

Министерство общего и профессионального образования Свердловской области
государственное автономное профессиональное образовательное учреждение
Свердловской области
«Ирбитский мотоциклетный техникум» (ГАПОУ СО «ИМТ»)

**ПРОГРАММА ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА
ПО СПЕЦИАЛЬНОСТИ
09.02.04 Информационные системы (по отраслям)**

**МЕЖДИСЦИПЛИНАРНЫЙ КУРС
МДК.02.01 Информационные технологии и платформы разработки
информационных систем**

**КОМПЛЕКС МЕТОДИЧЕСКИХ УКАЗАНИЙ К
ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

Ирбит 2017

Содержание

Пояснительная записка.....	3
Комплекс лабораторных работ.....	5

Пояснительная записка

Лабораторная работа - это важный элемент учебного процесса. Именно на таких занятиях студенты получают практические умения и навыки работы с программным обеспечением, лучше усваивают и закрепляют изученный теоретический материал.

Если лекция закладывает основы научных знаний в обобщенной форме, лабораторная работа призвана углубить, расширить и детализировать эти знания, содействовать выработке навыков профессиональной деятельности. Лабораторные работы развивают научное мышление и речь студентов при защите этой работы, позволяют проверить их знания, в связи с чем, лабораторные работы выступают важным средством достаточно оперативной обратной связи.

Для успешной подготовки к лабораторной работе студенту невозможно ограничиться слушанием лекций. Требуется предварительная самостоятельная работа студентов по теме планируемого занятия. Не может быть и речи об эффективности занятий, если студенты предварительно не поработают над конспектом, учебником, учебным пособием, чтобы основательно овладеть теорией вопроса.

Лабораторная работа служит своеобразной формой осуществления связи теории с практикой. Структура лабораторной работы в основном одинакова — вступление преподавателя, где осуществляется постановка задач на занятие, вопросы студентов по материалу, который требует дополнительных разъяснений, собственно практическая часть, защита выполненной работы и заключительное слово преподавателя. Цель занятий должна быть понятна не только преподавателю, но и студентам. Это придает учебной работе жизненный характер, утверждает необходимость овладения опытом профессиональной деятельности, связывает их с практикой жизни.

Студенты, как правило, отдают себе отчет в том, в какой мере им необходимы данные лабораторной работы для предстоящей профессиональной деятельности. Если студенты поймут, что все учебные возможности занятий исчерпаны, интерес к ним будет утрачен. Учитывая этот психологический момент, очень важно организовать занятия так, чтобы студенты постоянно

ощущали рост сложности выполняемых заданий, что ведет к переживанию собственного успеха в учении и положительно мотивирует студента. Если же студенты замечают «топтание на месте», уровень мотивации может заметно снизиться.

Преподаватель должен проводить занятия так, чтобы каждый студент получил возможность «раскрыться», проявить способности, поэтому при разработке плана занятий и индивидуальных заданий преподаватель должен учитывать подготовку и интересы каждого студента. Преподаватель при этом будет выступать в роли консультанта, наблюдающего за работой каждого студента и способного вовремя оказывать педагогически оправданную помощь. При такой организации проведения занятий в лаборатории не возникает мысли о том, что возможности занятий исчерпаны.

При проведении лабораторных занятий особенно важно, как, впрочем, и в учении вообще, учитывать роль повторений. Однообразие заданий, субъективное ощущение повторения как замедления движения вперед значительно ухудшают усвоение. Поэтому важно не проводить повторения в формировании заданий на лабораторных работах.

Существуют различные формы проведения лабораторной работы с применением компьютера:

1. Работа с готовой программой.
2. Самостоятельное решение предлагаемой преподавателем задачи.
3. Моделирование и усложнение предлагаемой преподавателем программы.

Преподаватель выполняет консультирующую, координирующую и направляющую функцию. Очень высока степень самостоятельности учащихся, на нее отводится 70% времени занятия.

Комплекс лабораторных работ

Лабораторная работа №1. Создание веб-страницы с помощью HTML-редактора

Основные элементы HTML

`<html>`

`</html>` – теги, определяющие начало и конец документа.

`<head>`

`</head>` – служебная секция. Здесь размещаются команды для браузеров, инструкции для поисковых роботов, ссылки на отдаленные файлы, скрипты.

`<title>`

`</title>` – теги, определяющие главный заголовок документа.

`<body>`

`</body>` – эта секция содержит всю видимую часть web-страницы.

Задание

1. Откройте блокнот, сохраните под своим именем в своей папке, с расширением html.
2. Наберите первые основные теги, отображающие Html структуру.
3. Создайте заголовок. В тэгах `<title>` `</title>` укажите свое название (например: Лабораторная №1).

Тэг тела документа

1. В теле документа наберите текст «Здравствуйте, это моя первая страница».
2. Добавьте в тело документа следующую фразу: «Добро пожаловать!»
3. Измените цвет фразы «Добро пожаловать!» со стандартного черного на "#CC0000". Используйте тэг font.

Заголовки и абзацы

1. Создайте заголовок первого уровня. Напишите «Заголовок 1». Создайте заголовок 5 уровня, с выравниванием по правому краю, используйте атрибут align.
2. Оформите абзацы и параграфы с помощью тегов `<p>` и `
` в тексте: «Структура HTML по большому счету лежит в основе каждой web - страницы. Сегодня существует несколько типов HTML документов. Каждому типу документа соответствует его собственная HTML структура. Правильная начальная структура HTML является залогом кросс-браузерности сайта.»
3. Измените структуру выше изложенного текста, используйте теги форматирования:

Теги `` `` → жирный HTML текст (жирный шрифт).

Теги `<code>` `</code>` → моноширинный HTML текст (моноширинный

шрифт).

Теги `<code> </code>` → моноширинный HTML текст (моноширинный шрифт).

Теги `<samp> </samp>` → моноширинный HTML текст (моноширинный шрифт).

Теги `<big> </big>` → HTML текст, размер больше обычного (крупный шрифт).

Теги `<small> </small>` → HTML текст, размер меньше обычного (мелкий шрифт).

Теги ` ` → наклонный HTML текст (наклонный шрифт).

Теги `<i> </i>` → наклонный HTML текст (наклонный шрифт).

Теги `<dfn> </dfn>` → наклонный HTML текст (наклонный шрифт).

Теги `<ins> </ins>` → подчеркнутый HTML текст (подчеркнутый шрифт).

Теги ` ` → зачеркнутый HTML текст (зачеркнутый шрифт).

Теги `` → HTML текст (шрифт) в нижнем индексе.

Теги `` → HTML текст (шрифт) в верхнем индексе.

4. Измените размер шрифта, используйте следующий тег с атрибутами: `<p style=«font-size:9px»>Текст высотой 9 пикселей</p>`

5. Измените цвет текста. Используйте следующие теги:

HTML цвет текста может быть задан в режиме RGB, например, `color:«#cc0000»`.

Цвет текста в HTML также может быть задан словом, например, `color:«Black»`.

Блоки и параграфы

HTML параграф определяется тегами `<p> </p>`.

HTML параграфы обыкновенно содержат текст, форматизирующие теги, изображения.

HTML параграф не может содержать внутри себя блочные элементы, такие как заголовки

`<h1> – </h6>`, блок `<div> </div>` и другие параграфы.

HTML блок определяется тегами `<div> </div>`.

HTML блок может вмещать в себя любые элементы web-страницы, в любом количестве.

HTML блоки прекрасно подходят для верстки интернет-страниц, ими легко манипулировать

1. Наберите в теле документа следующие строки и посмотрите результат:

```
<p>Первая строка</p>
<p>Вторая строка</p>
<div>Третья строка</div>
<div>Четвертая строка</div>
<div style="color:#ff0000"><p>Пятая строка</p></div>
<div>Шестая строка</div>
```

2. Создайте горизонтальную линию. Используйте код, указанный ниже:

```

<hr align="center" width="500" size="2" color="#ff0000">
<br>
<hr align="left" width="300" size="4" color="#ff9900">
<br>
<hr align="right" width="400" size="3" color="#0000dd">

```

3. Самостоятельно создайте горизонтальную линию зеленого цвета под текстом «Заголовок 1»

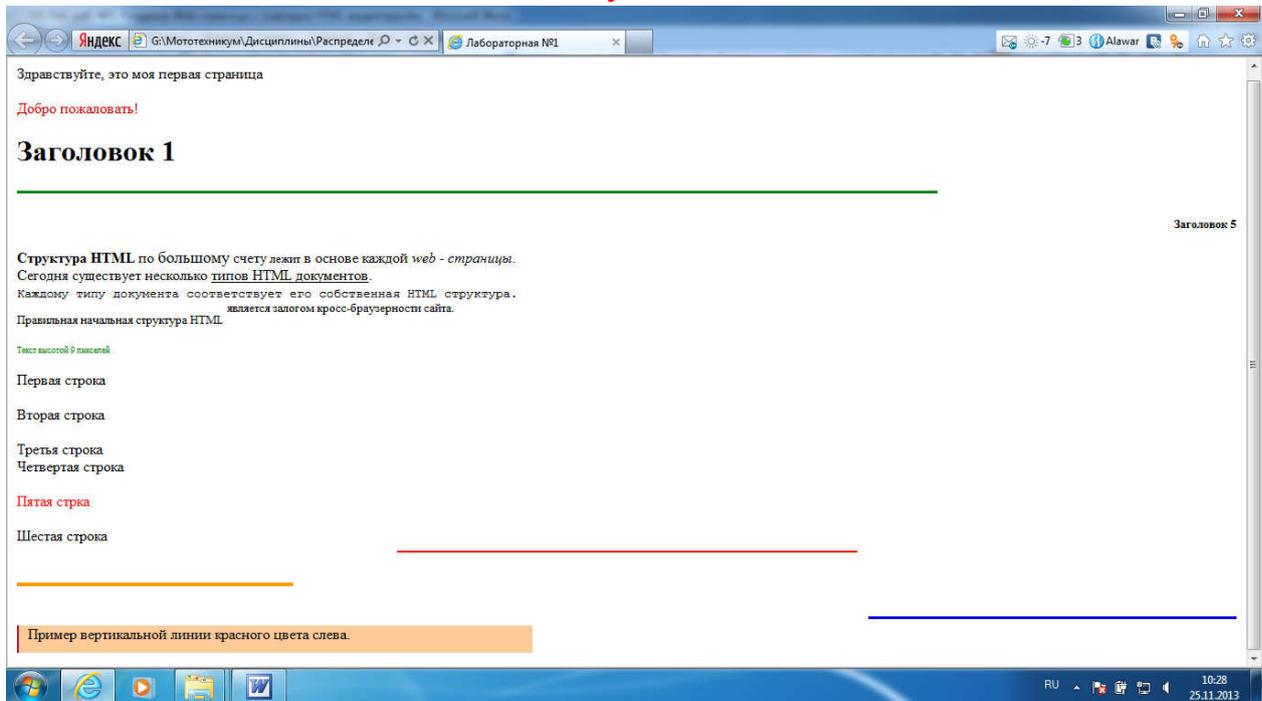
4. Создайте вертикальную линию. Используйте следующий код:

```

<div style="height:30px; width:560px; border-left:2px solid #cc0033; background-color:#fc9; padding-left:10px">Пример вертикальной линии красного цвета слева. </div>

```

Результат



Самостоятельная работа

1. Создайте страницу по примеру, предложенному ниже. Посмотрите результат.

```

<html>
<head>
<title>Пример 1.1</title>
</head>
<body>
<h1 align=center>Привет всем жителям Земли!</h1>
<hr size=4 width=100%>
<h3>Скоро все Вы узнаете как круто я создаю сайты!</h3>
<p>Это только начало, сейчас я просто учусь...</p>
<p align=left>Выравнивать текст по левому краю</p>
<p align=center>И по центру могу!</p>
<p align=right>И вправо легко!</p>
<hr size=2 width=50% color=#cc99ff>
<h2 align=center>Вот какой я молодец!!!<br>Это вовсе не конец!!!</h2>
</body>
</html>

```

2. Создание домашней страницы содержащей информацию о студенте. Обязательное содержание:

- Информация о том, чей сайт и фото студента.

- Электронный почтовый адрес.
- Ссылка на рабочую страницу или сайт техникума студента.
- Мой техникум.
- Моя группа.
- Моя будущая профессия.
- Мои увлечения или хобби.
- Любая другая информация.

Должно получиться 3 документа

Лабораторная работа №2. Форматирование текста, списков и таблиц в HTML-редакторе

Задание

1. Построить в HTML-редакторе таблицу:

Таблица задается тэгом: `<table></table>`.

Увы, это далеко не всё: таблица состоит из строк и столбцов (ячеек), поэтому нам надо еще указать и их: `<tr></tr>` - строчка таблицы; `<td></td>` - столбец (ячейка) таблицы.

1	2	3	
1x1	1x2	1x3	1
2x1	2x2	2x3	2

Итак, перед вами таблица из двух строк и трех столбцов (ячеек). Границы таблицы не заданы, поэтому вы их не видите. Как же получилась эта таблица? Выполним следующие действия:

Сначала мы зададим строки. В нашем примере их две. Теперь в каждой строке зададим по три столбца (ячейки).

Кстати, когда вы будете создавать свой сайт и таблицы, лучше рисуйте все сначала на бумаге, так вам будет легче потом верстать, т.к. вы все наглядно будете видеть на бумаге, и у вас будет меньше вероятности ошибиться.

Итак, теперь нам надо заполнить получившийся каркас.

Надеюсь вы поняли, что первая цифра в загадочных надписях - это номер ряда, а вторая номер ячейки (1x2 - первый ряд, вторая ячейка). Это опять же вам для наглядности. Если посмотреть то, что уже у нас с вами получилось, то это будет выглядеть так:

1x1 1x2 1x3
2x1 2x2 2x3

Увы, фона (голубого “#336699” или желтого “#FFCC33”, как в примере) еще не видно. Фон задается атрибутом `bgcolor="цвет_фона"`. Фон можно задать для таблицы в целом, для ряда, для ячейки (в пределах одного ряда). В нашем случае мы задаем фон для каждой ячейки.

Вот, что у нас получилось:

1x1	1x2	1x3
2x1	2x2	2x3

Если вы хотите задать фон для ряда, то атрибут `bgcolor` мы прописываем для тэга `<tr>`: `<tr bgcolor="#FFCC33">`;

Если вы хотите задать фон для всей таблицы, то атрибут `bgcolor` мы прописываем для тэга `<table>`: `<table bgcolor="#FFCC33">`.

Однако, если при заданном фоне для всей таблицы, вы задаете свой фон для ряда или ячейки, то этот ряд или ячейка будут иметь фон отличный от всей таблицы.

Итак, как помнится, мы решили с вами создать вот такую таблицу:

1	2	3	
1x1	1x2	1x3	1
2x1	2x2	2x3	2

А получилось у нас еще только вот такая таблица:

1x1	1x2	1x3
2x1	2x2	2x3

В чем же дело? А в том, что мы не задали высоту и ширину ячейкам таблицы). Вспомним об атрибутах `height` и `width` - вы можете их задать для всей таблицы, для одного ряда, для ячейки (столбца). Высота и ширина могут задаваться как в пикселях, так и процентах. В нашем случае мы зададим ширину и высоту в пикселях для столбцов (ячеек).

В нашем примере высота равняется 35, а ширина – 50. Высоту задаем только для первой ячейки в каждой строке.

Предвижу вопрос: - А почему высота задана только для двух ячеек (столбцов), а ширина для всех? Все очень просто: если в ряду вы задаете для какой-либо ячейки высоту большую, чем для других то, не смотря на это, все ячейки (столбцы) вашего ряда станут по высоте равны наибольшей. Тоже самое и с рядами, если вы зададите для какого-то ряда наибольшую длину, то все остальные ряды выровняются по этому наибольшему ряду (помните ряд - это не ячейка, поэтому в нашем примере ширина задана для каждой ячейки (столбца)).

Вы можете задать высоту и ширину для всей таблицы, тогда все ячейки (столбцы) и ряды поделят данное им пространство поровну, если не задавать им это пространство персонально (в процентах от общей ширины (высоты) таблицы или пикселях).

Кстати, атрибуты `height` и `width` можно задать в процентах: `<td width="30%">` содержимое ячейки `</td>`.

Если мы задали для ячейки ширину в 30 процентов, как в нашем примере, то это значит, что на остальные ячейки у нас осталось еще 70 процентов, т.е. сумма их ширин не должна превышать 70 процентов. Почему? Простая арифметика: в сумме это все должно равняться 100%. Считаем: $100 - 30 = 70$.

Вернемся к нашему примеру: теперь мы почти получили то, что хотели:

1x1	1x2	1x3
2x1	2x2	2x3

Теперь нам осталось лишь выровнять содержимое (текст) внутри таблицы. Выравниваем содержимое с помощью тега-контейнера `<center>`, как показано в приведенном ниже примере:

```
<td height="35" width="50" bgcolor="#FFCC33"> <center> 1x1 </center>
</td>
```

И... готово!

1x1	1x2	1x3
2x1	2x2	2x3

Поскольку содержимое каждой ячейки как бы обстановка отдельной комнаты, то и тэги для центрирования текста нам нужно было прописать в нашем

примере для содержимого (текста) каждой ячейки, что мы с вами и сделали.

Вы должны помнить и другие тэги для выравнивания содержимого, кроме `<center>`, а также атрибут `align`, который мы прописывали для них. Этот атрибут можно задать и для ячеек таблицы: `<td align="center">2x2</td>`.

Соответственно, атрибут `align` может принимать еще значения `right` и `left`. В нашем случае атрибут `align` выравнивает содержимое внутри ячейки (на всякий случай, если вы еще не поняли его назначение).

В каждой ячейке могут находиться и картинки, и текст (+тэги и атрибуты применяемые к ним), и даже таблицы (в этом случае они называются - вложенные таблицы). Т.е. тэги, которые мы применяем для форматирования содержимого (контента) - все те же.

Давайте шагнем на следующую ступеньку и продолжим разговор о таблицах.

2. Откорректируем таблицу:

Вернемся к нашему примеру, и поговорим о вертикальном выравнивании содержимого таблицы, т.е. о том, как можно сделать так, чтобы содержимое ячейки не только располагалось ровно посередине ее (как по умолчанию), а еще вверху или внизу. Вертикальное выравнивание задается следующим атрибутом - `valign="middle"` (`top`, `bottom`) - содержимое конкретной ячейки будет находится в середине ячейки (наверху или внизу). Откорректировать таблицу, как указано на рисунке ниже:

1x1	1x2	1x3
2x1	2x2	2x3

А теперь, когда мы знаем, что `align`, задает горизонтальное выравнивание содержимого ячейки, а `valign` – вертикальное, перейдем к следующей ступеньке, чтобы узнать о других атрибутах.

Задание выполняем в новом документе

Разберем атрибуты `colspan` и `rowspan`. `Colspan` - определяет количество столбцов, на которые простирается данная ячейка, а `rowspan` - количество рядов (эти параметры могут принимать значение от 2 и больше, т.е. наша ячейка может растягиваться на два и более столбца (ряда)). Теперь, чтобы было все понятно, обратимся к примерам.

1x1	1x2	
2x1	2x2	2x3

Итак, наша задача растянуть ячейку `1x1` на два столбца (ячейки). Для этого мы используем атрибут `colspan="2"`, прописав его для ячейки `1x1`. Код будет выглядеть следующим образом:

```
<td height="35" bgcolor="#FFCC33" colspan="2"> <center>1x1</center> </td>
```

Как вы видите, ячейка `1x1` простирается на длину двух ячеек. Соответственно, ее длина равняется суммированной длине этих двух ячеек (100 пикселей). Мы не прописали атрибут `width` для ячейки `1x1`, но если бы мы решили это сделать, то `width` мы задали бы равным ста пикселям. И еще, прошу вас

обратить внимание, что в нашем примере нет ячейки 1x3, т.е. в первом ряду всего лишь две ячейки, почему – мы уже это обсудили – ячейка 1x1 равна двум ячейкам, благодаря атрибуту colspan.

Если бы мы по оплошности прописали ячейку 1x3, тогда у нас получилась бы следующее:

1x1	1x2	1x3
2x1	2x2	2x3

Аккуратнее обращайтесь с параметрами colspan и rowspan. Ошибка может привести к тому, что ваш сайт «поедет». Теперь, когда мы разобрались с параметром colspan, разберемся с параметром rowspan. Принцип действия тут тот же:

1x1	1x2	1x3
2x1	2x2	

1

Попробуйте сами написать код для такой таблицы (подсказываю, тут у нас должна исчезнуть ячейка 2x3). А для полного закрепления полученных знаний написать еще и код для такой таблицы:

1x1	1x2	
2x1	2x2	1x2

2

Таблицы **1** и **2** должны быть на одной странице

Итак, мы остановились на такой таблице, редактируем только её:

1x1	1x2	
2x1	2x2	1x2

Сейчас можно избавиться от пространства между ячейками таблицы:

1x1	1x2	
2x1	2x2	1x2

Такая красота достигается с помощью атрибута cellspacing, равного нулю.

