

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Курганский государственный университет»

Кафедра программного обеспечения автоматизированных систем

WEB-ПРОГРАММИРОВАНИЕ

Методические указания
к выполнению лабораторных работ
для студентов направления подготовки 09.03.04
«Программная инженерия»

Курган 2015

Кафедра: «Программное обеспечение автоматизированных систем»

Направление подготовки: 09.03.04 «Программная инженерия»

Дисциплина: «Web-программирование»

Составил: канд. техн. наук, доцент А.М. Семахин.

Утверждены на заседании кафедры «30» сентября 2015 г.

Рекомендованы методическим советом университета «19» декабря 2015 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Протоколы и модели Internet-взаимодействия	5
1.1 Протоколы	5
1.2 Модель клиент-сервер	5
1.3 Формат описания адреса	6
2 Статическое web-программирование	6
2.1 Язык HTML	6
2.1.1 Структура HTML-документа	7
2.1.2 Создание гиперссылок	7
2.1.3 Лабораторная работа №1 «Разработка сайта, содержащего изображения и гиперссылки на web-страницы»	7
2.2 Работа с текстом и картой изображения	8
2.2.1 Форматирование текста	8
2.2.2 Создание Image Map и активных зон	9
2.2.3 Лабораторная работа №2 «Разработка сайта, содержащего карту-изображение с заданными активными зонами»	9
2.3 Работа с таблицами	11
2.3.1 Дескрипторы описания таблицы	11
2.3.2 Теги группирования элементов таблиц	11
2.3.3 Лабораторная работы №3 «Разработка сайта, содержащего таблицы»	12
2.4 Работа с формами и фреймами	13
2.4.1 HTML-элементы для создания формы	13
2.4.2 Описание фреймов	14
2.4.3 Лабораторная работа №4 «Разработка сайта “Электронный тест проверки знаний студентов изучаемых дисциплин»	14
2.5 Каскадные таблицы стилей (CSS)	16
2.5.1 Синтаксис CSS	16
2.5.2 Использование CSS	16
2.5.3 Свойства CSS	17
3 Динамическое web-программирование. JavaScript сценарии	18
3.1 Клиентские и серверные сценарии	18
3.1.1 Теоретическое введение	18
3.1.2 Лабораторная работа №5 «Разработка сайта с динамическими эффектами»	18
3.2 Операторы и функции JavaScript	19
3.2.1 Управление ходом выполнения сценария JavaScript	19
3.2.2 Функции JavaScript	20
3.2.3 Лабораторная работа №6 «JavaScript. Горизонтальное и вертикальное меню»	21

3.3 Встроенные объекты JavaScript	22
3.4 Динамический HTML	23
3.5 AJAX технология	23
3.5.1 Принципы AJAX технологии	23
3.5.2 Преимущества и недостатки	24
3.5.3 Лабораторная работа №7 «Технология фоновой обмена данными браузера с веб-сервером в информационных сетях»	24
4 Язык PHP и СУБД MySQL	25
4.1 СУБД MySQL	26
4.1.1 Особенности СУБД MySQL	26
4.1.2 Основные команды	26
4.1.3 Лабораторная работы №8 «Динамическая поддержка выбора значений из предлагаемого списка»	27
4.2 Язык PHP	29
4.2.1 Особенности языка PHP	30
4.2.2 Язык PHP и объектно-ориентированное программирование	30
4.2.3 Лабораторная работа №9 «Проектирование сайта электронной коммерции»	31
4.2.4 Лабораторная работы №10 «Разработка интернет-магазина»	33
5 Язык XML и PHP	34
5.1 Структура XML документа	35
5.2 Лабораторная работа №11 «Разработка SAX-парсера новостной ленты»	35
ЗАКЛЮЧЕНИЕ	37
СПИСОК ЛИТЕРАТУРЫ	38

ВВЕДЕНИЕ

Дисциплина «Web-программирование» имеет целью дать студентам теоретические знания и практические навыки в проектировании и реализации web-сайтов, разработке web-приложений.

Предмет дисциплины – технология разработки сценариев для интернета.

Задачи дисциплины – дать представление о принципах web-программирования и основных этапах создания объектно-ориентированных web-приложений.

Лабораторный практикум (34 часов).

Методические указания содержат теоретическое обоснование и варианты заданий для выполнения лабораторных работ по дисциплине web-программирование.

Методические указания разработаны в соответствии с требованиями государственного образовательного стандарта по подготовке бакалавров по направлению 09.03.04 «Программная инженерия».

1 Протоколы и модели Internet-взаимодействия

1.1 Протоколы

Протокол – набор правил, регламентирующий обмен информацией между устройствами. Интернет – множество взаимодействующих сетей. Для передачи информации в интернете используются протоколы:

- TCP (Transmission Control Protocol) – протокол управления передачей. Позволяет двум узлам сети установить соединение и обмениваться потоками данных.

- IP (Internet Protocol) – протокол Internet. Поддерживает адресацию узлов, позволяет определять тип сервиса и предпринять меры по защите данных.

- HTTP (Hypertext Transfer Protocol) – протокол передачи гипертекстовой информации. Используется браузерами и серверами для передачи данных.

- FTP (File Transfer Protocol) – протокол передачи файлов. Применяется для организации обмена файлами между компьютерами, подключенными к сети.

1.2 Модель клиент/сервер

Модель клиент/сервер описывает взаимодействие между двумя компьютерами, согласно которому клиент запрашивает у сервера услуги, а сервер обслуживает запрос. Модель клиент/сервер удобна для описания взаимодействия программ, находящихся в различных узлах сети. В интернете компьютеры, используемые для обращения к глобальной сети, выполняют функции клиентов, а компьютеры, на которых хранится информация, действуют как серверы. Аппаратные и программные средства, включая DNS,

брандмауэры, маршрутизаторы и шлюзы, обеспечивают попадание по назначению запроса клиента и информации, возвращаемой сервером.

1.3 Формат описания адреса

URL (Uniform Resource Locator) – унифицированный локатор ресурсов. URL – единый формат описания адреса web-узла или файла. В его состав входит информация об используемом протоколе web-узла, на котором расположен требуемый ресурс, подкаталог, в котором находится ресурс и имя файла. Структура URL показана на рисунке 1.1.

