

Федеральное агентство по образованию

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ
(ТУСУР)

Кафедра компьютерных систем в управлении
и проектировании (КСУП)

И.Г. Губин

**ТЕХНОЛОГИЯ СОЗДАНИЯ
ИНТЕРНЕТ-ПРИЛОЖЕНИЙ**

Учебное методическое пособие

ТОМСК - 2007

Губин И.Г.

Технология создания интернет-приложений: Учебное методическое пособие. - Томск: 2007. - 110 с.

В учебных методических указаниях подробно и полно излагается технология выполнения лабораторных работ по курсу "Технология создания интернет-приложений". В качестве инструментальных средств используются стандарты интернет-консорциума, такие как HTML и CSS 2.

Отдельная часть лабораторных работ посвящена языкам PHP, JavaScript и их взаимодействию со страницами – динамическому HTML.

Заключительные лабораторные работы выполняются с использованием средств создания БД на основе WEB-сервера Apache, сервера баз данных MySQL и языка структурированных запросов - SQL.

Предназначено для студентов, обучающихся на всех формах обучения.

© Губин И. Г.,
2007

© Кафедра компьютерных
систем в управлении и
проектировании, 2007

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 КРАТКАЯ ПРОГРАММА ЛЕКЦИОННОГО КУРСА.....	5
2 РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	8
3 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ	9
4 ЛАБОРАТОРНАЯ РАБОТА №1	
Создание простой HTML-страницы с гиперссылками.....	9
4.1 Задание на лабораторную работу №1	10
4.2 Отчетность по лабораторной работе.....	10
4.3 Порядок выполнения работы	11
5 ЛАБОРАТОРНАЯ РАБОТА №2	
Каскадные таблицы стилей, элементы уровня блока, фреймы	18
5.1 Задание на лабораторную работу №2	19
5.2 Отчетность по лабораторной работе	19
5.3 Порядок выполнения работы	20
6 ЛАБОРАТОРНАЯ РАБОТА №3	
Создание таблиц, форм	27
6.1 Задание на лабораторную работу №3	27
6.2 Отчетность по лабораторной работе.....	28
6.3 Порядок выполнения работы	28
7 ЛАБОРАТОРНАЯ РАБОТА №4	
Графика, мультимедиа, JavaScript.....	41
7.1 Задание на лабораторную работу №4	41
7.2 Отчетность по лабораторной работе.....	41
7.3 Порядок выполнения работы	42
8 ЛАБОРАТОРНАЯ РАБОТА №5	
Инсталляция программного обеспечения и работа с сервером Apache	47
8.1 Задание на лабораторную работу №5	47
8.2 Отчетность по лабораторной работе.....	48
8.3 Порядок выполнения работы	48
8.4 Общие сведения о пакете "Денвер-2"	48

8.5 Состав комплекса	50
8.6 Настройка компьютера для работы с сетью.....	51
8.7 Пользователю Windows 95	53
8.8 Установка дистрибутива.....	54
8.9 Работа с виртуальными хостами	57
8.10 Настройка Контроллера удаленного доступа	58
8.11 Настройка прокси-сервера.....	59
8.12 Запуск и остановка сервера	59
8.13 Размещение и тестирование своей личной Web- страницы.....	61
9 ЛАБОРАТОРНАЯ РАБОТА №6	
Создание сценария на языке PHP	61
9.1 Задание на лабораторную работу №6	61
9.2 Отчетность по лабораторной работе.....	62
9.3 Создание сценария на языке PHP	62
10 ЛАБОРАТОРНАЯ РАБОТА №7	
Создание базы данных.....	66
10.1 Задание на лабораторную работу №7	67
10.2 Отчетность по лабораторной работе	67
10.3 Создание пользователя и таблиц.....	68
10.4 Создание титульной страницы базы данных	71
10.5 Создание форм для ввода данных в базу.....	74
10.6 Передача данных в базу	77
10.7 Вывод данных на экран.....	82
11 ЛАБОРАТОРНАЯ РАБОТА №8	
Редактирование, поиск, сортировка и вывод данных на печать	87
11.1 Задание на лабораторную работу №8.....	87
11.2 Отчетность по лабораторной работе	87
11.3 Сортировка данных	87
11.5 Удаление записей из БД.....	92
11.6 Редактирование записей в БД.....	93
11.7 Ввод изменённых данных в БД.....	96
11.8 Поиск данных в БД.....	99
11.9 Вывод данных на печать	103
ПРИЛОЖЕНИЕ	110

ВВЕДЕНИЕ

Целью курса "Технология создания интернет-приложений" является изучение основ WEB-технологий, получение практических навыков при создании и опубликовании собственных WEB-страниц и интернет-программированию баз данных на основе использования в Web-страницах инструмента для реализации SQL-запросов для сервера MySQL.

В результате выполнения лабораторных работ должны быть созданы, в соответствии с заданием, WEB-страницы и WEB-сервер на основе сервера Apache, с использованием PHP (Personal Home Page) — популярного языка создания CGI-сценариев (Common Gateway Interface — общий шлюзовой интерфейс). Должны быть получены практические навыки по интернет-программированию баз данных на основе использования в Web-страницах инструмента для реализации SQL-запросов для сервера MySQL.

1 КРАТКАЯ ПРОГРАММА ЛЕКЦИОННОГО КУРСА

1. Введение.
2. Структура HTML-документа. Что такое гипертекстовый документ. Действительные документы HTML. HTML-редакторы. Гиперссылки. Форматирование документа. Синтаксис гипертекстовой разметки. Каскадные таблицы стилей. Типы представления документов. Правила оформления документа. Чего надо стараться избегать. Публикация.
3. Язык HTML 4.01. Структура документа – заголовок и тело. Информация о версии – !DOCTYPE. Элемент HTML. Группирующие элементы – DIV и SPAN. Заголовки. Фреймосодержащие документы. Ссылки. Текст – параграфы и списки. Структурирование текста. Таблицы. Формы. Графика, мультимедиа и внедренные объекты. Расширения Netscape и Microsoft. Таблицы стилей. Встраивание таблиц стилей. Псевдоклассы и псевдоэлементы. Определение классов и идентификаторов. Проблемы совместимости. Атрибуты, общие для большинства элементов.

4. Язык сценариев JavaScript и активные страницы Web. Об активных страницах Web, Java и JavaScript. Внедрение сценариев в документы HTML. Совместимость. Синтаксис языка JavaScript. Алфавит. Операции. Идентификаторы. Данные. Переменные. Литералы. Выражения. Ключевые слова. Комментарии. Операторы. Условные операторы. Операторы циклов. Операторы манипулирования объектами. Функции. Определение и вызов. Массивы аргументов. Предопределенные функции. Объекты, методы и свойства. Создание новых объектов. Некоторые методы объектов document и window. Предопределенные объекты языка JavaScript. Объектная модель документа. Иерархия объектов браузеров Netscape и Microsoft. Доступ к объектам документа из JavaScript-программы. События JavaScript. Всплывание событий и действия по умолчанию. Объект event. JavaScript в действии. Генерация контейнера фреймов при загрузке. Основы составления алгоритмов.
5. Практическая работа с сайтом. Где и как разместить сайт. Какой выбрать сайт – бесплатный или коммерческий. 20 наиболее популярных бесплатных серверов. Планирование сайта. Совместимость, содержание. Подготовка материалов. Стиль оформления. Программа MetaProduct Web Studio Workshop. Создание своего проекта. Разработка титульной страницы. Создание новых файлов. Внесение готовых файлов. Звуковое оформление страниц. Загрузка на сервер. Каталоги и поисковые системы. Счетчики посещений и участие в рейтингах. Способы повышения трафика. Создание корпоративного сайта. Исходные сведения. Создание проекта. Определение стилей. Первая страница. Слайд-шоу. Остальные страницы. Подгонка. Загрузка на сервер.
6. Основы синтаксиса языка PHP. Передача данных в браузер. Тестирование сценария в браузере. Передача простого текста в браузер. Передача страницы HTML в браузер. Использование пробельных символов в PHP и HTML. Добавление комментариев в сценарий.
7. Переменные. Синтаксис переменных. Типы переменных. Присвоение значений переменным. Предопределенные переменные.

8. HTML-формы и PHP. Использование методов Get и Post. Получение данных из формы в PHP. Ввод данных вручную.
9. Числовые переменные. Сложение, вычитание, умножение и деление. Форматирование чисел. Инкремент и декремент. Совместное использование различных операторов. Использование встроенных математических функций.
10. Использование строк. Удаление конечных пробелов. Соединение строк (сцепление, конкатенация). Кодирование и декодирование строк. Шифрование и дешифрование строк. Извлечение части строки.
11. Управляющие структуры. Условный оператор if. Другие операторы. Использование оператора if-else. Использование конструкции if-elseif. Условная конструкция switch. Цикл while. Цикл for.
12. Массивы. Создание массива. Добавление элементов в массив. Доступ к элементам массива. Сортировка массивов. Преобразование строк и массивов. Создание массива в экранной форме. Создание многомерных массивов.
13. Регулярные выражения. Что такое регулярные выражения. Создание простого шаблона. Сопоставление с шаблонами. Создание более сложных шаблонов. Сопоставление с шаблоном и его замена.
14. Создание функций. Создание и использование простых функций. Создание и вызов функций, принимающих аргументы. Создание и использование функций, возвращающих значение. Переменные и функции.
15. Файлы и каталоги. Права доступа к файлам. Запись данных в файл. Чтение файла. Каталоги. Загрузка файла на удаленный компьютер. Переименование и удаление файлов и каталогов.
16. Базы данных. Соединение с сервером и создание базы данных. Создание таблицы. Отправка данных. Извлечение данных. 16. Создание WEB-приложений. Использование функций include и require. Определение даты и времени. Использование HTTP-заголовков. Отправка электронной почты.

17. Отладка сценариев. Распространенные ошибки. Сообщения о возможных ошибках и их протоколирование. Отслеживание ошибок. Использование инструкции die.

2 РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Коржинский С.Н. Настольная книга Web-мастера: эффективное применение HTML, CSS и JavaScript. – 2-е изд., испр. и допол. – М.: Издательский торговый дом "КноРус", 2000. – 320 с.
2. Основы Web-технологий / П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин; под редакцией П.К. Храмцова. – И.: ИНТУИТ.РУ "Интернет-университет информационных технологий", 2003. – 512 с.
3. Айзекс С. Dynamic HTML. Секреты создания интерактивных WEB-страниц. BHV. – Санкт-Петербург, 1998.
4. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство. – М.: Символ, 2002.
5. Ливингстон Д., Браун М. CSS и XHTML Web-профессионалам. ИРМ – Киев, 2001.
6. Ульман Л. MySQL / Ларри Ульман: Пер. с англ. А. Слинкина. — М.: ДМК Пресс; СПб.: Питер, 2004. — 352 с.: ил.
7. Котеров Д.В. Самоучитель PHP 4. — СПб.: БХВ-Петербург, 2003. — 576 с.: ил.
8. Мазуркевич А. PHP: настольная книга программиста / А. Мазуркевич, Д. Еловой.— Мн.: Новое знание, 2003. — 480 с.: ил.
9. Ульман Л. Основы программирования на PHP: Пер. с англ. — М.: ДМК Пресс, 2001. — 288 с.: ил. (Самоучитель).
10. Орлов А.А. PHP: Полезные приёмы. — М.: Горячая линия — Телеком, 2004. — 224 с.: ил.
11. Томсон Лаура. Разработка Web-приложений на PHP и MySQL: Пер. с англ. / Лаура Томсон, Люк Веллинг. — 2-е изд., испр. — СПб: ООО "ДиаСофтЮП ", 2003. — 672 с.

3 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

Лабораторные и контрольные работы выполняются в последовательности, приведенной в методических указаниях по курсу. Перед выполнением лабораторных и контрольных работ необходимо изучить теоретический материал учебного пособия. Перед непосредственным выполнением конкретной работы необходимо детально изучить рекомендуемые разделы в методических указаниях. В результате выполнения всех работ должна быть создана домашняя (персональная) HTML-страница и разработан WEB-сервер, обеспечивающий обработку динамических страниц.

Предусмотрена следующая последовательность выполнения лабораторных работ.

1. Лабораторная работа №1. Создание простой HTML-страницы с гиперссылками.
2. Лабораторная работа №2. Каскадные таблицы стилей, элементы уровня блока, фреймы
3. Лабораторная работа №3. Создание таблиц, форм.
4. Лабораторная работа №4. Графика, мультимедиа, JavaScript.
5. Лабораторная работа №5. Установка программного обеспечения и работа с сервером Apache.
6. Лабораторная работа №6. Создание сценария на языке PHP.
7. Лабораторная работа №7. Создание базы данных.
8. Лабораторная работа №8. Редактирование, поиск, сортировка и вывод данных на печать.

4 ЛАБОРАТОРНАЯ РАБОТА №1

Создание простой HTML-страницы с гиперссылками

Цель работы: практическая работа по созданию простой HTML-страницы с гиперссылками.

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, текстовый редактор Блокнот.

4.1 Задание на лабораторную работу №1

Необходимо создать первую (титульную) страницу с гиперссылками своей будущей личной страницы. В последующих работах эта страница будет видоизменяться и дополняться. В результате выполнения всех лабораторных работ у вас должна быть создана личная (персональная) Web-страница, со всеми необходимыми разделами, иллюстрациями, гиперссылками, таблицами и т.д. В данном учебно-методическом пособии приводятся подробные пошаговые инструкции для достижения этой цели. В качестве приводимой информации на Web-странице вы должны использовать свой личные данные. В случае если вы захотите использовать на своей странице дополнительные элементы сверх обязательных по заданию, то это будет только приветствоваться.

4.2 Отчетность по лабораторной работе

В результате выполнения лабораторной работы должны быть предоставлены следующие материалы.

1. Файл отчета по лабораторной работе, оформленный в соответствии с существующими требованиями, который должен содержать следующие сведения:

1.1. Титульный лист.

1.2. Описание структуры HTML-документа. Информация о версии, типе документа HTML, языке. Детальное описание используемых при выполнении лабораторной работы всех элементов (тегов) языка HTML.

1.3. Детальное описание последовательности выполнения лабораторной работы.

1.4. Необходимые иллюстрации, таблицы, приложения и т.д.

1.5. Выводы.

2. Все файлы разработанной HTML-страницы с исходными текстами (файлы с расширением "html").

Пример оформления отчёта по лабораторной работе №1 приведён в приложении А.

4.3 Порядок выполнения работы

Перед выполнением лабораторной работы изучите главы 1, 2 и 4 учебного пособия "Технология создания интернет-приложений". Изучите раздел 4 данного учебно-методического пособия.

Выполните работу и оформите отчет в соответствии с требованиями к содержанию отчета.

1. Создайте на жёстком диске каталог, в котором вы намерены хранить все свои файлы страницы, например: "C:\MyWeb".
2. Запустите текстовый редактор Блокнот.
3. Введите следующий текст, который будет являться заготовкой для ваших будущих страниц:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE> </TITLE>
  </HEAD>
<BODY>
</BODY>
</HTML>
```

4. Сохраните этот документ под именем "welcome.htm".
5. Запустите любой браузер, например Internet Explorer, и загрузите этот файл. На экране будет пустое окно.

6. Модифицируем этот файл следующим образом:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>Моя личная страничка</TITLE>
  </HEAD>
<BODY>
```

Личная страница студента ТМЦ ДО Семёнова Сергея

Сергеевича. Я рад всех приветствовать на моей страничке. Из неё вы узнаете о моей жизни, моих радостях, огорчениях и увлечениях. Я вам расскажу о своей учёбе и специальности, которую выбрал. Это моя первая в жизни Web-страница, поэтому не судите строго за обнаруженные изъяны. Я надеюсь, что с её помощью я приобрету новых виртуальных знакомых и друзей. Если кто-то захочет связаться со мной, то он может это сделать по этому адресу. С уважением, Сергей!

</BODY>

</HTML>

Фамилия, имя и отчество студента ТМЦ ДО вымышлено.

7. Опять сохраним его на диске и загрузим в браузер. На экране монитора появится изображение, представленное на рисунке 4.1.

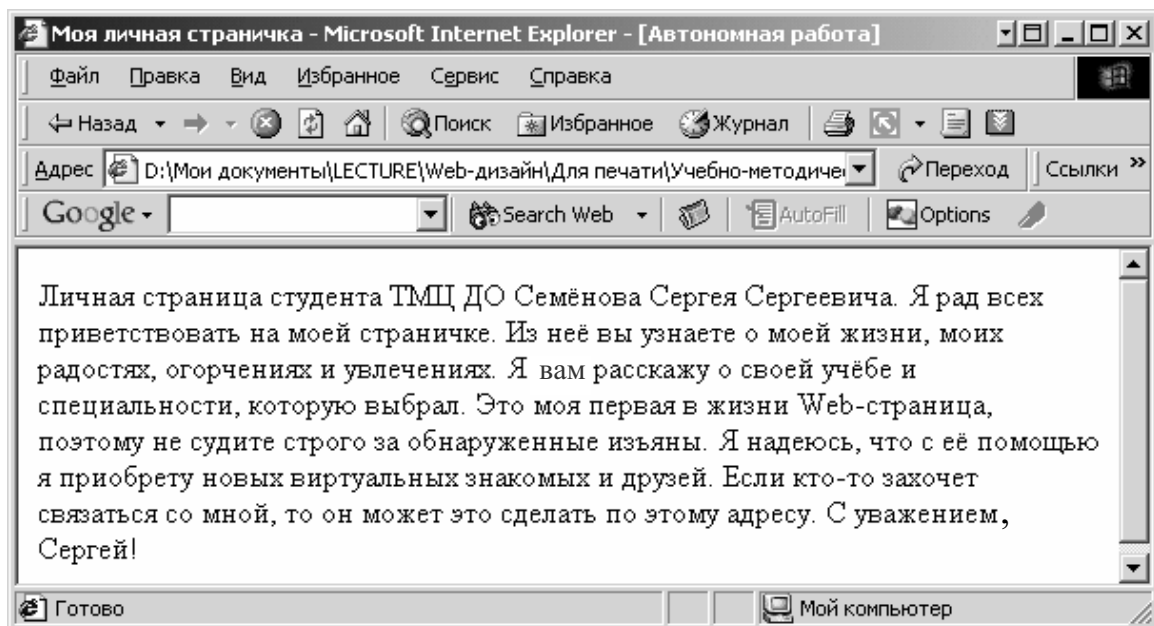


Рис. 4.1 – Простейший HTML-документ в MSIE

Из рисунка 4.1 мы видим, что текст " Моя личная страничка", набранный нами между тегами <TITLE> и </TITLE>, отобразился в заголовке окна браузера, а то, что написано внутри <BODY>, – в самом окне браузера. Сам элемент TITLE помещен внутри контейнера HEAD, который является заголовком документа.

Для более удобной работы, при создании вашей первой страницы с помощью редактора Блокнот можно при изменении файла не выходить из редактора, а просто сохранять файл. Для просмотра в броузере модифицированного варианта нужно нажать кнопку "Обновить".

8. Придадим нашей страничке более привлекательный вид – проведём выделение, выравнивание (центрирование) заголовков и разбивку текста на абзацы.

Для выделения заголовка используем элементы языка "H1...H6", а для абзаца – "P".

Фрагмент файла "welcome.htm" между тегами <BODY> </BODY> примет следующий вид:

```
<<H2>Личная страница студента ТМЦ ДО</H2>
<H1> Семёнова Сергея Сергеевича. </H1>
<P>Я рад всех приветствовать на моей страничке. Из неё вы
узнаете о моей жизни, моих радостях, огорчениях и увлече-
ниях. Я вам расскажу о своей учёбе и
специальности, которую выбрал. Это моя первая в жизни
Web-страница, поэтому не судите строго за обнаруженные
изъяны. Я надеюсь, что с её помощью я приобрету новых
виртуальных знакомых и друзей. Если кто-то захочет
связаться со мной, то он может это сделать по этому адресу.
</P> <P>С уважением, Сергей!</P>
```

Выглядеть это будет примерно так, как изображено на рисунке 4.2. Как видно на рисунке, текст, заключенный в элемент H1, выделен жирным шрифтом и имеет крупный размер. Текст, находящийся в заголовке 2-го уровня (H2), несколько мельче. После каждого заголовка строка обрывается и создается отступ перед следующей строкой.

Абзац, формируемый элементом P, также заставляет браузер создать отступ величиной приблизительно в одну строку до и после текста. Однако, если поставить несколько тегов P подряд, то эффект будет таким же, как и от одного. Поэтому для задания более значительного отступа применяют несколько элементов BR. Он заставляет браузер сделать то, чего вы добиваетесь, нажимая на клавиатуре клавишу Enter при редактиро-

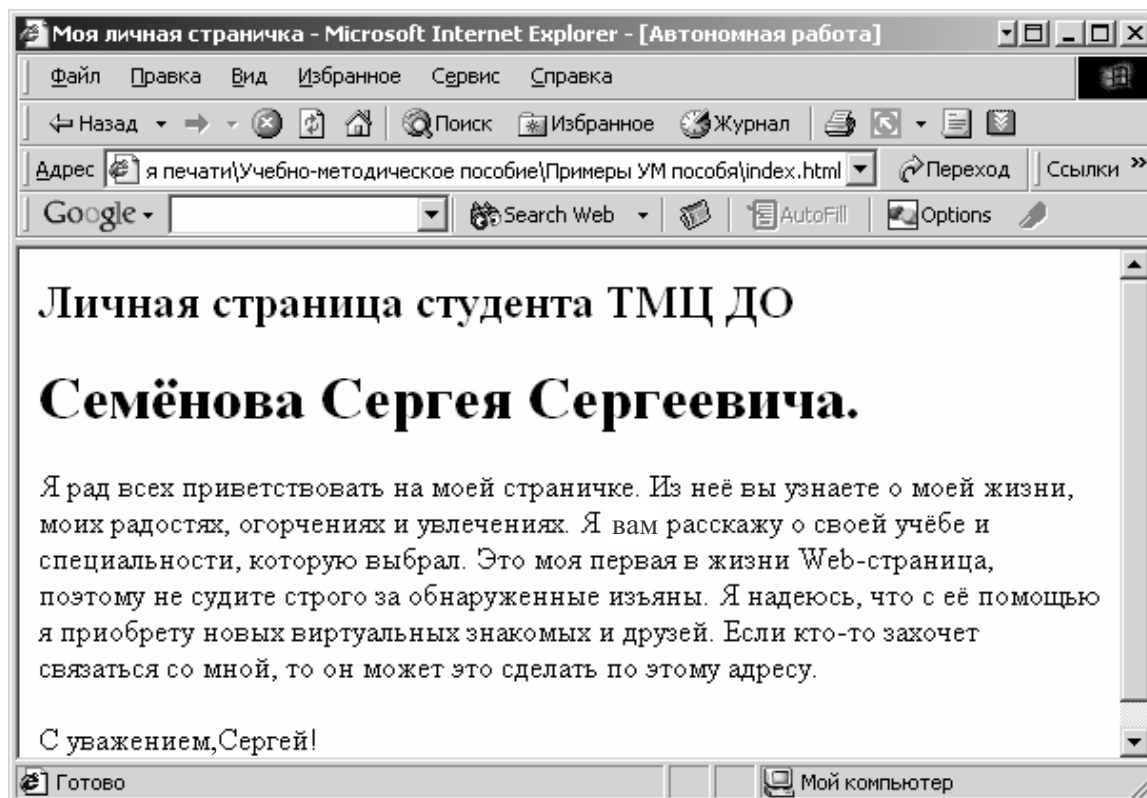


Рис. 4.2 – Заголовки H1 и H2, абзацы P

вании текста в обычном текстовом редакторе. Закрывающий тег `</P>` – необязателен.

Для выравнивания заголовков и абзацев используем атрибут "Align" с нужными параметрами ("Center" и "Justify").

Фрагмент файла "welcome.htm" между тегами `<BODY>` `</BODY>` примет следующий вид:

```
<H2 ALIGN="Center">Личная страница студента ТМЦ
ДО</H2>
<H2 ALIGN="Center">Семёнова Сергея Сергеевича. </H1>
<P ALIGN="Justify">Я рад всех приветствовать на моей
страничке. Из неё вы узнаете о моей жизни, моих радостях,
огорчениях и увлечениях. Я вам расскажу о своей учёбе и
специальности, которую выбрал. Это моя первая в жизни
Web-страница, поэтому не судите строго за обнаруженные
изъяны. Я надеюсь, что с её помощью я приобрету новых
виртуальных знакомых и друзей. Если кто-то захочет
```

связаться со мной, то он может это сделать по этому адресу.
 </P> <P>С уважением, Сергей!</P>

В браузере это будет так, как изображено на рисунке 4.3.

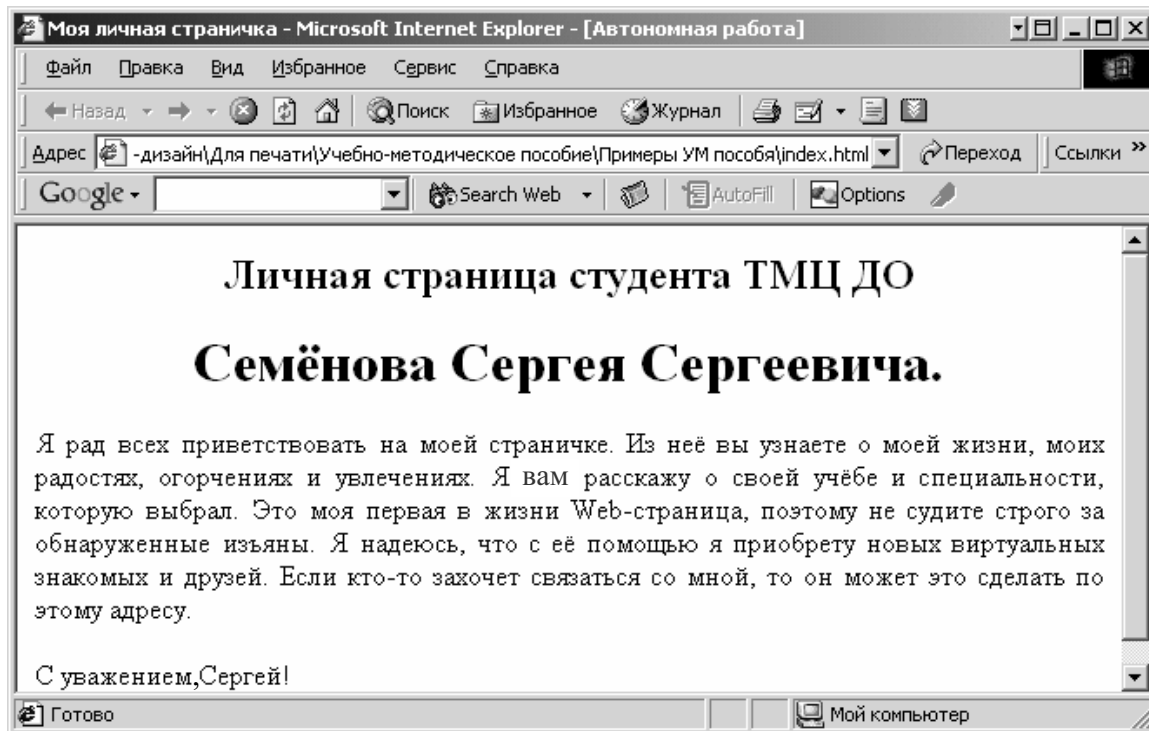


Рис. 4.3 – Выравнивание заголовков и абзацев

Как видно из рисунка 4.3, заголовки расположились по центру, а текст абзаца выровнялся по ширине. Спецификацией языка HTML 4.0 использование этого атрибута для горизонтального выравнивания текста является нежелательным. Вместо него рекомендуется использовать каскадные таблицы стилей (CSS). В последующих лабораторных работах мы вернёмся к нашей главной странице и выполним её в соответствии с этими рекомендациями.