Можно наоборот увеличить пространство между ячейками, допустим, пусть cellspacing="5", тогда получим следующее:

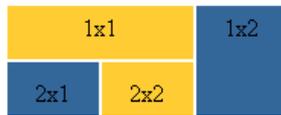
1x1	1x2	
2x1	2x2	1x2

Как видите, и то, и другое мы можем использовать как дизайнерский прием, т.к. даже уже на наших примерах это смотрится не стандартно.

Обычно атрибут cellspacing, рассматривается в руководствах и учебниках в паре с атрибутом cellpadding, который задает верхнее, нижнее, правое и левое поля ячейки. Чтобы было видно нагляднее, я для начала прижму текст ячеек первого ряда к верху, и нижнего - к низу, используя атрибут valign:

1x1	1x2	
2x1	2x2	1x2

Теперь зададим атрибут cellpadding="5":



В этих примерах мы отчетливо видим верхнее и нижнее поля (левое и правое тоже присутствуют, но на данном примере это не очевидно). Теперь вам должно быть понятно, что поля отделяют содержание ячейки от ее краев. Убрать поля можно, задав атрибут `cellpadding` равный нулю (по умолчанию небольшие поля все-таки у таблиц есть, как и пространство между ячеек).

3. Вложенные таблицы.

Это обычные таблицы, которые располагаются в ячейках другой таблицы, и больше ничего особенного в них нет... почти.

Допустим, у нас уже есть большая таблица, две колонки которой забиты текстом какого-то отчета, а третья между ними для красоты:

<p>С каждым днем в интернете появляется все больше бяк и бук. Это особенные существа, роль которых в развитии современного общества не понятна, но тем не менее само их присутствие заметно. Буки и бяки требуют особого обращения к себе, если обращаться к ним как к нормальным человеческим особям, то вы поняты не будете.</p>		<p>Вот таблица, которая показывает сколько бук, бяк и других обитает в интернете:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>буки</td> <td>65% населения</td> </tr> <tr> <td>бяки</td> <td>20% населения</td> </tr> <tr> <td>др.</td> <td>15% населения</td> </tr> </table> <p>За ним все</p>	буки	65% населения	бяки	20% населения	др.	15% населения
буки	65% населения							
бяки	20% населения							
др.	15% населения							

Задание выполняем в новом документе

Итак, перед нами таблица из одного ряда, с тремя ячейками (столбцами), в третьей ячейке нельзя не заметить вложенную таблицу (с буками и бяками). Теперь код:

```
<table border="2">
<tr>
<td width="200" valign="top" background="blue.gif" align="center">С каждым днем в интернете появляется все больше бяк и бук. Это особенные существа, роль которых в развитии современного общества не понятна, но тем не менее само их присутствие заметно. Буки и бяки требуют особого обращения к себе, если обращаться к ним как к нормальным человеческим особям, то вы поняты не будете.</td>
<td width="10" background="white.gif">&nbsp;</td>
<td width="200" valign="top" background="blue.gif" align="center">Вот таблица, которая показывает сколько бук, бяк и других обитает в интернете:
<br><br>
<table border="2" cellspacing="3">
<tr>
<td width="50" background="white.gif">буки</td>
<td background="white.gif">65% населения</td>
</tr>
<tr>
<td width="50" background="white.gif">бяки</td>
<td background="white.gif">20% населения</td>
</tr>
<tr>
<td width="50" background="white.gif">др.</td>
<td background="white.gif">15% населения</td>
</tr>
</table>
<br><br>
За ним все</td>
</tr>
</table>
```

Увы, код получился объемный, но зато в нем нет ничего сложного:

вложенная таблица от обычной ничем не отличается. Благодаря вложенным таблицам гораздо проще управлять содержимым сайта.

Теперь поговорим, о некоторых нюансах, на которые следует обратить внимание: `background="картинка.gif"` - раньше мы говорили об атрибуте `bgcolor`, который задает цвет фона для таблицы (ряда, ячейки), а атрибут `background` задает фоновую картинку для таблицы (ряда, ячейки).

А теперь сам нюанс: вроде было бы логичнее прописать просто белый фон для вложенной таблицы (`bgcolor="#ffffff"`), а не мучиться, создавая просто белую картинку и делая ее фоном (`background="white.gif"`), ведь так оно быстрее. Да, быстрее, но дело в том, что если IE (Internet Explorer) отображает атрибут `bgcolor` для вложенных таблиц, то некоторые другие браузеры отказываются этот атрибут для вложенных таблиц отображать, поэтому приходится идти обходным путем, используя `background`.

Вот так мы и убили двух зайцев сразу: узнали, что можно задать атрибут `background` для таблиц, и при этом ознакомились с некоторыми тонкостями. Теперь будем добивать оставшихся заек: `<td width="10" background="white.gif"> </td>`.

Давайте обратимся к коду нашей таблицы и вспомним, что эту ячейку мы ввели для красоты. Чтобы ячейка не пустовала, в нее введен ` ` (символ неразрывного пробела, это спецсимвол). Это не просто так. Дело в том, что есть любители использовать конструкцию типа: `<td></td>`. Вот такую конструкцию некоторые браузеры просто-напросто игнорируют, т.к. не любят пустых ячеек, поэтому для корректного отображения вашей таблицы везде вставляйте в пустые ячейки или ` `, или мелкую картинку размеров 1x1 пиксела.

Задание выполняем в новом документе

В следующей главе мы сделаем с вами несложный сайт с таблицами, используя полученные навыки, как показали несколько лет жизни учебника, необходимость в такой главе есть, чтобы вся информация уложилась в вашей голове окончательно.

Итак, возьмем за основу один, из рассмотренных нами примеров:

1x1	1x2
2x1	2x2

```
<table cellpadding="5">
<tr>
<td height="35" bgcolor="#FFCC33" colspan="2"> <center>1x1</center> </td>
<td width="50" bgcolor="#336699" rowspan="2"> <center>1x2</center> </td>
</tr>
<tr>
<td height="35" width="50" bgcolor="#336699"> <center>2x1</center> </td>
<td width="50" bgcolor="#FFCC33"> <center>2x2</center> </td>
</tr>
</table>
```

Давайте, попытаемся на основе его сделать полноценную страницу. Первое, что мы сделаем, введем обязательные тэги и атрибуты для них:

```

<html>
<head>
<title>Мой первый сайт с таблицами</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" link="#000000" alink="#000000"
vlink="#000000">
-----
</body>
</html>

```

Если вы посмотрите, что у нас получилось, то увидите, что таблица маленькая по размеру, а значит, что мы должны изменить значения атрибутов, отвечающих за высоту и ширину таблицы. Убираем все старые значения ширин и высот, и прописываем новые:

```

<table cellpadding="5" width="700">
<tr>
<td width="400" bgcolor="#FFCC33" colspan="2"> <center>1x1</center> </td>
<td width="300" bgcolor="#336699" rowspan="2"> <center>1x2</center> </td>
</tr>
<tr>
<td bgcolor="#336699"> <center>2x1</center> </td>
<td bgcolor="#FFCC33"> <center>2x2</center> </td>
</tr>
</table>

```

Мы задали для всей таблицы ширину 700 пикселей. Соответственно, общая сумма ширин ячеек не может превышать 700 пикселей: 300+400=700. Мы задали ширину только двум ячейкам, т.к. оставшиеся ячейки сами распределят выделенные под них 400 пикселей поровну. Мы не задали высоту (height), т.к. нельзя задать стопроцентную высоту (во весь экран) для таблиц, это не предусмотрено спецификацией, так что, наша таблица сама растянется, когда мы введем в нее какое-либо содержание (текст, картинки).

```

<center>
<table cellpadding="5" width="700">
<tr>
<td width="400" bgcolor="#ffcc33" colspan="2">
<div align="left">
Я совсем недавно начал(а) знакомиться с виртуальной жизнью, но мне по давней традиции тоже захотелось создать свою домашнюю страничку для моих новых виртуальных друзей и знакомых, чтобы они могли посмотреть мои фотографии, почитать обо мне, черкнуть пару строчек в мою гостевую книгу.
<br><br>А может и просто случайный посетитель вдруг захочет познакомиться со мной, и у меня появится<b>еще один виртуальный друг?</b><br><br>На фотографии изображен(а) я. Качество картинки не очень хорошее, к сожалению, поэтому она не четкая и разглядеть черты моего лица немного проблематично. Но в целом заметно, что я вполне ничего.<br><br>Если ты так тоже думаешь, то давай как-нибудь встретимся, поболтаем, чайку попьем в кафешке? Кто знает, может быть, мы и в реальной жизни станем друзьями.
</div>
</td>
<td width="300" bgcolor="#336699" rowspan="2"><center>1x2</center></td>
</tr>
<tr>
<td bgcolor="#336699"><center>2x1</center></td>
<td bgcolor="#ffcc33"><center>2x2</center></td>
</tr>
</table>
</center>

```

Итак, мы ввели в таблицу текст с виртуальным другом. Мы выровняли текст по левому краю при помощи тэга DIV. Саму таблицу мы расположили по центру экрана при помощи тэга CENTER.

Если вы обратите внимание, то заметите, что текст слишком близко расположен к краям нашей ячейки и это смотрится не красиво. Что мы можем сделать? Конечно, ввести атрибут cellpadding для таблицы.

```

<table cellspacing="5" width="700" cellpadding="20">
<tr>
<td width="400" bgcolor="#ffcc33" colspan="2" valign="top">
<div align="left">
<h3>Добро пожаловать на мою страницу!</h3>
Я совсем недавно начал(а) знакомиться с виртуальной жизнью, но мне по давней традиции
тоже захотелось создать свою домашнюю страничку
для моих новых виртуальных друзей и знакомых, чтобы они могли посмотреть мои фотографии,
почитать обо мне, черкнуть пару строчек в мою
гостевую книгу.
<br><br>А может и просто случайный посетитель вдруг захочет познакомиться со мной, и у
меня появится<b>еще один виртуальный друг?
</b><br><br>На фотографии изображен(а) я. Качество картинки не очень хорошее, к
сожалению, поэтому она не четкая и разглядеть черты
моего лица немного проблематично. Но в целом заметно, что я вполне ничего.<br><br>Если
ты так тоже думаешь, то давай как-нибудь
встретимся, поболтаем, чайку попьем в кафешке? Кто знает, может быть, мы и в реальной
жизни станем друзьями.
</div>
</td>
<td width="300" bgcolor="#336699" rowspan="2" valign="top">
<div align="center">

<br><br>

</div>

```

Кроме атрибута cellpadding мы ввели заголовок, чтобы ваш посетитель смог понять, куда он попал, а также мы ввели содержимое для второй колонки. Также, введя атрибут valign, мы расположили текст и картинки сверху наших ячеек (как мы помним, по умолчанию, если атрибут valign не вводить, содержимое располагается по центру ячейки).

Давайте введем содержание для оставшихся двух ячеек:

```

<tr>
<td bgcolor="#336699">
<div align="center"><a href="mailto:admin@postroika.ru">напиши мне письмо</a></div>
</td>
<td bgcolor="#ffcc33">
<div align="center"><a href="#">посмотри мои фотографии</a></div>
</td>
</tr>
</table>
</center>
</body>
</html>

```

Итак, в последние две ячейки мы ввели текст, который сделали ссылками на наш почтовый ящик, чтобы наш посетитель мог написать нам письмо, и на страницу с нашими фотографиями. Естественно, предварительно нужно создать документ с фотографиями, прежде, чем на него ссылаться.

 ... - заметьте, что мы не прописали путь к документу с фотографиями, и вместо него подставили значок решетки. Как вы помните, значок решетки используется при создании закладок внутри страницы. Однако, в нашем случае имя закладки не указано, и самой закладки в нашей странице нет. А это значит, что при нажатии на такую странную ссылку, которую мы прописали, ваш посетитель никуда не перейдет с вашей страницы. Т.е. это пустая ссылка.

Должно получиться 4 документа

Лабораторная работа №3. Создание гиперссылок. Вставка графических изображений

Вставка изображения

Для вставки изображения в html-документ используется непарный тег ``, который помещается в то место документа, где должно быть вставлено изображение.

Например, чтобы поместить на страничку изображение в нужном месте документа помещается строчка: ``.

Здесь мы указали полный путь (или абсолютный адрес) до изображения. А можно указать относительный адрес изображения: ``.

Размеры изображения

Размер каждого изображения задается двумя параметрами: ширина и высота. У тега `` есть соответствующие параметры: `width` и `height`. Эти параметры принимают значения в пикселах (px). Вы можете задать истинные размеры изображения: ``

Например: чтобы увеличить изображение в 1,5 раза, можно написать: `` или ``.

Например, если вы хотите, чтобы изображение по ширине занимало ровно половину странички, надо написать следующее: ``

Рамка вокруг изображения

У тега `` есть еще один необязательный параметр `border`. С его помощью можно задать толщину рамки вокруг изображения. По умолчанию толщина рамки изображения нулевая, т.е. рамки нет.

Например, так можно добавить к нашей картинке рамку толщиной 2 пиксела: ``

Вот что покажет нам браузер:



Выравнивание изображения

Для изображения, как и для параграфа, существует понятие выравнивания относительно текста и других изображений на странице. Задать тип выравнивания можно также с помощью параметра `align` тега ``.

Отступы вокруг изображения

Чтобы при обтекании текст не прилегал близко к изображению, можно задать отступы вокруг изображения. Сделать это можно с помощью параметров `hspace` (отступы слева и справа) и `vspace` (отступы сверху и снизу) тега ``

Выравнивание изображения

Значение	Описание	Результат
<code></code>	Верхняя граница изображения выравнивается по самому высокому текстовому элементу текущей строки.	Хотя Карлсон 😄 и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.
<code></code>	Верхняя граница изображения выравнивается по самому высокому элементу текущей строки.	Хотя Карлсон 😄 и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.
<code></code>	Выравнивание середины изображения по базовой линии текущей строки.	Хотя Карлсон 😄 и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.
<code></code>	Выравнивание середины изображения по середине текущей строки.	Хотя Карлсон 😄 и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.
<code></code>	Выравнивание изображения по базовой линии текущей строки.	Хотя Карлсон 😄 и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.
<code></code>	Выравнивание нижней границы изображения по окружающему тексту.	Хотя Карлсон 😄 и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.
<code></code>	Выравнивает изображение по левому краю окна.	😄 Хотя Карлсон и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.
<code></code>	Выравнивает изображение по правому краю окна.	Хотя Карлсон и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро. 😄

Ссылки

Для создания ссылки используется парный тег `<a>`. Чтобы сделать текст ссылкой, нужно поместить его между тегами `<a>` и `` и указать, куда ведет эта ссылка, т.е. адрес документа, где мы окажемся после нажатия на ссылку. Для этого нужно параметру `href` тега `a` присвоить адрес документа. Адрес документа, на который мы ссылаемся, называется адрес ссылки.

Вот, например: `Текст ссылки`

Адрес ссылки может быть абсолютным или относительным. В приведенном выше примере мы использовали абсолютный адрес, т.е. полный путь до документа `1.html`. Относительный адрес - путь к документу, составленный относительно текущего документа. Приведем примеры использования относительных адресов.

Пример 1: `Пример 1`

Такая строчка означает, что при нажатии на фразу Пример 1 мы перейдем на страничку `1.html`, которая лежит в том же каталоге, что и текущий документ.

Пример 2: `Пример 2` или `Пример 2`

Обе эти строчки означают, что в каталоге, где лежит текущий документ, есть подкаталог `about`, в котором лежит страничка `1.html`, на которую мы перейдем при нажатии на фразу Пример 2.

Пример 3: `Пример 3`

Эта строчка означает, что из каталога, где лежит текущий документ, нужно подняться на один уровень выше, и в каталоге, где мы оказались, лежит страничка `1.html`, на которую мы перейдем при нажатии на фразу Пример 3.

Пример 4: `Пример 4`

Эта строчка означает, что из каталога, где лежит текущий документ, нужно подняться на один уровень выше, и в каталоге, где мы оказались, есть подкаталог `about`, в котором лежит страничка `1.html`, на которую мы перейдем при нажатии на фразу Пример 4.

Пример 5: `Пример 5`

А если в адресе ссылки после папки не указать имя документа, то такая ссылка будет называться неполной. Она дает команду браузеру при нажатии на фразу Пример 5 открыть файл `index.html`, лежащий в каталоге `about`.

Если файл `index.html` отсутствует, браузер отобразит список файлов папки `about`, если в настройках вашего сервера есть разрешение. Иначе браузер поступит так же, как если бы вы сделали ссылку на любой другой несуществующий документ, т.е. покажет сообщение "Невозможно отобразить страницу".

Комбинируя эти примеры, вы сможете создавать сайты со структурой различной степени разветвленности.

В целях безопасности предпочтительнее указывать относительные адреса ссылок, кроме тех случаев, когда вы на своей страничке ссылаетесь на документы, расположенные на других сайтах.

Как сделать изображение ссылкой

Вместо текста между тегами <a> и вы можете вставить изображение, и изображение будет ссылкой:

Изображения, что у изображения, являющегося ссылкой, автоматически появляется рамка толщиной 1 пиксел такого же цвета, как текстовые ссылки, и знаете, как ее убрать в случае необходимости.

Цвет ссылок

Цвет ссылок можно задать с помощью параметров link, alink, vlink тега body. Свойства страницы - параметры тега body. Таким способом задается цвет всех ссылок на странице.

Если же вы хотите изменить цвет какой-то конкретной ссылки, вы можете сделать также, как и для обычного текста, т.е. воспользоваться тегом с параметром color. Только контейнер должен находиться внутри контейнера <a>: Текст ссылки

Ссылки внутри страницы

Бывают еще и внутренние ссылки, т.е. ссылки на какое-то место внутри текущего документа. Внутренние ссылки удобно использовать в больших документах для разбиения на главы. Итак, чтобы создать внутреннюю ссылку, надо выполнить 2 действия:

1. Сделать закладку в том месте документа, куда вы будете ссылаться, и дать этой закладке имя. Для этого в нужном вам месте вставьте контейнер <a> и его параметру name присвойте строковое значение:

Между тегами <a> и текст можно не писать, т.к. Вам нужно всего лишь пометить место в документе, куда ведет ссылка.

2. Ну и собственно сделать ссылку, которая будет ссылаться на эту закладку. Для этого вы делаете обычную ссылку, только вместо адреса документа пишете знак # и имя закладки: Ссылка на закладку

Вот пример использования внутренних ссылок:

```
<ol>
<li><a href="#chapter1">Раздел 1</a></li>
<li><a href="#chapter2">Раздел 2</a></li>
</ol>
<a name="chapter1"></a>
<h3>Раздел 1</h3>
<p>В городе Стокгольме, на самой обыкновенной улице, в самом обыкновенном доме живет самая обыкновенная шведская семья по фамилии Свантесон. Семья эта состоит из самого обыкновенного папы, самой обыкновенной мамы и трех самых обыкновенных ребят - Боссе, Бетан и Малыша.</p>
<a name="chapter2"></a>
<h3>Раздел 2</h3>
```

<p>Хотя Карлсон и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.</p>

А еще можно делать ссылку на закладку в другом документе. Например, в документе 1.html есть закладка с именем qqq. Чтобы сделать на нее ссылку из текущего документа, надо сделать обычную ссылку на документ 1.html и добавить к адресу знак # и имя закладки qqq: Ссылка на закладку

Ссылка на новое окно

Все ссылки, заданные выше описанными способами, по умолчанию открываются в текущем окне браузера. Для изменения этого свойства ссылки у тега <a> существует параметр target.

Если вы хотите, чтобы документ, на который вы ссылаетесь, открывался в новом окне браузера, присвойте параметру target значение _blank: Ссылка

Если вы хотите конкретно указать, что ссылка должна открыться в текущем окне, присвойте параметру target значение _self: Ссылка

Ссылки во фреймах

А теперь предположим, что вы используете на своей страничке фреймы. Тогда по умолчанию все ссылки будут открываться в текущем фрейме.

Присвоив параметру target значение _parent, вы укажете браузеру открывать ссылку во фрейме-родителе: Ссылка

Присвоив параметру target значение _top, вы укажете браузеру отменить все фреймы и открыть ссылку в полном окне браузера: Ссылка

А если вы хотите, чтобы ссылка открывалась в каком-то конкретном фрейме, присвойте параметру target имя нужного фрейма: Ссылка

Ссылка на адрес электронной почты

Ссылка на адрес электронной почты делается как и обычная ссылка, только вместо адреса документа пишется mailto:адрес электронной почты: Пишите письма

При нажатии на такую ссылку запустится почтовая программа, установленная по умолчанию, и откроется окно создания нового письма. В поле Кому уже будет стоять адрес, на который ведет ссылка.

Если же вы к адресу электронной почты присоедините ?subject=какая-то тема сообщения, то при нажатии на эту ссылку в окне нового сообщения в поле Тема будет стоять значение параметра subject: Пишите письма

Обратите внимание, что значение параметра subject не заключается в

кавычки. В кавычки заключается целиком адрес ссылки.