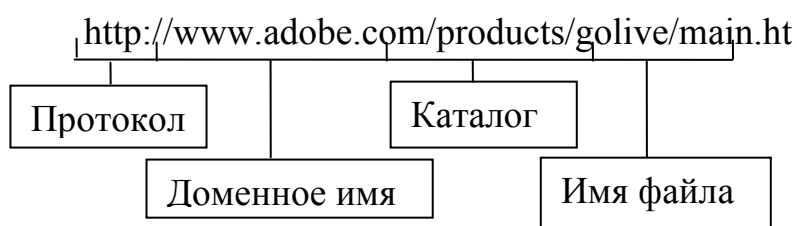


Рисунок 1.1 – Структура формата описания адреса

URL состоит из следующих компонентов:

- 1 `http` – тип протокола, используемый при хранении и передаче информации.
- 2 `www.adobe.com` – доменное имя, идентифицирующее узел сети, на котором хранится ресурс.
- 3 `products/golive` – каталог, в котором ресурс хранится на удаленном компьютере.
- 4 `main.html` – имя файла, который должен быть передан по сети [1].

2 Статическое web-программирование

2.1 Язык HTML

HTML (HyperText Markup Language) – язык разметки гипертекста. HTML – гипертекстовый язык разметки, позволяющий посредством специальных инструментов (тегов) определять структуру и внешний вид веб-страницы при ее отображении в браузере, создавать ссылки на другие веб-страницы и файлы.

Язык HTML разработан британским ученым Тимом Бернерсом Ли в 1989 г. в Женеве. HTML-документ – файл в формате ASCII. В основе лежат специальные дескрипторы (теги), которые определяют правила форматирования данных.

Тег (дескриптор) – специальный управляющий код, записанный в тексте в угловых скобках. HTML-файл имеет расширение `.htm` или `.html`. Для просмотра

HTML-документа необходимо специальное программное обеспечение (браузер).

Веб-страница – логическая единица всемирной паутины, определяемая адресом (URL) и представляющая собой файл HTML-формата.

Сайт – совокупность логически связанных веб-страниц, представляющих собой единое целое и находящихся, как правило, на одном компьютере (сервере) [2;3].

2.1.1 Структура HTML-документа

HTML-документ обозначается парными тегами <HTML> и </HTML>. Структурно HTML-документ разделяется на две части: заголовочную и основную. Заголовочная часть размещается между тегами <HEAD> и </HEAD>. Основная – между тегами <BODY> и </BODY>. Внутри заголовочной части <HEAD> и </HEAD> размещается контейнер из тегов <TITLE> и </TITLE>, содержащий текст заголовка [2].

2.1.2 Создание гиперссылок

Гиперссылка – слово или изображение на веб-странице, содержащее ссылку на другие страницы или файлы. Технология гиперссылок позволяет объединить множество документов в один.

Гиперссылки создаются с помощью парного дескриптора <A>:

гиперссылка.

В качестве адреса используют

- адрес URL на другую web-страницу;
- адрес URL на документ FTP-сервера;
- адрес URL на мультимедийный ресурс в интернете;
- закладка на текущей web-странице;
- адрес электронной почты;
- сценарий на языке JavaScript.

2.1.3 Лабораторная работа №1

«Разработка сайта, содержащего изображения и гиперссылки на web-страницы»

Цель: получить теоретические знания и практические навыки в использовании технологии HTML, CSS и гиперссылок.

Используемые приемы и технологии: HTML, слои, каскадные стилевые таблицы.

Ключевые термины: HTML-документ, тег, контейнер, браузер, кодировка, якорь, символ «#»

Варианты

- 1 Выдающиеся спортсмены России.
- 2 Выдающиеся музыканты России.
- 3 Выдающиеся художники России.
- 4 Выдающиеся политические деятели России.
- 5 Выдающиеся артисты эстрады России.
- 6 Выдающиеся писатели России.
- 7 Выдающиеся поэты России.
- 8 Выдающиеся певцы России.
- 9 Выдающиеся композиторы России.
- 10 Выдающиеся драматурги России.
- 11 Выдающиеся актеры кино, театра России.
- 12 Выдающиеся космонавты России.
- 13 Выдающиеся кинокритики России.
- 14 Выдающиеся кинорежиссеры России.
- 15 Выдающиеся бизнесмены России.
- 16 Выдающиеся депутаты Государственной Думы России.
- 17 Выдающиеся врачи России.
- 18 Выдающиеся полководцы России.
- 19 Почетные граждане г. Кургана.
- 20 Лауреаты Нобелевской премии.

Методические указания

1 Продумайте структуру сайта. Выполните поиск графического и текстового материала для информационного контента сайта.

2 Создайте web-страницу **Index.html**, на которой разместите 5-6 графических изображений, используя абсолютное позиционирование.

3 Создайте в файле **Style.css** требуемую таблицу стилей для оформления страниц по теме варианта задания. Введите в ней описание стилей **body** (фоновая картинка), **h2** (тип шрифта в ячейке таблицы), **.text** (заголовок), **.t1** (стиль рамки для первой ячейки), **.t2** (стиль рамки для второй ячейки), **.link** (стиль для шрифта текста, который является гиперссылкой), **.d1** (абсолютное положение таблицы на странице), **a:visited** и **a:hover** (цвет гиперссылок).

4 Создайте web-страницы с биографиями ученых, используя подключаемую к ним каскадную стилевую таблицу **Style.css**.

2.2 Работа с текстом и картой изображения

2.2.1 Форматирование текста

Теги, предназначенные для работы с текстом, позволяют форматировать текст, красиво располагать его на странице и выводить в виде всплывающих подсказок.

Тег позволяет с помощью атрибутов управлять размером и цветом шрифта. Атрибут SIZE управляет размером шрифта. Тег <P> позволяет разделить текст на абзацы. Тег <PRE> позволяет представлять на экране текст со специфическим форматированием. Тег <NOBR> позволяет отобразить весь текст в одной строке, не разрывая его. В листинге 2.1 приведен пример создания веб-страницы, содержащей текст в одной строке.