9. Создадим гиперссылки. Сделаем гиперссылки на прочие страницы, которые у нас вскоре появятся, прямо в тексте нашей главной страницы. Допустим, мы хотим сделать ссылки на другие HTML-файлы, которые расположены в том же самом каталоге на диске, что и главная страница. Определим в тексте нашей странички исходные якоря (текстовые выражения), с ко-

торых будут сделаны соответствующие ссылки. Пусть такими выражениями будут следующие: "о моей жизни", "увлечениях", "о своей учёбе", "по этому адресу".

С выражения "о моей жизни" должен быть выполнен переход на файл, содержащий автобиографические сведения. Назовём его "about_me.html". С выражения "увлечениях" – на файл "hobby.html". С выражения "о своей учёбе" – на файл "study.html". С выражения "по этому адресу" – на файл "address.html". Чтобы будущие ссылки корректно выполнялись, создадим в каталоге "C:\MyWeb", с помощью программы Блокнот, указанные файлы ("about_me.html", "study.html", "hobby.html", "address.html").

Файлы будут представлять собой заготовки с пустым наполнением. Отличаться друг от друга они пока будут только названием и информацией в разделе заголовка. Файл "about_me.html" будет иметь следующий вид:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE> Автобиография</TITLE>
  </HEAD>
  <BODY>
</BODY>
</HTML>
```

В файле "study.html" в заголовке напишите "Моя учёба". В файле "hobby.html" – соответственно "Мои увлечения". В файле "address.html" – "Мой адрес".

Создадим гиперссылку с исходного якоря "о моей жизни". Для этого в файле "welcome.html" выполните следующие изменения:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
```



```

<TITLE>Моя личная страничка</TITLE>
</HEAD>
<BODY>

<H2 ALIGN="Center">Личная страница студента ТМЦ
ДО</H2>
<H2 ALIGN="Center">Семёнова Сергея Сергеевича. </H1>
<P ALIGN="Justify">Я рад всех приветствовать на моей
страничке. Из неё вы узнаете <A title="Автобиография"
href="about_me.html">о моей жизни</A>, моих радостях,
огорчениях и увлечениях. Я вам расскажу о своей учёбе и
специальности, которую выбрал. Это моя первая в жизни
Web-страница, поэтому не судите строго за обнаруженные
изъяны. Я надеюсь, что с её помощью я приобрету новых
виртуальных знакомых и друзей. Если кто-то захочет
связаться со мной, то он может это сделать по этому
адресу. </P> <P>С уважением, Сергей!</P>
</BODY>
</HTML>

```

Как видно из приведённого листинга, он отличается от исходного лишь одним выражением:

```

<A TITLE = "Автобиография" HREF="about_me.html">о моей
жизни</a>

```

Здесь мы создали ссылку с помощью элемента "A" (якоря) с собственным атрибутом "HREF" и общим атрибутом "TITLE". Собственный атрибут "HREF" определяет якорь самой ссылки, а общий атрибут "TITLE" – заголовок ссылки. Другими словами, при наведении курсора мыши на эту ссылку атрибут "TITLE" будет отображать подсказку в виде текстовой строки – "Автобиография".

Создайте самостоятельно гиперссылки на остальные сходные якоря. Если вы всё сделаете правильно, то главная страница должна иметь вид, приведённый на рисунке 4.4.

Как видно из рисунка 4.4, все гиперссылки на странице выделены подчеркиванием и цветом. При наведении на них курсо-

ра мыши появляется подсказка. В конкретном случае при наведении курсора на якорь "по этому адресу" – появляется подсказка "Мой адрес". При активизации ссылки происходит переход

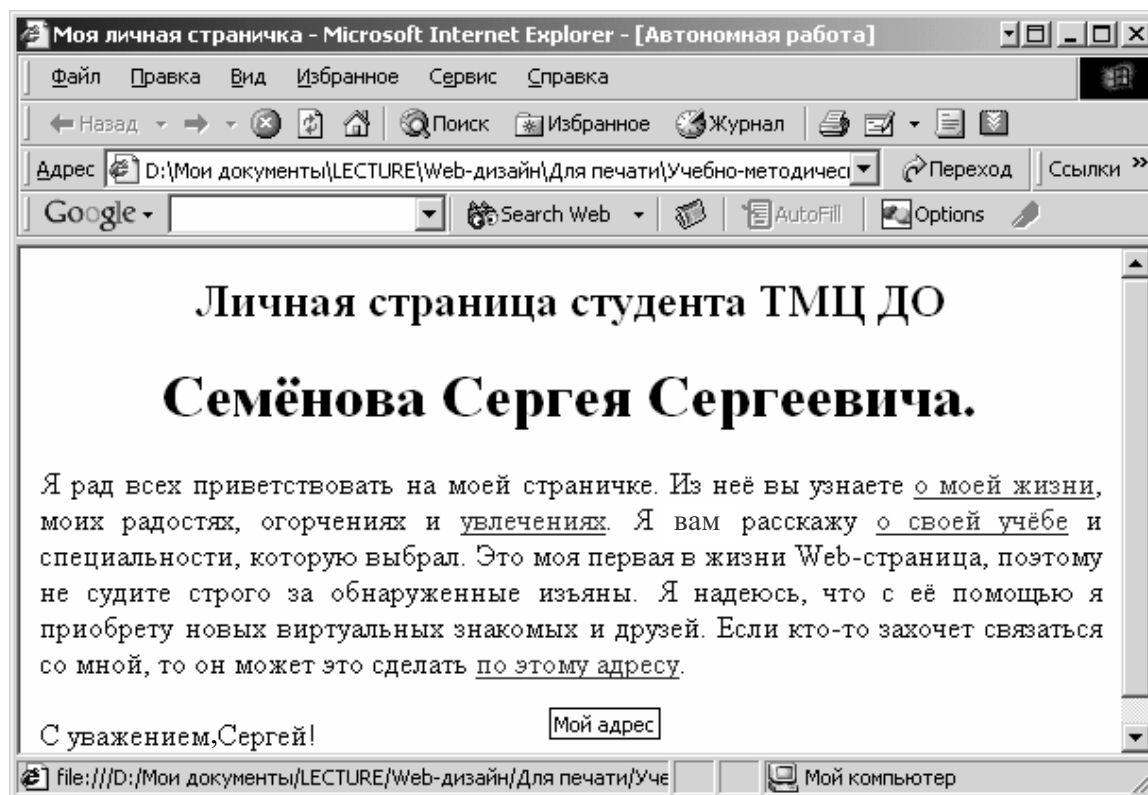


Рис. 4.4 – Страница с гиперссылками

на указанный URI (в нашем случае на Web-страницу).

5 ЛАБОРАТОРНАЯ РАБОТА №2

Каскадные таблицы стилей, элементы уровня блока, фреймы

Цель работы: практическая работа по использованию каскадных таблиц стилей, элементов уровня блока и фреймов.

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, текстовый редактор Блокнот.

5.1 Задание на лабораторную работу №2

Создание по образцу HTML-страниц с использованием каскадных таблиц стилей, находящихся во внешнем файле, элементов уровня блоков (контейнеров) и фреймов.

5.2 Отчетность по лабораторной работе

В результате выполнения лабораторной работы должны быть предоставлены следующие материалы.

1. Файл отчета по лабораторной работе, оформленный в соответствии с существующими требованиями, который должен содержать следующие сведения:

1.1. Титульный лист.

1.2. Детальное описание используемых при выполнении лабораторной работы всех элементов (тегов) языка HTML, свойств CSS.

1.2.1. Способы встраивания таблиц стилей в HTML-документ.

1.2.2. Свойства таблиц стилей "text-align".

1.2.3. Описание подклассов элемента "A" – "link, visited, active, hover".

1.2.4. Описание группирующего элемента "DIV" со всеми его атрибутами.

1.2.5. Описание элементов фреймов: "frameset, frame, noframes" со всеми атрибутами.

1.2.6. Описание элемента "base target".

1.3. Детальное описание последовательности выполнения лабораторной работы.

1.4. Необходимые иллюстрации, таблицы, приложения и т.д.

1.5. Выводы.

2. Все файлы разработанной HTML-страницы с исходными текстами (файлы с расширением "html").

5.3 Порядок выполнения работы

Перед выполнением лабораторной работы изучите главы 1, 2, 4 и Приложение Б учебного пособия "Технология создания интернет-приложений". Изучите раздел 5 данного учебно-методического пособия.

Выполните по приведённой ниже последовательности работы и оформите отчет в соответствии с требованиями к содержанию отчета.

Каскадные таблицы стилей

1. С помощью редактора "Блокнот" создайте на жёстком диске, в том же самом каталоге, где находится файл "welcome.html", текстовый файл "my.css". Этот файл будет являться внешним для хранения стилей CSS.

2. Используя синтаксис языка CSS, поместите в него информацию по оформлению наших страниц следующим образом:

2.1. Заголовки первого уровня H1 должны быть выровнены по центру (center) и иметь цвет 008080.

2.2. Заголовки второго уровня H2 должны быть выровнены по центру и иметь цвет FF0000.

2.3. Абзацы должны быть выровнены по ширине (justify).

2.4. Используя подклассы элемента "A" – ссылки, определите цвет ссылок. Первоначально ссылки (link) должны иметь цвет #0000F2, посещённая ссылка (visited) должна иметь цвет #7F007F, активная ссылка (active – последняя выбранная ссылка до того, как будет выбрана другая ссылка) должна иметь цвет #FF0000, ссылка в момент, когда курсор находится на ней (hover – выбранная), должна иметь цвет #00FF00.

3. Добавим в файл "welcome.html" ссылку на внешний файл таблиц стилей:

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

В отличие от элемента А элемент LINK может располагаться только в части заголовка (HEAD) документа.

4. Удаляем из файла "welcome.html" ненужные уже атрибуты для элементов заголовка "H1...H6" и для абзаца – "P". Файл "welcome.html" примет следующий вид:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>Моя личная страничка</TITLE>
    <link rel="stylesheet" href="my.css" type="text/css" />
  </HEAD>
  <BODY>
    <H2>Личная страница студента ТМЦ ДО</H2>
    <H1>Семёнова Сергея Сергеевича.</H1>
    <P>Я рад всех приветствовать на моей страничке. Из неё вы узнаете <a title="Автобиография" href="about_me.html">о моей жизни</a>, моих радостях, огорчениях и <a title="Мои увлечения" href="hobby.html">увлечениях</a>. Я вам расскажу <a title="Моя учёба" href="study.html">о своей учёбе</a> и специальности, которую выбрал. Это моя первая в жизни Web-страница, поэтому не судите строго за обнаруженные изъяны. Я надеюсь, что с её помощью я приобрету новых виртуальных знакомых и друзей. Если кто-то захочет связаться со мной, то он может это сделать <a title="Мой адрес" href="address.html">по этому адресу</a>. <P>С уважением, Сергей!

  </BODY>
</HTML>
```

5. Добавим на страницу "welcome.html", а затем и во все остальные файлы упоминание о ваших авторских правах. Для этого допишем в самом конце контейнера <BODY> файла "welcome.html", следующую строку:

```
<DIV CLASS="Footer">Copyright &copy; 2005 Сергей  
Семёнов</DIV>
```

Это элемент уровня блока, который применяется для форматирования соответствующей части документа с использованием каскадных таблиц стилей. В нашем случае мы используем класс "Footer".

6. Добавим этот элемент в контейнер `<BODY>` и в файлы: "about_me.html", "study.html", "hobby.html", "address.html".

7. Определим внешний вид добавленного элемента. Допустим, мы хотим, чтобы строка о вашем авторском праве отображалась в серой рамочке толщиной 2 пикселя (`border: solid 2px #808080; color: #808080`), серыми подчеркнутыми буквами размером 10pt и по центру на странице. Используем для этого каскадные таблицы стилей. С помощью блокнота откройте файл "my.css" и допишите следующую строку:

```
DIV.Footer {font-size: 10pt; color: #808080; width: 100%; text-align: center; text-decoration: underline; border: solid 2px #808080;}
```

Сохраните файл.

8. Чтобы выполненное оформление стало доступно для всех страниц, вставьте ссылку

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

в элемент `<HEAD>` файлов "about_me.html", "study.html", "hobby.html", "address.html".

В результате проделанных изменений страница приобретёт вид, приведённый на рисунке 5.1.

Создание фреймов

Создадим на нашей страничке два горизонтальных фрейма — нижний и верхний. В верхнем фрейме будет отображаться созданная страничка "welcome.html", а в нижнем — файл

"topic.html", содержащий ссылки для быстрой навигации по страницам.

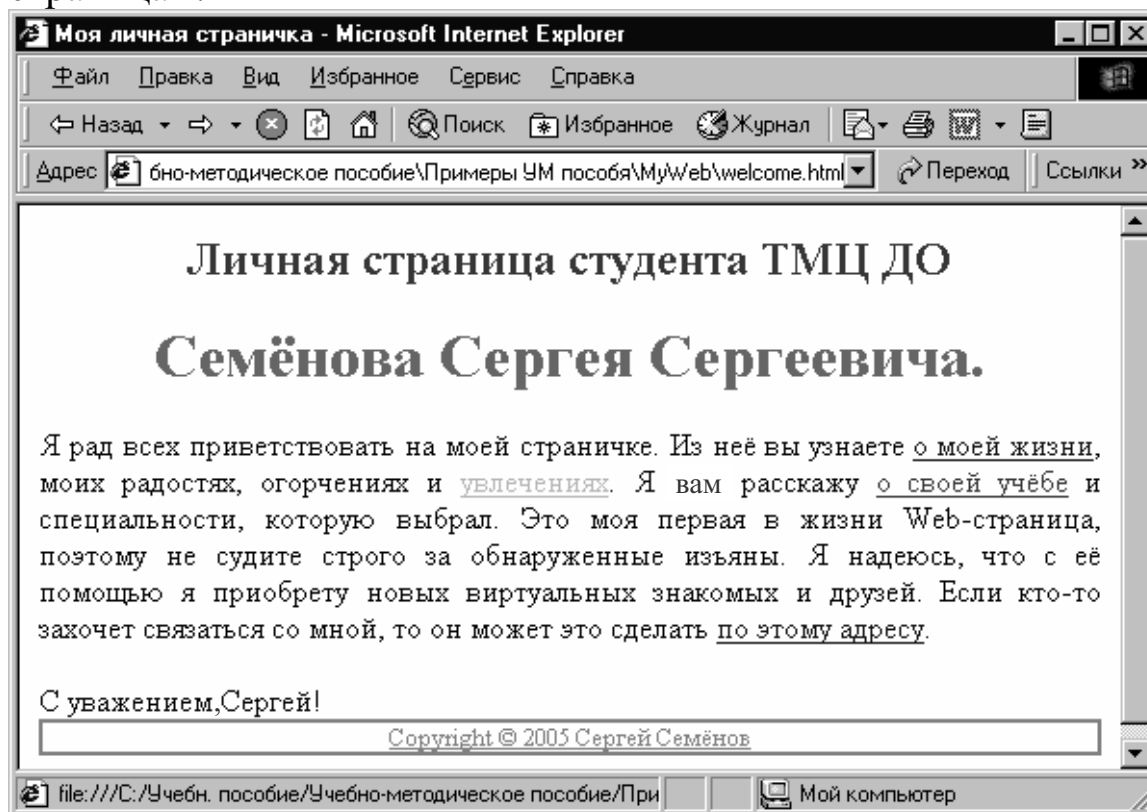


Рис. 5.1 – Изображение файла "welcome.html" после применения каскадных таблиц стилей

1. В редакторе "Блокнот" создадим фреймсодержащий файл "index.html" следующего содержания:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN">
<html>
<head>
<title>MyWeb</title>
</head>
<frameset rows="*,50">
<frame name="welcome" src="welcome.html" scrolling="Auto">
<frame name="topic" src="topic.html" scrolling="No" noresize>
</frameset>
<noframes>
<p>Your browser don't support frames.</p>
```

```
<p>Please, use <a href="http://home.netscape.com">Netscape Navigator</a> (Communicator) version 3.0 or higher,
or <a href="http://www.microsoft.com">MSIE</a> version 3.0 or
higher.</p>
```

```
</noframes>
</html>
```

2. Сохраним этот файл на жёстком диске в том же самом каталоге, где находятся и остальные файлы странички. Проанализируем содержимое этого файла.

В строке, содержащей информацию о версии HTML, указано – "HTML 4.0 Frameset". Следовательно, данный документ является контейнером для набора фреймов.

Элемент "<frameset rows="*,50">" – определяет набор создаваемых фреймов и их расположение в окне браузера. В нашем случае атрибут "rows" определяет набор из двух горизонтальных фреймов. На второй (нижний) фрейм отводится строка высотой 50 пикселей окна, а на первый (верхний) – все оставшееся место.

Элемент "<frame name="welcome" src="welcome.html" scrolling="Auto">". Здесь указывается, что первый фрейм из набора фреймов, заданного элементом FRAMESET, имеет имя "welcome" и должен быть загружен из файла "welcome.html". В дальнейшем имя фрейма может быть использовано для создания ссылок между фреймами (при помощи атрибута TARGET элемента A), а также в сценариях JavaScript.

Атрибут "scrolling="Auto"" – указывает браузеру, можно или нет создавать полосу прокрутки. В нашем случае используется параметр "Auto" (принят по умолчанию), который указывает браузеру на то, что полоса прокрутки должна появляться только в том случае, если документ целиком не помещается во фрейме.

Элемент "<frame name="topic" src="topic.html" scrolling="No" noresize>" – определяет второй фрейм. Отличие от первого фрейма состоит в том, что этот фрейм имеет другое имя, другой загружаемый файл, отсутствует полоса прокрутки

(scrolling="No"), запрещает пользователю изменять размеры фрейма (noresize).

"</frameset>" – закрывающий тег.

Элемент "<noframes>" – указывает браузеру, который не поддерживает фреймы, выводить альтернативное содержание документа-контейнера фреймов или пояснительный текст. В нашем случае таким текстом будет всё содержимое, находящееся внутри элемента "<noframes> ...</noframes>".

1. Создадим файл "topic.html", который будет загружаться в нижний фрейм. Для этого выполним следующие действия.

3.1. В редакторе "Блокнот" создадим файл "topic.html" следующего содержания:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
<head>
  <title>Содержание</title>
  <link rel="stylesheet" href="my.css" type="text/css" />
</head>
<body>
  <base target="welcome">
  <div class="Footer">
    <a href="about_me.html">Автобиография</a>|
    <a href="hobby.html">Мои увлечения</a>|
    <a href="study.html">Моя учёба</a> |
    <a href="address.html">Мой адрес</a> |
    <a href="welcome.html">Главная</a>
  </div>
</body>
</html>
```

3.2. Сохраним этот файл на жёстком диске в том же самом каталоге, где находятся и остальные файлы странички. Проанализируем содержимое этого файла.

Элемент "<link rel="stylesheet" href="my.css" type="text/css"/>" – ссылка на созданный ранее внешний файл

("my.css") каскадной таблицы стилей. Этот файл определяет оформление заголовков, абзацев, ссылок.

Элемент "`<base target="welcome">`" – указывает имя окна назначения для всех ссылок. В нашем случае все ссылки, которые будут активизированы из нижнего фрейма "topic.html", будут открываться в верхнем фрейме с именем "welcome".

Элемент "`<div class="Footer">`"

`Автобиография|`

`Мои увлечения|`

`Моя учёба |`

`Мой адрес |`

`Главная`

`</div>`" – элемент уровня блока, который применяется для

форматирования соответствующей части документа с использованием каскадных таблиц стилей. В нашем случае все перечисленные ссылки в контейнере `<div>` класса "Footer" будут

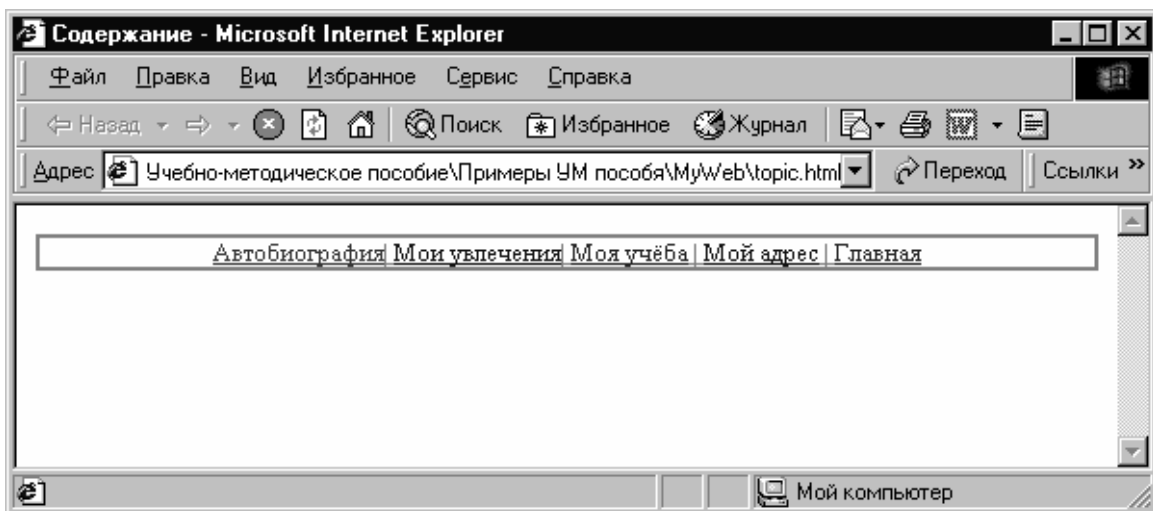


Рис. 5.2 – Изображение нижнего фрейма – файл "topic.html"

оформлены в соответствии со стилем файла "my.css".

Вид файла "topic.html" в MSIE 5.0 приведён на рисунке 5.2. Вид файла "index.html" приведён на рисунке 5.3.

6 ЛАБОРАТОРНАЯ РАБОТА №3

Создание таблиц, форм

Цель работы: практическая работа по созданию таблиц и форм.

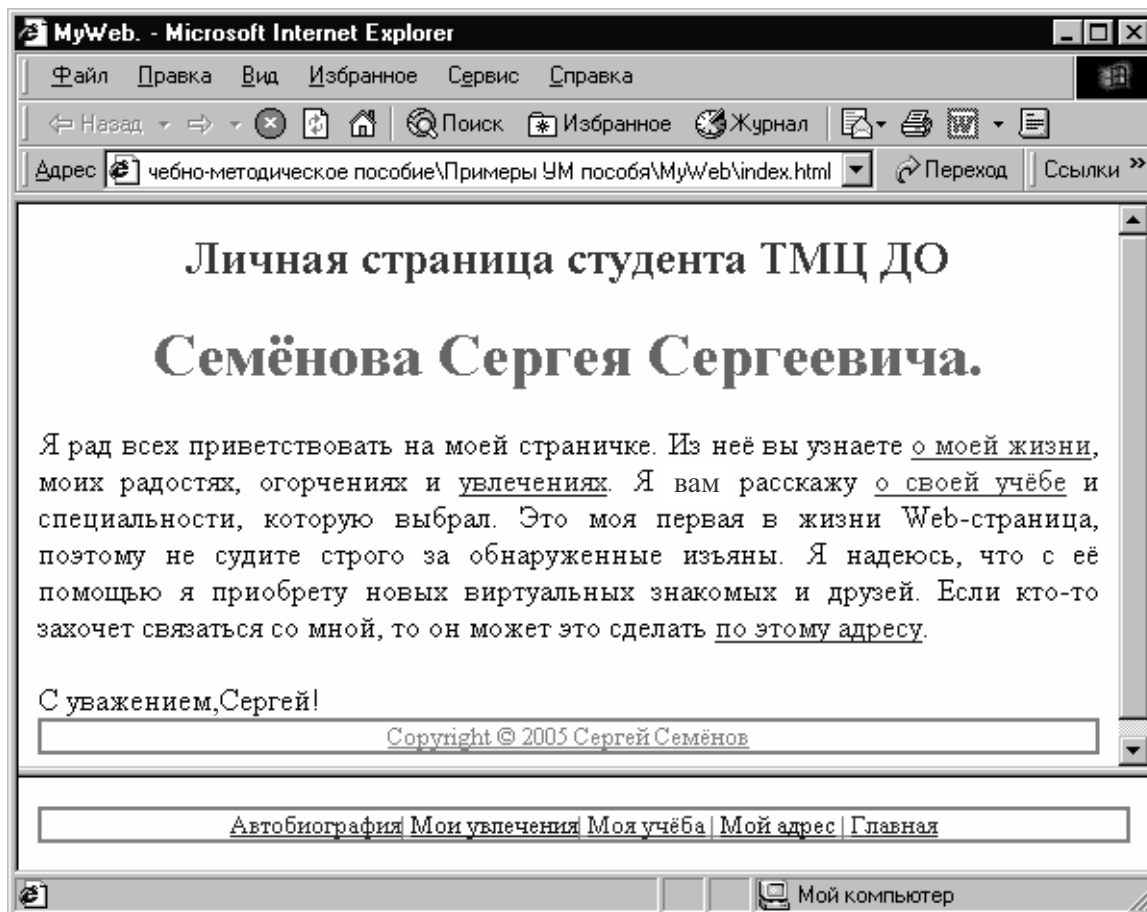


Рис. 5.3 – Изображение фреймсодержащего файла "index.html"

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, текстовый редактор "Блокнот".

6.1 Задание на лабораторную работу №3

Необходимо создать таблицы и формы своей личной страницы. В данном разделе учебно-методического пособия приводятся подробные пошаговые инструкции для достижения этой

цели. В качестве приводимой информации на Web-странице вы должны использовать свой личные данные. В случае, если вы захотите использовать на своей странице дополнительные элементы, сверх обязательных по заданию, то это будет только приветствоваться.

6.2 Отчетность по лабораторной работе

В результате выполнения лабораторной работы должны быть предоставлены следующие материалы.

1. Файл отчета по лабораторной работе, оформленный в соответствии с существующими требованиями, который должен содержать следующие сведения:

1.1. Титульный лист.

1.2. Детальное описание используемых при выполнении лабораторной работы всех элементов (тегов) языка HTML.

1.2.1. Описание элемента TABLE.

1.2.2. Описание элемента CAPTION.

1.2.3. Описание элемента TR.

1.2.4. Описание элементов TH и TD.

1.2.5. Описание элемента COLGROUP.

1.2.6. Описание элемента FORM.

1.2.7. Описание элемента INPUT.

1.2.8. Описание элемента SELECT.

1.2.9. Описание элемента OPTION.

1.2.10. Описание элемента TEXTAREA.

1.2.11. Описание элементов FIELDSET и LEGEND.

1.3. Детальное описание последовательности выполнения лабораторной работы.

1.4. Необходимые иллюстрации, таблицы, приложения и т.д.

1.5. Выводы.

2. Все файлы разработанной HTML-страницы с исходными текстами (файлы с расширением "html").

6.3 Порядок выполнения работы

Перед выполнением лабораторной работы изучите главы 1, 2 и 4 учебного пособия "Технология создания интернет-

приложений". Раздел 1 Изучите раздел 6 данного учебно-методического пособия.

Выполните работу и оформите отчет в соответствии с требованиями к содержанию отчета.

Создание таблиц

Создайте таблицу, содержащую информацию об успеваемости за текущий семестр Вашего обучения в ТМЦ ДО. Таблица должна быть помещена в файл "study.html". В качестве образца оформления таблицы используйте таблицу, приведённую на рисунке 6.1.

1. Проведём краткий анализ нашей таблицы. Таблица имеет несколько строк в заголовке (название таблицы), разных по размеру шрифта и расположению (центрированию). Таблица содержит 30 строк и 6 столбцов. Ячейки-заголовки таблицы имеют жирный шрифт и выровнены по центру. Некоторые поля таблицы имеют различный цвет (заливку). В некоторых строках столбцы объединены. В некоторых столбцах строки объединены. Столбцы таблицы имеют разную ширину. Все эти особенности таблицы необходимо учесть при её гипертекстовой разметке.

Я студент

Успеваемость студента ТМЦДО
Семёнова Сергея Сергеевича

1 семестр

№ п/п	Контроль	Оценка	Дата сдачи	Преподаватель	Аттеста- ция
1	Культурология (Общ)				
	Контрольная работа 1	Зач	18.02.04	Кобзева Л.В.	Зачёт Экзамен
	Контрольная работа 2	Зач	18.02.04	Кобзева Л.В.	
	Зачёт	Зач	18.02.04		
	Экзамен	5	29.03.05		
2	Физика - 1 (Общ)				
	Контрольная работа 1	Зач	27.02.04.	Компьютер	Зачёт Экзамен
	Контрольная работа 2	Зач	27.02.04.	Компьютер	
	Лабораторная работа 1	Зач	23.09.04	Галеева А.И.	
	Лабораторная работа 2	Зач	08.04.04.	Галеева А.И	
	Лабораторная работа 2	Зач	08.04.04.	Галеева А.И	

	Зачёт	Зач	23.09.04.		
	Экзамен	5	23.09.04.		
3	Иностранный язык - 1 (Общ)				
	Контрольная работа 1	Зач	18.02.04	Компьютер	Зачёт Экзамен
	Контрольная работа 2	Зач	19.02.04	Компьютер	
	Контрольная работа 2	Зач	19.02.04	Компьютер	
	Зачёт	Зач	20.02.04		
	Экзамен	3	29.03.04		
4	Информатика - 1 (ПрЭ)				
	Контрольная работа 1	Зач	18.04.04	Дмитриев И.В.	Зачёт Экзамен
	Контрольная работа 1	Зач	18.04.04	Дмитриев И.В.	
	Зачёт	Зач	20.02.04		
	Экзамен	3	29.04.04		
5	Высшая математика - 1 (ПрЭ)				
	Контрольная работа 1	Зач	27.12.04	Магазинников Л.И.	Зачёт Экзамен
	Контрольная работа 1	Зач	28.12.04	Магазинников Л.И.	
	Зачёт	Зач	20.12.04		
	Экзамен	4	29.12.04		

Рис.6.1 – Данные об успеваемости

2. Откройте в редакторе "Блокнот" файл "study.html" и внесите в него следующие изменения.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
<HEAD>
<TITLE>Моя учёба</TITLE>
<link rel="stylesheet" href="my.css" type="text/css" />
</HEAD>
<BODY>
<h2>Я студент!</h2>
```

```
<!--Определяем таблицу -->
```

```
<table width="100%" align="Center" frame="box" border="3">
```

```
<!--Определяем заголовок таблицы -->
```

```
<caption align="Top"><h3 align="Center">Успеваемость студента
ТМЦ ДО</h3><h1>Семёнова Сергея Сергеевича</h1><h3
align="Left">1 семестр</h3></caption>
```

```
<!--Определяем строки таблицы -->
```

```
<!-- Для данной таблицы таких элементов должно быть 30 шт.--
>
```

```
<tr>
<td></td><td></td><td></td><td></td><td></td><td></td>
</tr>
```

```
<tr>
<td></td><td></td><td></td><td></td><td></td><td></td>
</tr>
```

```
<tr>
<td></td><td></td><td></td><td></td><td></td><td></td>
</tr>
```

```
<tr>
<td></td><td></td><td></td><td></td><td></td><td></td>
</tr>
```

```
<tr>
<td></td><td></td><td></td><td></td><td></td><td></td>
</tr>
```

```
<tr>
<td></td><td></td><td></td><td></td><td></td><td></td>
</tr>
```

```
<!--Закрывающий тег для таблицы -->
```

```
</table>
```

```
<DIV CLASS="Footer">Copyright &copy; 2005 Сергей
```

```
Семёнов</DIV></BODY>
</HTML>
```