Задание

Создайте 2 html-документа. Вставьте в первый документ какой-нибудь текст, разбейте его на разделы. Создайте в начале страницы содержание со ссылками на соответствующие разделы. В конце документа вставьте ссылку на второй документ, которая будет открывается в текущем окне. Во втором документе сделайте ссылку на свой электронный адрес.

Должно получиться 1 документ

Лабораторная работа №4. Создание документа с фреймами. Создание документа с формами

Создание форм

В языке HTML формы применяются для отправки данных со странички на обработку в какую-либо программу, установленную на сервере.

Пользователь заполняет форму и нажимает кнопку Send (отправить), данные обрабатываются специальной программой (скриптом) на сервере, и мы получаем на адрес техподдержки вопросы пользователей.

Создание форм начинается с элемента `<form> </form>`.

У элемента формы `form` существуют два обязательных атрибута `method` и `action`.

`method` - атрибут определяет способ, с помощью которого будет осуществляться связь между формой и Web-сервером. Атрибут может принимать значение "get" и "post".

`action` - значением атрибута `action` является конкретный адрес скрипта, обрабатывающего данные. Атрибуты записываются в тег `<form>`, например так: `<form method="get" action="http://www.myserver.com/cgi-bin/maillscript">`

В данном уроке рассматривается только создание элементов форм, их виды. Для того, чтобы форма заработала, ее необходимо запрограммировать. Само программирование не входит в рамки данного урока.

Текстовое поле

Для создания поля ввода используют элемент `input` с атрибутом `type`. Type может принимать различные значения. Для самого простого поля ввода текста атрибут `type` будет иметь значение `text`. `<input type=text параметры>`.

Например:

```
<form method="get" action="/cgi-bin/handler.cgi">
```

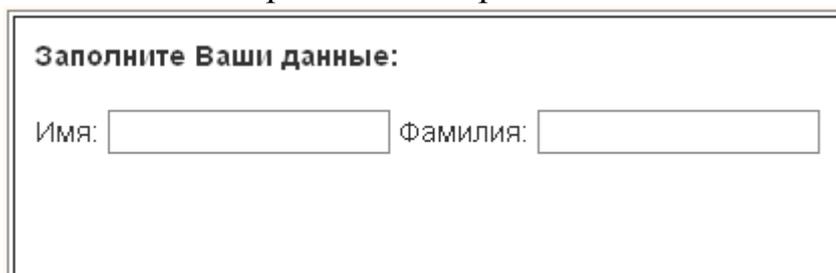
```
<b>Заполните Ваши данные:</b><br><br>
```

```
Имя: <input type=text maxlength=25 size=20>
```

```
Фамилия:<input type=text maxlength=25 size=20>
```

```
</form>
```

На страничке отобразится так:



The image shows a rectangular box representing a web form. At the top left of the box, the text "Заполните Ваши данные:" is displayed in a bold font. Below this text, there are two text input fields. The first field is preceded by the label "Имя:" and the second by "Фамилия:". Both input fields are empty and have a standard rectangular border.

Возможно задавать значения параметров:

Параметр	Описание
size=...	Ширина поля.
maxlength=...	Максимальное количество символов разрешенных в тексте. Если этот параметр опустить, то можно вводить строку длинее самого поля.
name=...	Имя поля. Предназначено для того, чтобы обработчик формы мог идентифицировать это поле.
value=...	Начальный текст содержащий в поле.
align=...	Выравнивание поля. Возможные значения: top, middle, bottom, right, left, texttop, baseline, absmiddle, absbottom. Результат их использования точно такой же, как и у изображений.

Поле для пароля

Поле для пароля — обычное текстовое поле, но отличается тем, что все символы показываются звездочками, чтобы никто не подглядел вводимый пароль. Задается кодом: `<input type=password параметры>`.

Например:

```
<form method="get" action="/cgi-bin/handler.cgi">
<b>Логин: </b><input type=text maxlength=25 size=20 name="text"><br>
<b>Пароль:</b> <input type=password maxlength=15 size=20 name="pass">
</form>
```

На страничке будет выглядеть так:

Многострочный текст

Для возможности введения в поле несколько строк текста используем тег `textarea`.

Например:

```
<form method="get" action="/cgi-bin/handler.cgi">
<b>Введите ваш отзыв: </b><br>
<textarea rows=10 cols=45></textarea>
</form>
```

На страничке отобразится так:

Для тега `textarea` используются следующие параметры:

Параметр	Описание
<code>name=...</code>	Имя поля. Предназначено для того, чтобы обработчик формы мог идентифицировать это поле.
<code>cols=...</code>	Количество столбцов текста.
<code>rows=...</code>	Число строк текста.
<code>value=...</code>	Значение поля определяет, что будет отправлено на сервер.
<code>wrap=...</code>	Параметры переноса строк. Возможные значения: <code>off</code> — отключает перенос строк; <code>virtuals</code> — показывает переносы строк, но отправляет текст как он введен; <code>physical</code> — переносы строк вставляются где указано и в таком виде текст отправляется.

Кнопки

Кнопка — это элемент интерфейса, на который нужно нажимать. Используем `type=button`, "button" переводится как "кнопка". `<input type=button параметры>`.

Например:

```
<form method="get" action="/cgi-bin/handler.cgi">
<p align=center><input type=button name=press value=" Нажми меня :-)">
</form>
```

На страничке отобразится следующим образом:



Для кнопок используются следующие параметры:

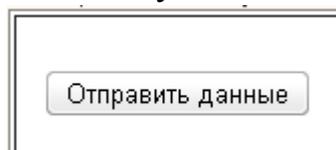
Параметр	Описание
<code>name=...</code>	Имя поля. Предназначено для того, чтобы обработчик формы мог идентифицировать это поле.
<code>value=...</code>	Надпись на кнопке, а также ее значение.
<code>align=...</code>	Выравнивание поля. Возможные значения: <code>top</code> , <code>middle</code> , <code>bottom</code> , <code>right</code> , <code>left</code> , <code>texttop</code> , <code>baseline</code> , <code>absmiddle</code> , <code>absbottom</code> . Результат их использования точно такой же, как и у изображений.

Кнопка SUBMIT. При нажатии на кнопку SUBMIT данные формы отправляются на сервер и обрабатываются программой, указанной параметром `action` тега `form`. Используем `type=submit`.

Например:

```
<form action="/cgi-bin/handler.cgi">
<p><input type=submit value="Отправить данные">
```

На страничке мы увидим следующее:



Кнопка RESET. При нажатии на эту кнопку, данные формы возвращаются

в первоначальное значение. Используем type=reset.

Например:

```
<form action="/cgi-bin/handler.cgi">  
<input type=text width=20> <input type=reset value="Сбросить данные">  
</form>
```

Отображение на страничке:

A screenshot of a web form. It consists of a rectangular border containing a text input field on the left and a button labeled "Сбросить данные" (Reset) on the right.

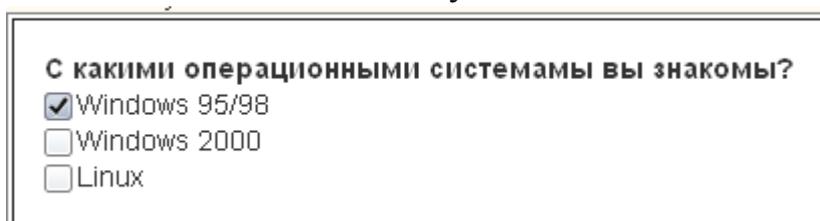
Флажки (checkbox)

Флажки используют, когда необходимо выбрать два или более варианта из предложенных. Используем type=checkbox.

Например:

```
<form action="/cgi-bin/handler.cgi"> <b>С какими операционными системами вы знакомы?</b><br>  
<input type=checkbox name=option1 value=a1 checked>Windows 95/98<br>  
<input type=checkbox name=option2 value=a2>Windows 2000<br>  
<input type=checkbox name=option3 value=a3>Linux  
</form>
```

Вот что получается:

A screenshot of a web form. The title is "С какими операционными системами вы знакомы?". Below the title are three checkboxes with labels: "Windows 95/98" (checked), "Windows 2000", and "Linux".

Переключатели (radiobutton)

Переключатели используют, когда необходимо выбрать один единственный вариант из предложенных. Используем type=radio.

Например:

```
<form action="/cgi-bin/handler.cgi"> <b>Какое время года Вы больше любите? :-)</b><br>  
<input type=radio name=answer value=a1>Весну<br>  
<input type=radio name=answer value=a2>Лето<br>  
<input type=radio name=answer value=a3>Осень<br>  
<input type=radio name=answer value=a4>Зиму  
</form>
```

На страничке можно будет увидеть следующее:

A screenshot of a web form. The title is "Какое время года Вы больше любите? :-)". Below the title are four radio buttons with labels: "Весну", "Лето", "Осень", and "Зиму".

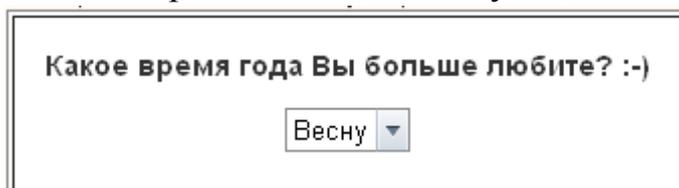
Поле со списком

В ниспадающем списке можно выбирать одно или несколько значений. Сделаем выбор времени года с помощью ниспадающего списка. Поля выбора начинаются тегом `<select>` и заканчиваются `</select>`. Каждое выбираемое поле заключено в теги `<option>` и `</option>`

Например:

```
<form action="/cgi-bin/handler.cgi">
<b>Какое время года Вы больше любите? :-)</b>
<p align=center>
<select name= время года>
<option>Весну</option>
<option>Лето</option>
<option>Осень</option>
<option>Зиму</option>
</select>
</form>
```

На страничке видим следующее:



Какое время года Вы больше любите? :-)

Весну ▼

Для списков используются следующие параметры:

Параметр select	Описание
name=...	Имя поля. Предназначено для того, чтобы обработчик формы мог идентифицировать это поле.
size=...	Количество видимых строк. По умолчанию одна строка.
multiple	Этот параметр позволяет выбрать сразу несколько строк. Выбор делается с помощью клавиши Control или Shift.
Параметр option	Описание
selected	Видимое поле по умолчанию.
value=...	Значение определяет, что будет отправлено на сервер при выбранном пункте списка.

Кнопка с изображением

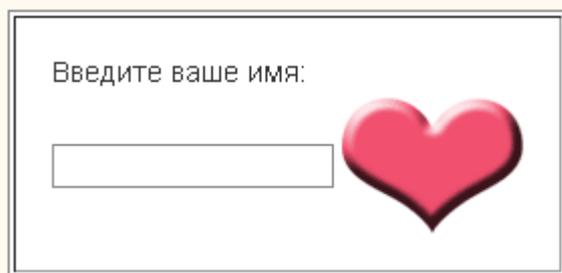
Кнопки с изображениями аналогичны по действию кнопке Submit, но представляют собой рисунок. Используем `type=image`.

Например:

```
<form action="/cgi-bin/handler.cgi">
<table align=center>
```

```
<tr><td colspan=2>Введите ваше имя: <tr>
<td><input type=text width=25></td>
<td><input type=image src=../school/examples3/ser.gif width=90 height=68
border=0></td>
</tr>
</table>
</form>
```

На страничке это будет выглядеть так:

A screenshot of a web form. The form is enclosed in a double-line border. At the top, it says "Введите ваше имя:". Below this text is a rectangular text input field. To the right of the input field is a red heart icon with a slight 3D effect and a shadow.

Картинка задается при помощи тега `src` и адреса картинки `src=../school/examples3/ser.gif`. Таким образом, комбинируя разные способы создания форм, мы можем на одной страничке разместить несколько форм сразу. Для отделения одного вида формы от другого будем использовать тег `<fieldset>`.

Задания

Задание 1: Создать комбинированную форму по предложенному примеру.

Пример комбинированной формы.

Пример использования разных форм на одной страничке.

Заполните, пожалуйста, эту форму вашими данными и нажмите кнопку "Отправить данные"! Вы сможете увидеть вариант обработки ваших данных.

The image shows a web form with three distinct sections, each enclosed in a box. The first section is titled "Заполните Ваши данные:" and contains a text input field labeled "Имя:". The second section is titled "Выберите любимый цвет:" and contains four radio buttons labeled "Желтый", "Голубой", "Розовый", and "Зеленый". The third section is titled "Напишите здесь что-нибудь хорошее :-)" and contains a text area. At the bottom of the form are two buttons: "Сбросить данные" and "Отправить данные".

```
<form method="post" action="8.php">
<fieldset>
<b>Заполните Ваши данные:</b><br><br>
Имя: <input type="text" name="name" maxlength=25 size=20 value=" ">
<br><br>
</fieldset>
<p>
<fieldset>
<b>Выберите любимый цвет:</b><br>
<p align="center"> <input type="radio" name="answer" value="#F4F090">Желтый<br>
<input type="radio" name="answer" value="#80ECF3">Голубой<br>
<input type="radio" name="answer" value="#F3ACC5">Розовый<br>
<input type="radio" name="answer" value="#A7F4AB">Зеленый<br>
</select>
<br><br>
</fieldset>
<p>
<fieldset>
<b>Напишите здесь что-нибудь хорошее :-)</b><br>
<textarea name="basictext" rows=7 cols=45 ></textarea>
</fieldset>
<p>
<fieldset>
<input type="reset" value="Сбросить данные">
<input type="submit" value="Отправить данные">
</fieldset>
</form>
```

Задание 2: Создать форму по примеру, приведенному ниже:

The image shows a web browser window with the title "Ter INPUT". The browser's address bar contains a search engine icon and the text "Google". The main content area of the browser displays a form with the following elements:

- A label "Ваше имя:" followed by a single-line text input field.
- A label "Каким браузером в основном пользуетесь:" followed by three radio button options: "Internet Explorer", "Опера", and "Firefox".
- A label "Комментарий" followed by a multi-line text area.
- Two buttons at the bottom: "Отправить" (Submit) and "Очистить" (Reset).

```

<form name="test" method="post" action="input1.php">
  <p><b>Ваше имя:</b><br>
  <input name="nick" type="text" size="40">
</p>
  <p><b>Каким браузером в основном пользуетесь:</b><br>
  <input type="radio" name="browser" value="ie"> Internet
  Explorer<br>
  <input type="radio" name="browser" value="opera">
  Opera<br>
  <input type="radio" name="browser" value="firefox">
  Firefox<br>
</p>
  <p>Комментарий<br>
  <textarea name="comment" cols="40"
  rows="3"></textarea></p>
  <p><input type="submit" value="Отправить">
  <input type="reset" value="Очистить"></p>
</form>

```

Задание 3: Символьные объекты. Что такое символьные объекты?

Наберите вот такую строчку:

```
<blockquote>#982;#978;#977;#969;#968;#967;#966;#960;#958;#950;#952;#948;#937;
</blockquote>
```

И такую:

```
<blockquote>
<p>#9834;#9835;#9834;#9834;#9835;....
</blockquote>
```

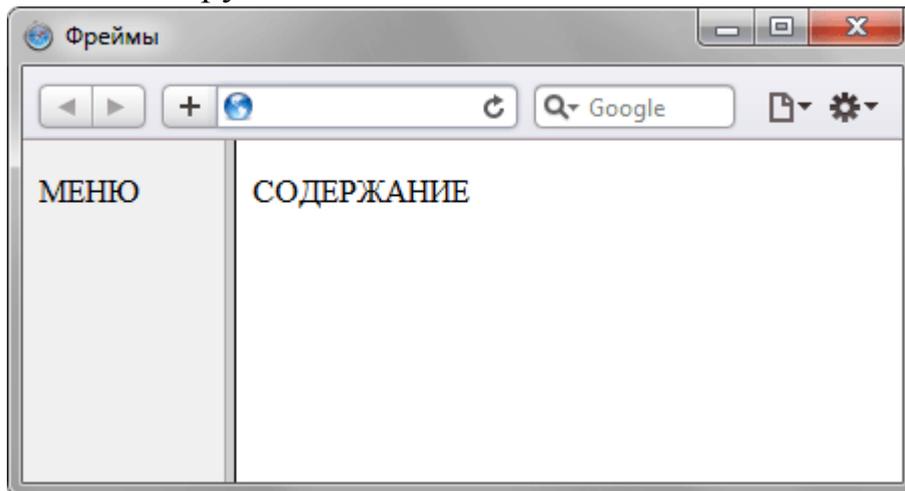
Посмотрите полученный результат. Сделайте выводы.

Должно получиться 3 документа

Лабораторная работа № 4. Создание фреймов

Несмотря на то, что сайты с фреймами встречаются все реже, изучение HTML было бы неполным без рассмотрения темы о фреймах. К тому же фреймы в каком-то смысле заняли свою нишу и применяются для систем администрирования и справки. Там, где недостатки фреймов не имеют особого значения, а преимущества наоборот, активно востребованы.

Для создания фрейма используется тег `<frameset>`, который заменяет тег `<body>` в документе и применяется для разделения экрана на области. Внутри данного тега находятся теги `<frame>`, которые указывают на HTML-документ, предназначенный для загрузки в область.



При использовании фреймов необходимо как минимум три HTML-файла: первый определяет фреймовую структуру и делит окно браузера на две части, а оставшиеся два документа загружаются в заданные окна. Количество фреймов не обязательно равно двум, может быть и больше, но никак не меньше двух, иначе вообще теряется смысл применения фреймов.

Рассмотрим этапы создания фреймов на основе страницы, продемонстрированной на рисунке. Нам понадобится три файла: `index.html` — определяет структуру документа, `menu.html` — загружается в левый фрейм и `content.html` — загружается в правый фрейм. Из них только `index.html` отличается по структуре своего кода от других файлов.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Фреймы</title>
</head>
<frameset cols="100,+">
  <frame src="menu.html" name="MENU">
  <frame src="content.html" name="CONTENT">
</frameset>
</html>
```

Формат документа, использующего фреймы, внешне очень напоминает формат обычного документа, только вместо тэга BODY используется контейнер FRAMESET, содержащий описание внутренних HTML-документов, содержащий собственно информацию, размещаемую во фреймах.

`<html>`

```
<head>...</head>
<frameset>...</frameset>
</html>
```

Создание фрейм-документа является специфичным видом HTML-документа, поскольку не содержит элемента BODY и какой-либо информационной нагрузки соответственно. Он описывает только фреймы, которые будут содержать информацию (кроме случая двойного документа, который мы рассмотрим позже).

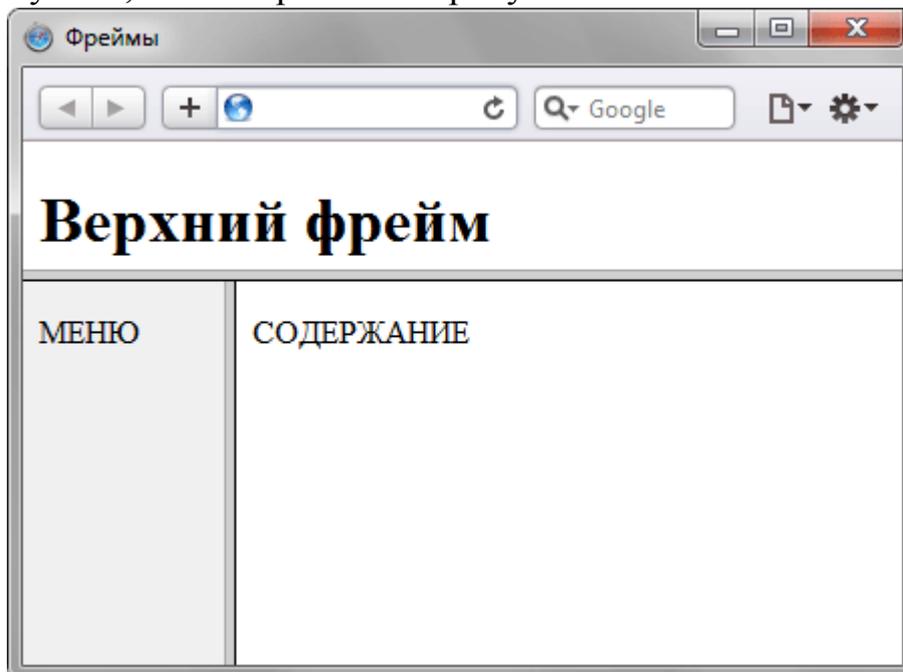
Представим общий синтаксис фреймов:

```
<frameset cols="value" | rows="value">
<frame src="url1">
<frame ...>
...
</frameset>
```

Общий контейнер FRAMESET описывает все фреймы, на которые делится экран. Вы можете разделить экран на несколько вертикальных или несколько горизонтальных фреймов. Тэг FRAME описывает каждый фрейм в отдельности. Рассмотрим более детально каждый компонент.

Задание

1. Создать файл меню.
2. Создать файл содержания.
3. Создать файл с фреймами.
4. Создать документ, как изображено на рисунке ниже:

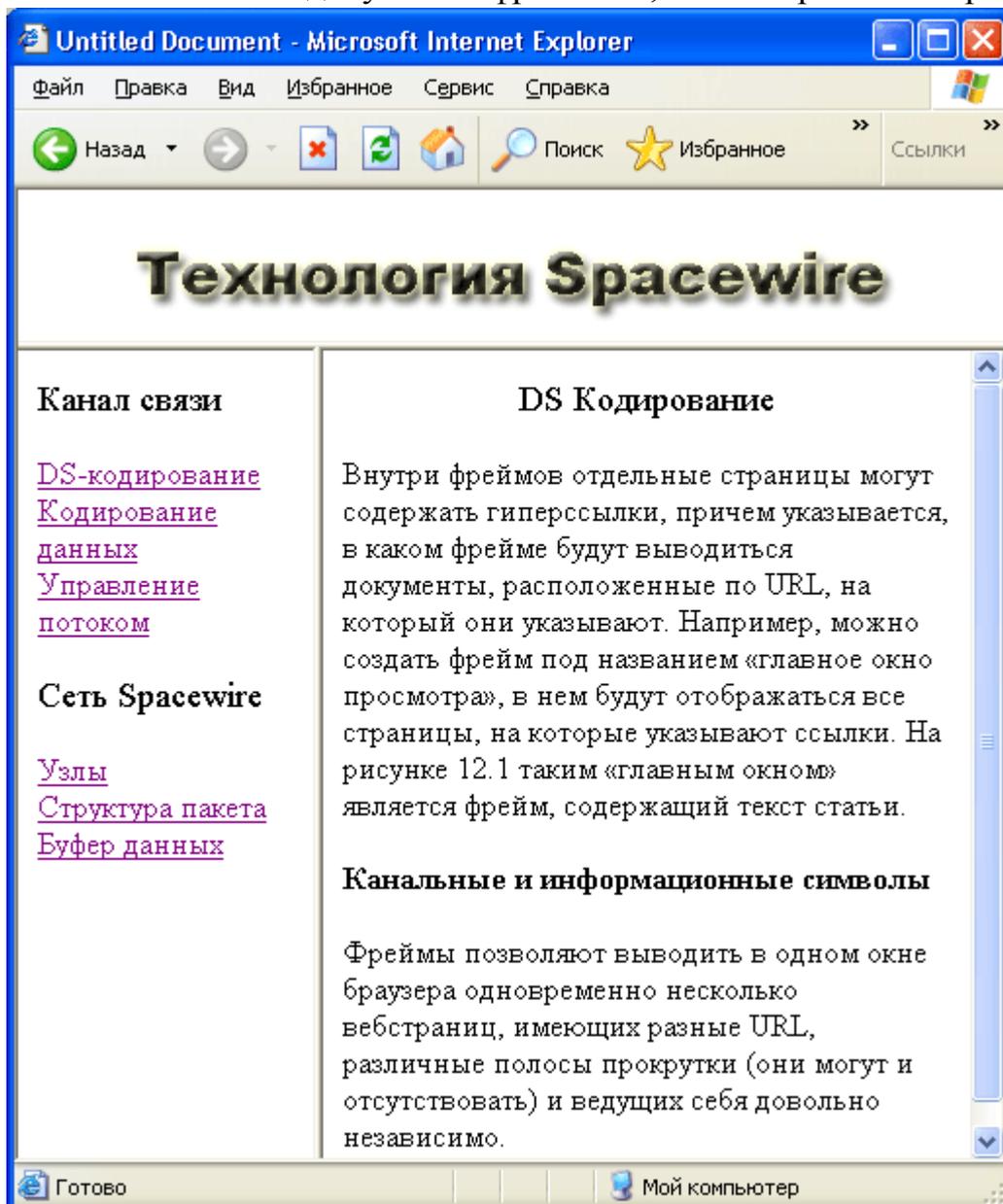


```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Фреймы</title>
</head>
<frameset rows="25%,75%">
  <frame src="top.html" name="TOP" scrolling="no" noresize>
  <frameset cols="100,*">
    <frame src="menu.html" name="MENU">
    <frame src="content.html" name="CONTENT">
  </frameset>
</frameset>
</html>

```

5. Создайте самостоятельно документ с фреймами, как изображено на рисунке:



Текст может быть любой. Основная задача сделать гиперссылки.

Лабораторная работа № 5. Работа с CSS-таблицами

Внешние таблицы стилей

Задание

1. Создайте обыкновенную html-страницу, примерно с таким кодом:

```
<html>
<head>
  <title>Подключение CSS к HTML</title>
</head>
<body>
  <h1>Это заголовок первого уровня</h1>
  Здесь просто текст
  <h2>Это заголовок второго уровня</h2>
  Здесь просто текст
</body>
</html>
```

2. Теперь создайте в блокноте страницу (пока пустую) и сохраните ее как style.css в ту же папку, где лежит html-страница.

3. Подключение файла css в html-странице. В созданной html-странице после контейнера <title> </title> вставить следующий код: <link rel="stylesheet" type="text/css" href="style.css">

Внутренние таблицы стилей

Так называют таблицу стилей, заданную внутри элемента HTML, при помощи атрибута style. Пример кода: <h1 style="color:red">Это заголовок красного цвета</h1>

Встроенные таблицы стилей

При этом способе таблица стилей встраивается в заголовок html-страницы. Для этого в HTML существуют теги <style></style>, с параметром type, который указывает, что подключается именно таблица стилей CSS (вообще-то существуют и другие).

Пример кода:

```
<html>
<head>
<title>Подключение CSS к HTML</title>
<style type="text/css">
h1{
color:red
}
</style>
</head>
<body>
<h1>Этот заголовок будет красного цвета</h1>
<h1>Этот заголовок будет красного цвета</h1>
<h1>Этот заголовок будет красного цвета</h1>
</body>
</html>
```



```
p{
  color:black;
}
p#pink{
  color:pink;
}
```

Посмотрите результат.

Селекторы по классу

В примере выше мы создали параграф с розовым цветом текста и указали, что такой id может быть только один. Для того чтобы розовый цвет текста был у двух или трех параграфов, существует параметр class, в качестве значения которого указывается его имя.

9. Добавить в html-страницу еще пару параграфов и присвоить им class="pink":

```
<html>
  <head>
    <title>CSS class</title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <h1>Это заголовок первого уровня</h1>
    Здесь просто текст
    <h2>Это заголовок второго уровня</h2>
    Здесь просто текст
    <p>Простой абзац</p>
    <p id="pink">Абзац с идентификатором</p>
    <p class="pink">Абзац с классом (class) pink</p>
    <p class="pink">Абзац с классом (class) pink</p>
  </body>
</html>
```

10. Для того чтобы указать стиль для этого класса, в таблице стилей напишем правило, где в качестве селектора снова будет использоваться элемент p и имя pink, но в данном случае это имя класса, поэтому в качестве разделителя будет использоваться точка (.):

```
p.pink{
  color:pink;
}
```

Контекстный селектор

Задание

1. Создать страницу со следующим кодом:

```

<html>
  <head>
    <title>Селекторы по элементу</title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <p>Этот текст находится в параграфе</p>
    Это просто текст.
    <i>Этот текст выделен курсивом</i>
    <p>Этот текст находится в параграфе, но <i>эта часть выделена курсивом</i></p>
  </body>
</html>

```

2. Необходимо выделить курсив зеленым цветом:

```

i{
  color:green;
}

```

Результат:

Этот текст находится в параграфе

Это просто текст. *Этот текст выделен курсивом*

Этот текст находится в параграфе, но *эта часть выделена курсивом*

3. Сделайте так, чтобы во втором предложении курсив был черного цвета.

Css + списки

Задание

1. Создать документ содержащий следующее:

```

<body>
  <ul id="spisok1">
    <li>Первый пункт</li>
    <li>Второй пункт</li>
    <li>Третий пункт</li>
  </ul>
  <ul id="spisok2">
    <li>Первый пункт</li>
    <li>Второй пункт</li>
    <li>Третий пункт</li>
  </ul>
  <ul id="spisok3">
    <li>Первый пункт</li>
    <li>Второй пункт</li>
    <li>Третий пункт</li>
  </ul>
</body>

```

2. Создать style.css:

```

#spisok1{
  list-style-type:disk;
}
#spisok2{
  list-style-type:circle;
}
#spisok3{
  list-style-type:square;
}

```

Не забудьте подключить данный файл в основной текст html документа.

Результат:

- Первый пункт
 - Второй пункт
 - Третий пункт
-
- Первый пункт
 - Второй пункт
 - Третий пункт
-
- Первый пункт
 - Второй пункт
 - Третий пункт

Добавление картинки в списки

3. Взять картинку: 

4. Наберите следующий текст:

```
<body>
  <ul id="spisok">
    <li>Первый пункт</li>
    <li>Второй пункт</li>
    <li>Третий пункт</li>
  </ul>
</body>
```

5. На странице style.css задать стиль:

```
#spisok{
list-style-image: url(marker.gif);
}
```

Результат:

-  Первый пункт
-  Второй пункт
-  Третий пункт

Свойства – шрифты

Задание

1. Создать следующий документ:

```
<html>
  <head>
    <title>css шрифты</title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <p id="sn">Текст в параграфе с идентификатором sn.</p>
    <p id="so">Текст в параграфе с идентификатором so.</p>
    <p id="si">Текст в параграфе с идентификатором si.</p>
  </body>
</html>
```

2. Зададим в таблице стилей шрифт для всех параграфов:

```
p#sn, p#so, p#si{
  font-family: Verdana, sans-serif;
}
```

3. Зададим каждому параграфу свой стиль:

```

p#sn, p#so, p#si{
  font-family: Verdana, sans-serif;
}
p#sn{
  font-style:normal;
}
p#so{
  font-style:oblique;
}
p#si{
  font-style:italic;
}

```

Различные шрифты включают в себя различные варианты начертания (bold, italic, regular и другие). Но, если в какой-либо шрифт не включен вариант italic, то он имитируется простым наклоном букв, что и соответствует значению oblique.

4. Сделайте второй абзац жирным. `font-weight:bold;`

<p>Текст в параграфе с идентификатором sn.</p> <p>Текст в параграфе с идентификатором so.</p> <p>Текст в параграфе с идентификатором si.</p>

Должно получиться 4 документа

Лабораторная работа №6. Создание и отображение XML-документа

Создание XML-документа

XML-документ представляет собой текст, который создается в текстовом редакторе, входящий в состав Microsoft Windows (блокнот, WordPad).

Задание

Создайте XML-документ:

1. Откройте новый файл в вашем текстовом редакторе и введите текст XML-документа, листинг которого представлен ниже. При желании можно опустить некоторые элементы типа BOOK. Элемент BOOK включает в себя теги <BOOK> и </BOOK> и весь содержащийся между ними текст.
2. Воспользуйтесь командой Save (Сохранить) текстового редактора, для сохранения документа на вашем жестком диске, присвоив ему имя Inventory.xml.

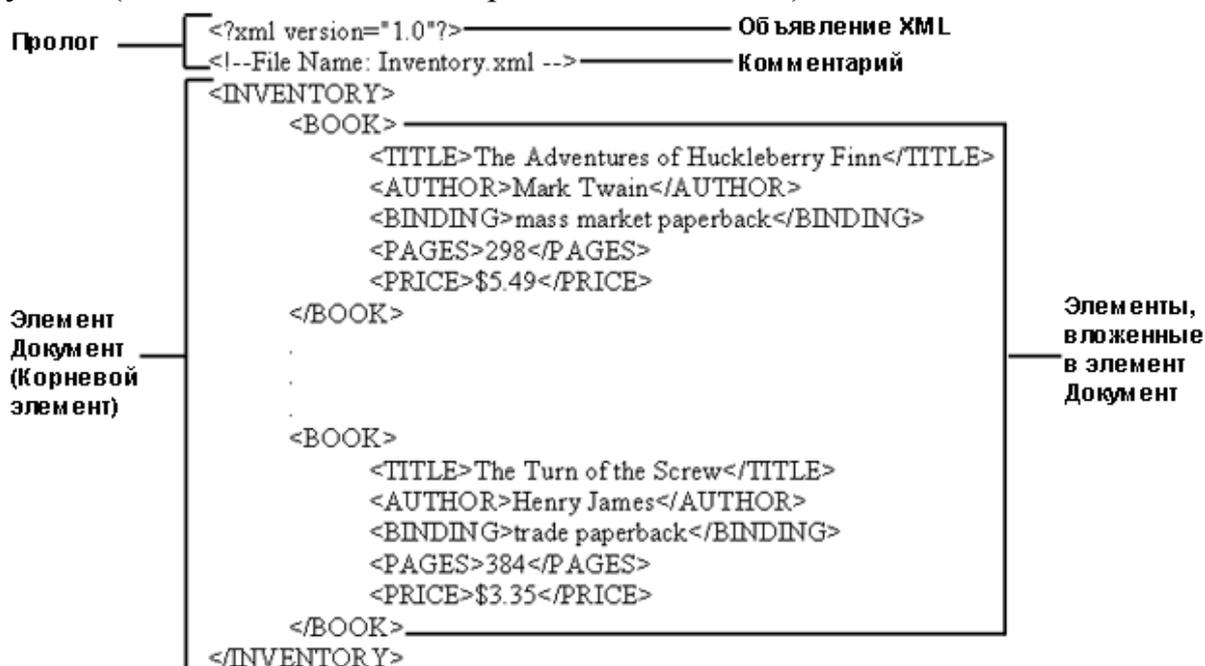
```

<?xml version="1.0"?>
<!-- File Name: Inventory.xml -->
<INVENTORY>
  <BOOK>
    <TITLE>The Adventures of Huckleberry Finn</TITLE>
    <AUTHOR>Mark Twain</AUTHOR>
    <BINDING>mass market paperback</BINDING>
    <PAGES>298</PAGES>
    <PRICE>$5.49</PRICE>
  </BOOK>
</INVENTORY>

```

Анатомия XML-документа.

XML-документ состоит из двух основных частей: пролога и элемента. Документ (его также называют корневым элементом).



Пролог состоит из следующих строк:

```
<?xml version="1.0"?>
<!-- File Name: Inventory.xml -->
```

Первая строка представляет собой объявление XML, указывающее на то, что это XML-документ и содержащее номер версии. Объявление XML не является обязательным, хотя спецификация требует его включения. Если вы включаете XML-объявление, оно должно находиться в начале документа.

Вторая строка пролога состоит из пробела. С целью улучшения внешнего вида документа вы можете вставлять любое количество пустых строк между элементами пролога. При обработке они будут игнорироваться.

Третья строка пролога представляет собой комментарий. Добавление комментариев в XML-документ не обязательно, но позволяет сделать его более понятным. Комментарий начинается с символов <!-- и заканчивается символами -->. Между этими двумя группами символов вы можете поместить любой текст (за исключением двойного тире --); XML-процессор проигнорирует его.

Пролог может также содержать следующие необязательные компоненты:

- объявление типа документа, определяющее тип и структуру документа. Объявление типа документа должно следовать после XML-объявления;
- одна или несколько инструкций по обработке, содержащих информацию о порядке проходов при обработке приложения XML-процессором. Далее в этой лекции вы познакомитесь с инструкцией по обработке для связывания таблицы стиля с XML-документом.

Примечание. XML-процессор – это программный модуль, считывающий XML-документ и обеспечивающий доступ к его содержимому. Он также предоставляет этот доступ другим программным модулям, или приложениям, которые манипулируют и отображают содержимое документа. Если вы отображаете XML-документ в Internet Explorer 5, браузер включает в себя как XML-процессор, так и приложение. (Если для отображения XML-документа вы используете HTML и сценарий (скрипт-код), то при этом самостоятельно создаете

часть приложения.) Обратите внимание, что термин приложение в данном случае отличен от термина, применяемого для обозначения XML-приложения (или словаря) как целевого набора элементов и структуры документа, которые используются для описания документов определенного типа.

Элемент Документ. Второй основной частью XML-документа является единый элемент Документ, или корневой элемент, который в свою очередь содержит дополнительные элементы.

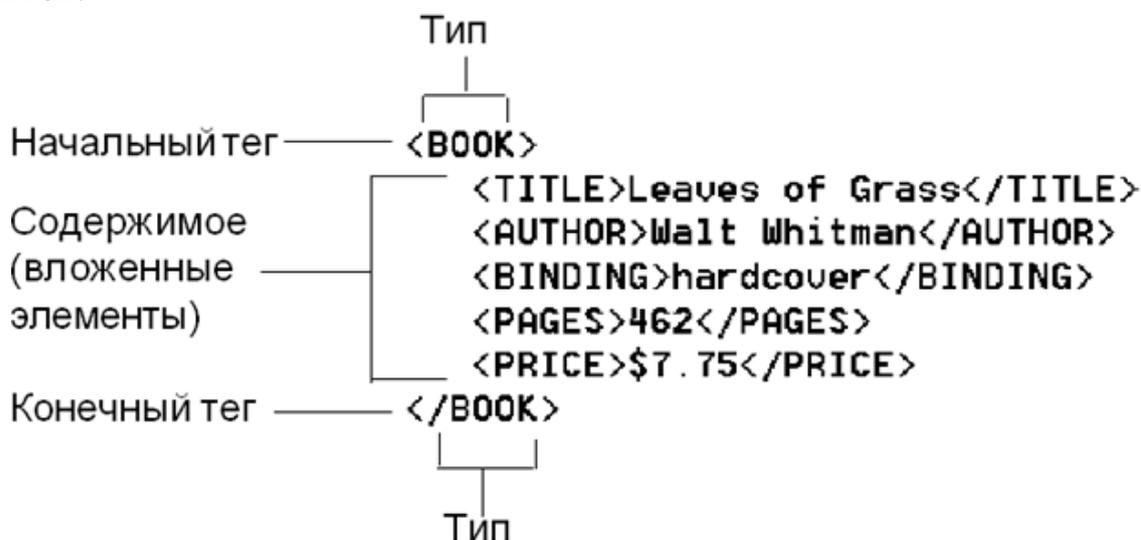
В XML-документе элементы определяют его логическую структуру и несут в себе информацию, содержащуюся в документе (в нашем примере это информация о книгах, такая как название, автор, жанр, страницы, цена). Типовой элемент состоит из начального тега, содержимого элемента и конечного тега. Содержимым элемента могут быть символьные данные, другие (вложенные) элементы, либо сочетание данных и вложенных элементов.

В рассматриваемом примере элемент Документ – INVENTORY. Его начальный тег – <INVENTORY>, конечный тег – </INVENTORY>, а содержимое – пять вложенных элементов BOOK.

Примечание. Текст в XML-документе представляет собой перемежающиеся символьные данные и данные, относящиеся к разметке. Разметка – это текст, ограниченный разделителями и описывающий структуру документа. А именно, начальный и конечный теги элемента, теги пустого элемента, объявления типа документа, инструкции по обработке, ограничители раздела CDATA, символьные ссылки, ссылки на примитивы (entity). Остальной текст представляет собой символьные данные – реальное информационное содержимое документа (в нашем примере это названия, фамилии авторов, цена и другая информация о книге).

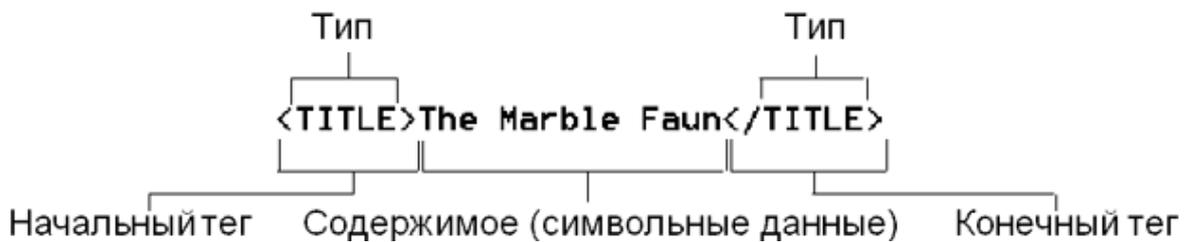
Примечание. Элемент Документ в XML-документе похож на элемент BODY на HTML-странице, за исключением того, что вы можете присвоить ему любое допустимое имя.

В свою очередь, каждый элемент BOOK содержит ряд вложенных элементов:



Примечание. Имя, которое содержится в начальном и конечном теге, есть тип элемента.

Каждый из элементов, вложенных в элемент BOOK, например, элемент TITLE, содержит только символьные данные, как показано на рисунке ниже:



Во второй части вы узнаете о дополнительных элементах XML-документа и о включении атрибутов в начальный тег элемента.

Некоторые базовые правила XML.

Ниже приведено несколько основных правил создания форматированного XML-документа. Форматированный документ соответствует минимальному набору правил, обеспечивающих возможность обработки документа браузером или другой программой. Документ, составленный вами ранее в этой лекции, является примером форматированного XML-документа, удовлетворяющего этим правилам:

- документ должен иметь только один элемент верхнего уровня (элемент Документ или корневой элемент). Все другие элементы должны быть вложены в элемент верхнего уровня;
- элементы должны быть вложены упорядоченным образом. То есть, если элемент начинается внутри другого элемента, он должен и заканчиваться внутри этого элемента;
- каждый элемент должен иметь начальный и конечный тег. В отличие от HTML, в XML не разрешается опускать конечный тег – даже в том случае, когда браузер в состоянии определить, где заканчивается элемент;
- имя типа элемента в начальном теге должно в точности соответствовать имени в соответствующем конечном теге;
- имена типов элементов чувствительны к регистру, в котором они набраны. В действительности весь текст внутри XML-разметки является чувствительным к регистру. Например, следующее описание элемента является неправильным, поскольку имя типа элемента в начальном теге не соответствует имени типа в конечном теге:

`<TITLE>Leaves of grass</Title> <!-- некорректный элемент -->`

Отображение XML-документа

Вы можете открыть XML-документ непосредственно через Internet Explorer, точно так же, как вы бы открыли HTML Web-страницу.

Если XML-документ не содержит связи с таблицей стилей, Internet Explorer помечает различные составные части документа различным цветом, чтобы облегчить их распознавание, а также представляет элемент Документ в виде иерархического дерева с возможностью свертывания и развертывания структуры и просмотра с меньшей или большей степенью детализации.

Если же XML-документ имеет связь с таблицей стиля, Internet Explorer отобразит только символьные данные из элементов документа, отформатировав их в соответствии с правилами, установленными в таблице стиля. Вы можете использовать либо таблицу каскадных стилей (CSS-таблицу, аналогичную той, которая используется для HTML-страниц), либо XSL-таблицу стилей, которая

является более мощным инструментом и строится в соответствии с синтаксисом, принятым для XML. Такие таблицы могут использоваться исключительно для XML-документов.

Обнаружение ошибок XML в Internet Explorer.

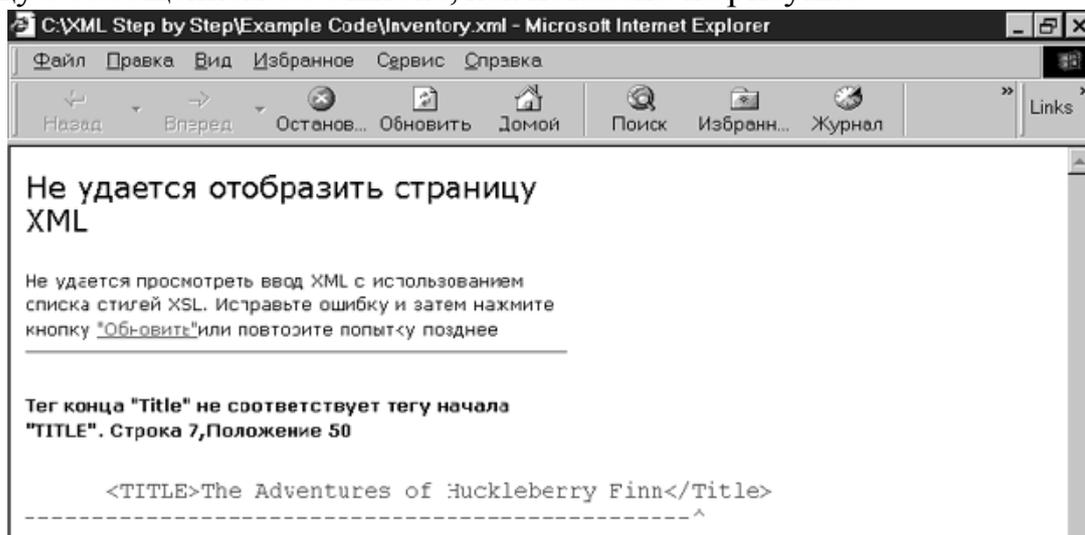
Прежде чем Internet Explorer отобразит ваш XML-документ, его встроенный синтаксический XML-анализатор (parser) просматривает содержимое документа. Если он обнаружит ошибку, Internet Explorer отобразит страницу с сообщением об ошибке, не предпринимая попытки отобразить документ. Internet Explorer отобразит страницу с сообщением об ошибке независимо от того, связан ли XML-документ с таблицей стилей.

Примечание. Синтаксический XML-анализатор является составной частью XML-процессора, который сканирует XML-документ, анализирует его структуру и обнаруживает синтаксические ошибки.

В следующем упражнении вы исследуете возможности Internet Explorer по обнаружению ошибок, искусственно введя ошибку в документ Inventory.xml.

Задание

1. В вашем текстовом редакторе откройте документ Inventory.xml, созданный вами в предыдущем упражнении. Измените первый элемент TITLE с `<TITLE>The Adventures of Huckleberry Finn</TITLE>` на `<TITLE>The Adventures of Huckleberry Finn</Title>`.
2. Сохраните внесенные изменения.
3. Дважды щелкните на имени файла документа Inventory.xml. Вместо того, чтобы отобразить XML-документ, Internet Explorer теперь отобразит следующую страницу с сообщением об ошибке, как показано на рисунке:



4. Поскольку вы еще будете работать с Inventory.xml в этой лекции, вам теперь нужно восстановить конечный тег для первого элемента TITLE, вернув ему первоначальный вид (`/TITLE`), а затем повторно сохранить документ.

Отобразите XML-документ с использованием таблицы каскадных стилей:

1. Откройте новый, пустой текстовый файл в вашем текстовом редакторе и заполните CSS-таблицу, как показано на рисунке ниже:

```

/* File Name: Inventory01.css */
BOOK
  {display:block;
  margin-top:12pt;
  font-size:10pt}
TITLE
  {font-style:italic}
AUTHOR
  {font-weight:bold}

```

2. С помощью команды Save (Сохранить) вашего текстового редактора сохраните таблицу стиля на жестком диске, задав имя файла Inventory01.css. Созданная вами CSS-таблица сообщает Internet Explorer, каким образом форматировать символьные данные элементов.

- отображать каждый элемент BOOK с отступом сверху в 12 пт (margin-top:12pt) с переводом строки сверху и снизу (display:block), используя размер шрифта 10 пт (font-size:10pt);
- отображать каждый элемент TITLE курсивом (font-style:italic);
- отображать каждый элемент AUTHOR полужирным (font-weight:bold).

3. В вашем текстовом редакторе откройте документ Inventory.xml, созданный вами в предыдущем упражнении. Добавьте в конце пролога документа (непосредственно над элементом INVENTORY) следующую инструкцию по обработке:

```
<?xml-stylesheet type="text/css" href="Inventory01.css"?>
```

Эта инструкция по обработке устанавливает связь созданной вами CSS-таблицы и XML-документа. В результате при открытии документа в Internet Explorer браузер отобразит содержимое документа в соответствии с инструкциями, записанными в таблице стилей.

4. Чтобы отразить новое имя файла, которое вы собираетесь присвоить, измените комментарий в начале документа с <!-- File Name: Inventory.xml --> на <!-- File Name: Inventory01.xml -->.

5. Воспользуйтесь командой Save As (Сохранить как) вашего текстового редактора, чтобы сохранить копию измененного документа под именем Inventory01.xml. Проверьте, чтобы этот файл был сохранен в той же папке, что и файл Inventory01.css.

6. Дополните файл Inventory01.xml новыми книгами:

```

<?xml version="1.0"?>
<!-- File Name: Inventory01.xml -->
<?xml-stylesheet type="text/css" href="Inventory01.css"?>
<INVENTORY>
  <BOOK>
    <TITLE>The Adventures of Huckleberry Finn</TITLE>
    <AUTHOR>Mark Twain</AUTHOR>
    <BINDING>mass market paperback</BINDING>
    <PAGES>298</PAGES>
    <PRICE>$5.49</PRICE>
  </BOOK>
  <BOOK>
    <TITLE>Leaves of Grass</TITLE>
    <AUTHOR>Walt Whitman</AUTHOR>
    <BINDING>hardcover</BINDING>
    <PAGES>462</PAGES>
    <PRICE>$7.75</PRICE>
  </BOOK>

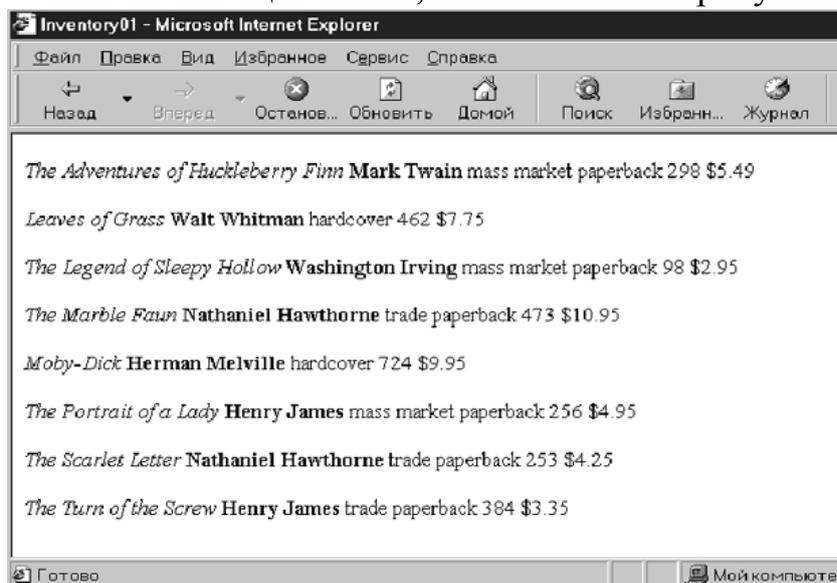
```

```

<BOOK>
  <TITLE>The Legend of Sleepy Hollow</TITLE>
  <AUTHOR>Washington Irving</AUTHOR>
  <BINDING>mass market paperback</BINDING>
  <PAGES>98</PAGES>
  <PRICE>$2.95</PRICE>
</BOOK>
<BOOK>
  <TITLE>The Marble Faun</TITLE>
  <AUTHOR>Nathaniel Hawthorne</AUTHOR>
  <BINDING>trade paperback</BINDING>
  <PAGES>473</PAGES>
  <PRICE>$10.95</PRICE>
</BOOK>
<BOOK>
  <TITLE>Moby-Dick</TITLE>
  <AUTHOR>Herman Melville</AUTHOR>
  <BINDING>hardcover</BINDING>
  <PAGES>724</PAGES>
  <PRICE>$9.95</PRICE>
</BOOK>
<BOOK>
  <TITLE>The Portrait of a Lady</TITLE>
  <AUTHOR>Henry James</AUTHOR>
  <BINDING>mass market paperback</BINDING>
  <PAGES>256</PAGES>
  <PRICE>$4.95</PRICE>
</BOOK>
<BOOK>
  <TITLE>The Scarlet Letter</TITLE>
  <AUTHOR>Nathaniel Hawthorne</AUTHOR>
  <BINDING>trade paperback</BINDING>
  <PAGES>253</PAGES>
  <PRICE>$4.25</PRICE>
</BOOK>
<BOOK>
  <TITLE>The Turn of the Screw</TITLE>
  <AUTHOR>Henry James</AUTHOR>
  <BINDING>trade paperback</BINDING>
  <PAGES>384</PAGES>
  <PRICE>$3.35</PRICE>
</BOOK>
</INVENTORY>

```

7. Дважды щелкните на файле Inventory01.xml, чтобы открыть документ. Internet Explorer откроет документ Inventory01.xml и отобразит его в соответствии с правилами из связанной таблицы стилей, как показано на рисунке:



8. Чтобы почувствовать, как можно изменять представление XML-документа путем модификации связанной таблицы стилей, откройте новый, пустой

текстовый файл в вашем текстовом редакторе и введите измененную CSS-таблицу, как показано на рисунке:

```
/* File Name: Inventory02.css */
BOOK
    {display:block;
    margin-top:12pt;
    font-size:10pt}

TITLE
    {display:block;
    font-size:12pt;
    font-weight:bold;
    font-style:italic}

AUTHOR
    {display:block;
    margin-left:15pt;
    font-weight:bold}

BINDING
    {display:block;
    margin-left:15pt}

PAGES
    {display:none}

PRICE
    {display:block;
    margin-left:15pt}
```

9. Воспользуйтесь командой Save (Сохранить) вашего текстового редактора, чтобы сохранить новую таблицу стилей на жестком диске, присвоив ей имя файла Inventory02.css. Созданная вами модифицированная таблица стилей сообщает Internet Explorer, каким образом форматировать символьные данные элементов.

- отображать каждый элемент BOOK с отступом сверху в 12 пт (margin-top:12pt) и с переводом строки сверху и снизу (display:block), используя размер шрифта 10 пт (font-size:10pt);
- отображать каждый из элементов TITLE, AUTHOR, BINDING и PRICE в отдельной строке (display:block);
- отображать элемент TITLE шрифтом с размером 12 пт (font-size:12pt), полужирным (font-weight:bold), курсивом (font-style:italic). Обратите внимание, что установка для элемента TITLE размера шрифта 12 пт превалирует над установкой размера 10 пт для родительского элемента BOOK;
- отступ слева для каждого из элементов AUTHOR, BINDING и PRICE на 15 пт (margin-left:15pt);
- отображать элемент AUTHOR полужирным (font-weight:bold);
- не отображать элемент PAGES (display:none).

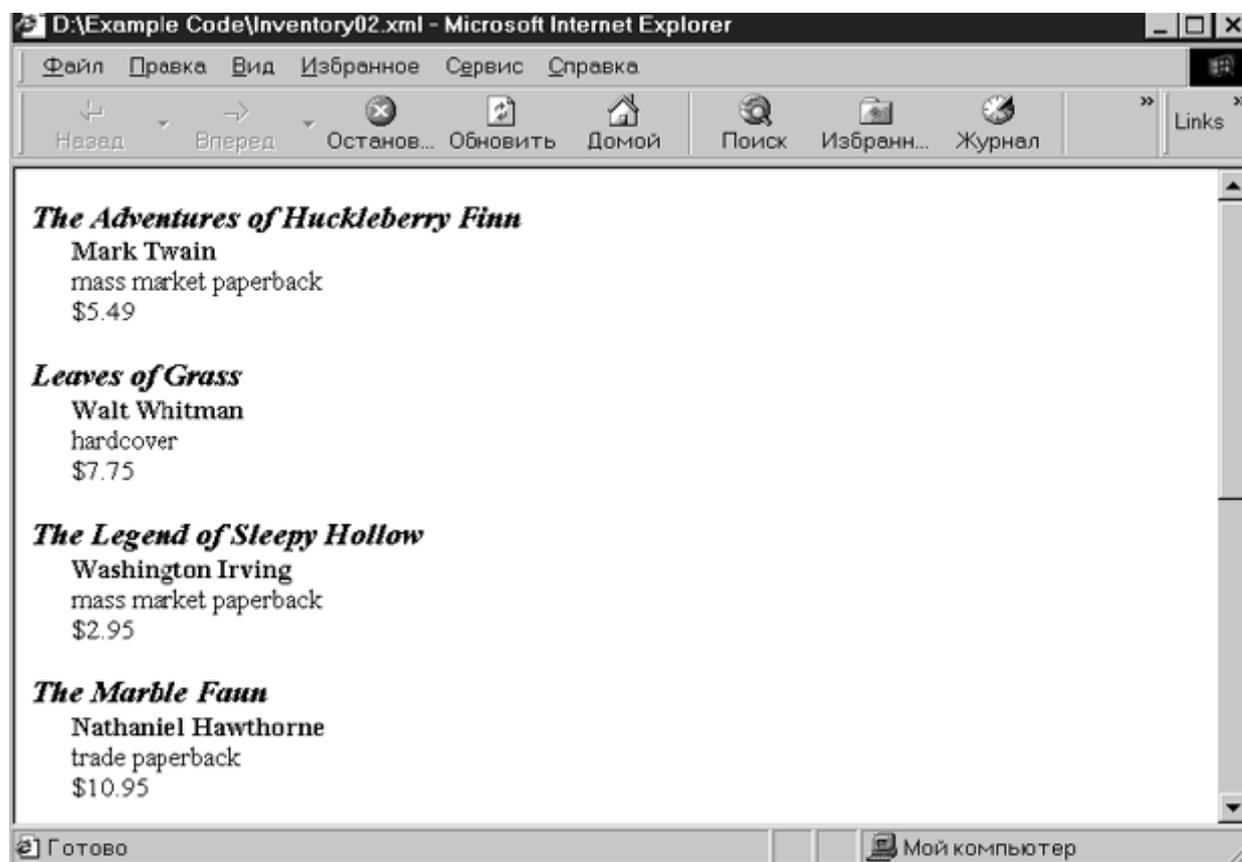
10. В вашем текстовом редакторе откройте документ Inventory01.xml. Замените в конце пролога документа (над элементом INVENTORY) следующую инструкцию по обработке:

```
<?xml-stylesheet type="text/css" href="Inventory02.css"?>
```

Эта инструкция по обработке устанавливает связь между новой CSS-таблицей, созданной вами, и XML-документом.

11. Чтобы отразить новое присвоенное вами имя файла, измените комментарий в начале документа с <!-- File Name: Inventory.xml --> на <!-- File Name: Inventory02.xml -->.

12. Воспользуйтесь командой Save As (Сохранить как), чтобы сохранить копию измененного документа под именем Inventory02.xml. Проверьте, чтобы этот файл был сохранен в той же папке, что и файл Inventory02.css.
13. Дважды щелкните на имени файла Inventory02.xml, чтобы открыть его. Internet Explorer откроет документ Inventory02.xml и отобразит его в соответствии с правилами, установленными в связанной таблице стилей Inventory02.css, как показано на рисунке ниже (здесь представлено лишь четыре книги; осуществив прокрутку вниз, вы увидите остальные):



Самостоятельная работа

1. Создайте XML-документ с инвентарем компьютерного магазина (5 элементов).
2. Сделайте отображение XML-документа «Компьютерный магазин» с произвольными символьными данными элементов (данные не должны повторяться с приведенным выше примером).

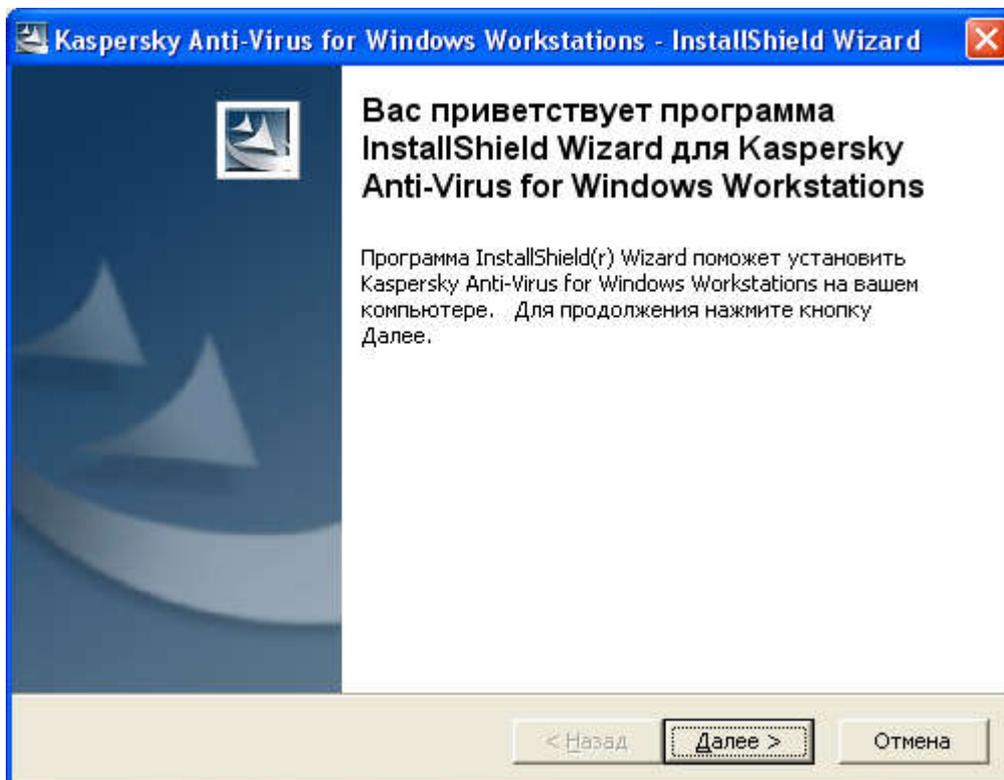
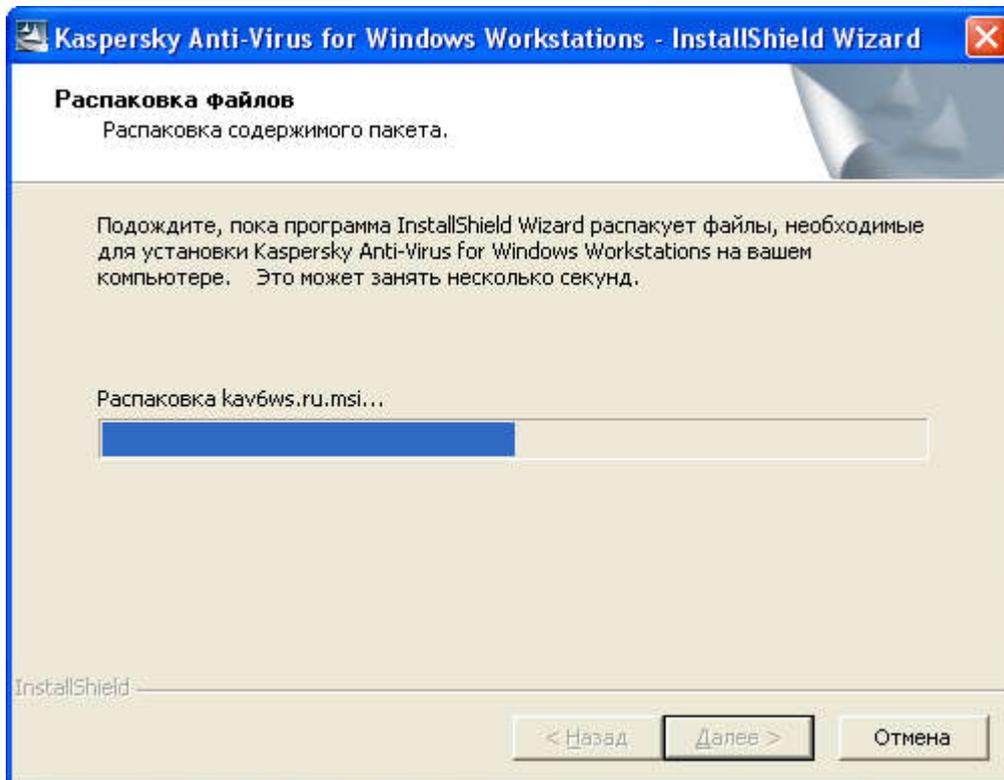
ПОДГОТОВИТЬСЯ К КОНТРОЛЬНОЙ РАБОТЕ ПО РАЗДЕЛУ 2

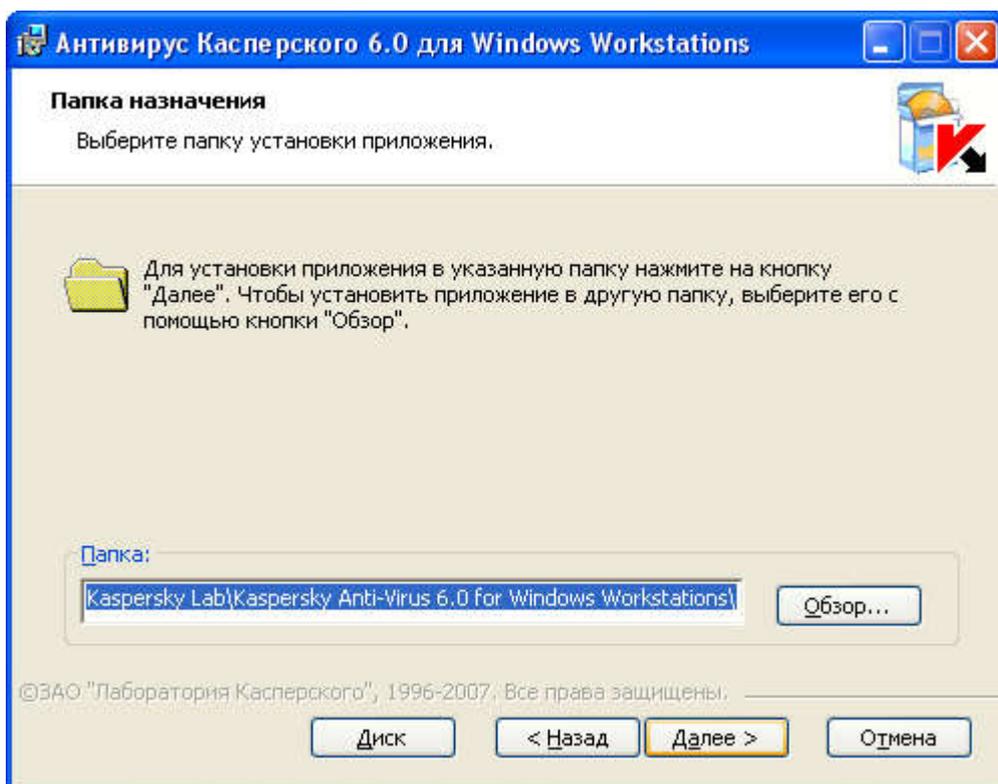
Лабораторная работа № 7

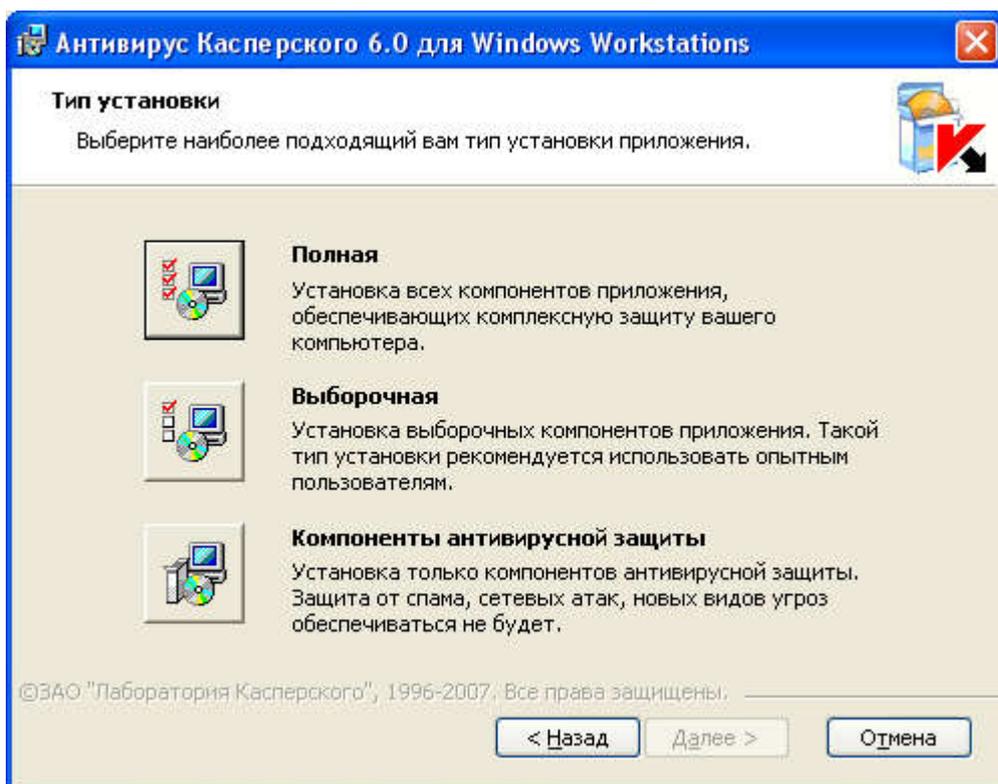
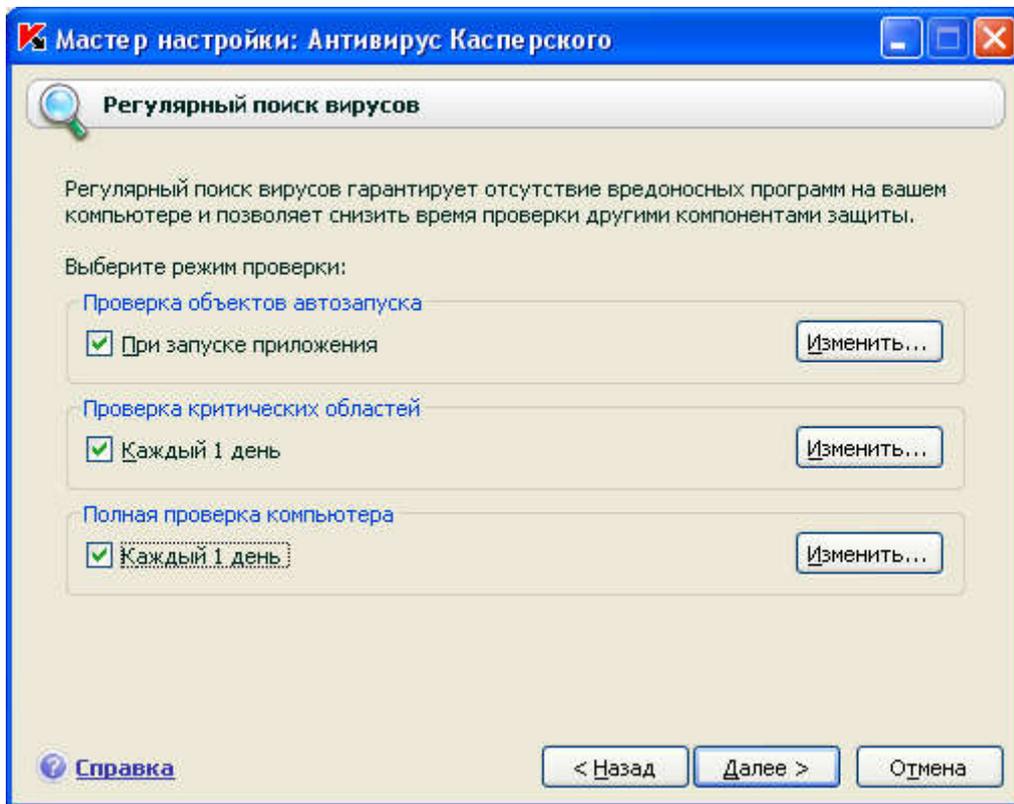
Сделать отчет в виде скриншотов и описаний по «созданию и изменению учетных записей» и по «созданию и администрированию групп пользователей».

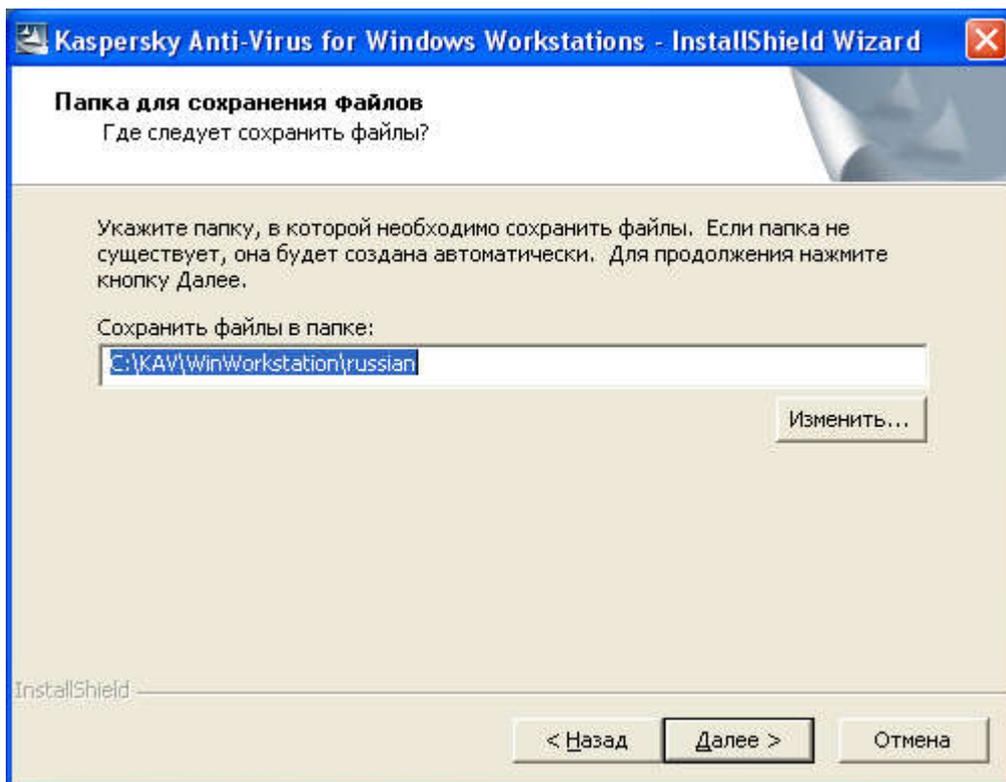
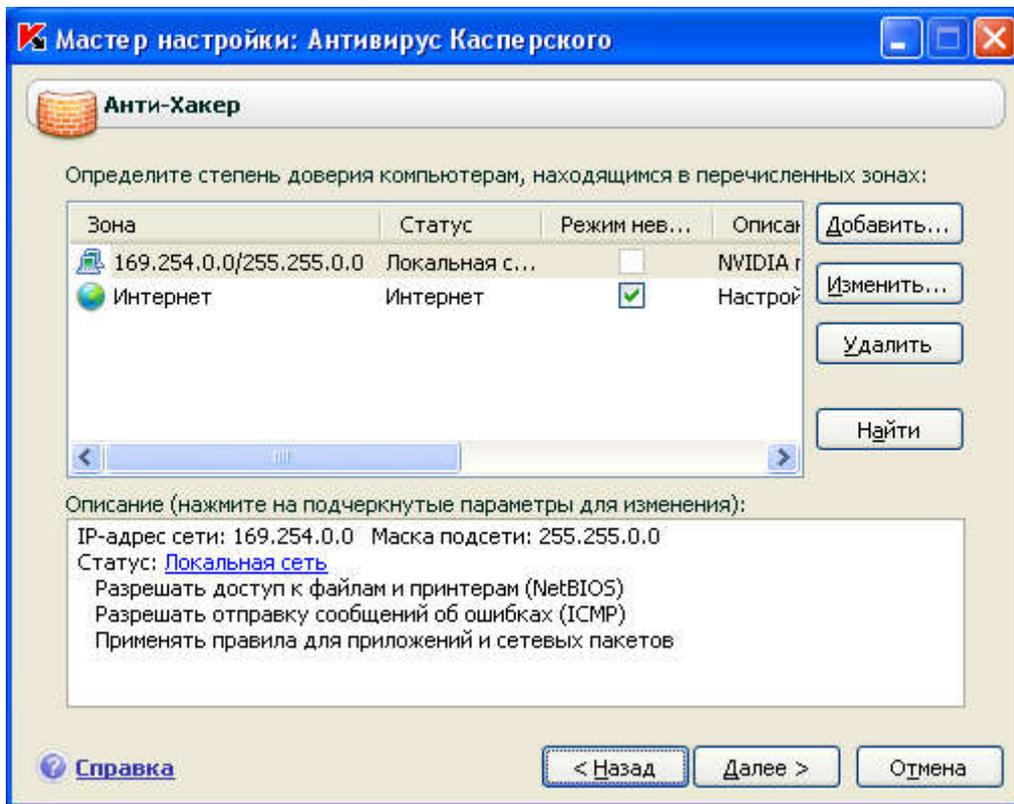
Лабораторная работа № 8