Листинг 2.1 Веб-страница, содержащая текст в одной строке.

```
<HTML><HEAD><TITLE> Строки без разрыва </TITLE>  
</HEAD><BODY><NOBR> Строка является длинной строкой  
документа, которая не допускает разбиения </NOBR>  
</BODY></HTML>
```

2.2.2 Создание Image Map и активных зон

Карта изображений (Image Map) — это графический объект языка разметки HTML, связанный с изображением и содержащий специальные области (активные зоны), при нажатии на которые происходит переход по определённому URL.

Для задания описания карты изображения используется парный дескриптор <MAP>. Например, <MAP name=имя_карты>...</MAP> .

Для описания активной зоны применяется дескриптор <AREA>.

- <AREA shape=rect coords="координаты верхнего левого и правого нижнего угла">//Описание прямоугольной активной зоны;
- <AREA shape=circle coords="координаты центра и радиус"> //Описание активной зоны в форме круга;
- <AREA shape=poly coords="координаты вершин в порядке построения">//Описание активной зоны в форме многоугольника;
- <AREA shape=default">//По умолчанию активная зона – изображение.

2.2.3 Лабораторная работы №2

«Разработка сайта, содержащего карту-изображение с заданными активными зонами»

Цель: получить теоретические знания и практические навыки в использовании технологии HTML, CSS и активных зон.

Используемые приемы и технологии: HTML, слои, каскадные стилевые таблицы, карта-изображение (ImageMap).

Ключевые термины: гиперссылка, гипермедиаадрес, карта-изображение.

Варианты

1 План экспозиции музея боевой Славы ветеранов Великой Отечественной войны 1941-1945 годов.

- 2 План экспозиции картинной галереи современных художников России.
- 3 План экспозиции выставки достижений народного хозяйства.
- 4 Карта метрополитена г.Москвы.
- 5 План экспозиции международной выставки достижений науки и техники.
- 6 План экспозиции международной выставки современной авиационной техники.
- 7 План экспозиции международной выставки вооружений, боевой техники и средств связи.
- 8 Олимпийские объекты зимней Олимпиады Сочи-2014.
- 9 Достопримечательности и памятники Курганской области.
- 10 Высшие и средние технические учебные заведения Курганской области.
- 11 Города России, входящие в «Золотое кольцо».
- 12 План экспозиции международной выставки достижений автомобильной промышленности.
- 13 План экспозиции международной выставки достижений вычислительной техники.
- 14 Крупнейшие предприятия и организации Курганской области.
- 15 Материнская плата персонального компьютера.
- 16 План экспозиции Кремля г. Москвы.
- 17 Устройство системного блока персонального компьютера.
- 18 Достопримечательности и памятники г. Москвы.
- 19 Карта метрополитена г.Санкт-Петербурга.
- 20 Карта научно-исследовательских станций Антарктиды.

Методические указания

1 Продумайте структуру сайта. Выполните поиск графического и текстового материала для информационного контента сайта.

2 Загрузите рисунок *.jpg в графический редактор **Microsoft Paint**. Наведите курсор мыши на место изображения, которое будет являться опорной точкой активной области создаваемой карты-изображения (ImageMap). В строке состояния окна **Paint** прочтите координаты точки. Решите, какая область (прямоугольник, окружность или многоугольник) будет являться активной областью, и определите координаты опорных точек.

3 Создайте стилевую таблицу в отдельном файле с именем **Style.css**. Введите описание стилей **body** (фоновая картинка), **h2** (тип шрифта в ячейке таблицы), **.text** (заголовок), **.t1** (стиль рамки для первой ячейки), **.t2** (стиль рамки для второй ячейки), **.link** (стиль для шрифта текста, который является гиперссылкой), **.d1** (абсолютное положение таблицы на странице), **a:visited** и **a:hover** (цвет гиперссылок).

4 Создайте требуемые web-страницы, используя стилевую таблицу **Style.css**.

2.3 Работа с таблицами

2.3.1 Дескрипторы описания таблицы

Парный тег `<TABLE>` и `</TABLE>` используется для описания таблиц. Ячейки таблиц в языке HTML могут содержать любые HTML-элементы: заголовки, списки, текстовые абзацы, графику и элементы форм.

Тег `<TABLE>` имеет атрибуты:

- `align="left|center|right"` – определяет положение подписи таблицы;
- `bgcolor="Цвет"` – задает фоновый цвет;
- `border="Толщина границы в пикселах"` – определяет вид границы таблицы;
- `cellpadding="Расстояние между содержимым ячейки и ее краев"` – определяет расстояние от границ ячейки до ее содержимого в пикселах;
- `cellspacing="Расстояние между ячейками таблицы"` – определяет расстояние между ячейками в пикселах;
- `width="Ширина"` – определяет ширину таблицы.

Для создания строк и столбцов таблицы служат парные теги `<TR>` и `</TR>`, `<COL>` и `</COL>`. Для создания ячейки таблицы служит парный тег `<TD>` и `</TD>`. Для создания заголовка таблицы служит парный тег `<TH>` и `</TH>`. Для группирования элементов служат теги `<COLGROUP>`, `<TBODY>` и `<TFOOT>`. В листинге 2.2 приведен пример создания таблицы.

Листинг 2.2 Создание таблицы из двух строк и семи столбцов, в которой надписи центрированы.

```
<HTML><HEAD><TITLE> Таблица из двух строк и семи столбцов  
</TITLE></HEAD>  
<BODY><H3> Таблица из двух строк и семи столбцов </H3>  
<TABLE CELLSPACING=0 CELLPADDING=0 WIDTH="60%"  
BORDER=4><TR ALIGN=CENTER> <TH BGCOLOR=BLACK>  
<TH BGCOLOR=WHITE>a12 <TH BGCOLOR=BLACK>  
<TH BGCOLOR=WHITE>a14 <TH BGCOLOR=BLACK>  
<TH BGCOLOR=WHITE>a16 <TH BGCOLOR=BLACK></TR>  
<TR ALIGN=CENTER> <TH BGCOLOR=WHITE>a21  
<TH BGCOLOR=BLACK> <TH BGCOLOR=WHITE>a23  
<TH BGCOLOR=BLACK><TH BGCOLOR=WHITE>a25  
<TH BGCOLOR=BLACK><TH BGCOLOR=WHITE>a27</TR>  
<TH BGCOLOR=BLACK></TR></TABLE></BODY></HTML> [2].
```

2.3.2 Теги группирования элементов таблиц

Для группирования элементов таблиц применяют теги `<COLGROUP>`, `<TBODY>`, `<TFOOT>`.