Остальные строки таблицы необходимо дописать самостоятельно. В приведённом тексте введены следующие дополнения. Добавлен заголовок второго уровня "<h2>Я студент!</h2>".

Применены комментарии, помещённые в угловые скобки <!--...-->. Любые символы внутри комментариев браузером игнорируются. Комментарии используются для документирования программы.

Определена таблица – "<table width="100%" align="Center" frame="box" border="3">" шириной во весь экран (width="100%"), выровненная по центру (align="Center"), с отображением рамок пустых ячеек (frame="box"), с рамкой шириной 3 пикселя (border="3").

Определен заголовок таблицы – "<caption align="Top"><h3 align="Center">Успеваемость студента ТМЦ ДО </h3> <h1> Семёнова Сергея Сергеевича </h1><h3 align="Left">1 семестр</h3></caption>". Заголовок определен элементом "<caption></caption>", имеет верхнее расположение относительно таблицы (align="Top") и состоит из двух заголовков третьего и одного – первого уровня. Последний заголовок выровнен по левому краю (<h3 align="Left">1 семестр</h3>).

Определены строки таблицы – "<tr></tr>" и ячейки столбцов – "<td></td>". В нашем примере мы определили первые шесть строк таблицы с шестью ячейками. Внешний вид таблицы должен быть таким, каким он изображён на рисунке 6.2.

Рис. 6.2 – Внешний вид таблицы на начальном этапе создания

3. Выполним преобразование таблицы в соответствии с образцом, а именно сделаем объединение необходимых столбцов в строках и строк в столбцах. В результате выполненных преобразований мы должны получить таблицу, изображённую на рисунке 6.3.

Рис. 6.3 – Внешний вид таблицы после преобразования ячеек

В браузере таблица должна смотреться примерно так, как она изображена на рисунке 6.4. Исходный текст файла "Study.html" будет иметь вид:

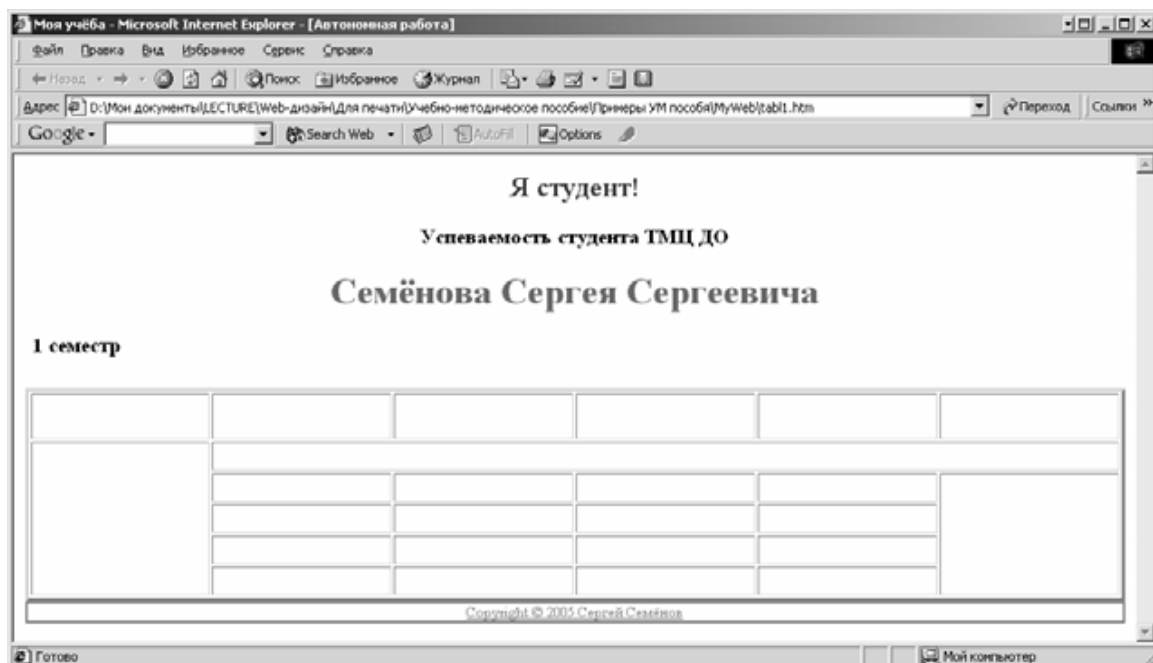


Рис. 6.4 – Внешний вид таблицы в MSIE5

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0
```

```

Transitional//EN">
<HTML>
<HEAD>
<TITLE>Моя учёба</TITLE>
  <link rel="stylesheet" href="my.css" type="text/css" />
</HEAD>
<BODY>
<h2>Я студент!</h2>
<table width="100%" align="Center" frame="box" border="3" >
  <caption align="Top"><h3 align="Center">Успеваемость студента
  ТМЦ  ДО</h3><h1>Семёнова  Сергея  Сергеевича</h1><h3
  align="Left">1 семестр</h3></caption>

  <tr>
    <td height="40"
  </td><td></td><td></td><td></td><td></td><td></td>
  </tr>

  <tr>
    <td rowspan="5" ></td><td height="25" colspan="5"></td>
  </tr>

  <tr>
    <td height="25" ></td><td></td><td></td><td></td><td></td><td
    rowspan="4" ></td>
  </tr>

  <tr>
    <td height="25"></td><td></td><td></td><td></td>
  </tr>

  <tr>
    <td height="25"></td><td></td><td></td><td></td>
  </tr>

  <tr>
    <td height="25"></td><td></td><td></td><td></td>
  </tr>

```

```
</table>
```

```
<DIV CLASS="Footer">Copyright &copy; 2005 Сергей  
Семёнов</DIV></BODY>  
</HTML>
```

Обратите внимание, что в первой строке таблицы присутствуют все шесть элементов ячеек `<td>`. Для второй строки таблицы используется всего лишь два элемента ячейки `<td>` с атрибутами `"rowspan"` и `"colspan"`. Дело в том, что атрибут `"colspan"` определяет протяжённость ячейки в длину, т.е. на сколько столбцов данная ячейка протягивается в длину. В данном случае она распространяется на пять строк, т.к. имеет числовой параметр `"5"`. Иными словами, он описывает все пять ячеек, хотя по расположению находится на месте второй.

Атрибут `"rowspan"` определяет протяжённость ячейки в высоту, т.е. количество строк, на которые ячейка распространяется по вертикали. Так как в нашем случае он имеет параметр `"5"`, то будет описывать пять первых ячеек таблицы в строках 2 – 6.

В третьей строке также используется атрибут `"rowspan"`, но с параметром `"4"`. Этот атрибут объединит шестые ячейки в строках 3 – 6.

Строки 4 – 6 таблицы будут иметь одинаковое описание, состоящее всего из четырёх элементов `"<td></td>"`.

3. Выделим ячейки-заголовки таблицы жирным шрифтом и разным фоном, а также дополним таблицу необходимой текстовой информацией в соответствии с образцом и проведём выравнивание текста в колонках по горизонтали. После проделанных действий фрагмент текста, заключённый между элементом `"<table></table>"`, будет иметь следующий вид.

```
<table width="100%" align="Center" frame="box" border="3">  
<caption align="Top"><h3 align="Center">Успеваемость студента  
ТМЦ ДО</h3><h1>Семёнова Сергея Сергеевича</h1>1  
семестр</h3></caption>  
<colgroup align="Center">  
<colgroup align="Left">  
<colgroup span="4" align="Center" >
```

```

<tr bgcolor="#0099FF">
  <th>№<br>п/п</th><th align="Center">Контроль</th>
  <th>Оценка</th> <th>Дата сдачи </th> <th>Преподаватель</th>
  <th>Аттестация</th>
</tr>

<tr>
  <td rowspan="5">1</td><th colspan="5" align="Center"
  bgcolor="#0099FF">Культурология (Общ)</th>
</tr>

<tr>
  <td>Контрольная работа
  1</td><td>Зач</td><td>18.02.04</td><td>Кобзева Л.В.</td>
  <td rowspan="4">Зачёт <br />Экзамен </td>
</tr>

<tr>
  <td>Контрольная работа 1</td> <td>Зач</td> <td>18.02.04</td>
  <td>Кобзева Л.В.</td>
</tr>

<tr>
  <td>Зачёт</td><td>Зач</td><td>18.02.04</td><td></td>
</tr>

<tr>
  <td>Экзамен</td><td>5</td><td>29.03.05</td><td></td>
</tr>

</table>

```

В браузере страница будет иметь вид, приведённый на рисунке 6.5.

В файле выполнены следующие изменения.

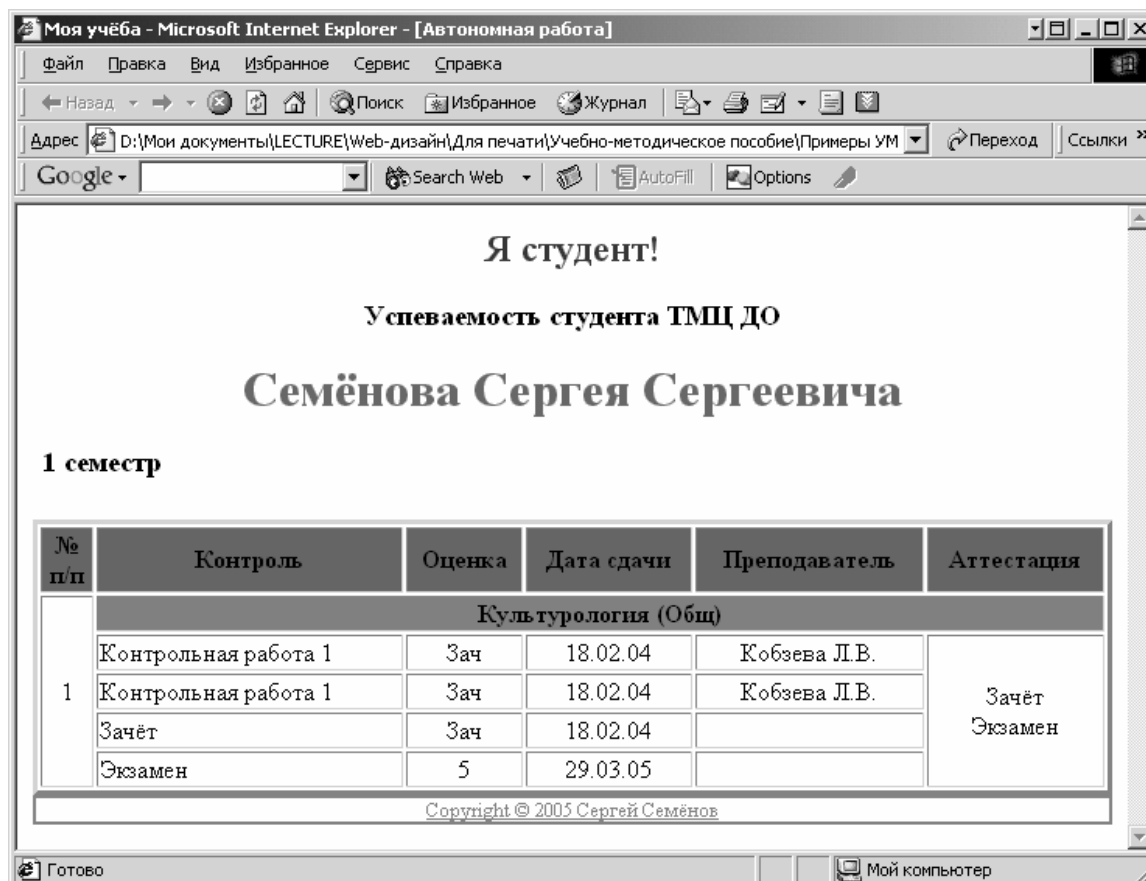


Рис. 6.5 – Окончательный вид фрагмента таблицы

С помощью элемента `<colgroup>` выполнено форматирование (выравнивание) текста в колонках по горизонтали. Первый элемент выравнивает первый столбец таблицы по центру (`<colgroup align="Center">`). Второй элемент выравнивает второй столбец по левому краю (`<colgroup align="Left">`). Третий элемент выравнивает оставшиеся четыре столбца по центру (`<colgroup span="4" align="Center">`). В последнем случае используется атрибут "span" с параметром, равным "4", т.е. его действие будет распространяться на все ячейки четырёх столбцов.

Для задания фона первой строки и второго столбца второй строки используется атрибут "bgcolor" соответственно с параметром "#0099FF" и "#0099FF". В первом случае он находится в теле элемента строки "`<tr>`", и поэтому его действие распространяется на всю строку целиком. Во втором случае он распо-

лагается в элементе ячейки-заголовка "<th>" и определяет только фон этой ячейки.

Для выделения текста ячеек-заголовков первой и второй строки таблицы используется элемент "<th>". Текст, расположенный внутри этого элемента, будет выводиться жирным шрифтом с выравниванием по центру.

Создание форм

Создайте простую форму и поместите её в файле "address.html". Вид в браузере этого файла приведён на рисунке 6.6. Данный файл содержит в себе практически все типы форм. Чтобы эти формы реально могли работать, т.е. отправлять помещённые в них данные, необходимо, чтобы на удалённом сервере были установлены соответствующие программы для их обработки. Такими программами обычно являются CGI (Common Gateway Interface) и ISAPI – приложения. CGI-приложения могут быть написаны на разных языках, например PERL, PHP, C или языке пакетных заданий, и размещены на WWW-сервере. Это позволяет создавать динамические интерактивные страницы. Написание таких серверных приложений выходит за рамки данных лабораторных работ, поэтому мы ограничимся созданием только собственно форм на локальной клиентской машине.

Приведённый на рисунке 6.6 файл содержит в себе восемь форм.

В самом начале своего файла "address.html", сразу после заголовка – "Давайте познакомимся поближе", поместите следующую строку:

```
<FORM action="mailto:foo@somehost.ru" method="post">
```

Эта строка указывает браузеру использовать для отправки формы почтового клиента, а в качестве метода отправки использовать универсальный метод " post ". Хотя в нашем случае практически ничего отправляться не будет и присутствие этой строки ни на что не повлияет.

Для реализации первой формы с заголовком "Мой адрес" используйте элемент ввода текстовой строки – "INPUT" совместно с атрибутом "TYPE="TEXT"".

Для реализации второй формы с заголовком "Представьте, пожалуйста" используйте элементы "FIELDSET", "LEGEND" и "INPUT" совместно с атрибутом "TYPE" и значением "Text". Элементы "FIELDSET" и "LEGEND" позволяют связывать элементы форм в группы, создавать подписи к ним и объединять их рамкой. В нашем случае созданы три группы с

Мой адрес - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Назад Поиск Избранное Медиа

Адрес: C:\DISTANT\WEB\Учебно-методическое пособие\Примеры УМ пособия\MyWeb\ Переход Ссылки

Давайте познакомимся поближе

Мой адрес
 Это мой адрес: 647954, Приобский край, г. Затон, ул. Осетровая, д.13, кв.77

А теперь расскажите немного о себе...

Представьте, пожалуйста

ФИО
 Фамилия: Имя: Отчество:

Возраст
 Какой у вас возраст
☐ Школьный ☐ Студенческий ☐ Средний ☐ Пенсионный ☐ Не имеет значения

Ваше любимое увлечение: Баня с пивом

Введите дополнительную информацию о себе

Дополнительная информация
☐ Состою в браке ☐ Есть судимость ☐ Состою в партии ☐ Имею награды ☐ Бывал за границей

Для ввода сообщения воспользуйтесь окном ниже

Ввод сообщения
 Первая строка текста...
 Вторая строка...

Для отправки заранее подготовленного файла воспользуйтесь окном ниже

Выбор файла
 Обзор...

Отправка или очистка данных

Готово Мой компьютер

Рис. 6.6 – Внешний вид формы файла "address.html" в MSIE5

подписями: "Фамилия: ", "Имя: " и "Отчество:".

Для реализации третьей формы с заголовком "Возраст" так же используйте элементы "FIELDSET", "LEGEND" и "INPUT" совместно с атрибутом "TYPE" и значением "Radio". Такое сочетание позволяет создавать "исключающий переключатель". В нашем случае переключатель может быть установлен в одно из пяти значений: "Школьный", "Студенческий", "Средний", "Пенсионный", "Не имеет значения".

Для реализации четвертой формы с заголовком "Ваше любимое увлечение" используйте элементы "SELECT" и "OPTION", позволяющие создавать раскрывающиеся списки. В нашем случае создан раскрывающийся список из шести строк: "Баня с пивом", "Баня без пива", "Пиво без бани", "Женщины", "Мужчины", "Другое".

Для реализации пятой формы с заголовком "Введите дополнительную информацию о себе" используйте элементы "LEGEND" и "INPUT" совместно с атрибутом "TYPE" и значением "Checkbox". Такое сочетание позволяет создавать "включатель". В нашем случае "включатель" содержит пять пунктов: "Состою в браке", "Есть судимость", "Состою в партии", "Имею награды", "Бывал за границей".

Для реализации шестой формы с заголовком "Для ввода сообщения воспользуйтесь окном ниже" используйте элементы "LEGEND" и "TEXTAREA".

Для реализации седьмой формы с заголовком "Для отправки заранее подготовленного файла воспользуйтесь окном ниже" используйте элементы "LEGEND" и "INPUT" совместно с атрибутом "TYPE" и значением "File". Такое сочетание позволяет создавать элемент для выбора локальных файлов.

Для реализации восьмой формы с заголовком "Отправка или очистка данных" используйте элементы "LEGEND" и "INPUT" совместно с атрибутом "TYPE" и значениями "SUBMIT" и "RESET". Такое сочетание позволяет создавать кнопки "отправить" и "очистить".

7 ЛАБОРАТОРНАЯ РАБОТА №4

Графика, мультимедиа, JavaScript

Цель работы: практическая работа по внедрению графики, звука и сценариев JavaScript.

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, текстовый редактор "Блокнот".

7.1 Задание на лабораторную работу №4

Необходимо дополнить страницы графическими изображениями, звуковым сопровождением и сценарием на языке JavaScript. В данном разделе учебно-методического пособия приводятся подробные пошаговые инструкции для достижения этой цели. В качестве приводимой информации на Web-странице вы должны использовать свои личные данные. В случае, если вы захотите использовать на своей странице дополнительные элементы сверх обязательных по заданию, то это будет только приветствоваться.

7.2 Отчетность по лабораторной работе

В результате выполнения лабораторной работы должны быть предоставлены следующие материалы.

1. Файл отчета по лабораторной работе, оформленный в соответствии с существующими требованиями, который должен содержать следующие сведения:

1.1. Титульный лист.

1.2. Детальное описание используемых при выполнении лабораторной работы всех элементов (тегов) языка HTML.

1.2.1. Описание элемента BODY.

1.2.2. Описание элемента IMG.

1.2.3. Описание элемента EMBED.

1.2.4. Описание элементов SCRIPT.

1.2.5. Описание объекта Function.

1.3. Детальное описание последовательности выполнения лабораторной работы.

1.4. Необходимые иллюстрации, таблицы, приложения и т.д.

1.5. Выводы.

2. Все файлы разработанной HTML-страницы с исходными текстами (файлы с расширением "html").

7.3 Порядок выполнения работы

Перед выполнением лабораторной работы изучите главы 2, 3 и 4 учебного пособия "Технология создания интернет-приложений". Раздел 1. Изучите раздел 7 данного учебно-методического пособия.

Выполните работу и оформите отчет в соответствии с требованиями к содержанию отчета.

Внедрение в HTML-страницы фона и графики

1. Установите цвет фона. Для того чтобы он был одинаковым для всех страничек, вставьте в начало файла таблиц стилей "my.css" строку:

```
BODY { background: #A7D3DA color: #000000;}
```

В этом выражении атрибут "background" с параметром "#A7D3DA" определяет фон страничек, а атрибут "color" – цвет шрифта (в нашем случае он чёрный, что принято по умолчанию).

2. Разместите на страничках фоновый рисунок. В качестве рисунка для фона можете использовать любой из каталога "Media\Images-Texture\". В нашем случае неплохо подойдет рисунок "blumrbld.jpg". Дополним только что введенную строку в файле "my.css" таким образом, чтобы она приняла следующий вид:

```
BODY { background: #A7D3DA fixed URL("blumrbld.jpg ")
left repeat-y; margin-left:150px; margin-right: 10px;color:
#000000;}
```

В этом выражении мы в качестве фона используем рисунок "blumrbld.jpg". Чтобы браузер нашел этот файл, необходимо ли-

бо скопировать его в каталог, где находятся остальные файлы, либо указать полный путь доступа к файлу. Параметр "Fixed" предписывает фоновому рисунку оставаться неподвижным в окне навигатора при просмотре документа. Параметр "left repeat-y" определяет левое расположение фонового рисунка в окне. Параметры "margin-left" и "margin-right" сдвигают текст соответственно слева и справа на определённое число пикселей.

3. Разместите на титульной странице "welcome.html" свою фотографию. Для этого отсканируйте её в соответствии с рекомендациями, приведёнными в разделе 4.2.3 учебного пособия "Технология создания интернет-приложений". Если со сканированием возникли непреодолимые проблемы, то вместо своей фотографии используйте любой другой рисунок.

Для размещения на странице фотографии дополните файл "welcome.html" следующим выражением:

```
<IMG SRC="photo_me.jpg" HEIGHT="200" WIDTH="200"
ALIGN="LEFT" ALT="Студент ТМЦДО Сергей Семёнов, г. За-
тон" />
```

Эту строку необходимо вставить сразу после заголовка "<H1>Семёнова Сергея Сергеевича</H1>".

В этом выражении применяется элемент "IMG" с обязательным атрибутом "SRC", который предписывает браузеру использовать фотографию с именем "photo_me.jpg". Данная фотография взята с личной страницы студента ТМЦ ДО Алексея Шевченко, г. Райчихинск. Его и другие личные Web-страницы студентов вы можете посмотреть на сайте ТМЦ ДО. Чтобы браузер нашел этот файл, необходимо, как и в предыдущем примере, либо скопировать его в каталог, где находятся остальные файлы, либо указать полный путь доступа к нему.

Собственные атрибуты "HEIGHT" и "WIDTH" с параметрами "200" задают соответственно высоту и ширину фотографии в пикселях.

Собственный атрибут "ALIGN" со значением "LEFT" указывает на расположение рисунка относительно обтекающего его текста. В данном случае изображение находится слева, текст обтекает его по правой границе.

Общий атрибут "ALT" содержит краткое описание рисунка, которое появляется на экране при наведении на него указателя мыши. В нашем случае появляется надпись "Студент ТМЦДО Сергей Семёнов, г. Затон".

Внедрение в HTML-страницы звукового сопровождения

Дополните свою страничку фоновым звуковым сопровождением. Для этого вставьте в файл "welcome.html" следующее выражение:

```
<EMBED SRC="MoneyMoney.mid" HIDDEN="False"
TYPE="audio/midi"></EMBED>
```

Эту строку можно вставить в любом месте файла. Если ваш компьютер снабжён аудиосистемой, то сразу же после загрузки файла должна воспроизводиться указанная мелодия.

В этом выражении используется элемент "EMBED" с собственным атрибутом "SRC", который предписывает браузеру использовать звуковой файл с именем "MoneyMoney.mid". Чтобы браузер нашел этот файл, необходимо, как и в предыдущих примерах, либо скопировать его в каталог, где находятся остальные файлы, либо указать полный путь доступа к нему. В качестве звукового файла можете использовать любой из каталога "Media\Sound-MIDI" компакт-диска.

Собственный атрибут "HIDDEN" со значением "False" указывает браузеру отображать на странице объект, который должен использоваться для воспроизведения указанного типа файла. В нашем случае на странице будет отображаться панель управления проигрывателя Windows Media, который установлен в качестве устройства по умолчанию для воспроизведения данного типа файлов. Если указать значение "True", то панель управления проигрывателя будет скрыта, но звуковое сопровождение всё равно будет происходить.

Собственный атрибут "TYPE" со значением "audio/midi" указывает на тип данных, URI которых определен в атрибуте SRC. Типы данных перечислены в "Приложении А". В нашем случае это звуковой файл формата "midi".

Окончательный вид файла "welcome.html" в MSIE5 будет соответствовать рисунку 6.7.

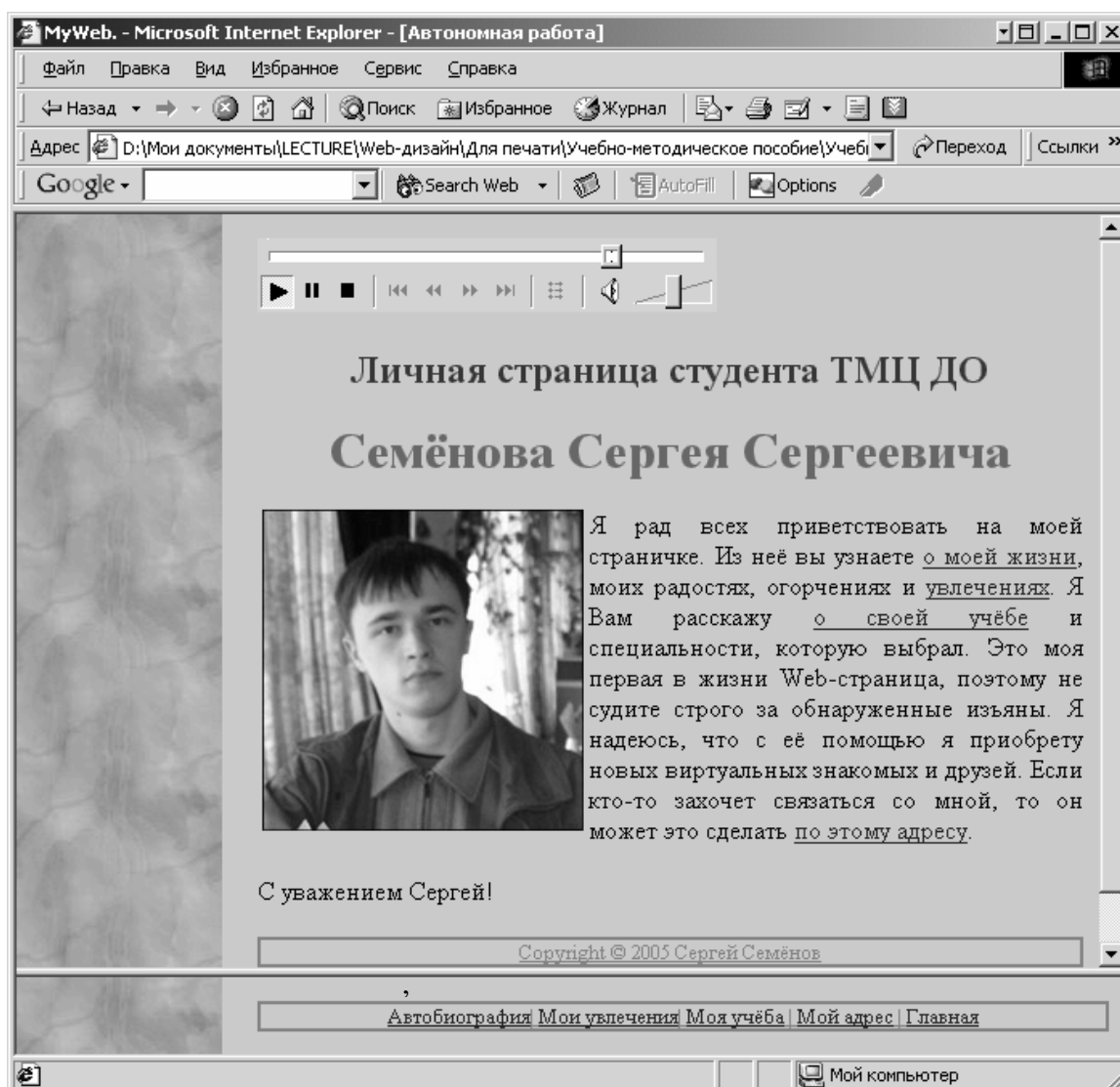


Рис. 6.7 – Окончательный вид файла "welcome.html" в MSIE5

Внедрение в HTML-страницы сценариев JavaScript

Поместите сценарий JavaScript из примера раздела 3.1.2 методических указаний на страницу "hobby.html". В этом примере создаётся приветствие, изменяющееся в зависимости от времени суток. Исходный текст файла "hobby.html" должен иметь следующее содержание.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
<HEAD>
<TITLE>Мои увлечения</TITLE>
  <link rel="stylesheet" href="my.css" type="text/css"/>
<SCRIPT LANGUAGE="JavaScript"><!--
function gethello() {
var nowdate = new Date();
var nowtime = nowdate.getHours();
var hellostr = ' ';
if (nowtime>=0 && nowtime<6) {hellostr="<h1>Доброй ночи!
</h1>";}
if (nowtime>=6 && nowtime<11) {hellostr="<h1>С добрым
утром! </h1>";}
if (nowtime>=11 && nowtime<18) {hellostr="<h1>Добрый
день!</h1>";}
if (nowtime>=18) {hellostr="<h1>Добрый вечер!</h1>";}
return hellostr;
}
//--></SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript"><!--
document.write(gethello());
//--></SCRIPT>
  <DIV CLASS="Footer">Copyright &copy; 2005 Сергей
Семёнов</DIV>
</BODY>
</HTML>

```

В отличие от примера раздела 3.1.2 в этом файле выполнено форматирование выводимых приветствий с помощью заголовков первого уровня.

Вид в браузере MSIE5 всех созданных файлов вы можете найти на компакт-диске в каталоге "WEB-дизайн\Лабораторные".

Окончательное заполнение страниц "hobby.html" и "about_me.html "

Дополните файлы "hobby.html" и "about_me.html" своей личной информацией. В файл "hobby.html" поместите по своему усмотрению сведения о своих увлечениях. В файл "about_me.html" – сведения о себе. Постарайтесь сделать эти страницы интересными и красивыми.

8 ЛАБОРАТОРНАЯ РАБОТА №5

Инсталляция программного обеспечения и работа с сервером Apache

Цель работы: практическая работа по инсталляции сервера Apache.

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, комплекс программ — "Денвер-2", который позволяет запустить полноценный веб-сервер Apache с поддержкой PHP, Perl и сервер MySQL на компьютере, работающем под управлением операционной системы MS Windows 95/98/Me/NT/2000/XP.

8.1 Задание на лабораторную работу №5

1. Изучить и установить пакет ПО Денвер-2 на компьютере, работающем под управлением ОС MS Windows 95/98/Me/NT/2000/XP.

2. Разместить и протестировать свою личную Web-страничку, созданную при выполнении предыдущих четырёх лабораторных работ, в структуре каталогов сервера Apache (каталог:

"C:\WebServers\home\localhost\www").

8.2 Отчетность по лабораторной работе

В результате выполнения лабораторной работы должны быть предоставлены следующие материалы.

1. Текстовый файл отчета по лабораторной работе, оформленный в соответствии с существующими требованиями, который должен содержать следующие сведения:

1.1. Титульный лист.

1.1. Детальное описание последовательности всех шагов по установке сервера Apache.

1.2. Работоспособный сервер Apache, содержащий в своей структуре каталогов личные WEB-страницы.

8.3 Порядок выполнения работы

Перед выполнением лабораторной работы изучите раздел 7 данного учебно-методического пособия.

Выполните работу и оформите отчет в соответствии с требованиями к содержанию отчета.

8.4 Общие сведения о пакете "Денвер-2"

При выполнении лабораторной работы используется пакет "Денвер-2" версии — Денвер-2 2005-10-07. Данный пакет находится на ftp-сервере кафедры в каталоге .../Денвер-2 (саморазархивирующийся файл "Base_2005-10-07_a1.3.31_p4.4.0_m4.1.8_pma 2.6.1.exe").

Последнюю версию можно загрузить с сервера:

"http://denwer.ru" или "http://www.denwer.ru/dis/Base"

Предлагаемый комплекс программ позволяет запустить полноценный веб-сервер Apache с поддержкой PHP, Perl и сервер MySQL на машине, работающей под управлением MS Windows 95/98/Me/NT/2000/XP. Как правило, при установке не требуется никакой настройки, комплекс может использоваться даже неподготовленными пользователями. Тем не менее он может найти применение не только для обучения азам веб-программирования, но и для отладки интерактивных сайтов,

программ, использующих базу данных MySQL, написанных на языках PHP и Perl.

Основные отличия комплекса от аналогов следующие.

1. Модульность и расширяемость. Нет необходимости скачивать многомегабайтные дистрибутивы отдельных компонентов. Базовая версия предлагаемого комплекса Apache+PHP+Perl+MySQL имеет размер всего около 3.1 Мб и при этом полностью функциональна. В случае необходимости Вы можете докачать с сайта разработчиков дополнительные комплекты, содержащие интерпретатор Perl с различными модулями, библиотеки, расширяющие возможности PHP и обеспечивающие работу с архивами, графикой, базами данных, отличными от MySQL, и т.д.

2. Компоненты комплекса уже сконфигурированы для работы. Конечно, для оптимизации работы и включения дополнительных возможностей необходимо редактировать конфигурационные файлы, но базовые возможности доступны сразу же после установки! В состав пакета включен также установщик, который значительно упрощает процедуру конфигурирования при "переезде" пакета в другую директорию.

3. Программы, входящие в состав комплекса, написаны различными разработчиками. Их особенности иногда могут затруднить совместное функционирование этих продуктов. Возможны также принципиальные различия различных версий одной и той же программы, что зачастую затрудняет установку и настройку. Компоненты предлагаемого пакета конфигурировались и тестировались для обеспечения полноценной работы в составе комплекса.

4. PHP работает в качестве модуля веб-сервера Apache, что позволяет отлаживать программы авторизации пользователей и открывает доступ к ряду дополнительных возможностей. Необходимо отметить, что сторонними разработчиками чаще всего предлагается работать с интерпретатором PHP как CGI-программой, что несколько ограничивает возможности. Пакет "Денвер-2" предусматривает такую возможность, но штатным режимом все же является режим функционирования как модуля Apache, что наиболее распространено на платформе UNIX.

8.5 Состав комплекса

Базовый комплект имеет следующий состав:

1. Сервер Apache: выполняемые файлы, дистрибутивные и адаптированные конфигурационные файлы.

2. Язык сценариев PHP: выполняемые файлы, модуль для веб-сервера Apache, дистрибутивный и адаптированный конфигурационный файл, библиотека GD — пока без поддержки формата GIF.

3. Сервер баз данных MySQL: выполняемые файлы, файлы сообщений об ошибках на русском и английском, база данных mysql.

4. Панель управления базой данных MySQL — phpMyAdmin, а также скрипт, упрощающий добавление нового пользователя MySQL.

5. Язык Perl: выполняемые файлы, модули отсутствуют.

6. Отладочная "заглушка" для sendmail (/usr/sbin/sendmail), не отправляющая письма, а записывающая их в директорию /tmp/!sendmail.

7. Система автоматического поиска виртуальных хостов и обновления системного файла hosts, а также конфигурации сервера Apache. Благодаря ей добавление нового виртуального хоста (или домена третьего уровня) заключается в простом создании каталога в /home (см. по аналогии с уже существующими хостами) и перезапуске комплекса. Все изменения вносятся в конфигурационные и системные файлы автоматически, но вы можете управлять этим процессом при помощи механизма шаблонов хостов (см. /usr/local/apache/conf/httpd.conf для детальных разъяснений).

На официальном сайте доступны (либо будут доступны) следующие дополнения, расширяющие возможности базового комплекта:

- 1) PHP-версии 3 в виде CGI-программы;
- 2) PHP-версии 5 в виде CGI-программы;
- 3) модули и документация для Apache;
- 4) модули и документация для PHP;
- 5) модули и документация Perl;
- 6) интерпретатор ActivePython.

- 7) сервер MySQL с поддержкой транзакций, документация;
- 8) модули поддержки технологии Parser;
- 9) другие популярные модули.

8.6 Настройка компьютера для работы с сетью

Начиная с версии Денвер-2, все дистрибутивы поставляются в виде самораспаковывающихся архивов 7-zip (<http://www.7-zip.org>), содержащих также и автоматически запускаемый инсталлятор. Из-за особенностей некоторых версий Windows в случае ручной распаковки рекомендуется выбирать имя временной папки не очень длинным (длина пути должна быть в пределах 10—20 символов). Иначе инсталлятор может не запуститься.

Перед установкой сервера необходимо убедиться, что ваш компьютер действительно настроен для работы с сетью. Обычно этот режим устанавливается при инсталляции операционной системы по умолчанию. Многие ассоциируют слово "сеть" с Интернетом, локальной сетью или хотя бы модемом. И совершенно напрасно. Фраза "настроен для работы с сетью" может иметь смысл даже в том случае, когда ни одного из перечисленных устройств у компьютера нет! Здесь имеется в виду лишь установка драйверов и сетевых протоколов, которые позволят Apache запуститься и работать на локальной машине. Для проверки откройте меню "Пуск" — "Выполнить" и введите там команду: `ping 127.0.0.1`. Окно ввода этой команды показано на рисунке 8.1. После нажатия клавиши "Enter" вы должны увидеть примерно картину, представленную на рисунке 8.2. Процесс продолжается несколько секунд. Если вы это видите, то все в порядке и можно приступать к инсталляции дистрибутива. Если же, например, окно лишь "мигнет" (откроется и тут же закроется) либо же будут выведены какие-нибудь другие сообщения, значит, сетевые протоколы у вас не установлены и придётся их устанавливать вручную.

Для этого пользователям Windows 98/ME необходимо зайти в меню "Пуск" — "Настройка" — "Панель управления" — "Установка и удаление программ" — "Установка Windows" и добавить там "Удаленный доступ к сети". При этом автоматически установятся и все нужные протоколы.

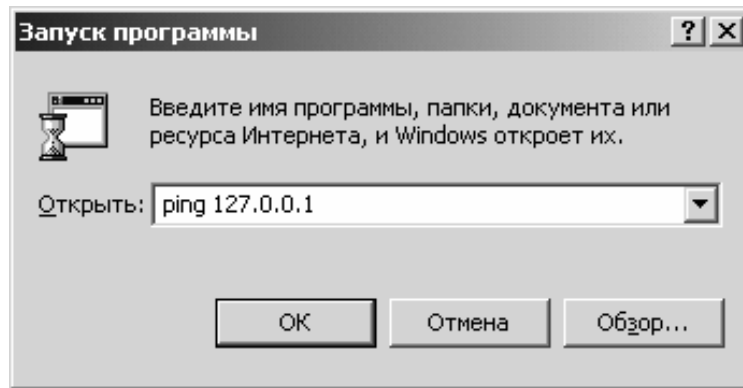


Рис. 8.1 — Окно ввода команды для проверки работы с сетью

Если вы являетесь пользователем ОС Windows 95, то перед установкой вам необходимо будет проделать ещё и некоторые дополнительные шаги, которые описаны в следующем разделе.

Пользователям Windows 95 необходимо открыть "Панель управления" — "Сеть" и добавить там "Протокол TCP/IP" (от Microsoft). Также, возможно, потребуется добавить и "Клиент для сети Microsoft". В этом же окне из выпадающего списка необходимо выбрать "Способ входа в сеть" значение "Обычный вход в Windows".

После перезагрузки повторите ввод команды "ping" снова. Если адрес по-прежнему недоступен, значит, протоколы так и не установились, а вам необходимо проделать все сначала. Если же "ping" пошел — все отлично!



Рис. 8.2 — Результат выполнения команды "ping"

8.7 Пользователю Windows 95

Для настройки компьютера, работающего под операционной системой Windows 95, необходимо скачать и установить несколько дополнительных утилит, предоставляемых компанией Microsoft.

Для начала нужно установить драйвер WinSock2, доступный по адресу <http://www.denwer.ru/dis/other/W95ws2setup.exe> (или на сайте Microsoft). Он необходим для работы всех компонентов (характерный признак отсутствия драйвера — запись в файле журнала Apache "WSASocket failed to open the inherited socket"). Этот драйвер также входит в комплект Win95 Service Pack 1.

Затем придется установить еще несколько драйверов. Все они поставляются вместе с Microsoft Internet Explorer версий 5.0 и выше. Однако если вы не хотите устанавливать этот браузер (пожалуй, для слабых машин он тяжеловат), скачайте драйверы вручную, воспользовавшись следующими советами. Во-первых, необходимо установить Microsoft Office любой версии. Дело в том, что работа с PHP и MySQL подразумевает наличие драйверов ODBC в системе, которые, в частности, ставятся вместе с Офисом. Второй способ — скопировать с машины, на которой установлен Office, все файлы из каталога C:\Windows\System\odbc*.*, это тоже работает.

Во-вторых, из-за того, что в Windows 95 по умолчанию идет старая версия библиотеки ole32.dll, PHP не работает. Как написано на сайте разработчиков, для установки новой версии библиотеки вам придется скачать обновление DCOM с сайта <http://microsoft.com/> или вот отсюда:

"<http://www.denwer.ru/dis/other/dcom95.exe>".

В-третьих, Perl и Apache отказываются работать, если в папке C:\Windows\System нет файла msvcr7.dll (а его там по умолчанию нет). Его можно скачать и скопировать в положенное из следующего ресурса:

"<http://www.denwer.ru/dis/other/msvcr7.dll>".

8.8 Установка дистрибутива

Запускаем самораспаковывающийся архив Денвера-2 — "Base_2005-10-7_a1.3.31_p5.0.4._m4.1.8_rma2.6.1.exe". Вначале архив будет распакован во временную директорию (нужно немного подождать), а затем автоматически запустится инсталлятор.

Инсталлятор написан на языке Perl. Это, однако, не означает, что вам нужно знать что-то об этом языке: все необходимые компоненты уже входят в дистрибутив, а чтобы программа могла запуститься, она имеет расширение BAT и содержит команды, стартующие Perl.

Через некоторое время на экране монитора должно появиться следующее изображение (рис. 8.3).

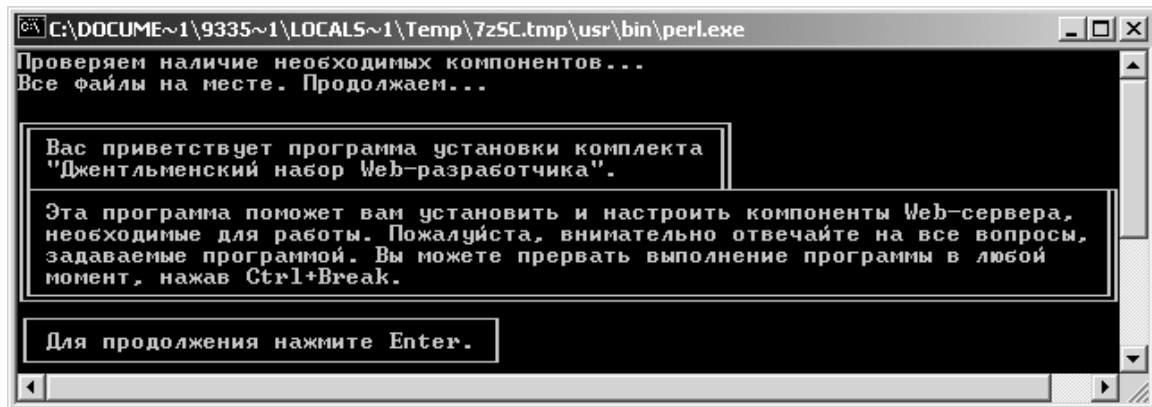


Рис. 8.3 — Начальный этап установки Web-сервера

Комплекс по умолчанию устанавливается в каталог "C:\WebServers", для продолжения установки вам нужно лишь нажать клавишу "Enter", чтобы согласиться с этим выбором. В указанном каталоге будут расположены абсолютно все компоненты системы, и вне него никакие файлы в дальнейшем не создаются (исключая ярлыки на Рабочем столе).

Рекомендуется устанавливать комплекс в каталог первого уровня — то есть, "C:\WebServers", а не в каталог, например, "C:\My\WebServers". Дело в том, что инсталляторы пакетов расширений ищут базовый комплект именно на первом уровне по всем дискам. И если не находят, вам придётся ввести имя директории вручную.

Далее вам предложат ввести имя виртуального диска, который будет связан с только что указанной директорией. Рекомендуем вам согласиться со значением по умолчанию (Z:). Важно, чтобы в системе не было диска с этим именем.

После этого начнется копирование файлов дистрибутива, а под конец вам будет задан вопрос, как именно вы собираетесь запускать и останавливать комплекс. У вас есть две альтернативы:

1. Создавать виртуальный диск при загрузке машины (естественно, инсталлятор позаботится, чтобы это происходило автоматически), а при остановке серверов его (диск) не отключать.

2. Создавать виртуальный диск только по явной команде старта комплекса (при щелчке по ярлыку запуска на Рабочем столе). И, соответственно, отключать диск от системы — при остановке серверов.

После этого установка сервера будет завершена. На Рабочем столе должны появиться новые логотипы (рис. 8.4). В операционных системах Windows 2000/XP такие логотипы могут не создаваться, а также не создастся и виртуальный диск при загрузке (если выбран первый вариант запуска и остановки комплекса). Одна из причин — присутствие русских букв в аккаунте при входе в Windows. В этом случае можно разместить логотипы на Рабочем столе и в автозагрузке вручную. Для этого воспользуйтесь ярлыками для стандартного варианта установки из Zip-архива (shortcuts_for_odd_russian_login.zip). Просто скопируйте ярлыки, приведённые на рисунке 8.4, из Zip-архива на Рабочий стол, а ярлык "Create virtual drive" — в автозагрузку. После этого щелкните по ярлыку "Start servers" на Рабочем столе, а затем, дождавшись, когда все консольные окна исчезнут, запустите браузер и наберите в нем адрес: "http://localhost".



Рис. 8.4 — Новые логотипы рабочего стола

При успешном старте сервера в броузере должно появиться изображение, подобное приведённому на рисунке 3.5. Некоторые ОС имеют обыкновение при первом запуске Internet Explorer вызывать "Мастер подключения". Если это произошло на вашей машине, то выйдите из мастера.

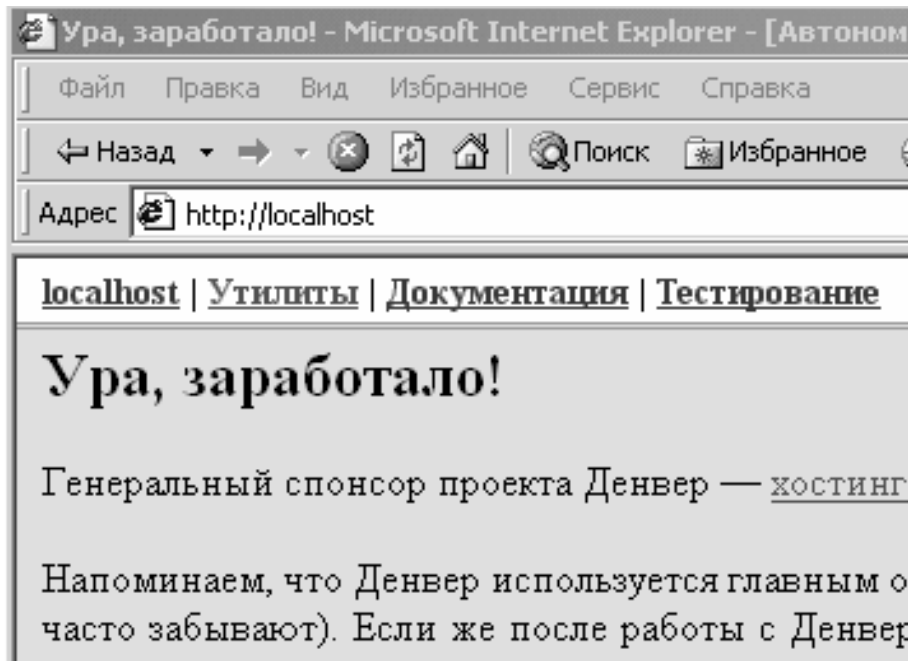


Рис. 8.5 — Успешный запуск сервера Apache

Если тестовая страница все же не загрузится, проверьте:

- Отключен ли у вас прокси-сервер в настройках броузера?
- Запущен ли Денвер? Если да, то нет ли ошибок при его работе? Об этом можно узнать, если щелкнуть левой кнопкой мыши на пиктограмме сервера Apache (изображение пера) на панели задач внизу справа.
- Не запущен ли у вас какой-то другой Web-сервер, который мешает Денверу (часто бывает в Windows XP)? Например, Microsoft IIS? Если да, отключите его.
- Если у вас Windows 95, проверьте, проделали ли вы рекомендуемые действия по подготовке ОС к инсталляции комплекса.