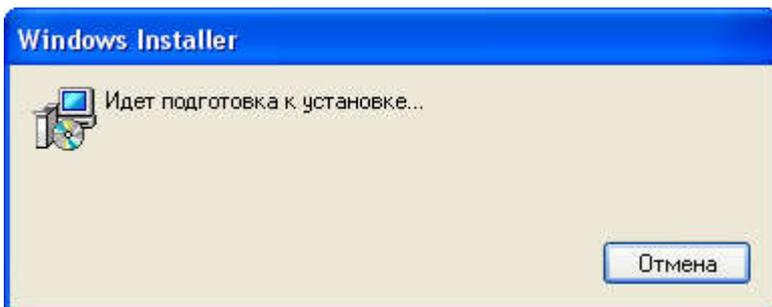
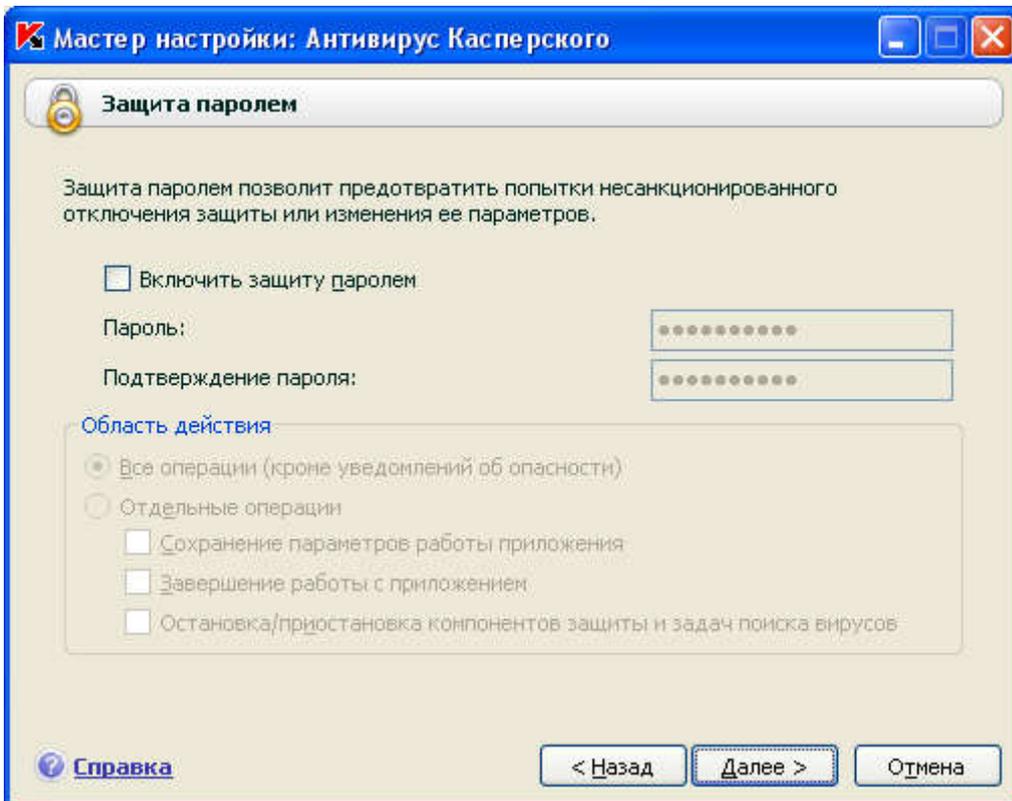
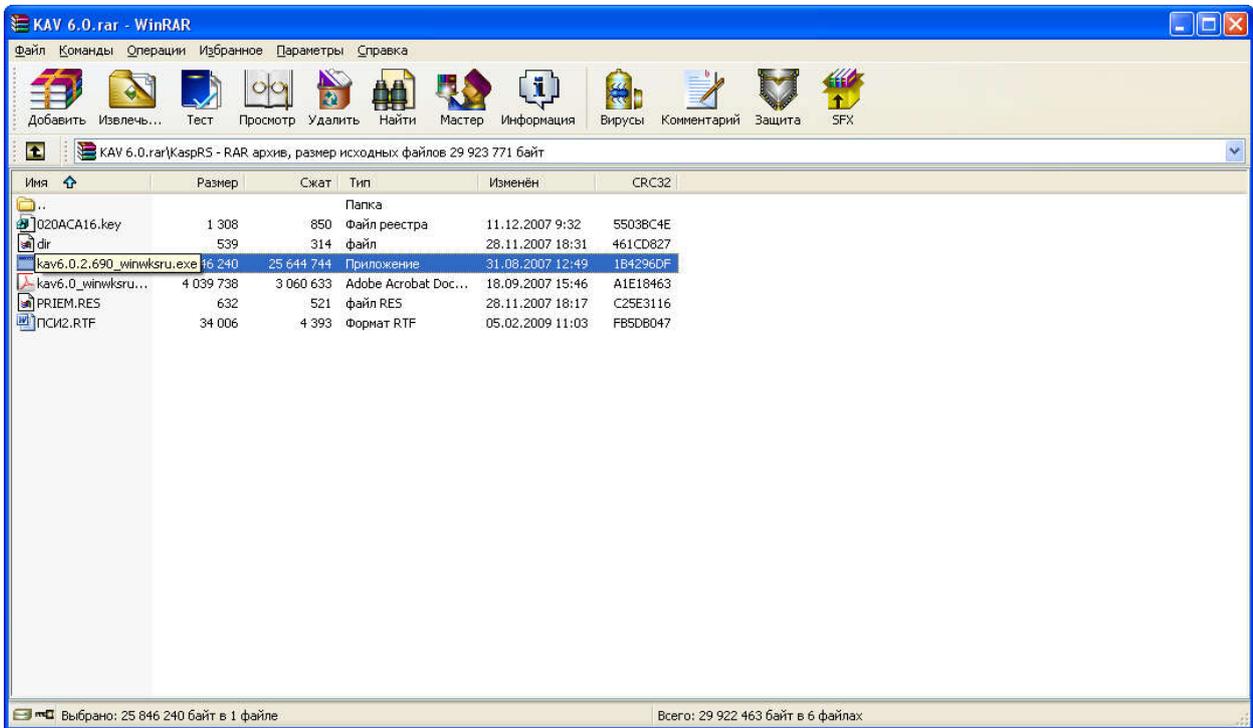
Создать инструкцию по установке антивируса Касперского с описанием скриншотов

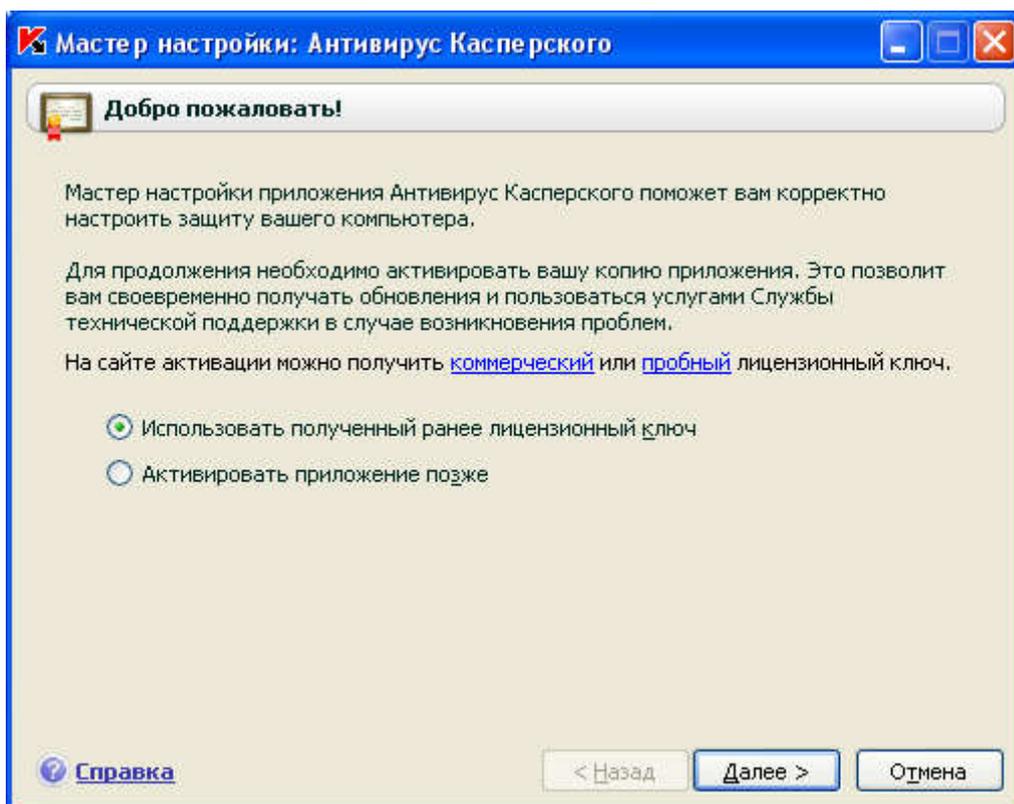
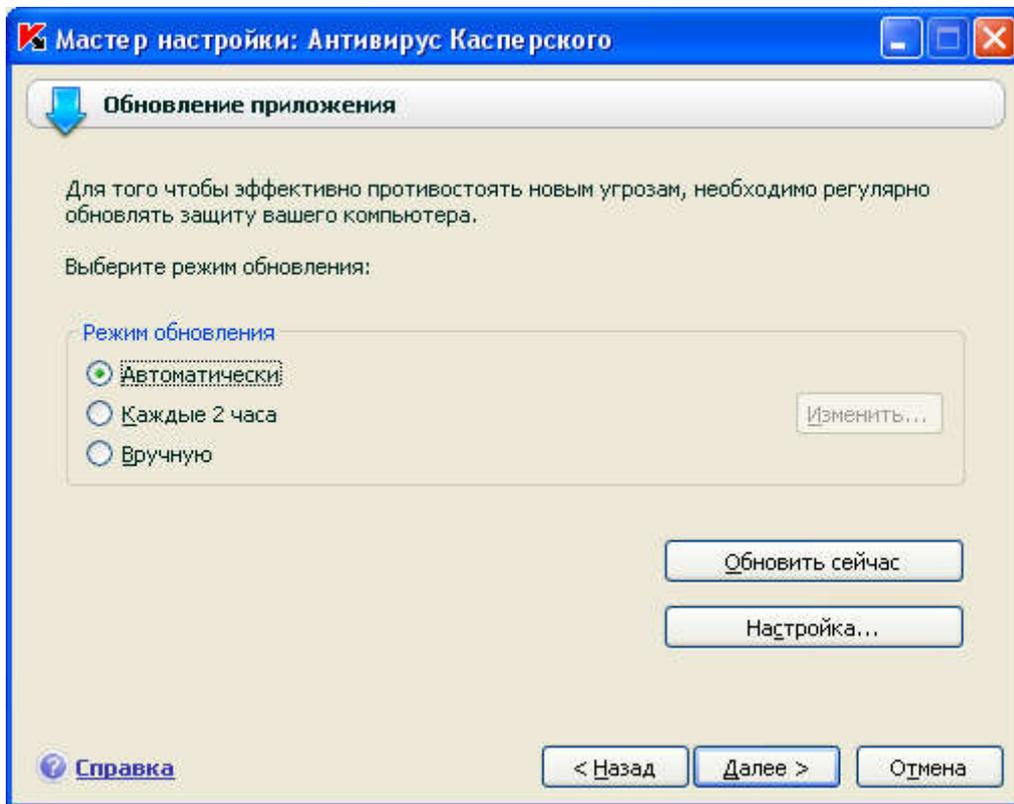


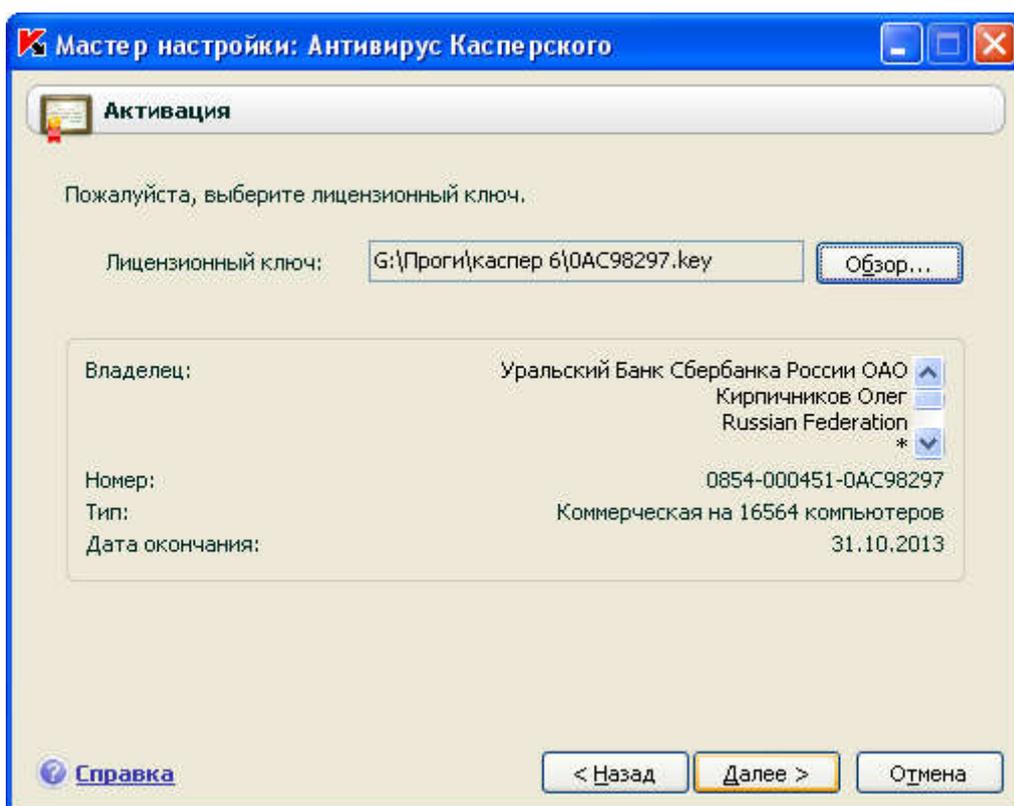
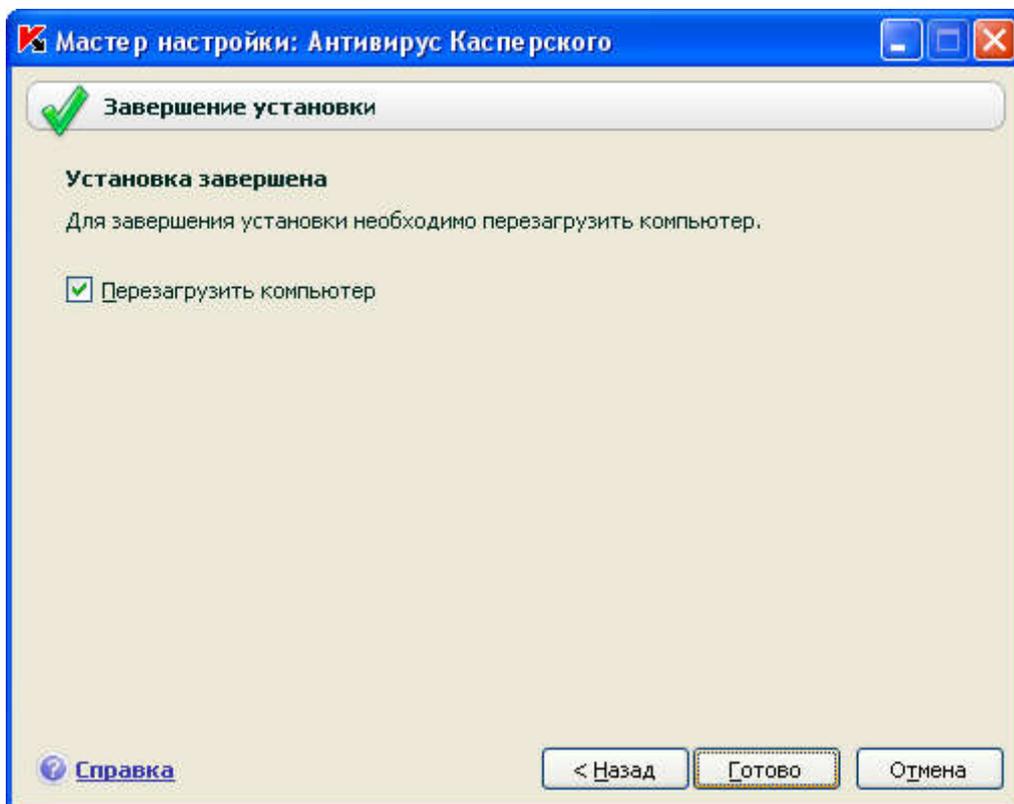


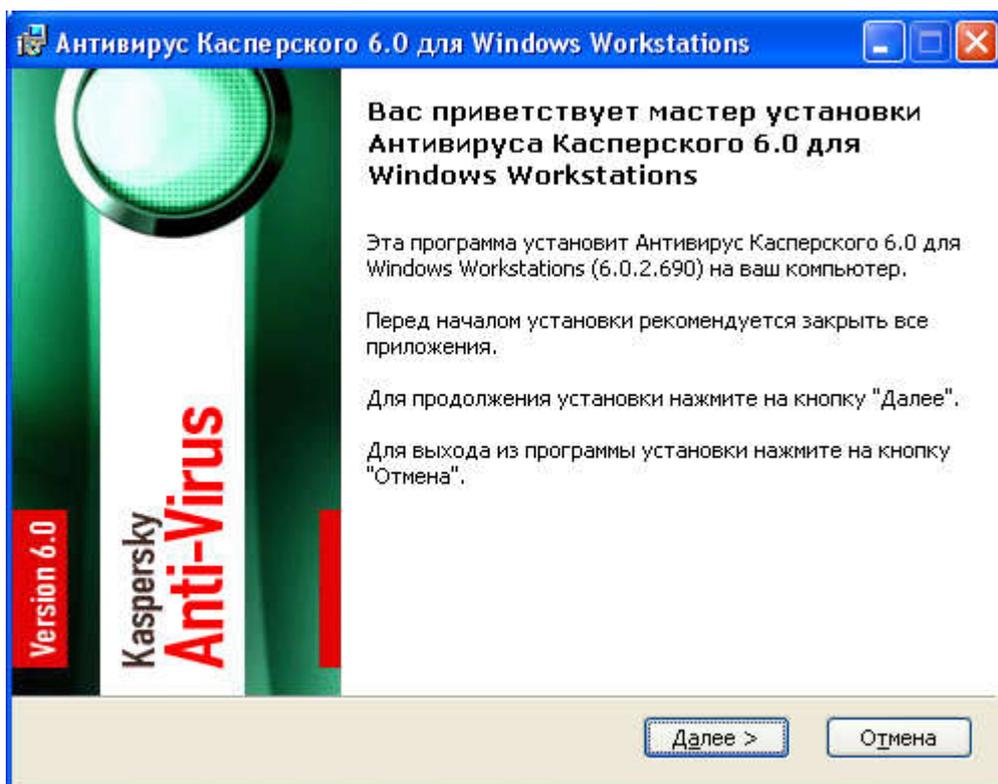
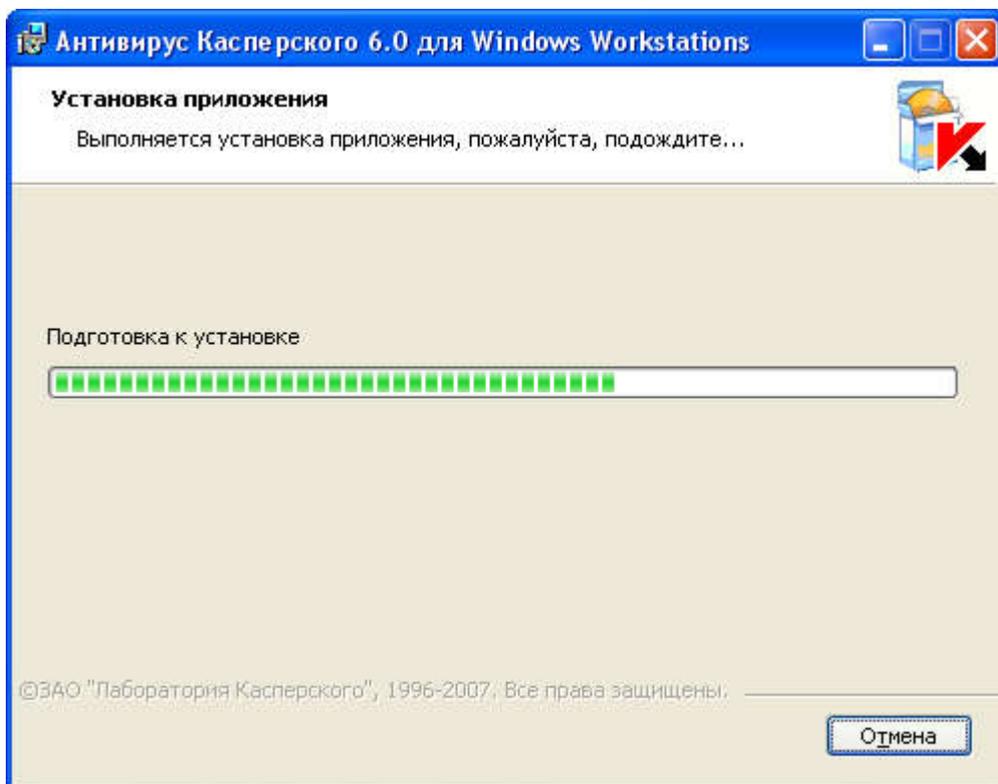


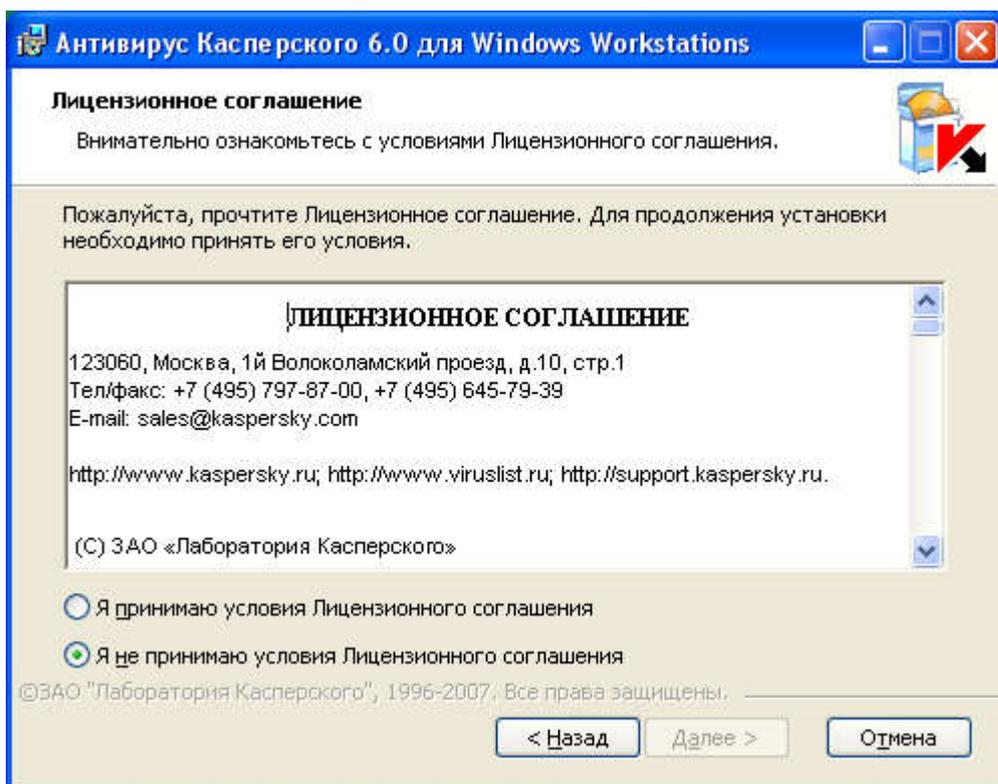
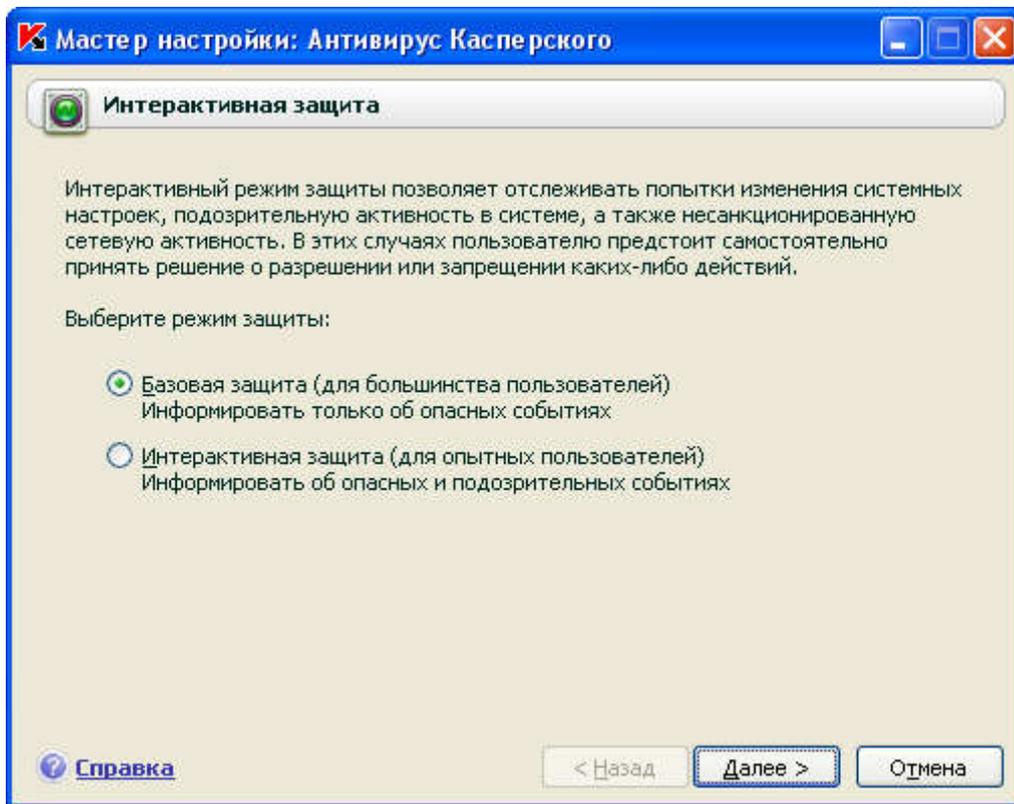


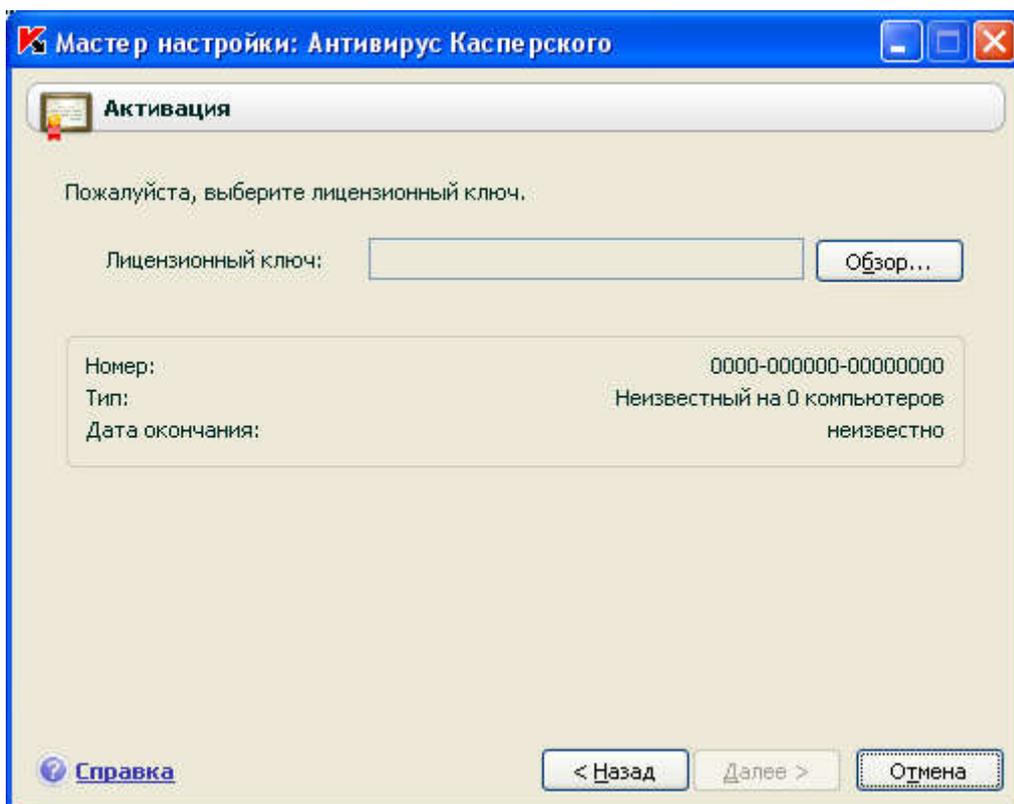
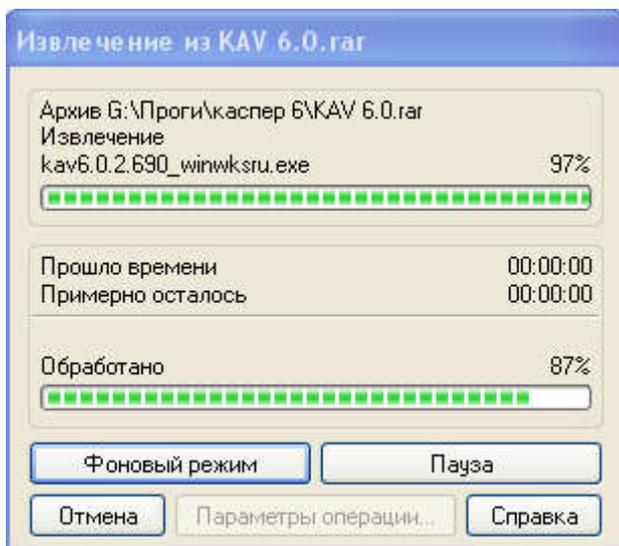


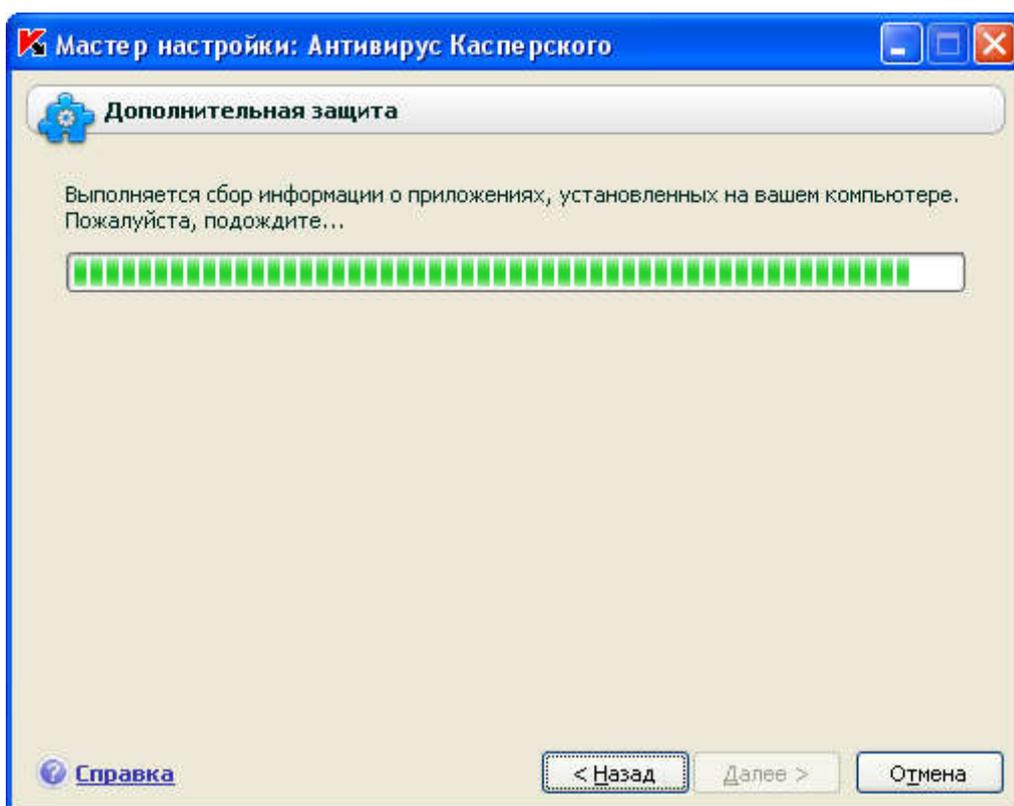
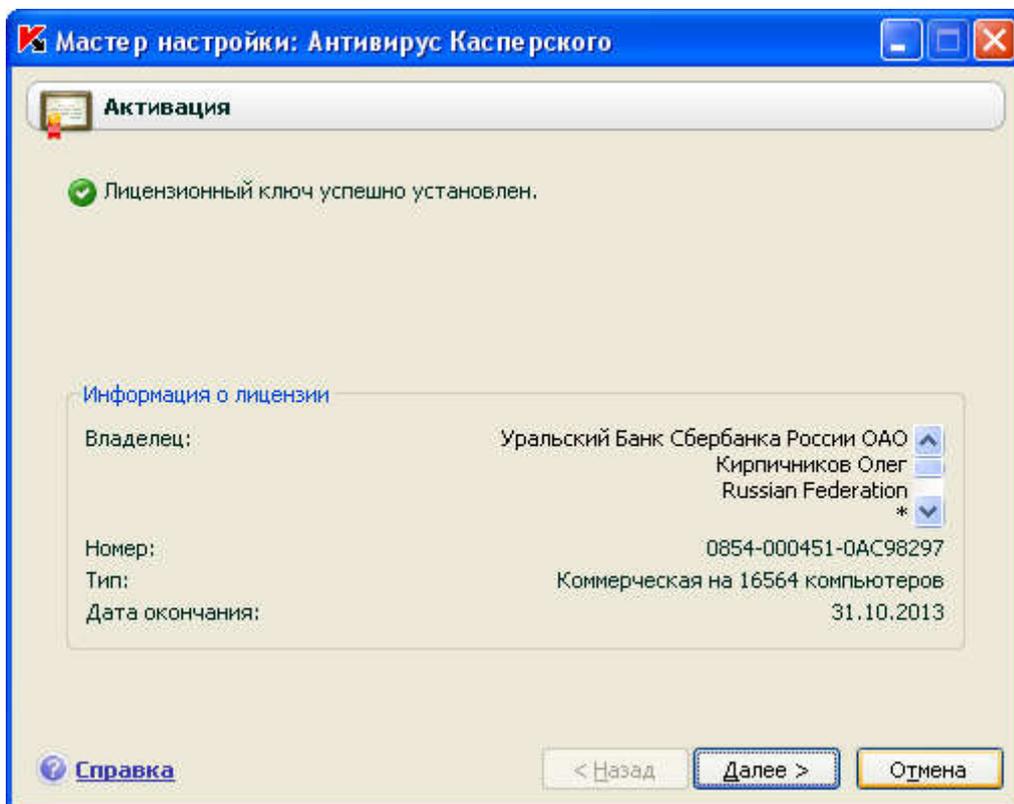


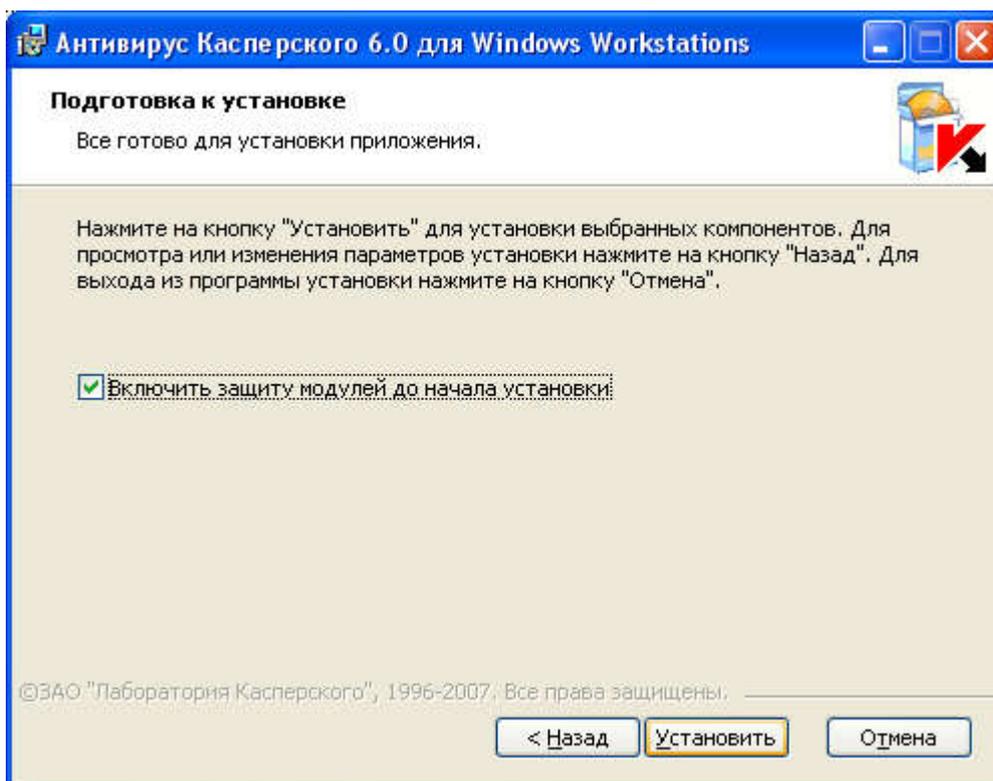
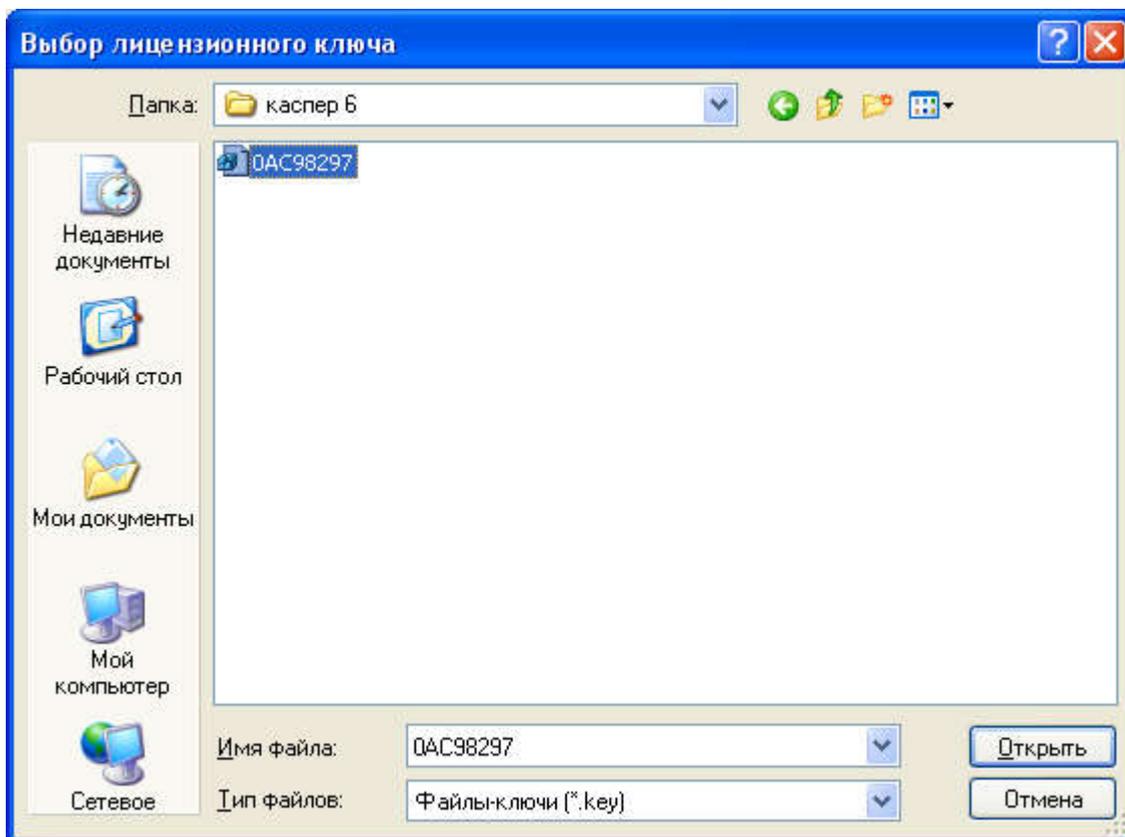












 **Антивирус Касперского 6.0 для Windows Workstations** 

Сетевой экран Microsoft Windows 

Отключение сетевого экрана Microsoft Windows.

Вы выбрали для установки компонент Анти-Хакер при включенном сетевом экране Microsoft Windows. Во избежание конфликтов сетевой экран будет отключен. Вы в любой момент можете включить его в Панели управления.

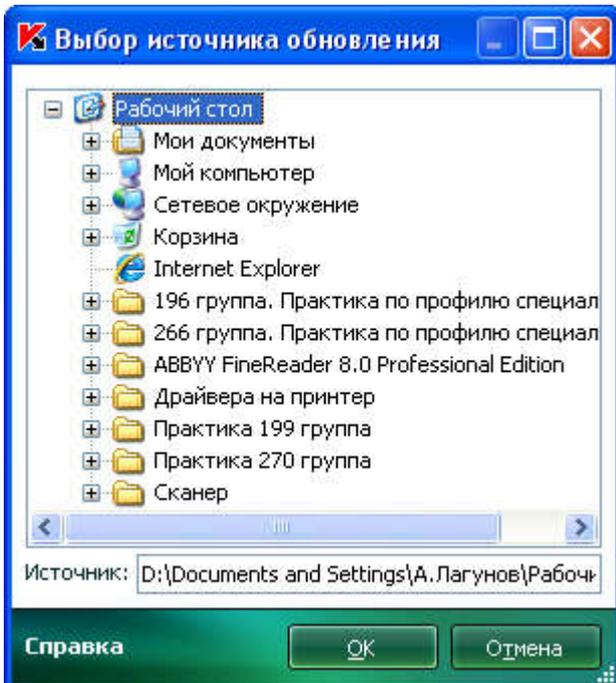
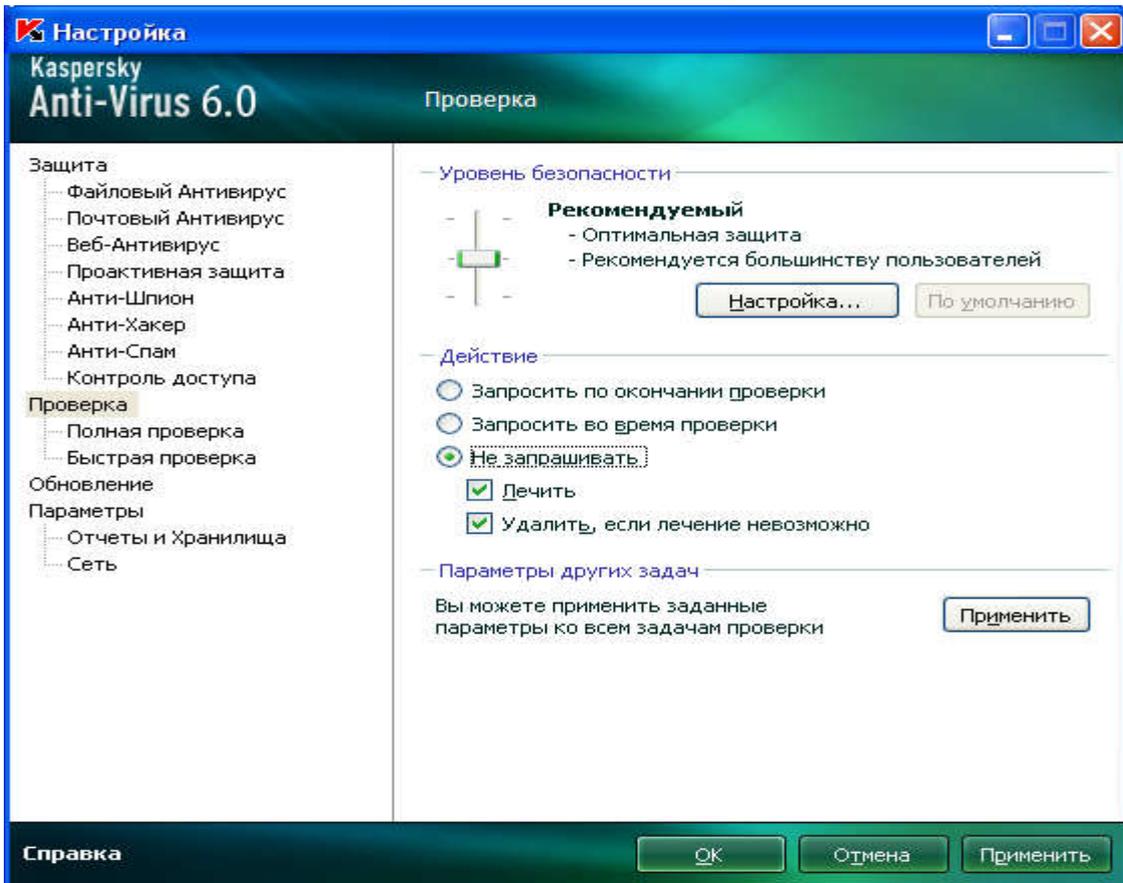
 **Отключить сетевой экран Microsoft Windows**
Сетевой экран Microsoft Windows будет отключен. Анти-Хакер обеспечит полную защиту при работе в сети.

 **Использовать сетевой экран Microsoft Windows**
Сетевой экран Microsoft Windows продолжает защищать ваш компьютер. Во избежание конфликтов Анти-Хакер будет отключен. Вы сможете в любой момент включить его.

©ЗАО "Лаборатория Касперского", 1996-2007. Все права защищены.

Лабораторная работа №9

