый файл. За одну операцию загрузки можно переместить на сайт до 10 файлов. Выбрав файлы, нужно щелкнуть кнопку «Загрузить файлы» (рис. П8).

Мастерская / Управление файлами / Загрузка файлов			
Загрузка в папку « <b>Корневая папка</b> » <a href="#">Выбрать папку</a>			
<input type="text"/>	<input data-bbox="397 512 517 539" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="784 512 904 539" type="button" value="Browse..."/>
<input type="text"/>	<input data-bbox="397 547 517 574" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="784 547 904 574" type="button" value="Browse..."/>
<input type="text"/>	<input data-bbox="397 582 517 609" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="784 582 904 609" type="button" value="Browse..."/>
<input type="text"/>	<input data-bbox="397 617 517 644" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="784 617 904 644" type="button" value="Browse..."/>
<input type="text"/>	<input data-bbox="397 652 517 679" type="button" value="Browse..."/>	<input type="text"/>	<input data-bbox="784 652 904 679" type="button" value="Browse..."/>
<p>Объём каждого файла не должен превышать 5 Мб. При этом многотомный архив считается одним файлом.</p>			
<input data-bbox="324 930 591 962" type="button" value="Загрузить файлы"/>		<input data-bbox="604 930 746 962" type="button" value="Очистить"/>	

Рис. П8. Загрузка файлов на сайт

## П2.5. Управление файлами сайта

Выбор команды «Управление файлами и HTML-редактор» приводит к переходу на страницу управления файлами, где можно выполнить следующие действия (рис. П9):

- Копировать, переместить или удалить файл
- Создать новую папку или удалить имеющуюся

Мастерская / Управление файлами		
		<a href="#">Создать папку</a>
Отмеченное:	<input type="checkbox"/> Копировать	<input type="checkbox"/> Переместить <input type="checkbox"/> Удалить
<input type="checkbox"/>	Имя	Описание
<input type="checkbox"/>	Папки	
<input type="checkbox"/>	<a href="#">Images</a>	
<input type="checkbox"/>	<a href="#">News</a>	
<input type="checkbox"/>	Страницы	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">CompTech 00.htm</a>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">CompTech 01.htm</a>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">CompTech 02.htm</a>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">CompTech 03.htm</a>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">CompTech 04.htm</a>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">CompTech 05.htm</a>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">Frameset CompTech 00.htm</a>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">Frameset CompTech 01.htm</a>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> <a href="#">Frameset CompTech 02.htm</a>	

Рис. П9. Управление файлами сайта



## 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. *Дмитриева М.В.* Самоучитель JavaScript. — СПб.: БХВ-Петербург — 2001. — 512 с.
5. *Дмитриева М.В.* JavaScript. Быстрый старт. — СПб.: БХВ-Петербург, 2002. — 336 с.
6. *Дронов В.А.* JavaScript и AJAX в Web-дизайне: 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2008. — 736 с.
7. *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.
8. *Керов Л.А., Афанасьева С.В.* Компьютерные технологии офисной деятельности. Учебное пособие. — СПб: Издательство «Ютас», 2010. — 90 с.
9. *Леонтьев В.П.* Новейшая энциклопедия Интернета 2010. — М.: ОЛМА Медиа Групп, 2009. — 640 с.
10. *Пауэрс Л.* Microsoft Visual Studio 2008: Пер. с англ. — СПб.: БХВ-Петербург, 2009. — 1200 с.
11. *Стоцкий Ю., Васильев А., Телина И.* Office 2007. Изучаем самостоятельно. — СПб.: Питер, 2008. — 524 с.
12. *Close R.C., Humphreys R., Ruttenbug B.W.* E-Learning and Knowledge Technology & The Internet Are Changing the Way We Learn. — Sun Trust Equitable Securities, 2000.
13. *Johnson G., Northrup T.* Microsoft .NET Framework 2.0 Web-Based Client Development. MCTS Self-Paced Training Kit (Exam 70-528). — Redmond: Microsoft Press, 2007.
14. *Stoecker M.A., Stein S.J., Northrup T.* Microsoft .NET Framework 2.0 Windows-Based Client Development. MCTS Self-Paced Training Kit (Exam 70-526). — Redmond: Microsoft Press, 2007.