Тег `<COLGROUP>` используется для группирования столбцов таблицы и имеет атрибуты:

- BGCOLOR – задает цвет фона ячеек.
 - ALIGN – задает режим горизонтального выравнивания содержимого внутри ячейки.
 - VALIGN – задает режим вертикального выравнивания содержимого внутри ячейки.
 - SPAN – задает количество столбцов в группе.
 - WIDTH – задает ширину ячеек в столбцах.
 - ID – задает имя для ссылки.
- Тег <TBODY> предназначен для создания логически связанных групп ссылок в теле таблицы.
- Тег <TFOOT> применяется для логического группирования строк в нижней части таблицы [2].

2.3.3 Лабораторная работа №3

«Разработка сайта, содержащего таблицы»

Цель: получить теоретические знания и практические навыки в использовании технологии HTML, CSS и таблиц.

Используемые приемы и технологии: HTML, слои, каскадные стилевые таблицы.

Ключевые термины: таблица, строка, ячейка, фон, разлиновка, обрамление.

Варианты

- 1 Устройства ввода-вывода информации.
- 2 Модельный ряд видеокарт.
- 3 Модельный ряд материнских карт.
- 4 Модельный ряд модемов.
- 5 Модельный ряд сетевых карт.
- 6 Модельный ряд косметики.
- 7 Модельный ряд персональных компьютеров.
- 8 Модельный ряд автомобилей фирмы Toyota.
- 9 Модельный ряд спутниковых ресиверов.
- 10 Модельный ряд карт спутникового интернета.
- 11 Модельный ряд авиационной техники.
- 12 Модельный ряд оборудования кабельного телевидения.
- 13 Модельный ряд бытовых источников электроэнергии.
- 14 Модельный ряд музыкальных инструментов.
- 15 Модельный ряд боевой техники РФ.
- 16 Модельный ряд современной космической техники.
- 17 Модельный ряд современных средств связи.
- 18 Устройства обработки информации.

- 19 Устройства хранения информации.
- 20 Конфигурация персонального компьютера.

Методические указания

1 Продумайте структуру сайта. Выполните поиск графического и текстового материала для информационного контента сайта.

2 Создайте стилевую таблицу в отдельном файле с именем **Style.css**. Введите описание стилей **body** (фоновая картинка), **h2** (тип шрифта в ячейке таблицы), **.text** (заголовок), **.t1** (стиль рамки для первой ячейки), **.t2** (стиль рамки для второй ячейки), **.link** (стиль для шрифта текста, который является гиперссылкой), **.d1** (абсолютное положение таблицы на странице), **a:visited** и **a:hover** (цвет гиперссылок).

3 Создайте web-страницы, используя подключаемую стилевую таблицу **Style.css**.

4 Разместите на страницах графические изображения.

2.4 Работа с формами и фреймами

2.4.1 HTML-элементы для создания формы

Форма – набор элементов, предназначенных для ввода пользователем информации и обмена этой информацией с web-сервером. Форма создается при помощи тегов:

<FORM> – для создания формы внутри HTML-документа;

<INPUT> – для добавления элемента ввода;

<OPTION> – для создания элементов внутри списка SELECT;

<SELECT> – для создания списков выбора из ограниченного числа элементов;

<TEXTAREA> – для создания многострочного текстового поля.

Атрибуты тега <FORM>:

NAME – определяет имя формы;

ACTION – задает URL-адрес программы;

METHOD – определяет способ отправки параметров формы;

TARGET – определяет окно, в которое возвращается результат обработки отправленной формы.

В листинге 2.3 приведен пример создания формы.

Листинг 2.3 Создание веб-страницы, содержащей форму.

```
<HTML><HEAD><TITLE> Формы </TITLE></HEAD>
```

```
<BODY><FORM><CENTER><H3> Элементарная форма </H3>
```

```
ПЕРВОЕ ПОЛЕ <INPUT NAME="ПЕРВОЕ ПОЛЕ" SIZE="40"><BR>
```

```
<INPUT NAME="ВТОРОЕ ПОЛЕ" SIZE="60">ВТОРОЕ ПОЛЕ<BR>
```

```
</CENTER></FORM></BODY></HTML> [2].
```

2.4.2 Описание фреймов

Фрейм – элемент языка HTML, позволяющий разделить web-страницу на несколько независимых окон, в каждом из которых размещают отдельную web-страницу. Для создания фреймов используются дескрипторы <FRAMESET> и <FRAME>.

Тег <FRAMESET> описывает все фреймы, на которые делится экран. Экран можно разделить на несколько вертикальных или несколько горизонтальных фреймов. Для описания каждого фрейма в отдельности служит тег <FRAME>.

Тег <FRAMESET> имеет атрибуты:

- COLS – задает количество и размеры столбцов в создаваемом наборе фреймов;
- ROWS – задает количество и размеры строк в создаваемом наборе фреймов в пикселах, в процентах или связанных масштабных коэффициентах;
- BORDER – задает толщину обрамления в пикселах для всех фреймов, входящих в данный набор фреймов;
- FRAMEBORDER – позволяет отменить вывод обрамления у фреймов независимо от значения атрибута BORDER.

В листинге 2.4 приведен пример создания фрейма.

Листинг 2.4 Создание веб-страницы, содержащей два горизонтальных фрейма.

```
<HTML><HEAD><TITLE> Два горизонтальных фрейма </TITLE>  
</HEAD> <FRAMESET FRAMEBORDER=1 BORDER="5"  
ROWS="480, 200"><FRAME SRC="1.htm">  
<FRAME SRC="2.htm"></FRAMESET></HTML>[2].
```

2.4.3 Лабораторная работа №4

«Разработка сайта “Электронный тест проверки знаний студентов изучаемых дисциплин”»

Цель: получить теоретические знания и практические навыки в использовании технологии HTML, CSS, форм, фреймов и JavaScript-сценариев.