Наконец, если ничего не помогает, загляните в список часто задаваемых вопросов — файл "FAQ denwer.rar" на компакт-диске.

Денвер-2 прошел тестирование в следующих ОС:

- Windows 95/98/ME;
- Windows NT;
- Windows 2000/XP.

8.9 Работа с виртуальными хостами

Вниманию пользователей Windows NT, 2000 или XP (и старше). Прежде чем продолжить, убедитесь, что у вас запущена служба "DNS-клиент". Это можно сделать, открыв "Панель управления — Администрирование — Службы". В противном случае виртуальные хосты работать не будут.

Если вы занимаетесь разработкой Web-сайтов, вам наверняка хотелось бы обслуживать одним сервером сразу несколько хостов. Иными словами, введя в броузере путь "http://localhost", вы попадете на один сайт, а, напечатав "http://test.ru", — совсем на другой (но тоже на локальной машине). Чтобы добавить новый хост, вам нужно проделать следующее:

- Создать в папке "/home" директорию с именем, совпадающим с именем виртуального хоста (в нашем случае test.ru).

Обратите внимание, что имя директории содержит точку. Эта директория будет хранить директории документов доменов третьего уровня для "test.ru". Например, имя "abc.test.ru" связывается сервером с директорией "/home/test.ru/abc/", а имя

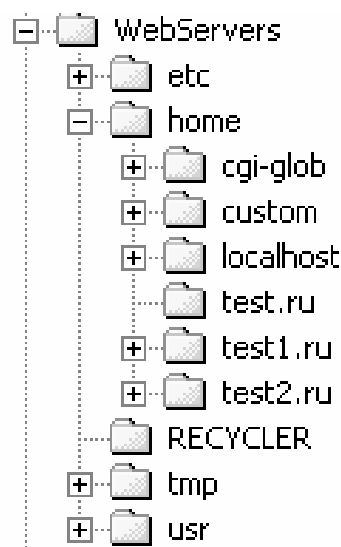


Рис. 8.6 — Структура каталогов сервера Apache

"abc.def.test.ru" — с "/home/test.ru/abc.def/". Ну и, конечно, поддиректория "www" соответствует адресам "www.test.ru" и просто "test.ru". На рисунке показано, как может выглядеть директория "/home". Не забудьте создать папку www в директории виртуального хоста, ведь именно в ней будут храниться его страницы и скрипты!

- Перезапустить сервер, воспользовавшись, например, ярлыком "Restart servers" на Рабочем столе.

Файл "hosts" модифицируется автоматически, подстраиваясь под текущую конфигурацию каталогов в /home. При этом используется специальный алгоритм для распознавания, какие хосты были внесены Денвером, а какие пользователь добавил самостоятельно, вручную (если он, конечно, захочет это сделать).

8.10 Настройка Контроллера удаленного доступа

Как только вы начнете создавать виртуальные хосты, Контроллер удаленного доступа на некоторых системах может при первом открытии хоста предлагать вам действия, изображённые на рисунке 8.7. В этом случае необходимо всегда выбирать "Подключиться" или "Повторить". Если всё же после этого Контроллером удаленного доступа начинается набор номера на модеме, необходимо зайти в меню "Сервис — Свойства обозревателя —

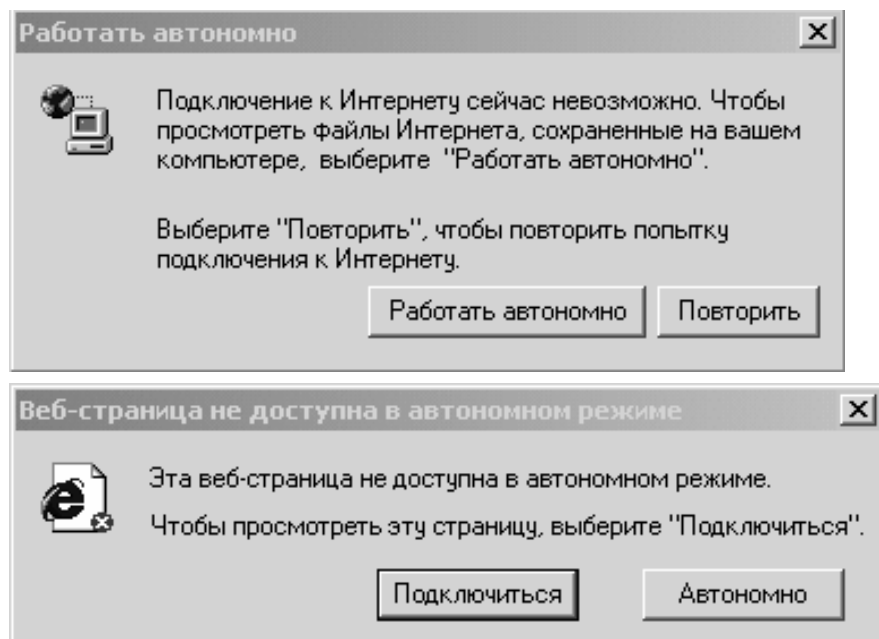


Рис. 8.7 — Сообщения Контроллера удалённого доступа

Подключение" и в разделе "Настройка удаленного доступа" поставить флажок "Не использовать это соединение в программах Интернета" (или "Never Dial a connection"). Это рекомендации для пользователей Windows 2000. На всех остальных системах пункты меню и кнопки могут называться немного по-другому, но смысл остается тот же.

8.11 Настройка прокси-сервера

Многие версии Windows поставляются с включенным по умолчанию прокси-сервером. Это может вызвать некоторые проблемы при работе с Денвером. Для их устранения рекомендуется выполнить следующие действия.

Если после запуска Денвера страница "http://localhost" не работает, вероятнее всего, необходимо отключить прокси-сервер в настройках браузера. Для "простых" хостов (вида "localhost", "test", "dklab" и т.д.) обычно достаточно установить флажок "Не использовать прокси-сервер для локальных адресов" в меню "Сервис — Свойства обозревателя — Подключение — Настройка сети — Дополнительно".

Если страница "http://localhost" отображается, а хосты, имена которых состоят из нескольких частей (например: "http://test1.ru"), — нет, то, вероятно, браузер не может распознать последний хост как локальный. В этом случае необходимо либо полностью отключить прокси-сервер, либо же перечислить хосты в списке исключения. Для этого необходимо зайти в меню "Подключение — Настройка сети — Дополнительно — Исключения" и в открывшемся окне "Не использовать прокси-сервер для адресов, начинающихся с:" перечислить все локальные адреса.

8.12 Запуск и остановка сервера

Управление запуском и остановкой сервера осуществляется либо с Рабочего стола, выбором соответствующего ярлыка, либо непосредственным запуском файлов "etc\Run.exe" и "etc\Stop.exe". Эти файлы находятся в соответствующих директориях комплекса.

Старт сервера осуществляется файлом "etc\Run.exe".

Остановка сервера — файлом "etc\Stop.exe".

Остановка и отключение виртуального диска — файлом "etc\utils\SwitchOff.exe".

Рекомендуется сервер останавливать исключительно по команде "Stop.exe", так как в противном случае возможна потеря данных в базе сервера MySQL.

После установки и первого запуска комплекса рекомендуется сразу же запустить браузер и набрать в нем следующий URL: "http://localhost". После этого вы должны попасть на тестовую страницу системы, изображённую на рисунке 3.8. Выбрав те или иные ссылки, вы можете протестировать каждый компонент отдельно.

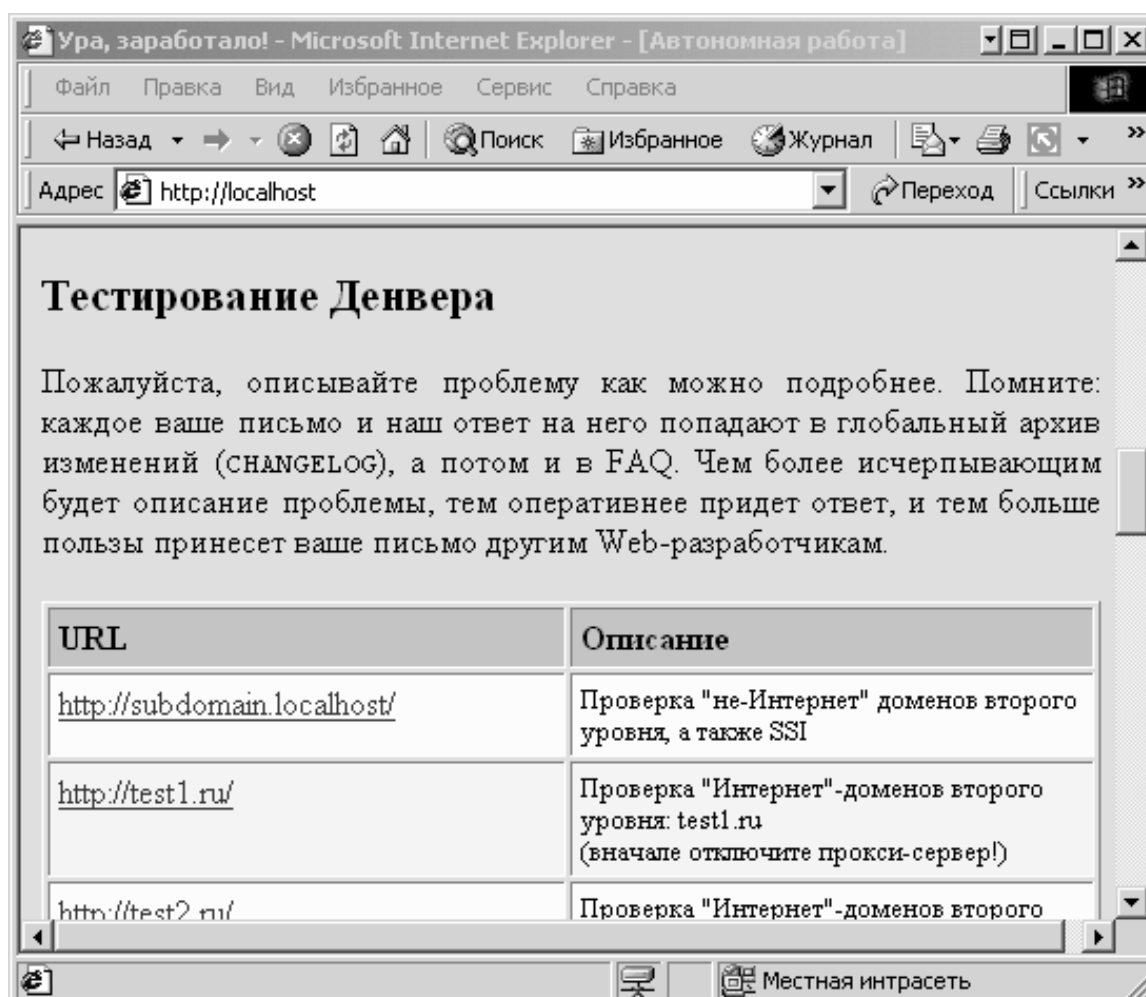


Рис. 8.8 — Страница тестирования Денвера

8.13 Размещение и тестирование своей личной Web-страницы

Необходимо скопировать все файлы, созданные при выполнении лабораторных работ, в каталог C:\WebServers\home\localhost\www". Это были следующие файлы: about_me.html, address.html, blumrbld.jpg, hobby.html, index.html, JavaScr.htm, MoneyMoney.mid, my.css, photo_me.jpg, study.html, tabl.htm, topic.html, welcome.html, witewall.jpg. Указанный каталог создаётся автоматически по умолчанию при инсталляции ПО "Денвер-2". Перед тестированием своей странички необходимо запустить сервер "Apache" и в окне браузера ввести адрес: <http://localhost/index>. Хотя визуально вы и не обнаружите никакой разницы в отображении страниц, но существует одно отличие — сейчас они загружаются с вашего сервера.

9 ЛАБОРАТОРНАЯ РАБОТА №6 Создание сценария на языке PHP

Цель работы: практическая работа по созданию первого сценария на языке PHP.

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, комплекс программ — "Денвер-2".

9.1 Задание на лабораторную работу №6

1. Изучить язык сценариев PHP.
2. Написать сценарий на языке PHP, обрабатывающий запрос формы вашей Web-страницы. В результате обработки формы сервер должен сгенерировать сообщение типа:

Добрый день (утро, вечер) Михаил!

Я приветствую тебя на своём сервере!

Мой адрес: 647954, Приобский край....

.....

и т.д. (далее должны быть обработаны все данные из формы). Этот сценарий должен обрабатывать данные, вводимые в соз-

данную ранее форму (файл "address.html"), выводить их на экран и сохранить во внешнем текстовом файле.

3. Разработать PHP-сценарий, записывающий все данные в текстовый файл.

9.2 Отчетность по лабораторной работе

В результате выполнения лабораторной работы должны быть предоставлены следующие материалы.

1. Текстовый файл отчета по лабораторной работе, оформленный в соответствии с существующими требованиями, который должен содержать следующие сведения:

1.1. Титульный лист.

1.2. Детальное описание последовательности всех шагов по выполнению лабораторной работы, с приведением всех экранных форм.

9.3 Создание сценария на языке PHP

После изучения языка создания сценариев — PHP, создайте свой первый сценарий с именем "form.php". Этот сценарий должен обрабатывать данные, вводимые в созданную ранее форму (файл "address.html"), выводить их на экран и сохранить во внешнем текстовом файле.

Пример получения данных из созданной формы приведён в учебном пособии "Технология создания Интернет-приложений. Раздел 3. Основы PHP и MySQL".

Как вы помните, файл "address.html" предписывал использовать для отправки формы почтового клиента с адресом "foo@somehost.ru", а в качестве метода отправки — универсальный метод "post". Для упрощения задачи просто отправим данные из формы в файл сценария "Form.php". Этот файл и необходимо создать. Для этого в файле "address.html" необходимо изменить строку, отвечающую за перенаправление данных, следующим образом:

до изменения

```
<FORM action="mailto:foo@somehost.ru" method="post">
```

после изменения

```
<FORM ACTION="Form.php" method="post" >
```

Остальной текст файла должен остаться без изменения.

Создайте в текстовом редакторе файл "Form.php".

Этот сценарий должен получить из нашей формы и обработать следующие параметры.

1. Приветствие в зависимости от времени суток и с указанием имени зарегистрировавшего.

2. Значения текстовых полей с фамилией, именем и отчеством.

3. Значения исключающего переключателя с параметрами о возрасте. Будет передано только одно из значений ("Школьный", "Студенческий", "Средний", "Пенсионный" или "Не имеет значения").

4. Параметры списка выбора о любимом увлечении, содержащего одно значение.

5. Параметры включателя о дополнительной информации. Эти параметры будут переданы только в том случае, если соответствующие им переключатели активны. Может быть передано одно значение, несколько значений или же ни одного.

6. Значение текстового поля, содержащее сообщение.

7. Выполнение записи всех обработанных данных в текстовой файл.

Пример приветствия в зависимости от времени суток приведён в учебном пособии по основам PHP и MySQL.

Каждый параметр из формы передаётся с помощью соответствующей переменной. Имена переменных могут быть любыми из числа допустимых в PHP. Желательно, чтобы они несли также и смысловую нагрузку. Например, в качестве имён можно использовать следующие.

В качестве имён переменных для значений текстовых полей с фамилией, именем и отчеством соответственно — "\$name1", "\$name2" и "\$name3".

В качестве имени переменной для значения исключающего переключателя с параметрами о возрасте — "\$Radio".

В качестве имени переменной для значения параметра списка выбора о любимом увлечении — "\$Hobby".

В качестве имён переменных для значений параметров включателя, о дополнительной информации — "\$special1", "\$special2", "\$special3", "\$special4" и "\$special5".

В качестве имени переменной для значения параметра текстового поля, содержащего сообщение — "\$Description".

Интерпретатор создаёт для каждого параметра переменную с соответствующим именем. Для отображения их значений на экране монитора достаточно использовать функции "echo()" или "printf()", которые являются синонимом "print".

Параметры текстовых полей (переменные "\$name1", "\$name2", "\$name3", "Description"), исключающего переключателя (переменные "\$Radio") и списка выбора (переменная "\$Hobby") обрабатываются легко. Например, для отображения значения параметра текстового поля о фамилии достаточно написать:

```
print ("Ваше фамилия $name1.<BR>\n");
```

С параметрами включателя дело обстоит несколько сложнее. Если переключатель не активен, то параметры вообще не будут переданы на сервер, как будто бы вообще их и не было. Следовательно, не будет создана глобальная переменная и при попытке обращения к такой переменной мы получим сообщение о том, что переменная не существует. Поэтому просто написать строчку для отображения информации о том, состоите ли в браке — "print("\$special1
");" будет не корректно. Нужно предварительно проверить, существует ли переменная:

```
if (isset($special1))
    print("$special1 <BR>");
```

Только после такой проверки для каждой из переменных для включателя можно начинать с ними работать.

Для записи всех данных, обработанных сценарием, во внешний текстовый файл воспользуйтесь примером, приведённым в учебном пособии.

Сначала определите функцию, которая будет принимать все аргументы. В нашем случае их количество будет равно двена-

дцати. Строка, определяющая функцию, может иметь следующий вид:

```
function WriteToFile($day,$name1, $name2,$name3, $Radio1,
$Hobby, $special1, $special2,$special3, $special4, $special5, $De-
scription) {
```

В качестве аргументов в ней перечислены все переменные, которые были определены в HTML-форме, а также добавлена новая — "\$day ". Эта переменная в сочетании с переменной "\$name2" (имя) должна выводить на экран и в текстовый файл приветствие в зависимости от времени суток. Например, такое — "Добрый вечер Виктор!".

После фигурной скобки будет следовать сама функция, которая должна содержать в себе все этапы — открытие файла, собственно запись данных, закрытие файла. В качестве имени текстового файла можно использовать, например, имя "data.txt".

Функция записи в текстовый файл может иметь следующий вид:

```
fwrite($Open,$day\t"."$name2!\n"."Ваша фамилия-
\t"."$name1\n"."Ваше имя-\t"."$name2\n"."Ваше отчество-
\t"."$name3\n"."Ваш возраст-\t"."$Radio1\n"."Ваше лю-
бимое увлечение-\t"."$Hobby\n"."Дополнительная ин-
формация о вас-\t\n"."$special1\n"."$special2\n".
"$special3\n"."$special4\n"."$special5\n"."Ваше сообще-
ние-\t"."$Description");
```

В ней перечислены все необходимые аргументы, обеспечивающие корректное сохранение данных.

При вызове функции необходимо указывать её аргументы в той же последовательности, в которой они были перечислены при её определении:

```
$CallFunction=WriteToFile($day,$name1, $name2,$name3,
$Radio1,$Hobby, $special1, $special2,$special3, $special4, $spe-
cial5,$Description);
```

Пример заполненной формы приведён на рисунке 9.1. На рисунке 9.2 приведён результат выполнения файла сценария

Мой адрес - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Назад Поиск Избранное Журнал

Адрес http://localhost/kursovoy/address.html

Переход Ссылки

Это мой адрес: 647954 Приобский край, г. Зетон, ул.Осетровая, д.13, кв.77

А теперь расскажите немного о себе...

Представьтесь, пожалуйста

ФИО

Фамилия Сорокин Имя Виктор Отчество: Иванович

Возраст

Какой у вас возраст

☐ Школьный ☒ Студенческий ☐ Средний ☐ Пенсионный ☐ Не имеет значения

Ваше любимое увлечение Пиво без бани

Введите дополнительную информацию о себе

Дополнительная информация

☒ Состою в браке ☐ Есть судимость ☒ Состою в партии ☒ Имею награды ☒ Был за границей

Для ввода сообщения воспользуйтесь окном ниже

Ввод сообщения

Рад познакомиться с вами

Для отправки заранее подготовленного файла воспользуйтесь окном ниже

Выбор файла

Обзор

Отправка или очистка данных

Отправить Очистить

Готово Местная интрасеть

Рис. 9.1 — Внешний вид заполненной формы в броузере "form.php". На рисунке 9.3 и 9.4. приведено содержание текстового файла "data1.txt" в текстовом редакторе MS Word и Блокноте. Редактор Блокнот не корректно интерпретирует управляющую последовательность начала новой строки — "\n". Вместо перевода на новую строку, он отображает символ четырёхугольника.

10 ЛАБОРАТОРНАЯ РАБОТА №7

Создание базы данных

Цель работы: практическая работа по созданию базы данных с помощью языка запросов SQL.

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, комплекс программ — "Денвер-2".

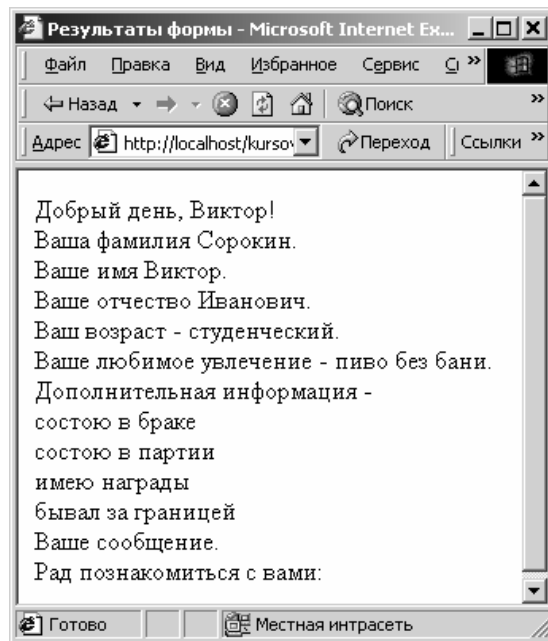


Рис. 9.2 — Результат выполнения файла сценария "form.php" в браузере

10.1 Задание на лабораторную работу №7

1. Изучить язык запросов к базе данных (БД) — SQL.
2. Создать с помощью языка запросов SQL БД "Мои родственники" и "Мои друзья" (Ф.И.О., год рождения, степень родства, адрес, профессия, увлечения). Наполнение может быть реальное или вымышленное.

10.2 Отчетность по лабораторной работе

В результате выполнения лабораторной работы должны быть предоставлены следующие материалы.

1. Текстовый файл отчета по лабораторной работе, оформленный в соответствии с существующими требованиями, который должен содержать следующие сведения:

1.1. Титульный лист.

1.2. Детальное описание последовательности всех шагов по выполнению лабораторной работы, с приведением всех экранных форм.

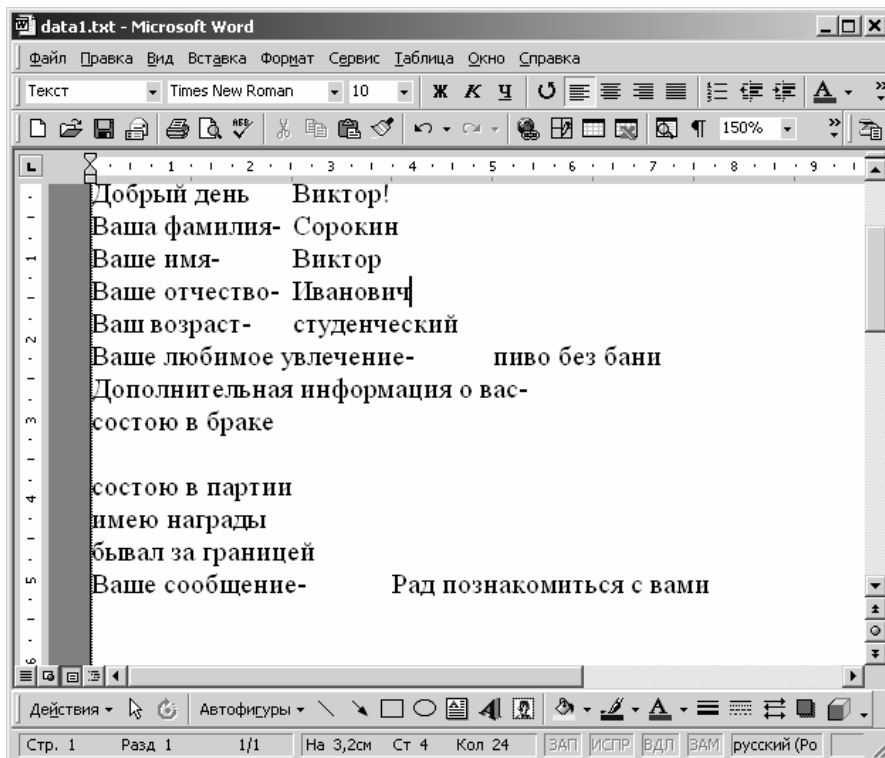


Рис. 9.3 — Вид текстового файла в редакторе MS WORD

10.3 Создание пользователя и таблиц

При создании БД используйте материал методических указаний. Создайте базы данных под названиями "Мои родствен-

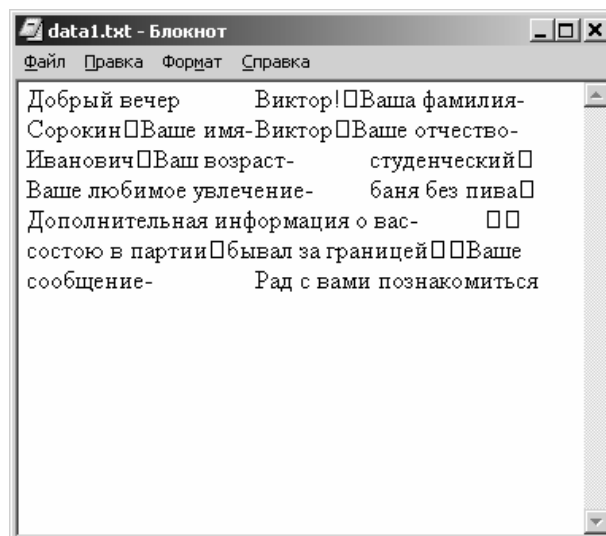


Рис. 9.4 — Вид текстового файла в редакторе Блокнот

ники" ("Relative") и "Мои друзья" ("Friend").