## СОДЕРЖАНИЕ

<b>ГЛАВА 1. СОЗДАНИЕ WEB-САЙТА В MICROSOFT VISUAL STUDIO .....</b>	<b>3</b>
<b>1.1. СОЗДАНИЕ САЙТА С HTML-СТРАНИЦАМИ .....</b>	<b>3</b>
1.1.1. Создание Web-сайта.....	3
1.1.2. Добавление HTML- страницы .....	5
1.1.3. Разметка заголовков .....	6
1.1.4. Разметка абзацев .....	8
1.1.5. Вложенность тегов.....	9
1.1.6. Просмотр страницы в браузере.....	10
<b>1.2. РАБОТА С ТЕКСТОМ HTML-СТРАНИЦЫ .....</b>	<b>11</b>
1.2.2. Нумерованные списки.....	13
1.2.3. Списки определений.....	14
1.2.4. Управление переносом строк.....	15
1.2.5. Специальные символы .....	16
<b>1.3. РАБОТА С ГИПЕРССЫЛКАМИ .....</b>	<b>17</b>
1.3.1. Создание гиперссылки.....	17
1.3.2. Функционирование гиперссылки .....	19
1.3.3. Адресация в World Wide Web .....	20
1.3.4. Создание закладок.....	21
<b>1.4. РАБОТА С ГРАФИКОЙ HTML-СТРАНИЦЫ .....</b>	<b>22</b>
1.4.1. Графические файлы .....	22
1.4.2. Вставка изображения.....	23
1.4.3. Изображение-гиперссылка.....	25
1.4.4. Создание карт навигации .....	26
<b>1.5. РАБОТА С ТАБЛИЦАМИ HTML-СТРАНИЦЫ.....</b>	<b>30</b>
1.5.1. Создание таблицы .....	30
1.5.2. Атрибуты тега <table> .....	31
1.5.3. Атрибуты тегов <tr> и <td>.....	32
1.5.4. Объединение ячеек в строке .....	34
1.5.5. Объединение ячеек в столбце.....	35

<b>1.6. Всплывающие подсказки .....</b>	<b>37</b>
<b>1.7. Горизонтальные линии .....</b>	<b>38</b>
<b>1.8. Комментарии.....</b>	<b>39</b>
<b>1.9. Пролог HTML-страницы.....</b>	<b>40</b>
<b>1.10. РАБОТА С ФРЕЙМАМИ HTML-СТРАНИЦЫ .....</b>	<b>41</b>
1.10.1. Разделение окна браузера.....	41
1.10.2. Задание параметров фреймов .....	43
1.10.3. Структура загружаемых страниц .....	43
1.10.4. Страница с гиперссылками .....	45
1.10.5. Навигация внутри страницы .....	47
 <b>ГЛАВА 2. СТИЛЕВОЕ ОФОРМЛЕНИЕ HTML-СТРАНИЦ.....</b>	<b>52</b>
<b>2.1. Понятие и виды стилей.....</b>	<b>52</b>
2.1.1. Встроенный стиль .....	52
2.1.2. Внутренняя таблица стилей .....	53
2.1.3. Стилиевой класс.....	54
2.1.4. Стилиевой селектор .....	56
2.1.5. Стил для нескольких тегов.....	57
2.1.6. Комбинированный стил .....	58
2.1.7. Внешняя таблица стилей .....	59
2.1.8. Связь HTML-страницы с таблицей.....	60
2.1.9. Каскадное определение стилиа.....	62
 <b>2.2. Редактор и атрибуты стилей .....</b>	<b>63</b>
2.2.1. Запуск редактора стилей.....	63
2.2.2. Атрибуты шрифта .....	64
2.2.3. Атрибуты блока.....	65
2.2.4. Атрибуты фона .....	66
2.2.5. Атрибуты границы.....	67
 <b>2.2.6. Атрибуты поля.....</b>	<b>68</b>
2.2.7. Атрибуты положения .....	68
2.2.8. Атрибуты макета .....	69
2.2.9. Атрибуты списка.....	70
2.2.10. Атрибуты таблицы.....	71
 <b>2.3. Стили ГИПЕРССЫЛОК .....</b>	<b>72</b>