Используемые приемы и технологии: HTML, каскадные стилевые таблицы, слои, фреймовая структура, форма.

Ключевые термины: фрейм, полоса прокрутки, бегущая строка, гиперссылка, форма, интерактивная форма, флажок, переключатель («радиокнопка»), группа флажков, группа переключателей, кнопка сброса, кнопка подтверждения.

Варианты

- 1 Электронный тест по компьютерной графике.
- 2 Электронный тест по объектно-ориентированному программированию.
- 3 Электронный тест по объектно-ориентированному проектированию.

- 4 Электронный тест по web-программированию.
- 5 Электронный тест по структуре и алгоритмам обработки данных.
- 6 Электронный тест по инженерной графике.
- 7 Электронный тест по интерактивным графическим системам.
- 8 Электронный тест по информатике.
- 9 Электронный тест по теории вероятности.
- 10 Электронный тест по математическому программированию.
- 11 Электронный тест по вычислительным сетям.
- 12 Электронный тест по компьютерному моделированию.
- 13 Электронный тест по проектированию баз данных.
- 14 Электронный тест по обработке экспериментальных данных на ЭВМ.
- 15 Электронный тест по высшей математике.
- 16 Электронный тест по программированию на языке высокого уровня.
- 17 Электронный тест по архитектуре вычислительных систем.
- 18 Электронный тест по человеко-машинному взаимодействию.
- 19 Электронный тест по электротехнике и электронике.
- 20 Электронный тест по технологии разработки программного обеспечения.

Методические указания

1 Продумайте структуру сайта. Выполните поиск графического и текстового материала для информационного контента сайта. Главная страница **Index.html** должна состоять из двух фреймов, в первом из которых размещается меню (список вопросов), а во втором – изначально выводится иллюстрация по теме тестирования, а при выборе вопросов в меню выводится текст вопросов.

2 Создайте страницу с именем **1.html**, соответствующую первому вопросу. Она должна состоять из двух фреймов, в которые загружаются страницы с именами **1-1.html**, **2-1.html**.

3 Аналогичным способом создайте страницы, соответствующие остальным вопросам. Они также должны состоять из двух фреймов, в которые загружаются страницы с соответствующими именами.

4 Для автоматизации подсчета правильных ответов на вопросы ограниченные возможности HTML дополняют сценариями. Сценарий (скрипт) может быть написан на языке JavaScript или VBScript и представляет собой программный код для создания динамических web-страниц. Преимущество сценариев – простота. Чтобы добавить код сценария в HTML-документ, используют тег `<script>`, который располагается в пределах тегов `<head>` `</head>` или `<body>` `</body>`.

5 После последнего вопроса необходимо создать две кнопки: кнопку **Проверить** (при нажатии которой в текстовой строке **Результат** выводится количество правильных ответов) и кнопку сброса **Стереть**, нажатие которой отменяет все сделанные изменения.

2.5 Каскадные таблицы стилей (CSS)

Стиль определяет особенность форматирования и отображения элементов веб-страниц. Стили HTML-, XHTML- и XML-документов задаются с помощью каскадных таблиц стилей (Cascading Style Sheets, CSS).

CSS позволяют:

- 1 Создать файл, содержащий листы стилей, и ссылаться на него из HTML-документов.
- 2 Осуществить контроль над шрифтами, цветом фона и другими характеристиками, влияющими на отображение документа.
- 3 Быстро и легко изменять внешний вид страницы.
- 4 Уменьшить время загрузки документов.

2.5.1 Синтаксис CSS

Каскадные таблицы стилей состоят из правил, отображающих особенности отображения элементов web-страниц. Правило состоит из двух частей: селектора и декларации. Селектор определяет элемент, на который воздействует правило. Декларация содержит инструкции о том, как должен быть обработан элемент, свойство и значение. Правило записывается следующим образом: Селектор {свойство: значение}.

Селектор можно рассматривать как обозначение дескриптора, к которому применяется правило, а свойство – как атрибут, который необходимо изменить. Свойство и его значение помещаются в фигурные скобки и разделяются двоеточием.

Формат записи стилевых правил CSS похож на табличное представление данных. Заголовок таблицы напоминает описание элемента, класса или идентификатора стиля, в качестве ячеек и рядов таблицы выступают свойства и значения стилей.

В листинге 3.1 приведен пример записи CSS.

Листинг 3.1 Формат записи стилевых правил CSS.

A.menu:link

```
{COLOR: #c2ala0; FONT-FAMILY: Arial; FONT-SIZE: 14px;  
FONT-WEIGHT: bold; TEXT-DECORATION: none }
```

При использовании стилей в коде необходимо включить в заголовок документа META-определение: <META http-equiv=Content- Style-Type content="text/css">.

Стили предоставляют разработчику максимальный уровень контроля над особенностями отображения веб-страниц [3].

2.5.2 Использование CSS

Существуют четыре варианта использования листов стилей в HTML-документе: встраивание, подключение, связывание и импорт.

Встраиваемые стили определяются в HTML-теге и оказывают влияние только на этот тег.

Подключаемые (внутренние) стили применяются ко всей веб-странице. Для этого служит тег <STYLE>, помещаемый в заголовок HTML-документа.

Связываемые (внешние) стили применяются к документам всего web-сайта.

Импортируемые стили используют импорт в сочетании с другими методами определения стилей. В листинге 3.2 приведен пример импорта внешних стилей.

Листинг 3.2 Импорт внешних стилей.

```
<HTML><HEAD><TITLE> Пример импорта внешних стилей
</TITLE><style type="text/css">
  <!--
  @import C:\PRIMER\Листинг.css
  H3 {color: maroon; font-family: courier;}
  ->
</style><HEAD><BODY>
<H3> Современные технологии форматирования </H3>
<p> Импорт внешних стилей <BR>
  позволяет сократить <BR>
  размеры HTML-файлов<BR>
</P></BODY></HTML>
```

Внешний файл для веб-страницы представлен в листинге 3.3.