Создать инструкцию по настройке антивируса Касперского с описанием скриншотов



Antivirus Касперского 6.0 для Windows Workstations

Kaspersky Anti-Virus 6.0

Настройка

Безопасность компьютера под угрозой Исправить!

Защита

Проверка
Полная проверка
Быстрая проверка

Обновление

Лицензия

Обновление
Сервис обновления помогает поддерживать защиту вашего компьютера в актуальном состоянии, чтобы противостоять новым вирусам и другим угрозам.

Статус баз: Актуальны [Обзор вирусной активности](#)

Типы угроз:	Всего:	Дата выпуска баз:
Вредоносные программы	8152901	20.06.2012 6:04:00
Баннеры	28859	19.06.2012 20:25:00
Фишинговые сайты	353995	19.06.2012 6:36:00
Спам	123048	20.06.2012 2:12:00
Вредоносные скрипты	27296	18.06.2012 15:28:00
Сетевые атаки	1120	13.06.2012 7:10:00

Режим запуска: [Каждый 1 день в 10:00](#)

Последнее обновление: [20.06.2012 10:03:37](#)

[Откат к предыдущим базам](#)

Выполнить обновление

Справка | Поддержка

Обнаружено | Отчеты

Antivirus Касперского 6.0 для Windows Workstations

Kaspersky Anti-Virus 6.0

Настройка

Безопасность компьютера под угрозой Исправить!

Защита

Защита
Антивирус Касперского обеспечивает комплексную защиту компьютера от вирусов, сетевых атак, спама, шпионского ПО и других вредоносных программ.

- ✓ **Файловый Антивирус**
- ✓ **Почтовый Антивирус**
- ✓ **Веб-Антивирус**
- ✓ **Анти-Спам**
- ✗ **Проактивная защита**
Анализ активности
Мониторинг реестра
- ✓ **Анти-Шпион**
Анти-Баннер
Анти-Дозвон
- ✓ **Анти-Хакер**
Сетевой экран
Обнаружение вторжений
- ✓ **Контроль доступа**
Контроль устройств

Всего проверено	1322
Активные угрозы	1
Объекты карантина	0
Объекты хранилища	12
Заблокировано атак	0

Диск аварийного восстановления

Справка | Поддержка

Обнаружено | Отчеты