Для этого сначала необходимо создать встроенными средствами администрирования Денвера нового пользователя, например с именем "mydb".

Затем в текстовом редакторе создайте новый РНР-документ под именем "CreateMydb.php", который выполнит подключение к MySQL и создание базы данных. Для создания БД можно использовать также встроенные средства администрирования, но для учебных целей необходимо создать её вручную. После этого зайдите в программу "phpMyAdmin" и убедитесь, что база данных действительно создана. В случае успешного создания БД в каталоге "C:\WebServers\usr\local\mysql4\data\mydb" должен появиться новый файл с именем "db.opt".

Создайте в этой БД две таблицы — "Мои родственники" ("Relative") и "Мои друзья" ("Friend"). Эти таблицы должны иметь следующую текстовую схему:

```
Relative (Relative_ID, FIO, BIRTHDFAY, ADDRESS, FONE,
PROFESSIONAL, FASCINATION, DEGREE_KINSHIP);
```

```
Friend(Friend_ID, FIO, BIRTHDFAY, ADDRESS, FONE,
PROFESSIONAL, FASCINATION).
```

Первые столбцы в таблицах являются первичными ключами. Как вам известно из методических указаний — это специальный столбец с уникальными значениями, используемыми для обращения к табличным строкам. Этот столбец индексируется в базе данных для того, чтобы быстрее передвигаться по таблице. Он может иметь только один первичный ключ, который обычно задаётся как автоматически увеличивающееся на единицу целое число.

Далее перечислены все остальные столбцы таблицы. Столбец " BIRTHDFAY" имеет тип "DATE", остальные "TXT". Дата должна вводиться и отображаться в формате YYYY-MM-DD (ГГГГ-ММ-ДД).

Файл для создания таблицы "Мои друзья" (CreateTable-Friend.php) может иметь следующий вид.

```

1. <HTML>
2. <HEAD>
3. <TITLE>Создание таблицы друзей </TITLE>
4. </HEAD>
5. <BODY>
6. <?php
7. // Установка значений переменных для доступа к ба-
   зе данных.
8. $Host = "localhost";
9. $User = "mydb";
10. $Password = "mydb";
11. $DBName = "mydb";
12. $TableName = "Friend";
13. $Link = mysql_connect ($Host, $User, $Password);
14. $Query = "CREATE table $TableName
   (id INT UNSIGNED NOT NULL
   AUTO_INCREMENT PRIMARY KEY,
   FIO TEXT, BIRTHDAY DATE, ADDRESS TEXT,
   FONE TEXT, PROFESSIONAL TEXT,
   FASCINATION TEXT )";
15. if (mysql_db_query ($DBName, $Query, $Link)) {
16. print ("Запрос успешно выполнен!<BR>\n");
17. } else {
18. print ("Запрос не был выполнен!<BR>\n");
19. }
20. mysql_close ($Link);
21. ?>
22. </BODY>
23. </HTML>

```

Файл для создания таблицы "Мои родственники" (CreateTableRelative.php) отличается тем, что имеет на один столбец больше. В нём присутствует столбец "Степень родства" (DEGREE_KINSHIP).

Протестируйте оба сценария. При успешном выполнении в каталоге "C:\WebServers\usr\local\mysql4\data\mydb" должны создаться шесть новых файлов — "Relative.frm", "Relative.MYD", "Relative.MYI", "Friend.frm", "Friend.MYD", "Friend.MYI".

10.4 Создание титульной страницы базы данных

Создайте титульную страницу вашей базы данных и сохраните её под именем "base.php". Титульная страница должна содержать в себе элементы навигации по базе, а также ссылки (нижний фрейм) для обращения к ранее созданной личной странице.

Титульная страница должна содержать следующие элементы навигации:

1. Просмотр и редактирование базы данных.
2. Добавление данных.
3. Поиск данных в базе.
4. Нижний фрейм со ссылками.

Чтобы форма смотрелась опрятно, рекомендуется исполь-

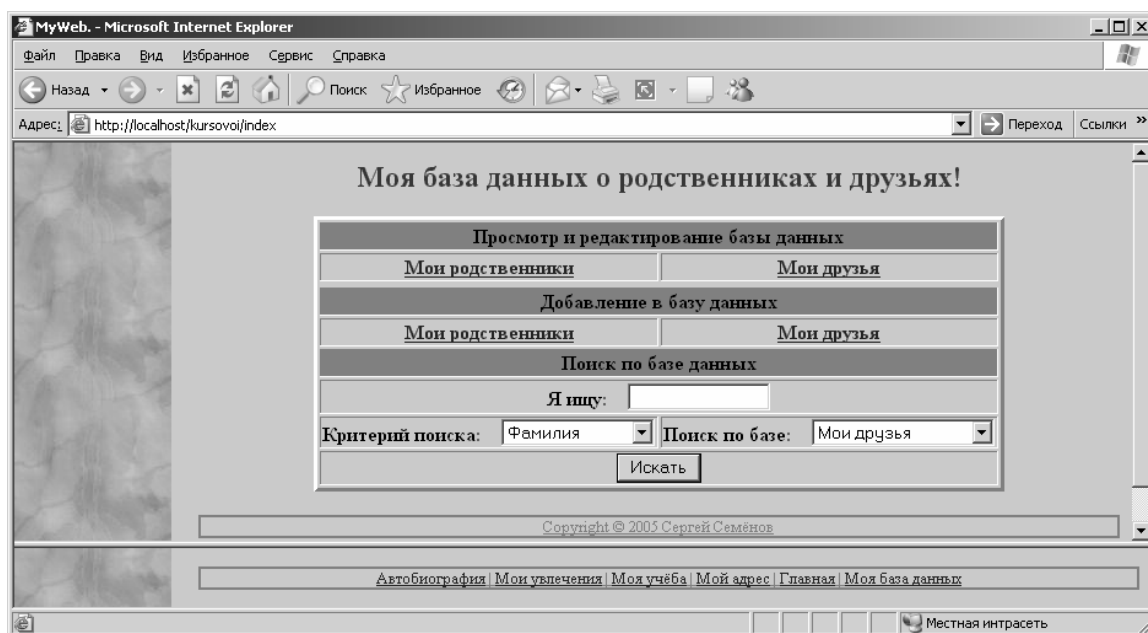


Рис. 10.1 — Титульная страница базы данных

зовать таблицу для размещения в неё необходимой информации.

Такая титульная страница может иметь вид, приведённый на рисунке 10.1.

Как видно из рисунка 10.1, для доступа к БД с целью их просмотра и редактирования служат две ссылки — "Мои родственники" и "Мои друзья".

Для добавления новых записей служат соответственно одноимённые ссылки, но размещённые в разделе "Добавление в базу данных".

Для поиска в базе данных предназначены окна ввода "Я ищу", "Критерий поиска" и "Поиск по базе".

В текстовом поле "Я ищу" вводится информация для поиска по базе данных. Это, например, может быть фамилия или её часть, дата рождения или профессия.

В поле с выпадающим меню "Критерий поиска" указывается критерий по которому будет произведён поиск. Такими критериями в нашем случае являются "Фамилия", "День рождения" или "Профессия".

В поле с выпадающим меню "Поиск по базе" указывается имя БД в которой необходимо осуществить поиск — "Мои друзья" или "Мои родственники".

Для обеспечения перехода с главной страницы сайта на титульную страницу базы данных необходимо добавить в файл "Welcome.html" строку со ссылкой:

Информацию о моих родственниках и друзьях можно посмотреть [Моя база данных по родственникам и друзьям](basa.php "Моя база данных по родственникам и друзьям") здесь.

В файл нижнего фрейма "Topic.html" добавляем строку со ссылкой:

[Моя база данных](basa.php) .

Файл "Basa.php" может иметь следующий вид.

1. <HTML>
2. <HEAD>
3. <TITLE>Моя база данных</TITLE>
4. <link rel="stylesheet" href="my.css" type="text/css" />
5. <form action="search.php" method="get">
6. </HEAD>
7. <BODY>
8. <h2>Моя база данных о родственниках и друзьях!</h2>
9. <table width="75%" align="Center" border="3">
10. <colgroup align="Center">

[illegible]

[illegible]

Следующий шаг, который необходимо выполнить, — это реализовать возможность ввода новых данных в базу.

10.5 Создание форм для ввода данных в базу

Для ввода новых данных в базу необходимо создать две формы. Одна форма должна пополнять данные о друзьях, другая — о родственниках. Присвоим этим формам соответственно имена "FormFriend.php" и "FormRelative.php". Внешний вид таких форм приведён на рисунках 10.2 и 10.3. Кроме кнопок ввода данных и очистки, они также содержат по две ссылки. По ссылке — "Возврат на главную страницу БД" происходит переход на титульную страницу базы данных. По ссылке "Просмотр БД "Мои родственники"" — должен осуществляться переход на страницу, содержащую информацию о родственниках. Эти ссылки пока не работают.

Сценарий "FormRelative.php" может иметь следующий вид.

1. <HTML>
2. <HEAD>

MyWeb. - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Назад Поиск Избранное

Адрес: <http://localhost/kursovoy/index> Переход Ссылки

Добавление данных в базу "Мои друзья"

ФИО:
 Дата рождения:
 Адрес:
 Номер телефона:
 Профессия:
 Увлечение:

Отправка или очистка данных

[Возврат на главную базы данных](#)
[Просмотр БД "Мои друзья"](#)

[Автобиография](#) |
 [Мои увлечения](#) |
 [Моя учёба](#) |
 [Мой адрес](#) |
 [Главная](#) |
 [Моя база данных](#)

Готово Местная интрасеть

Рис. 10.2 — Форма, для ввода данных в базу "Мои друзья"

3. <TITLE>Форма для заполнения БД "Мои родственники"</TITLE>
4. <link rel="stylesheet" href="my.css" type="text/css" />
5. </HEAD>
6. <BODY>
7. <FORM ACTION="HandleFormRelative.php" METHOD="POST">
8. <h3 ALIGN="Center">Добавление данных в базу "Мои родственники"</h3>
9. <table align="Center">
10. <tr>
11. <td>ФИО:</td> <td><INPUT TYPE="TEXT" NAME="Array[FIO]" SIZE="40"> </td>
12. </tr>
13. <tr>
14. <td>Дата рождения: </td> <td><INPUT TYPE="DATE" NAME="Array[BIRTHDAY]" SIZE="40"> </td>

Форма для заполнения БД "Мои родственники" - Microsoft Internet Explorer

Адрес: http://localhost/kursovoy/formrelative.php

Добавление данных в базу "Мои родственники"

ФИО:
 Дата рождения:
 Адрес:
 Номер телефона:
 Профессия:
 Увлечение:
 Степень родства:

Отправка или очистка данных

[Возврат на главную страницу БД](#)
[Просмотр БД "Мои родственники"](#)

Местная интрасеть

Рис. 10.3 — Форма для ввода данных в базу "Мои родственники"

15. </tr>
16. <tr>
17. <td>Адрес: </td> <td> <INPUT TYPE="TEXT" NAME="Array[ADDRESS]" SIZE="40"></td>
18. </tr>
19. <tr>
20. <td>Номер телефона: </td> <td> <INPUT TYPE="TEXT" NAME="Array[FONE]"SIZE="40"></td>
21. </tr>
22. <tr>
23. <td>Профессия: </td> <td> <INPUT TYPE="TEXT" NAME="Array[PROFESSIONAL]"SIZE="40"></td>
24. </tr>
25. <tr>
26. <td>Увлечение: </td> <td> <INPUT TYPE="TEXT" NAME="Array[FASCINATION]"SIZE="40"></td>

```

27. </tr>
28. <tr>
29. <td>Степень родства:</td> <td> <INPUT TYPE="TEXT"
    NAME="Array[degree_kinship]"SIZE="40"></td>
30. </tr>
31. <tr>
32. <td colspan="2" align="Center" align="Center">
    <b>Отправка или очистка данных</b> <br><br> </td>
33. </tr>
34. <tr>
35. <td align="Center"> <INPUT TYPE="SUBMIT"
    NAME="Submit" VALUE="Ввести!"></td> <td
    align="Center"> <INPUT TYPE="RESET"
    NAME="Reset" VALUE="Очистить!"> </td>
36. </tr>
37. <tr>
38. <td align="Center"> <a href="Basa.php"><b>Возврат на
    главную страницу БД</b></a></td><td align="Center"><a
    href="EditRelative.php"><b>Просмотр БД "Мои родст-
    венники"</b></a></td></p>
39. </tr>
40. </table>
41. </table>
42. </FORM>
43. </BODY>
44. </HTML>

```

При вводе данных в поле "Дата рождения " необходимо учитывать, что формат, используемый по умолчанию, имеет вид — ГГГГ-ДД-ММ. При ошибочном формате ввода в этом поле будет внесена дата — 000-00-00.

Написание таких форм не должно вызывать затруднений.

10.6 Передача данных в базу

Для передачи данных из форм серверу — необходимо написать еще два сценария. Эти сценарии должны помещать данные в базу данных (таблицы).

Сценарий для обработки данных о родственниках — форма "FormRelative.php", назовём "HandleFormRelative.php". Он может иметь следующий вид.

```

1. <HTML>
2. <HEAD>
3. <TITLE>Добавление данных в БД "Мои родственни-
   ки"</TITLE></HEAD>
4. <BODY>
5. <DIV align="center">
6. <?php
7. /* Эта страница получает и обрабатывает данные, приня-
   тые от формы "FormRelative.php". */
8. /* Удаление пробелов в начале и в конце строк.*/
9. $Array["FIO"] = trim ($Array["FIO"]);
10. $Array["BIRTHDAY"] = trim ($Array["BIRTHDAY"]);
11. $Array["ADDRESS"] = trim ($Array["ADDRESS"]);
12. $Array["FONE"] = trim ($Array["FONE"]);
13. $Array["PROFESSIONAL"] = trim ($Ar-
   ray["PROFESSIONAL"]);
14. $Array["FASCINATION"] = trim ($Ar-
   ray["FASCINATION"]);
15. $Array["degree_kinship"] = trim ($Ar-
   ray["degree_kinship"]);
16. /* Установка значения переменных для доступа к базе
   данных.*/
17. $Host = "localhost";
18. $User = "mydb";
19. $Password = "mydb";
20. $DBName = "mydb";
21. $Link = mysql_connect ($Host, $User, $Password
22. $TableName = ("Relative");
23. $Query = "INSERT into $TableName values ('0',
   '$Array[FIO]','$Array[BIRTHDAY]', '$Array[ADDRESS]',
   '$Array[FONE]','$Array[PROFESSIONAL]','$Array[FASCI
   NATION]','$Array[degree_kinship]')";
24. if (mysql_db_query ($DBName, $Query, $Link)) {

```

```

25. print ("Запрос добавления данных успешно выпол-
    нен!<BR>\n");
26. print ("Добавлен родственник $Array[FIO]<BR>\n");
27. print ("Вы можете продолжить ввод данных<BR>\n");
28. $data_h=date('H');
29. if ($data_h>=4 and $data_h<12) {$privet='Доброе утро';}
    // утро
30. if ($data_h>=12 and $data_h<17) {$privet='Добрый день';}
    // день
31. if ($data_h>=17 and $data_h<24) {$privet='Добрый ве-
    чер';} // вечер
32. if ($data_h>=0 and $data_h<4) {$privet='Доброй ночи';}
    // ночь
33. print '<script LANGUAGE="JavaScript">';
34. print 'alert ("'. $privet. '! \nДобавлен ваш родствен-
    ник:'. $Array[FIO]. ' \nДата рождения:
    '. $Array[BIRTHDAY]. ' \nАдрес жительства:
    '. $Array[ADDRESS]. ' \nТелефон:'. $Array[FONE]. '
    \nПрофессия: '. $Array[PROFESSIONAL]. ' \nУвлечение:
    '. $Array[FASCINATION]. "');";
35. print '</script>';
36. } else {
37. print ("Запрос добавления данных не может быть выпол-
    нен!<BR>\n");
38. }
39. mysql_close ($Link);
40. Include("FormRelative.php");
41. ?>
42. </DIV>
43. </BODY>
44. </HTML>

```

Строка № 5 задаёт выравнивание блока по центру.

Строки с № 9 по № 15 удаляют пробелы в начале и в конце строк.

Строки с № 17 по № 21 осуществляют стандартную процедуру присвоения значений переменных для доступа к базе данных и обеспечивают соединение с сервером. Такая процедура

может многократно использоваться и в других PHP-сценариях. Поэтому целесообразно поместить её в отдельном файле, например "Connect.php", а затем при необходимости вызывать в нужном месте с помощью функции "Include". В этом случае содержание файла "Connect.php" должно быть следующим:

1. <?
2. \$Host = "localhost";
3. \$User = "mydb";
4. \$Password = "mydb";
5. \$DBName = "mydb";
6. \$Link = mysql_connect (\$Host, \$User, \$Password);
7. ?>

В случае использования такого файла строки с № 16 по № 22 в сценарии "HandleFormRelative.php" необходимо заменить строкой вызова функции "Include".

```
Include("Connect.php");
```

Строка № 23 создаёт запрос и добавляет информации в таблицу.

Строки с № 24 по № 27 направляют на сервер MySQL запрос и создают условную конструкцию, которая выполняет определенные действия, исходя из результатов запроса. Все функции обращения к БД, кроме действий собственно с данными, выдают значение "истина" при успешном выполнении и "ложь" во всех остальных случаях.

Строки с № 28 по № 32 создают условную конструкцию, которая присваивает переменной "\$privet" значение приветствия в зависимости от времени суток.

Строки с № 33 по № 35 представляют собой JavaScript, который создаёт на экране модальное диалоговое окно с приветствием и информацией, введённой в форму.

Строки с № 36 по № 38 завершают условную конструкцию в случае возникновения ошибки.

Строка № 39 закрывает соединение с сервером.

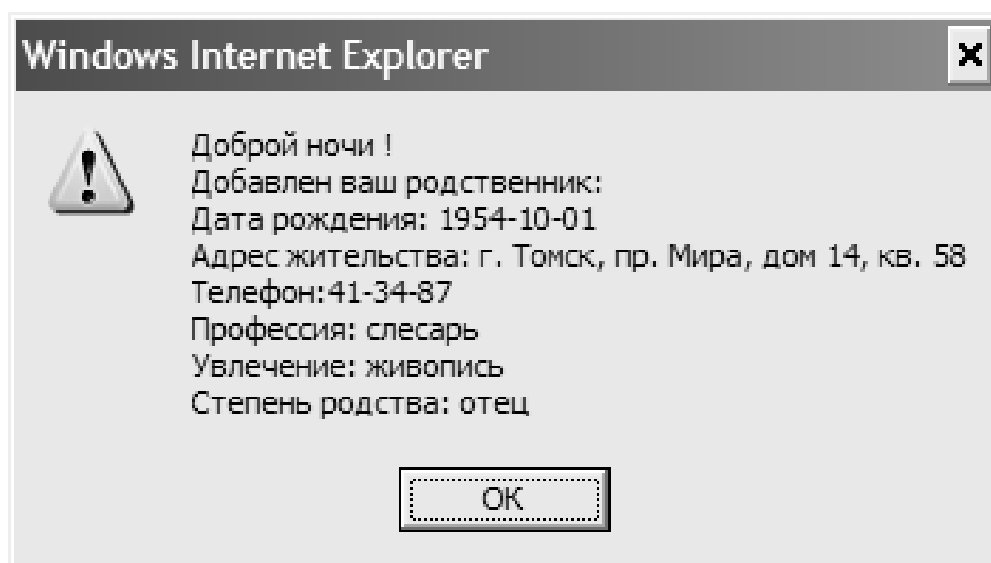


Рис. 10.4 — Модальное окно после выполнения сценария "HandleFormRelative.php"

Строка № 40 с помощью функции "Include" загружает HTML-страницу "FormRelative.php".

Сценарий для обработки формы "FormFriend.html" назовём "HandleFormFriend.php". Он будет иметь аналогичный вид, но в

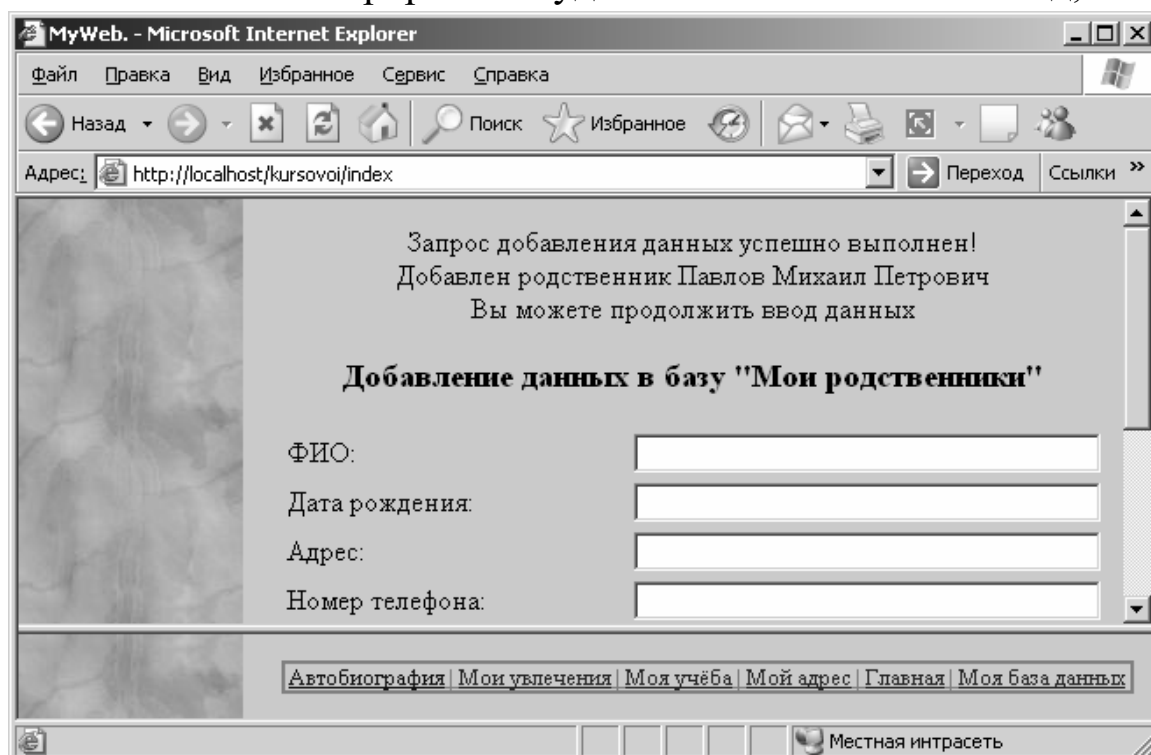


Рис. 10.5 — Исходное окно формы после выполнения запроса о вводе данных

нём будет отсутствовать поле для ввода информации о степени родства.

Протестируйте оба сценария. Для этого необходимо заполнить формы, приведённые на рисунках 10.2 и 10.3, и нажать кнопку "Ввести!". После выполнения сценария по добавлению родственника "HandleFormRelative.php" на экране появится окно, изображённое на рисунке 10.4.

После нажатия кнопки "ОК" форма вернётся в исходное состояние, изображённое на рисунке 10.5.

Чтобы просмотреть содержимое базы данных, необходимо написать дополнительно ещё два сценария.

10.7 Вывод данных на экран

Создадим два сценария, которые обеспечат просмотр БД. Для удобства работы с БД разместим в этом сценарии также ссылки для редактирования, сортировки и удаления данных из баз "Мои родственники" и "Мои друзья". Присвоим сценариям вывода данных на экран соответственно имена "DisplayRelative.php" и "DisplayFriend.php". Файл для вывода базы данных "DisplayRelative.php" может иметь следующий вид.

1. <HTML>
2. <HEAD>
3. <TITLE>Извлечение данных из БД "Мои родственники"</TITLE></HEAD>
4. <link rel="stylesheet" href="my.css" type="text/css" />
5. <BODY>
6. <h2>Информация о моих родственниках </h2>
7. <?php
8. // Установка значения переменных для доступа к базе данных.
9. Include("Connect.php");
10. \$TableName = "Relative";
11. \$Query = "SELECT * from \$TableName";
12. \$Result = mysql_db_query (\$DBName, \$Query, \$Link);
13. // Создание таблицы.

```

14. print("<TABLE BORDER=1 CELLSPACING=2
    CELLPADDING=2 ALIGN=CENTER>\n");
15. print("<TR ALIGN=CENTER VALIGN=TOP>\n");
16. print("<TH ALIGN=CENTER VALIGN=TOP> Ф.И.О<a
    href='SortRelative.php?sort=fio_relative'>sort</a></TH>\n");
17. print("<TH ALIGN=CENTER VALIGN=TOP>Дата рож-
    дения</TH>\n");
18. print("<TH ALIGN=CENTER
    VALIGN=TOP>Адрес</TH>\n");
19. print("<TH ALIGN=CENTER VALIGN=TOP>Номер те-
    лефона</TH>\n");
20. print("<TH ALIGN=CENTER
    VALIGN=TOP>Профессия</TH>\n");
21. print("<TH ALIGN=CENTER
    VALIGN=TOP>Увлечение</TH>\n");
22. print("<TH ALIGN=CENTER VALIGN=TOP> Степень
    родства</TH>\n");
23. print("<TH ALIGN=CENTER
    VALIGN=TOP>Ред</TH>\n");
24. print("<TH ALIGN=CENTER
    VALIGN=TOP>Удал</TH>\n");
25. print("</TR>\n");
26. // Получение результатов из базы данных.
27. while ($Row = mysql_fetch_array ($Result)) {
28. print("<TR ALIGN=CENTER VALIGN=TOP>\n");
29. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[FIO]</TD>\n");
30. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[BIRTHDAY]</TD>\n");
31. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[ADDRESS]</TD>\n");
32. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[FONE]</TD>\n");
33. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[PROFESSIONAL]</TD>\n");
34. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[FASCINATION]</TD>\n");

```

```

35. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[degree_kinship]</TD>\n");
36. print("<td><a
    href='FormEditRelative.php?id=$Row[id]'>edit</a>
    </td>\n");
37. print("<td><a href='DelRelative.php?id=$Row[id]'>del
    </a></td>\n");
38. print("</TR>\n");
39. }
40. mysql_close($Link);
41. print("</TABLE>\n");
42. echo '<br><Center>[ <a target="_blank"
    href="printversion.php?table_print='.$TableName.'&
    sort='.$sort.'">Версия для печати</a> ]<br><br>';
43. ?>
44. <table width="75%" align="Center" >
45. <tr>
46. <td > <a href="Basa.php"><b>Возврат на главную Страницу БД</b></a></td><td ></a></td></p><td ><a
    href="FormRelative.php"><b>Добавить данные</b></a></td></p>
47. </tr>
48. </table>
49. </BODY>
50. </HTML>

```

Строки с № 1 по № 12 вам уже хорошо известны. Они создают стандартный HTML-заголовок, открывают РНР-раздел, задают значения переменных для доступа к базе данных и осуществляют соединение с сервером.