Листинг 3.3 Внешний файл Листнг.css.

```
<STYLE type="text/css">
H3 {color:crimson font-family:Arial Black}
P {background:cornsilk; font-family: times}
</STYLE>
```

В примере браузер сначала импортирует правила, содержащиеся в файле Листинг.css, а затем формирует набор правил для веб-страницы [3].

2.5.3 Свойства CSS

Свойства CSS позволяют управлять фоном, отображением текста, шрифтами, обрамлением, списками, позиционированием, определять границы, размеры области заполнения, отображение строк и тестовых блоков.

Свойства CSS для управления фоном позволяют разработчику задавать цвет фона и фоновые изображения, а также управлять размещением фоновых изображений. Для создания фона документа используются свойства:

- background-color – устанавливает цвет фона для выбранных элементов. Например, highlighted {background-color: yellow};
- background-image – задает фоновое изображение для выбранных элементов. Например, .flowery {background-image: url(flowers.gif);};
- background-position – позиционирует фоновое изображение;

- background-repeat – определяет, должно ли повторяться фоновое изображение для указанного элемента;
- background-attachment – указывает, должно ли фоновое изображение перемещаться вместе с элементом или оставаться фиксированным. Например, BODY {background-attachment: fixed} [3].

3 Динамическое web-программирование. JavaScript сценарии

JavaScript – объектно-ориентированный язык программирования сценариев. Используется для создания встраиваемых в веб-страницы сценариев, позволяющих полностью управлять веб-страницами и браузерами. JavaScript используется для создания интерактивных веб-страниц и веб-приложений [4].

3.1 Клиентские и серверные сценарии

3.1.1 Теоретическое введение

Сценарии – программы, созданные на сценарном языке программирования.

Клиентский JavaScript расширяет ядро языка, добавляя объекты управления браузером и объектную Модель Документа (Document Object Model, DOM). Они позволяют приложению размещать элементы на HTML-форме и реагировать на пользовательские события, такие как щелчок мышью, ввод данных в форму или навигация по страницам.

Серверный JavaScript расширяет ядро языка, добавляя объекты, относящиеся к запуску JavaScript на сервере. Например, серверные расширения позволяют приложению взаимодействовать с реляционной базой данных, сохраняя информацию между вызовами приложения, или выполнять манипуляции с файлами на сервере [5].

3.1.2 Лабораторная работа №5

«Разработка сайта с динамическими эффектами»

Цель: получить теоретические знания и практические навыки в использовании JavaScript, интерактивных форм, работе с изображениями и анимационными эффектами.

Используемые приемы и технологии: HTML, JavaScript сценарии, слои, каскадные стилевые таблицы, динамическая обработка форм.

Ключевые термины: анимационный эффект, форма, интерфейс, браузер, сценарии, фильтры.

Варианты

- 1 Фирма, осуществляющая продажу вычислительной техники.
- 2 Фирма, осуществляющая продажу теле-, видеоаппаратуры.

- 3 Фирма, осуществляющая продажу косметики.
- 4 Фирма, осуществляющая продажу музыкальных инструментов.
- 5 Фирма, осуществляющая продажу вело-, мототехники.
- 6 Фирма, осуществляющая продажу кабельного оборудования.
- 7 Фирма, осуществляющая продажу канцелярских принадлежностей.
- 8 Фирма, осуществляющая продажу спутникового оборудования.
- 9 Фирма, осуществляющая продажу легковых автомобилей.
- 10 Фирма, осуществляющая продажу сельскохозяйственной техники.
- 11 Фирма, осуществляющая продажу электротоваров.
- 12 Фирма, осуществляющая продажу лакокрасочной продукции.
- 13 Фирма, осуществляющая продажу книгопечатной продукции.
- 14 Фирма, предоставляющая риэлтерские услуги.
- 15 Фирма, осуществляющая продажу торгового оборудования.
- 16 Фирма, предоставляющая юридические услуги.
- 17 Фирма, осуществляющая продажу средств связи.
- 18 Фирма, предоставляющая туристические услуги.
- 19 Фирма, осуществляющая продажу строительных материалов.
- 20 Фирма, осуществляющая продажу продуктов питания.

Методические указания

1 Продумайте структуру сайта. Выполните поиск графического и текстового материала для информационного контента сайта.

2 Создайте интерактивные формы (текстовые поля, списки, флажки и т.п.). Реализуйте контроль данных, введенных пользователем в форму на стороне клиента (браузера) посредством JavaScript-сценария.

3 Осуществите предварительную загрузку и эффективное использование изображений клиентскими сценариями.

4 Создайте анимационные эффекты. Фильтры Wave, Fade, Light.

3.2 Операторы и функции JavaScript

3.2.1 Управление ходом выполнения сценария JavaScript

Операторы служат для управления потоком команд в JavaScript.

В JavaScript используются следующие операторы: break, comment, continue, for, for...in, function, if...else, return, var, while, with.

Оператор break прерывает цикл for и while, передавая управление первому оператору после цикла. В листинге 5.1 приведен пример применения оператора break.

Оператор comment – комментарии. JavaScript поддерживает следующий стиль комментариев:

- однострочные комментарии начинаются с символа (//);
- многострочные комментарии начинаются /* и заканчиваются */.

Оператор `continue` передает управление оператору проверки истинности условия в цикле `while` и оператору обновления значения счетчика в цикле `for`. Важное отличие от оператора `break` заключается в том, что оператор `continue` не прерывает цикл, а делает следующее.

В цикле `while` это переходит назад к условию.

В цикле `for` это переходит к модернизированному выражению.

Оператор `for` создает цикл с тремя необязательными выражениями, заключенными в круглых скобках и разделенными точкой с запятой, сопровождаемыми блоками утверждений, выполненных внутри цикла.

Например, *for (initialExpression; condition; updateExpression)*

{...строки кода...}

Выражение `initialExpression` используется для инициализации переменной счетчика. Выражение `condition` – условие выполнения цикла. Выражение `updateExpression` используется для изменения значения переменной счетчика.

Оператор `for...in` – присваивает переменной поочередно свойства объекта. Для каждого свойства цикл `for...in` выполняет операторы, содержащиеся в теле цикла:

Синтаксическое описание оператора

for (var in obj) {Statements}

Оператор `function` объявляет функции языка JavaScript, требует указания имени `name` и списка параметров `param`.