<b>2.4. КОНТЕЙНЕРЫ HTML-СТРАНИЦЫ .....</b>	<b>73</b>
<b>2.5. ПРИМЕР WEB-ДИЗАЙНА САЙТА .....</b>	<b>74</b>
2.5.1. Графика для домашней страницы.....	74
2.5.2. Стилизовое оформление страниц .....	75
2.5.3. Разметка домашней страницы.....	76
2.5.4. Разметка страницы контента.....	78
2.5.5. Проверка функционирования сайта.....	80
<b>ГЛАВА 3. ПРОГРАММНОЕ УПРАВЛЕНИЕ HTML-СТРАНИЦАМИ.....</b>	<b>83</b>
<b>3.1. СЦЕНАРИИ .....</b>	<b>83</b>
3.1.1. Понятие сценария .....	83
3.1.2. Литералы и операции .....	85
3.1.3. Понятие переменной .....	86
3.1.4. Оператор присваивания .....	88
3.1.5. Файлы сценариев.....	89
<b>3.2. ОБРАБОТКА СОБЫТИЙ СТРАНИЦЫ .....</b>	<b>90</b>
3.2.1. События и их обработчики.....	90
3.2.2. Обработчик с параметром .....	93
<b>ГЛАВА 4. РАБОТА С WEB-ФОРМАМИ .....</b>	<b>96</b>
<b>4.1. ПОНЯТИЕ И СОЗДАНИЕ WEB-ФОРМЫ .....</b>	<b>96</b>
4.1.1. Понятие Web-формы .....	96
4.1.2. Текстовые поля и кнопки .....	97
4.1.3. Возврат нескольких значений на форму .....	100
4.1.4. Возврат значений на форму несколькими функциями.....	101
<b>4.2. ИСПОЛЬЗОВАНИЕ ВЕТВЛЕНИЙ И ЦИКЛОВ.....</b>	<b>103</b>
4.2.1. Решение квадратного уравнения.....	103
4.2.2. Тестирование решения уравнения .....	105
4.2.3. Оператор цикла «for» .....	106
4.2.4. Расчет таблицы факториалов.....	107
4.2.5. Тестирование расчета таблицы .....	108
4.2.6. Оператор цикла «while» .....	109
4.2.7. Нахождение общего делителя.....	110
<b>4.3. ИСПОЛЬЗОВАНИЕ ПЕРЕКЛЮЧАТЕЛЕЙ, ФЛАЖКОВ И СПИСКОВ.....</b>	<b>112</b>
4.3.1. Использование переключателей на форме .....	112
4.3.2. Тестирование переключателей.....	113
4.3.3. Использование флажков на форме .....	114

4.3.4. Тестирование флажков .....	117
4.3.5. Использование списков на форме.....	117
4.3.6. Тестирование списков.....	120
<b>ПРИЛОЖЕНИЕ.....</b>	<b>124</b>
<b>РАЗМЕЩЕНИЕ WEB-САЙТА В ИНТЕРНЕТЕ .....</b>	<b>124</b>
<b>СПИСОК ЛИТЕРАТУРЫ .....</b>	<b>133</b>
<b>ОГЛАВЛЕНИЕ.....</b>	<b>134</b>

*Л. Керов*

**ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ  
ТЕХНОЛОГИИ В УПРАВЛЕНЧЕСКОЙ ПРАКТИКЕ**

Часть 1. Web-дизайн

---

Подписано в печать .10.2011.  
Бумага мелованная. Гарнитура Myriad Pro. Печать цифровая.  
Тираж 70 экз. Заказ 128.

---

Подготовлено к печати и отпечатано  
Отделом оперативной полиграфии  
НИУ ВШЭ — Санкт-Петербург.  
198099, Санкт-Петербург, ул. Промышленная, д. 17а  
Тел./факс (812) 786-58-95