Строки с № 14 по № 25 создают заголовок таблицы для вывода данных. Причём в строке № 16 размещена ссылка на файл "SortRelative.php", который должен обеспечивать сортировку столбца в алфавитном порядке. Строки № 24 и № 25 создают дополнительно два столбца, которые содержат надпись "Ред" и "Удал", что подразумевает соответственно — редактирование и удаление. Содержимое файла "SortRelative.php" будет рассмотрено ниже.

Строка № 27 создаёт цикл с использованием запроса "mysql_fetch_array", который возвращает массив, соответствующий извлечённому ряду. В нашем случае этот запрос будет извлекать все данные из таблицы.

Строки с № 28 по № 35 отображают соответствующие данные в таблице.

Строки № 36 и № 37 создают два дополнительных столбца со ссылками на файлы, которые обеспечивают редактирование и удаление записей. Обратите внимание на то, что эти ссылки также передают параметры номера строки "id" в соответствующий файл. Именно по номеру этой строки будет производиться изменение в них данных. Файлы "FormEditRelative.php" и "DelRelative.php" также будут рассмотрены ниже.

Строки с № 38 по № 41 заканчивают цикл и закрывают соединение с сервером.

Строка № 42 создаёт ссылку для перехода на сценарий, обеспечивающий подготовку таблицы к печати. Атрибут "target" задаёт имя окна, где эта ссылка будет использоваться. При активизации этой ссылки в сценарий "printversion.php" передаются два параметра — "table_print='.\$TableName.'& sort='.\$sort'", которые необходимы для правильного выбора данных для печати. Сценарий подготовки данных к печати "printversion.php" будет рассмотрен ниже.

Строки с № 44 по № 48 создают ссылки для обеспечения навигации. Для их форматирования используется таблица. Строка № 46 создаёт ссылки для возврата на главную страницу БД и для добавления данных в базу "Мои родственники".

Вид извлечённых данных из базы "Мои родственники", с помощью сценария "DisplayRelative.php", представлен на рисунке 10.6. В эту базу была внесена всего одна запись. При повторном заполнении формы данные добавятся в следующую строку таблицы.

Как видно из приведённого рисунка, в таблицу добавились два столбца со ссылками для редактирования и удаления записей. Ссылка для сортировки разместилась в столбце "Ф.И.О." заголовка таблицы. Внизу окна разместилась ссылка вывода на печать.

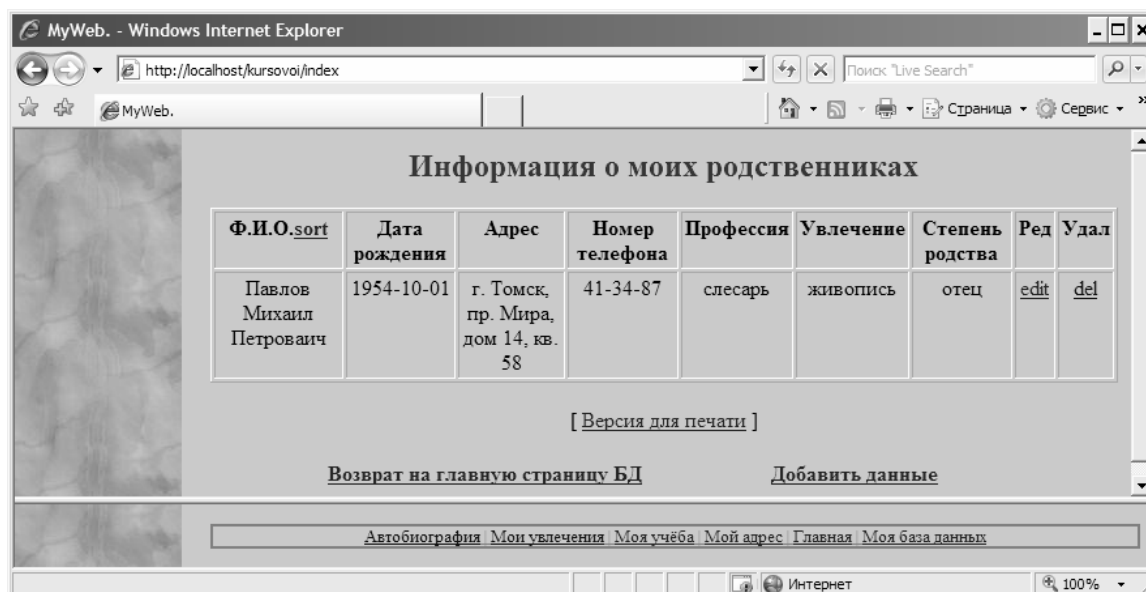


Рис. 10.6 — Данные, извлечённые из базы "Мои родственники" после первого ввода

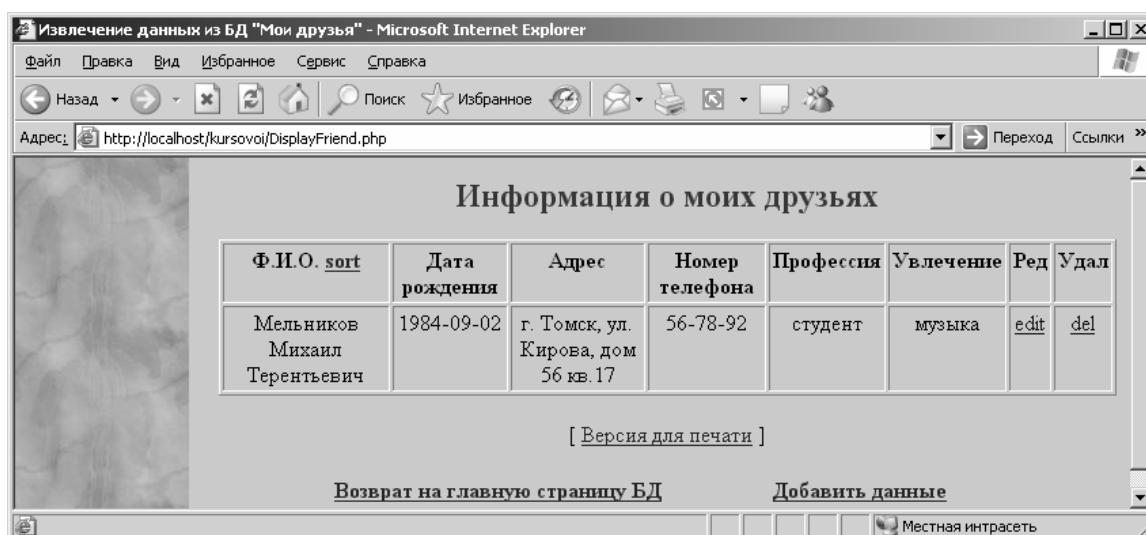


Рис. 10.7 — Данные, извлечённые из базы "Мои друзья" после первого ввода

Вид извлечённых данных из базы "Мои родственники", с помощью сценария "DisplayFriend.php", представлен на рисунке 10.7.

Теперь необходимо написать ещё несколько сценариев, которые позволяли бы выполнять элементарные операции по редактированию, удалению, поиску, сортировке и выводу данных на печать.

11 ЛАБОРАТОРНАЯ РАБОТА №8

Редактирование, поиск, сортировка и вывод данных на печать

Цель работы: практическая работа по сортировке, редактированию и выводу данных на печать.

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, комплекс программ — "Денвер-2".

11.1 Задание на лабораторную работу №8

1. Создать сценарий для сортировки данных в столбцах "Ф.И.О." .
2. Создать сценарий для удаления записей из БД.
3. Создать сценарий для редактирования записей в БД.
4. Создать сценарий для поиска данных в БД.
5. Создать сценарий для вывода данных на печать.

11.2 Отчетность по лабораторной работе

В результате выполнения лабораторной работы должны быть предоставлены следующие материалы.

1. Текстовый файл отчета по лабораторной работе, оформленный в соответствии с существующими требованиями, который должен содержать следующие сведения:

1.3. Титульный лист.

1.4. Детальное описание последовательности всех шагов по выполнению лабораторной работы, с приведением всех экранных форм.

11.3 Сортировка данных

Создадим сценарий для сортировки данных в столбцах "Ф.И.О." наших таблиц. Назовём их именами "SortRelative.php" и "SortFriend.php" соответственно для сортировки в таблице о родственниках и друзьях. Такой сценарий для сортировки дан-

ных в БД "Мои родственники" может быть реализован в следующем виде.

```

1. <HTML>
2. <HEAD>
3. <TITLE>Сортировка данных БД "Мои родственни-
   ки"</TITLE></HEAD>
4. <link rel="stylesheet" href="my.css" type="text/css" />
5. <BODY>
6. <h2>Информация о моих родственниках </h2>
7. <?php
8. // Установка значения переменных для доступа к базе
   данных.
9. Include("Connect.php");
10. $TableName = "Relative";
11. $Query = "SELECT * from $TableName ORDER BY FIO";
12. $Result = mysql_db_query ($DBName, $Query, $Link);
13. // Создание таблицы.
14. print("<TABLE BORDER=1 CELLSPACING=2
   CELLPADDING=2 ALIGN=CENTER>\n");
15. print("<TR ALIGN=CENTER VALIGN=TOP>\n");
16. print("<TH ALIGN=CENTER VALIGN=TOP>Ф.И.О.<a
   href=SortRelative.php > sort</a></TH>\n");
17. print("<TH ALIGN=CENTER VALIGN=TOP>Дата рож-
   дения</TH>\n");
18. print("<TH ALIGN=CENTER
   VALIGN=TOP>Адрес</TH>\n");
19. print("<TH ALIGN=CENTER VALIGN=TOP>Номер те-
   лефона</TH>\n");
20. print("<TH ALIGN=CENTER
   VALIGN=TOP>Профессия</TH>\n");
21. print("<TH ALIGN=CENTER
   VALIGN=TOP>Увлечение</TH>\n");
22. print("<TH ALIGN=CENTER VALIGN=TOP> Степень
   родства</TH>\n");
23. print("<TH ALIGN=CENTER
   VALIGN=TOP>Ред</TH>\n");

```



```

24. print("<TH ALIGN=CENTER
    VALIGN=TOP>Удал</TH>\n");
25. print("</TR>\n");
26. // Получение результатов из базы данных.
27. while ($Row = mysql_fetch_array ($Result)) {
28. print("<TR ALIGN=CENTER VALIGN=TOP>\n");
29. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[FIO]</TD>\n");
30. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[BIRTHDAY]</TD>\n");
31. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[ADDRESS]</TD>\n");
32. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[FONE]</TD>\n");
33. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[PROFESSIONAL]</TD>\n");
34. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[FASCINATION]</TD>\n");
35. print("<TD ALIGN=CENTER
    VALIGN=TOP>$Row[degree_kinship]</TD>\n");
36. print("<td><a
    href='FormEditRelative.php?id=$Row[id]'\>edit</a></td>\n"
    );
37. print("<td><a
    href='DelRelative.php?id=$Row[id]'\>del</a></td>\n");
38. print("</TR>\n");
39. }
40. mysql_close ($Link);
41. print("</TABLE>\n");
42. echo '<br><Center>[ <a target="_blank"
    href="printversion.php?table_print='.$TableName.'&sort='.$
    sort.'">Версия для печати</a> ]<br><br>';
43. ?>
44. <table width="75%" align="Center" >
45. <tr>
46. <td > <a href="Basa.php"><b>Возврат на главную базы
    данных</b></a></td><td ></a></td></p><td ><a

```

href="FormRelative.html">Добавить дан-
ные</td></p>

47. </tr>

48. </table>

49. </BODY>

50. </HTML>

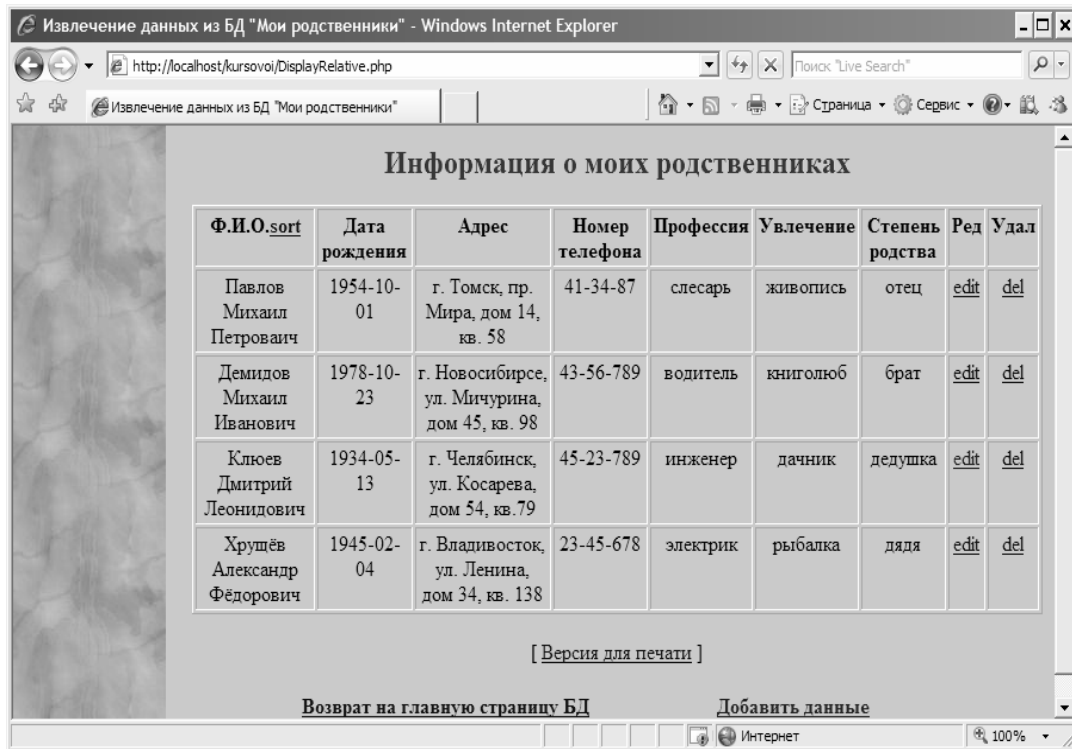


Рис. 11.1 — Данные базы "Мои родственники" до сортировки

Если посмотреть на этот скрипт внимательно, то можно заметить, что он как две капли воды похож на сценарий "DisplayRelative.php". Единственное отличие заключается в строке запроса:

51. \$Query = "SELECT * from \$TableName ORDER BY FIO";

Эта команда выбирает все записи в таблице с именем "\$TableName" из поля "FIO" в алфавитном порядке. Причём символ "*" предписывает в отобранных записях извлечь все поля.

Чтобы этот запрос выполнялся, необходимо, чтобы в таблице было несколько записей. Введём в таблицу "Мои родст-

венники" несколько записей и отсортируем их. На рисунке 11.1 представлено содержание БД "Мои родственники" до сортировки. На рисунке 11.2 то же самое содержание, но после сортировки. Как видно из рисунков, сортировка выполнена — значит, запрос сформирован правильно.

Ф.И.О.	Дата рождения	Адрес	Номер телефона	Профессия	Увлечение	Степень родства	Ред	Удал
Демидов Михаил Иванович	1978-10-23	г. Новосибирск, ул. Мичурина, дом 45, кв. 98	43-56-789	водитель	книголюб	брат	edit	del
Клюев Дмитрий Леонидович	1934-05-13	г. Челябинск, ул. Косарева, дом 54, кв. 79	45-23-789	инженер	дачник	дедущка	edit	del
Павлов Михаил Петрович	1954-10-01	г. Томск, пр. Мира, дом 14, кв. 58	41-34-87	слесарь	живопись	отец	edit	del
Хрущёв Александр Фёдорович	1945-02-04	г. Владивосток, ул. Ленина, дом 34, кв. 138	23-45-678	электрик	рыбалка	дядя	edit	del

[[Версия для печати](#)]

[Возврат на главную страницу БД](#) [Добавить данные](#)

Рис. 11.2 — Данные "Мои родственники" после сортировки

В данном примере, для реализации операции сортировки, мы создали отдельный файл сценария и запустили его со ссылкой. Во многих случаях бывает удобно реализовывать некоторые операции в одном и том же файле. То есть вместо того, чтобы делать две страницы, содержащие разные сценарии для разных действий, мы все действия можем поместить в одну страницу. Тогда с помощью условий "if-else" можно будет управлять ходом выполнения сценария, гоня одну и ту же страницу по кругу. Это, конечно, увеличит размер страницы и время, необходимое на ее обработку, но в некоторых случаях такой способ очень удобен. В этом случае ссылка должна содержать имя и URL текущей страницы и использовать переменную языка PHP — "\$PHP_SELF", которая эквивалентна имени текущей страницы. Обычно это имя и URL содержатся в этой переменной, к которой затем многократно и обращаются. В этом случае,

при активизации ссылки, к имени самой страницы на лету должна быть добавлена некоторая информация: переменные и их значения. Переменные, которые передаются в строке, — автоматически создаются PHP-движком, и к ним можно обращаться так, как если бы вы их создавали в сценарии сами. В примерах, которые приведены в данном учебно-методическом пособии, такой приём не используется. Но он может быть рекомендован для продвинутых разработчиков.

Сценарий для сортировки данных в БД "Мои друзья" будет иметь аналогичный вид.

11.5 Удаление записей из БД

Для удаления записей из БД создадим два сценария с именами "DelRelative.php" и "DelFriend.php". Файл "DelRelative.php" может иметь вид.

```

1. <?php
2. $Host = "localhost";
3. $User = "mydb";
4. $Password = "mydb";
5. $DBName = "mydb";
6. $Link = mysql_connect ($Host, $User, $Password);
7. $TableName = "Relative";
8. $Query = "DELETE FROM Relative WHERE
   id=".$_GET['id'];
9. $Result = mysql_db_query ($DBName,$Query,$Link);
10. print '<h2>Удалён!</h2> ';
11. include("DisplayRelative.php");
12. ?>

```

Назначение строк с № 1 по № 7 должно быть вам уже хорошо известно.

Строка № 8 создаёт запрос удаления записи из таблицы "Relative", отвечающей условию указанного в предложении "WHERE". В данном случае этому условию будет отвечать порядковый номер выбранной записи "id". Значение этого параметра передаётся из сценария "DisplayRelative.php" с помощью

ссылкм "del". В сценарии "DelRelative.php" оно присваивается с помощью переменной "\$_GET['id']".

Строка № 9 выполняет запрос.

Строка № 10 выводит сообщение "Удалён!".

Строка № 11 передаёт управление обратно сценарию "DisplayRelative.php".

Загрузите сценарий на сервер и протестируйте.

Сценарий для удаления данных в БД "Мои друзья" будет иметь аналогичный вид.

11.6 Редактирование записей в БД

Создадим два сценария, которые позволят изменять данные в БД. Назовём эти сценарии именами "FormEditRelative.php" и "FormEditFriend.php". Эти сценарии должны выбирать из соответствующей таблицы указанную запись, выводить её на экран и после редактирования обеспечить её сохранение.

Сценарий для редактирования данных в БД "Мои родственники" может быть реализован в следующем виде.

1. <HTML>
2. <HEAD>
3. <TITLE>Редактирование БД "Мои родственники"
4. </TITLE>
5. <link rel="stylesheet" href="my.css" type="text/css" />
6. </HEAD>
7. <BODY>
8. <FORM ACTION="HandleEditFormRelative.php"
9. METHOD="POST">
10. <h3 ALIGN="Center">Изменение данных в базе "Мои
11. родственники"</h3>
12. <?php
13. \$Host = "localhost";
14. \$User = "mydb";
15. \$Password = "mydb";
16. \$DBName = "mydb";
17. \$Link = mysql_connect (\$Host, \$User, \$Password);

```

15. $TableName = "Relative";
16. $Result = mysql_db_query ($DBName,"SELECT * FROM
    Relative WHERE id=".$_GET['id'],$Link);
17. $Row = mysql_fetch_array ($Result);
18. ?>
19. <table align="Center">
20. <tr>
21. <td>Ф.И.О.:</td><td><INPUT TYPE="TEXT"
    value="<?=$Row['FIO']?>"NAME="Array[FIO]"
    SIZE="40">
22. </tr>
23. <tr>
24. <td>Дата рождения:</td><td><INPUT TYPE="DATE"
    value="<?=$Row['BIRTHDAY']?>"NAME="Array[BIRTH
    DAY]" SIZE="40"> </td>
25. </tr>
26. <tr>
27. <td>Адрес:</td> <td> <INPUT TYPE="TEXT"
    value="<?=$Row['ADDRESS']?>" NAME= "Ar-
    ray[ADDRESS]" SIZE="40"></td>
28. </tr>
29. <tr>
30. <td>Номер телефона:&nbsp;   </td><td> <INPUT
    TYPE="TEXT" value="<?=$Row['FONE']?>" NAME=
    "Array[FONE]"SIZE="40"></td>
31. </tr>
32. <tr>
33. <td>Профессия:</td><td> <INPUT TYPE="TEXT"
    value="<?=$Row['PROFESSIONAL']?>" NAME= "Ar-
    ray[PROFESSIONAL]"SIZE="40"></td>
34. </tr>
35. <tr>
36. <td>Увлечение:</td> <td> <INPUT TYPE="TEXT"
    value="<?=$Row['FASCINATION']?>" NAME= "Ar-
    ray[FASCINATION]"SIZE="40"></td>
37. </tr>
38. <tr>

```

```

39. <td>Степень родства:</td> <td> <INPUT TYPE="TEXT"
    value="<?=$Row['degree_kinship']?>" NAME="Array [de-
    gree_kinship]"SIZE="40"></td>
40. </tr>
41. <tr>
42. <td colspan="2" align="Center" align="Center">
    <b>Отправка или очистка данных</b> </td>
43. </tr>
44. <tr>
45. <td align="Center"> <INPUT TYPE="SUBMIT"
    NAME="Submit" VALUE="Ввести!"></td> <td
    align="Center"> <INPUT TYPE="RESET"
    NAME="Reset" VALUE="Очистить!"> </td>
46. </tr>
47. <tr>
48. <td align="Center"> <a href="Basa.php"><b>Возврат на
    главную базы данных</b></a></td><td align="Center"><a
    href="EditRelative.php"><b>Просмотр БД "Мои родст-
    венники"</b></a></td></tr>
49. </tr>
50. </table>
51. <input type="hidden" name="id" value="<? echo
    $Row["id"]?>">
52. </table>
53. </FORM>
54. </BODY>
55. </HTML>

```

В этом сценарии строка № 7 указывает в качестве обработчика формы сценарий "HandleFormEditRelative.php". В качестве метода используется метод "POST".

Строка № 16 формирует запрос и устанавливает соединение с сервером. В качестве запроса из таблицы "Relative" выбирается запись, номер которой был получен из сценария "DisplayRelative".

Строка № 17 превратит строку информации из запроса (сохраненную в переменной "\$Result") в массив с названием "\$Row". В дальнейшем можно обращаться к каждому отдельно-

му столбцу в извлеченной из базы данных строке точно так же, как к массиву. Ключи массива — названия столбцов таблицы.

Строки с № 19 по № 40 формируют таблицу и помещают в неё найденную информацию указанной записи.

Строка № 45 создаёт элементы управления формой для отправки и очистки данных.

Строка № 48 создаёт навигационные ссылки.

Строка № 51 передаёт невидимый параметр — номер отредактированной строки в сценарий "HandleEditFormRelative.php". Этот сценарий должен обработать изменённые данные, введённые в форму, и разместить их в БД.

Сценарий для редактирования данных таблицы "Мои друзья" — "FormEditFriend.php" будет иметь аналогичный вид.

11.7 Ввод изменённых данных в БД

Создадим сценарий ввода отредактированной записи в БД — "HandleEditFormRelative.php". По своему содержанию он будет очень похож на сценарий первичного ввода данных — "HandleFormRelative.php". Существенным отличием будет только то, что он должен обновлять существующие записи. Для этого необходимо сформировать запрос следующего вида:

```
$Query = "UPDATE Relative set FIO='$Array[FIO]'
,BIRTHDAY= '$Array[BIRTHDAY]', ADDRESS=
'$Array[ADDRESS]',FONE='$Array[FONE]', PROFESSIONAL=
'$Array[PROFESSIONAL]', FASCINATION ='$Array
[FASCINATION]',degree_kinship='$Array[degree_kinship]'
WHERE id='$id';
```

Этот запрос будет обновлять только ту запись, номер которой совпадает с записью, указанной в запросе. В данном случае номер обновленной строки указывается в предложении "WHERE id='\$id' функции "UPDATE". Ниже приводится содержание сценария "HandleEditFormRelative.php" без дальнейших комментариев.

1. <HTML>


```

2. <HEAD>
3. <TITLE>Ввод изменённых данных в БД "Мои родствен-
   ники"</TITLE></HEAD>
4. <BODY>
5. <DIV align="center">
6. <?php
7. /* Удаление пробелов в начале и в конце строк.*/
8. $Array["FIO"] = trim ($Array["FIO"]);
9. $Array["BIRTHDAY"] = trim ($Array["BIRTHDAY"]);
10. $Array["ADDRESS"] = trim ($Array["ADDRESS"]);
11. $Array["FONE"] = trim ($Array["FONE"]);
12. $Array["PROFESSIONAL"] = trim ($Ar-
    ray["PROFESSIONAL"]);
13. $Array["FASCINATION"] = trim ($Ar-
    ray["FASCINATION"]);
14. $Array["degree_kinship"] = trim ($Ar-
    ray["degree_kinship"]);
15. // Установка значения переменных для доступа к базе
    данных.
16. $Host = "localhost";
17. $User = "mydb";
18. $Password = "mydb";
19. $DBName = "mydb";
20. $TableName = "Relative";
21. $Link = mysql_connect ($Host, $User, $Password);
22. mysql_select_db($DBName,$Link);
23. $Query = "UPDATE Relative set FIO='$Array[FIO]'
    ,BIRTHDAY= '$Array[BIRTHDAY]'
    ,ADDRESS='$Array[ADDRESS]',FONE='$Array[FONE]',
    PROFESSIONAL='$Array[PROFESSIONAL]',
    FASCINATION ='$Array[FASCINATION]'
    ,degree_kinship='$Array[degree_kinship]' WHERE
    id='$id';
24. $Result= mysql_db_query($DBName,$Query,$Link) ;
25. $Row = mysql_fetch_array ($Result) ;
26. if (mysql_db_query($DBName, $Query, $Link)) {
27. printf ("Запрос изменения данных успешно выпол-
    нен!<BR>\n");

```

```

28. printf ("Изменены данные о родственнике
    $Array[FIO]<BR>\n");
29. printf ("Вы можете продолжить ввод данных<BR>\n");
30. $data_h=date('H');
31. if ($data_h>=4 and $data_h<12) {$privet='Доброе утро ';}
    // утро
32. if ($data_h>=12 and $data_h<17) {$privet='Добрый день
    ';} // день
33. if ($data_h>=17 and $data_h<24) {$privet='Добрый вечер
    ';} // вечер
34. if ($data_h>=0 and $data_h<4) {$privet='Доброй ночи ';}
    // ночь
35. print '<script LANGUAGE="JavaScript">';
36. print 'alert (".$privet.'!\nДобавлен ваш родствен-
    ник:'. $Array[FIO].'\nДата рождения:
    '.$Array[BIRTHDAY].'\nАдрес жительства:
    '.$Array[ADDRESS].'\nТелефон:'. $Array[FONE].'
    \nПрофессия: '.$Array[PROFESSIONAL].'\nУвлечение:
    '.$Array[FASCINATION].'\nСтепень родства:
    '.$Array[FASCINATION].")';';
37. print '</script>';
38. } else {
39. printf ("Запрос изменения данных не может быть выпол-
    нен!<BR>\n");
40. }
41. mysql_close ($Link);
42. Include("DisplayRelative.php");
43. ?>
44. </DIV>
45. </BODY>
46. </HTML>

```

Для проверки созданных сценариев загрузите их на сервер и протестируйте. В качестве примера изменим запись о родственнике Демидове следующим образом:

Демидович Максим Исакевич
1945-01-07

г. Челябинск ул. Макушина дом 14, кв.234
 34-62-789
 инженер
 филателист
 дядя

Загрузим сценарий "Relative.php", выберем ссылку "edit" на строке таблицы с данными о Демидове, введём новые данные и нажмём кнопку "Ввести". Вид таблицы после редактирования приведён на рисунке 11.3. Как видно из рисунка, все введенные данные правильно отобразились в таблице.