Синтаксическое описание `function`.

function name([param] [, param] [... , param]) { statements }

Оператор `return` указывает значение, которое будет возвращено функцией.

Синтаксическое описание оператора `return`:

return expression;

Оператор `var` объявляет переменную `varname` и может присвоить значение. Синтаксическое описание оператора `var`:

var VariableName [= value] [... , VariableName [= value]]

Оператор `with` устанавливает объект по умолчанию для набора операторов. Синтаксическое описание оператора `with`:

with (object) { statment }[4].

3.2.2 Функции JavaScript

Функция – поименованная часть приложения, формализующая решение подзадачи. Функции разделяются на группы:

- методы класса – функции, которые могут быть вызваны только по отношению к экземпляру JavaScript-объекта или класса;
- функции верхнего уровня – функции, не принадлежащие ни одному объекту;
- пользовательские функции – функции, разработанные пользователем.

Для создания функции используют ключевое слово `function`. Синтаксис объявления функции:

function имя_функции(параметры) { / тело функции */ } [5].*

3.2.3 Лабораторная работа №6 «JavaScript. Горизонтальное и вертикальное меню»

Цель: получить теоретические знания и практические навыки в использовании JavaScript и создании горизонтальных и вертикальных меню.

Используемые приемы и технологии: JavaScript (DHTML), HTML, CSS.

Ключевые термины: горизонтальное и вертикальное меню.

Варианты

- 1 Фирма, осуществляющая продажу продуктов питания.
- 2 Фирма, осуществляющая продажу строительных материалов.
- 3 Фирма, предоставляющая туристические услуги.
- 4 Фирма, осуществляющая продажу средств связи.
- 5 Фирма, предоставляющая юридические услуги.
- 6 Фирма, осуществляющая продажу торгового оборудования.
- 7 Фирма, предоставляющая риэлторские услуги.
- 8 Фирма, осуществляющая продажу книгопечатной продукции.
- 9 Фирма, осуществляющая продажу лакокрасочной продукции.
- 10 Фирма, осуществляющая продажу электротоваров.
- 11 Фирма, осуществляющая продажу сельскохозяйственной техники.
- 12 Фирма, осуществляющая продажу легковых автомобилей.
- 13 Фирма, осуществляющая продажу спутникового оборудования.
- 14 Фирма, осуществляющая продажу канцелярских принадлежностей.
- 15 Фирма, осуществляющая продажу кабельного оборудования.
- 16 Фирма, осуществляющая продажу вело-, мототехники.
- 17 Фирма, осуществляющая продажу музыкальных инструментов.
- 18 Фирма, осуществляющая продажу косметики.
- 19 Фирма, осуществляющая продажу теле-, видеоаппаратуры.
- 20 Фирма, осуществляющая продажу вычислительной техники.

Методические указания

1 Продумайте структуру сайта. Выполните поиск графического и текстового материала для информационного контента сайта.

2 Создайте сайт по варианту задания с горизонтальным и вертикальным меню.

2.1 Горизонтальное меню.

Простая, удобная и красивая навигация на сайте с небольшим количеством разделов.

Маркированный список со ссылками выстраиваем в линию и делаем выделение той ссылки, на которую наводим курсор.

2.1.1 Создадим маркированный список со ссылками внутри.

2.1.2 Выстроим его в линию и уберем маркер.

2.1.3 Поскольку мы не можем использовать блочные свойства стилей для inline элементов, сделаем из тега a блочный элемент, а чтобы они выстроились в одну строку, добавим свойство float:left.

2.1.4 Описание стилей для текста ссылок и добавление отступов и фона:

2.1.4.1 Описываем стили для текста ссылок.

2.1.4.2 Добавляем внутренние отступы и фон для ссылки.

2.1.4.3 Описываем псевдокласс hover.

2.2 Вертикальное выпадающее меню.

Вертикальное выпадающее меню позволяет разместить большое количество ссылок на ограниченной площади.

2.2.1 Создадим тело нашего меню.

2.2.2 При помощи стилей уберем подменю.

2.2.3 Описание стилей.

2.3 Вертикальное раскрывающееся меню.

2.3.1 Создадим тело нашего меню.

2.3.2 Напишем функцию на JavaScript, которая будет открывать и закрывать подменю.

2.3.3 Позаботимся о пользователях старых браузеров и вместо того, чтобы писать в стилях #menu li ul{display:none}, напишем еще одну функцию, которая будет закрывать все подменю при загрузке страницы. Если по каким-либо причинам наш JavaScript код не сработает, меню будет полностью открыто и позволит пользователю кликать на ссылки подменю.

2.3.4 Поставим вызов этой функции на загрузку страницы.

3.3 Встроенные объекты JavaScript

Объекты JavaScript объединяют переменные, свойства и функции (методы). Пользователь может создавать свои объекты. JavaScript содержит базовые объекты:

- Global – набор высокоуровневых свойств и методов, у которых нет родительского объекта.
- String – позволяет форматировать и изменять текстовые строки.
- Number – дает возможность обрабатывать числа.
- Boolean – позволяет создавать логические величины.
- Array – дает возможность создавать и управлять массивами.
- Function – позволяет создать функции.
- Date – дает возможность работать со временем и датой.
- Math – предоставляет некоторые математические функции и константы.
- RegExp – позволяет выполнять функции регулярных выражений для заданных строк.
- Object – является прототипом для всех объектов [4].

3.4 Динамический HTML

Динамический HTML (Dynamic HTML, DHTML) – способ создания интерактивного веб-сайта. DHTML включает язык HTML, каскадную таблицу стилей CSS, язык JavaScript. Объектная модель документа (Document Object Model, DOM) связывает три компонента, придавая документу возможность динамического изменения страницы без перезагрузки. Динамический HTML предоставляет контроль JavaScript над элементами документа, параметрами их оформления и размещения, текстом страницы. Страницы могут динамически менять стили, содержание и оформление. Можно создавать интерактивные документы, взаимодействующие с пользователем в реальном масштабе времени [4].