Ввод изменённых данных в БД "Мои родственники" - Windows Internet Explorer

http://localhost/kursovoi/HandleEditFormRelative.php

Имя	Дата рождения	Адрес	Телефон	Профессия	Хобби	Роль	edit	del
Петров	1945-01-07	г. Челябинск, ул. Макушина, дом 14, кв. 234	34-62-789	инженер	филателист	дядя	edit	del
Демидович Максим Исаевич	1945-01-07	г. Челябинск, ул. Макушина, дом 14, кв. 234	34-62-789	инженер	филателист	дядя	edit	del
Клюев Дмитрий Леонидович	1934-05-13	г. Челябинск, ул. Косарева, дом 54, кв. 79	45-23-789	инженер	дачник	дедушка	edit	del
Хрущёв	1945-02-	г. Владивосток	23-45-678	электрик	рыбалка	дядя	edit	del

Рис.11.3 — Данные "Мои родственники" после редактирования

11.8 Поиск данных в БД

Реализуем возможность поиска данных в нашей БД по различным критериям. Для этого создадим сценарий под именем "Search.php". Этот сценарий должен осуществлять поиск по таблицам "Мои друзья" или "Мои родственники" в зависимости от выбранных параметров в выпадающих меню, размещённых на титульной странице БД. Один из вариантов такого сценария может иметь следующий вид.

1. <HTML>
2. <HEAD>
3. <TITLE>Поиск данных в БД </TITLE></HEAD>
4. <link rel="stylesheet" href="my.css" type="text/css" />

```

5. <BODY>
6. <h2>Результаты поиска </h2>
7. <?php
8. // Установка значения переменных для доступа к базе
   данных.
9. Include("Connect.php");
10. $fi_select[0]="";
11. $fi_select[1]="";
12. $fi_select[2]="";
13. $ta_select[0]="";
14. $ta_select[1]="";
15. if ($field=='Фамилия')      {$field_search='FIO';
   $fi_select[0]='selected';}
16. if ($field=='День рождения')
   {$field_search='BIRTHDAY'; $fi_select[1]='selected';}
17. if ($field=='Профессия')
   {$field_search='PROFESSIONAL';
   $fi_select[2]='selected';}
18. if ($table=='Мои друзья')   {$table_search='Friend';
   $rela=""; $ta_select[0]='selected';}
19. if ($table=='Мои родственники') {$table_search='Relative'; $rela=',degree_kinship';
   $ta_select[1]='selected';}
20. // ответ на поиск
21. $lenid=0;
22. $Query = "SELECT FIO, BIRTHDAY, ADDRESS, FONE,
   PROFESSIONAL, FASCINATION ".$rela." FROM
   ".$table_search." WHERE ".$field_search." like
   '".$search."'%";
23. $Result= mysql_db_query($DBName,$Query,$Link) ;
24. print '<b><Center>Результат поиска в БД "'.$table.'" по
   критерию "'.$field.'" и запросу: "'.$search.'"</b><br><br>';
25. if ($table=='Мои друзья') {
26. print"<TABLE BORDER=1 CELSPACING=2
   CELLPADDING=2 ALIGN=CENTER>
27. <TR ALIGN=CENTER VALIGN=TOP>
28. <TH ALIGN=CENTER VALIGN=TOP>Ф.И.О.</TH>
29. <TH ALIGN=CENTER VALIGN=TOP>Дата рожде-
   ния</TH>

```

```

30. <TH ALIGN=CENTER VALIGN=TOP>Адрес</TH>
31. <TH ALIGN=CENTER VALIGN=TOP>Номер телефо-
    на</TH>
32. <TH ALIGN=CENTER VALIGN=TOP>Профессия</TH>
33. <TH ALIGN=CENTER VALIGN=TOP>Увлечение</TH>
34. </TR>\n"; }
35. if ($table=='Мои родственники') {
36. print "<TABLE BORDER=1 CELLSPACING=2
    CELLPADDING=2 ALIGN=CENTER>
37. <TR ALIGN=CENTER VALIGN=TOP>
38. <TH ALIGN=CENTER VALIGN=TOP>Ф.И.О.</TH>
39. <TH ALIGN=CENTER VALIGN=TOP>Дата рожде-
    ния</TH>
40. <TH ALIGN=CENTER VALIGN=TOP>Адрес</TH>
41. <TH ALIGN=CENTER VALIGN=TOP>Номер телефо-
    на</TH>
42. <TH ALIGN=CENTER VALIGN=TOP>Профессия</TH>
43. <TH ALIGN=CENTER VALIGN=TOP>Увлечение</TH>
44. <TH ALIGN=CENTER VALIGN=TOP> Степень родст-
    ва</TH>
45. </TR>\n"; }
46. while ($line = mysql_fetch_array($Result,
    MYSQL_ASSOC)) {
47. print "\t<tr>\n";
48. foreach ($line as $col_value) {
49. print "<td> $col_value</td>\n";
50. }
51. print "\t</tr>\n";
52. }
53. print "</table>";
54. //mysql_close ($Link);
55. echo '<br><Center>[ <a target="_blank"
    href="printversion.php?table_print=search&table='.$table.'&
    field='.$field.'&search='.$search.'">версия для печати</a>
    ]<br><br>';
56. ?>
57. <Center > <a href="Basa.php"><b>Возврат на главную
    БД</b></a></td>
58. </BODY>

```

59. </HTML>

Строки с № 10 по № 14 присваивают соответствующим переменным значения из выпадающих списков, которые возвращаются формой после выбора пользователем данного пункта.

Строки с № 15 по № 19 проверяют совпадение фактических выбранных пользователем значений из списков со значениями переменных. При совпадении значений — переменным, которые в дальнейшем участвуют в поиске, присваиваются определённые значения.

Строка № 22 формирует запрос к серверу, в котором происходит подстановка значений переменных. В этом запросе предложение

WHERE ".\$field_search." like ".\$search."%";

выводит информацию из выбранного поля целиком, даже если в строке поиска был указан лишь первый символ.

Строка № 23 отправляет запрос серверу.

Строка № 24 отображает сообщение о том, в какой таблице, по какому критерию и по какому запросу будет осуществлён

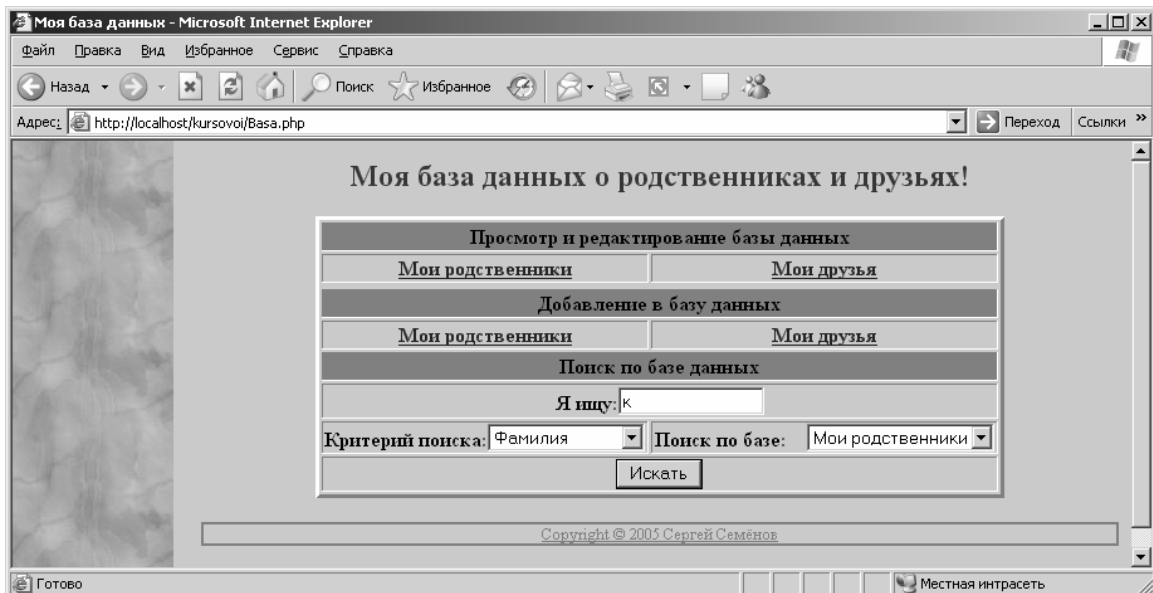


Рис. 11.4 — Титульная страница БД с установленными параметрами поиска

поиск.

Строки с № 23 по № 45 в зависимости от выбранной таблицы отображают соответствующий заголовок таблицы.

Строка № 46 организует цикл, в котором используется тип цикла "foreach (\$line as \$col_value)", предназначенный специально для перебора всех элементов ассоциативного массива.

Строки с № 49 по № 53 выводят на экран всю строку, по полю, указанному в переменной "\$col_value".

Строка № 55 формирует ссылку для подготовки найденных данных к печати.

Протестируем наш сценарий поиска. Для этого установим параметры поиска, приведенные на рисунке 11.4 и нажмем кнопку "Искать". Результат поиска приведён на рисунке 11.5. Как видно из рисунка, поиск осуществлён правильно.

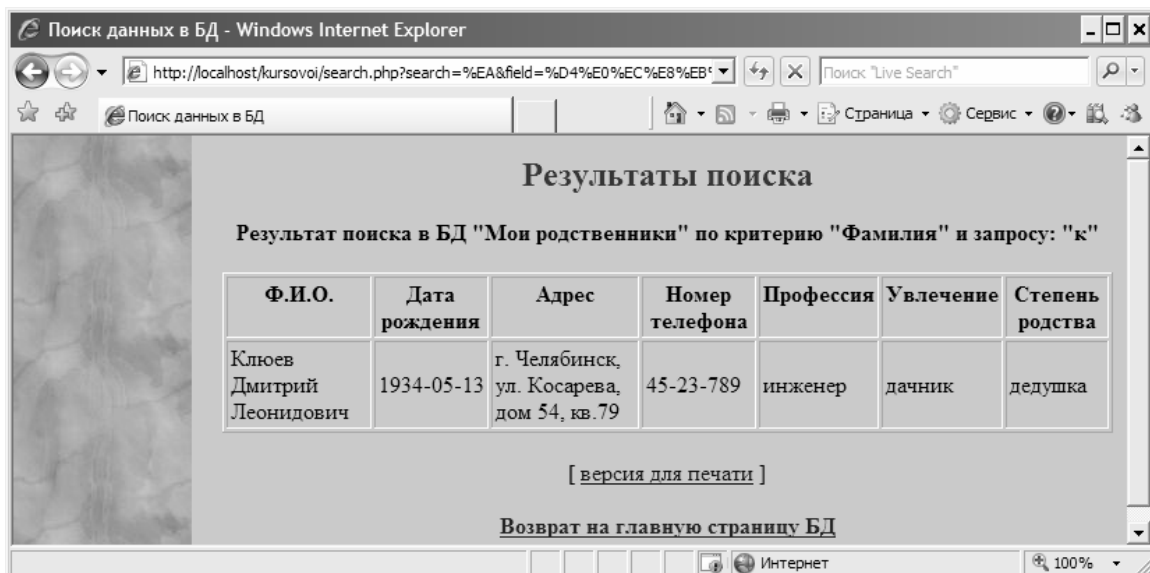


Рис. 11.5 — Результат поиска по БД "Мои родственники"

11.9 Вывод данных на печать

Последний сценарий, который нам необходимо написать, чтобы БД была функционально законченной, это сценарий вывода на печать. Такой сценарий может иметь следующий вид.

```

1. <html>
2. <head>
3. <meta http-equiv="Content-Type" content="text/html;
   charset=windows-1251">
4. <title>Версия для печати</title>
5. <script language="JavaScript">
6. <!--
7. function printThisPage() {
8.   if (document.all) {
9.     var OLECMDID_PRINT = 6;
10.    var OLECMDEXECOPT_DONTPROMPTUSER = 2;
11.    var OLECMDEXECOPT_PROMPTUSER = 1;
12.    var WebBrowser = "<OBJECT ID=\"WebBrowser1\"
      WIDTH=0 HEIGHT=0 CLASSID=\"CLSID:8856F961-
      340A-11D0-A96B-00C04FD705A2\"></OBJECT>";
13.    document.body.insertAdjacentHTML("beforeEnd", Web-
      Browser);
14.    WebBrowser1.ExecWB(OLECMDID_PRINT,
      OLECMDEXECOPT_PROMPTUSER);
15.    WebBrowser1.outerHTML = "";
16.   } else {
17.     self.print();
18.   }
19. }
20. file:!-->
21. </script>
22. </head>
23. <link rel="stylesheet" href="my.css" type="text/css" />
24. <body>
25. <center>
26. <?
27. include('connect.php');
28. global $sort, $table;
29. if ($table_print=='Friend') {
30.   if ($sort=='fio_friend')
31.   {$Query = "SELECT FIO, BIRTHDAY, ADDRESS, FONE,
      PROFESSIONAL, FASCINATION FROM Friend ORDER
      BY FIO asc;" or die; }

```



```

32. else
33. {$Query = "SELECT FIO, BIRTHDAY, ADDRESS, FONE,
    PROFESSIONAL, FASCINATION FROM Friend;";}
34. $Result= mysql_db_query($DBName,$Query,$Link) ;
35. print '<b>Мои друзья:</b><br><br>';
36. print"<TABLE BORDER=1 CELLSPACING=2
    CELLPADDING=2 ALIGN=CENTER>\n";
37. print "<TR ALIGN=CENTER VALIGN=TOP>
38. <TH ALIGN=CENTER VALIGN=TOP>Ф.И.О.</TH>
39. <TH ALIGN=CENTER VALIGN=TOP>Дата рождения
    </TH>
40. <TH ALIGN=CENTER VALIGN=TOP>Адрес</TH>
41. <TH ALIGN=CENTER VALIGN=TOP>Номер телефона
    </TH>
42. <TH ALIGN=CENTER VALIGN=TOP>Профессия</TH>
43. <TH ALIGN=CENTER VALIGN=TOP>Увлечение</TH>
44. </TR>";
45. while ($line = mysql_fetch_array($Result,
    MYSQL_ASSOC)) {
46. print "\t<tr>\n";
47. foreach ($line as $col_value) {
48. print "\t\t<td> $col_value</td>\n";
49. }
50. print "</tr>";
51. }
52. print "</table>";
53. }
54. if ($table_print=='Relative') {
55. $Query = "SELECT FIO, BIRTHDAY, ADDRESS, FONE,
    PROFESSIONAL, FASCINATION, degree_kinship FROM
    relative;";
56. if ($sort=='fio_relative')
57. {$Query = "SELECT FIO, BIRTHDAY, ADDRESS, FONE,
    PROFESSIONAL, FASCINATION, degree_kinship FROM
    relative ORDER BY FIO;";}
58. $Result= mysql_db_query($DBName,$Query,$Link) ;
59. print '<b>Мои родственники:</b><br><br>';
60. print "<TABLE BORDER=1 CELLSPACING=2
    CELLPADDING=2 ALIGN=CENTER>\n";

```

```

61. print "<TR ALIGN=CENTER VALIGN=TOP>
62. <TH ALIGN=CENTER VALIGN=TOP>Ф.И.О.</TH>
63. <TH ALIGN=CENTER VALIGN=TOP>Дата рожде-
    ния</TH>
64. <TH ALIGN=CENTER VALIGN=TOP>Адрес</TH>
65. <TH ALIGN=CENTER VALIGN=TOP>Номер телефо-
    на</TH>
66. <TH ALIGN=CENTER VALIGN=TOP>Профессия</TH>
67. <TH ALIGN=CENTER VALIGN=TOP>Увлечение</TH>
68. <TH ALIGN=CENTER VALIGN=TOP> Степень родст-
    ва</TH>
69. </TR>";
70. while ($line = mysql_fetch_array($Result,
    MYSQL_ASSOC)) {
71. print "\t<tr>\n";
72. foreach ($line as $col_value) {
73. print "\t\t<td>$col_value</td>\n";
74. }
75. print "\t</tr>";
76. }
77. print "</table>";
78. }
79. if ($table_print=='search')
80. {
81. $fi_select[0]="";
82. $fi_select[1]="";
83. $fi_select[2]="";
84. $ta_select[0]="";
85. $ta_select[1]="";
86. if ($field=='Фамилия')          {$field_search='FIO';
    $fi_select[0]='selected';}
87. if ($field=='День рождения')
    {$field_search='BIRTHDAY'; $fi_select[1]='selected';}
88. if ($field=='Профессия')        {$field_search=
    'PROFESSIONAL'; $fi_select[2]='selected';}
89. if ($table=='Мои друзья')      {$table_search='Friend';
    $rela=""; $ta_select[0]='selected';}

```

```

90. if ($table=='Мои родственники') {$table_search='Relative'; $rela=',degree_kinship'; $table_select[1]='selected';}
91. //$lenid=0;
92. $Query = "SELECT FIO, BIRTHDAY, ADDRESS, FONE, PROFESSIONAL, FASCINATION".$rela." FROM ".$table_search." WHERE ".$field_search." LIKE '".$search.'"%";";
93. $Result= mysql_db_query($DBName,$Query,$Link) or die (mysql_error());
94. print '<b>Результат поиска:по "'.$field.'" в "'.$table.'"</b><br><br>';
95. print "<TABLE BORDER=1 CELLSPACING=2 CELLPADDING=2 ALIGN=CENTER>\n";
96. if ($table=='Мои друзья'){
97. print "<tr><th ALIGN=CENTER VALIGN=TOP>Ф.И.О.</th>
98. <th ALIGN=CENTER VALIGN=TOP>Дата рождения</th>
99. <th ALIGN=CENTER VALIGN=TOP>Адрес</th>
100. <th ALIGN=CENTER VALIGN=TOP>Номер телефона</th>
101. <th ALIGN=CENTER VALIGN=TOP>Профессия</th>
102. <th ALIGN=CENTER VALIGN=TOP>Увлечение</th>
103. </tr>";}
104. if ($table=='Мои родственники') {
105. print "<tr><th ALIGN=CENTER VALIGN=TOP>Ф.И.О.</th>
106. <th ALIGN=CENTER VALIGN=TOP>Дата рождения</th>
107. <th ALIGN=CENTER VALIGN=TOP>Адрес</th>
108. <th ALIGN=CENTER VALIGN=TOP>Номер телефона</th>
109. <th ALIGN=CENTER VALIGN=TOP>Профессия</th>
110. <th ALIGN=CENTER VALIGN=TOP>Увлечение</th>
111. <th ALIGN=CENTER VALIGN=TOP>Степень родст-

```

```

        va</TH>
112. </TR>"; }
113. while ($line = mysql_fetch_array($Result,
        MYSQL_ASSOC)) {
114.     print "\t<tr>\n";
115.     foreach ($line as $col_value) {
116.         print "<td> $col_value</td>\n";
117.     }
118.     print "\t</tr>\n";
119. }
120. print "</table>";
121. }

```

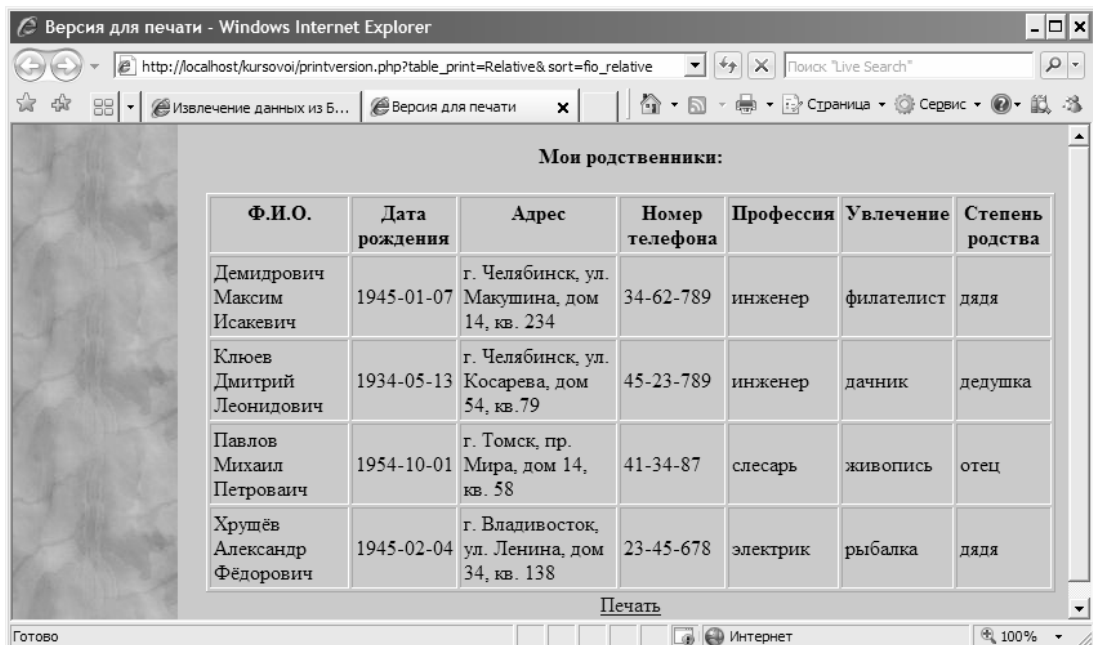


Рис. 11.6 — Данные базы "Мои родственники",
подготовленные для вывода на печать

```

122. mysql_close ($Link);
123. ?>
124. <a href='#' onclick='printThisPage()>Печать</a>
125. </body>
126. </html>

```

Строки с № 5 по № 21 представляют из себя сценарий на языке "JavaScript", который выводит файл собственно на принтер.

Остальная часть листинга, которая вам хорошо известна, подготавливает информацию для вывода на принтер. Протестируем этот сценарий. Загрузим таблицу "Мои родственники", отсортируем столбец "Ф.И.О." и активизируем ссылку "версия для печати". В результате на экране будет представлена информация, приведённая на рисунке 11.6. Вам осталось выполнить окончательную проверку — вывод на принтер.

12 ПЕРЕЧЕРЬ ТЕМ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Директива `@import` языка разметки HTML 4.01.
2. Установка Apache и PHP на сервер Windows.
3. Протокол безопасных соединений SSL.
4. Написание безопасного PHP-кода.
5. Расширения языка разметки HTML 4.01. для браузеров компаний Netscape и Microsoft.
6. Назначение обработчиков событий в JavaScript.
7. Определение классов и идентификаторов в HTML.
8. Права доступа к файлу на сервере-UNIX.
9. Права доступа к файлу на Windows-сервере.
10. Проблемы совместимости HTML-кода в различных браузерах.
11. Способы повышения трафика WEB-сервера.

ПРИЛОЖЕНИЕ**Пример оформления отчёта по лабораторной работе
№1**

Федеральное агентство по образованию РФ
ТОМСКИЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ

ОТЧЁТ
по лабораторной работе №1

**Создание простой HTML-страницы с гиперссыл-
ками**

Выполнил студент гр. 24-058 З/Ф
Семёнов С.С.

Проверил доцент каф. КСУП
Губин И.Г.

" ____ " _____ 2005 г.

ТОМСК 2007

Цель работы: практическая работа по созданию простой HTML-страницы с гиперссылками.

Используемое методическое и программное обеспечение: учебное и учебно-методическое пособия, текстовый редактор "Блокнот".

1. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ В ЛАБОРАТОРНОЙ РАБОТЕ ЭЛЕМЕНТОВ HTML

(В этом разделе необходимо детально описать следующие элементы языка HTML со всеми атрибутами).

1. Структура HTML-документа.
2. Информация о версии, типе документа HTML, языке.
3. Элемент HTML.
4. Элемент HEAD.
5. Элемент TITLE.
6. Элемент BODY.
7. Заголовки – элементы H1, H2,...H6.
8. Ссылки – элемент A.
9. Параграфы – элемент P.

2. ОПИСАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

(В этом разделе должны быть подробно описаны в хронологическом порядке все действия по выполнению задания по лабораторной работе).

1. Создаем на жёстком диске каталог "C:\MyWeb", в котором будут храниться все HTML-файлы.

2. В редакторе "Блокнот" создаём заготовку для всех новых HTML-файлов следующего содержания:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0  
Transitional//EN">
```

```
<HTML>  
  <HEAD>  
    <TITLE> </TITLE>  
  </HEAD>  
<BODY>  
</BODY>
```

3. Сохраняем этот документ под именем "index.htm".
(Дальше вы должны привести описание остальных пунктов выполнения лабораторной работы).

3. РЕЗУЛЬТАТЫ РАБОТЫ

В результате выполнения лабораторной работы были изучены структура HTML-документа, элементы языка HTML – HEAD, TITLE, BODY, заголовки (H1, H2,...H6), А, Р и создана титульная страница HTML-документа со ссылками на другие страницы.

В качестве результатов необходимо предоставить.

1. Отчёт по лабораторной работе №1 – текстовый файл в формате Word.

2. HTML-файлы (welcome.html, study.html, about_me.html, hobby.html, address.html).