DHTML может быть использован для создания приложения в веб-браузере: например для простой навигации, придания интерактивности форм, динамического перетаскивания элементов по экрану. Также он может служить как инструмент для создания основанных на браузере видеоигр [5].

В листинге 4.1 приведен пример динамической подсветки ссылок.

Листинг 4.1 Динамическая подсветка ссылок на веб-странице.

```
<a href="javascript:
for (nIx5Bs=0; nIx5Bs<document.all.length; nIx5Bs++) {
if(document.all[nIx5Bs].tagName=='A') {
with(document.all[nIx5Bs].style) {if(backgroundcolor=='yellow'){
void(backgroundcolor=document.bgColor)}
else {void(backgroundcolor='yellow')}}}}}>
Подсветить ссылки (вкл/выкл)</a> [7].
```

3.5 AJAX технология

3.5.1 Принципы AJAX технологии

AJAX (англ. Asynchronous Javascript and XML – «асинхронный JavaScript и XML») – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся более быстрыми и удобными.

Технология AJAX – концепция использования нескольких смежных технологий. AJAX базируется на двух основных принципах:

1 Динамическое обращение к серверу «на лету», без перезагрузки всей страницы полностью.

2 Использование DHTML для динамического изменения содержания страницы.

В качестве формата передачи данных обычно используются JSON (JavaScript Object Notation) или XML (eXtensible Markup Language).

3.5.2 Преимущества и недостатки

Технология AJAX имеет преимущества и недостатки.

Преимущества:

- 1 Экономия трафика.
- 2 Уменьшение нагрузки на сервер.
- 3 Ускорение реакции интерфейса.

Недостатки:

- 1 Отсутствие интеграции со стандартными инструментами браузера.
- 2 Динамически загружаемое содержимое недоступно поисковикам.
- 3 Старые методы учёта статистики сайтов становятся неактуальными.
- 4 Усложнение проекта.
- 5 Требуется включенный JavaScript в браузере [5; 6].

3.5.3 Лабораторная работа №7

«Технология фоновой передачи данных браузером с веб-сервером в информационных сетях»

Цель: получить теоретические знания и практические навыки в использовании технологии AJAX.

Используемые приемы и технологии: AJAX, JavaScript, HTML, CSS

Ключевые термины: веб-приложение, методы объекта XMLHttpRequest.

Варианты

Разработайте веб-приложение «Калькулятор», предусматривающее возможность ввода в поле арифметических выражений типа «23*6» с асинхронной передачей данных между браузером и сервером с помощью AJAX и отображением результата.

Методические указания

1 Теоретические сведения

Важнейшей составляющей технологии AJAX является класс XMLHttpRequest.

Использование объекта класса XMLHttpRequest реализуется посредством методов и свойств.

Методы объекта XMLHttpRequest:

- метод open() создает запрос для Web-сервера;
- метод send() посылает запрос серверу;
- метод abort() отменяет текущий запрос к серверу.

Свойство onreadystatechange определяет обработчик события, которое происходит при каждой смене состояния объекта. Имя должно быть записано в нижнем регистре. Свойство readyState определяет текущее состояние объекта XMLHttpRequest. Свойство responseText содержит присланный сервером текст.

Приложения, использующие AJAX каждый раз, когда вводится символ, предусматривают выполнение следующих действий.

Вызывается функция обработчик.

Формируется *url* с необходимыми параметрами для отправки веб-серверу. Создается объект класса XMLHttpRequest.

Указывается имя функции, подлежащей исполнению при наступлении события ввода символа. Она обеспечит отображение изменяющейся части страницы, когда ответ сервера будет готов.

Открывается созданный объект класса XMLHttpRequest с указанным значением *url*.

Отправляется HTTP запрос веб-серверу.

2 Ход работы

Реализация сетевого взаимодействия с использованием технологии AJAX предусматривает наличие клиентской и серверной частей веб-приложений.

2.1 Клиентская часть

Предполагается, что пользователь может вводить в текстовое поле формы вариант подбираемого логина, получая при этом динамически ответ о существовании или отсутствии такого логина в списке зарегистрированных пользователей без перезагрузки страницы.

2.2 Серверная часть

Разместите на веб-сервере в файле с именем `server_bearbeiter.php` следующий PHP сценарий:

```
<?php
header("Cache-Control: no-cache, must-revalidate");
// Прошедшая дата
header("Expires: Mon, 1 Sep 2008 07:30:00 GMT");
// Инициализация массива названий
$login[]="aazc"; $login[]="bblk"; $login[]="cclk";
//получение параметра log из URL
$log = $_GET["log"];
//поиск соответствий из массива если длина log > 0
$antvort = ""; if (strlen($log) > 0)
{for($i = 0; $i<count($login); $i++)
{if (strtolower($log) == strtolower(substr($login[$i],0,strlen($log))))
$antvort = "УЖЕ СУЩЕСТВУЕТ";
else $antvort = "ВОЗМОЖЕН";
}} //вывод результата
echo $antvort;?>
```

4 Язык PHP и СУБД MySQL

PHP – язык сценариев общего назначения с открытым исходным кодом.

MySQL разработал Михаэль Видениус (Michael Widenius, monty@analytikerna.se). MySQL является небольшой и быстрой реляционной СУБД основанной на традициях Hughes Technologies Mini SQL (mSQL) [7].

Последнюю версию MySQL можно скачать с www.tcx.se.

4.1 СУБД MySQL

4.1.1 Особенности СУБД MySQL

СУБД MySQL характеризуется свойствами:

- 1 Многопоточность. Поддержка нескольких одновременных запросов.
- 2 Оптимизация связей с присоединением данных за один проход.
- 3 Записи фиксированной и переменной длины.
- 4 ODBC драйвер в комплекте с исходником.
- 5 Гибкая система привилегий и паролей.
- 6 До 16 ключей в таблице. Каждый ключ может иметь до 15 полей.
- 7 Поддержка ключевых полей и специальных полей в операторе CREATE.
- 8 Поддержка чисел длиной от 1 до 4 байт (ints, float, double, fixed), строк переменной длины и меток времени.
- 9 Интерфейс с языками C и perl.
- 10 Основанная на потоках, быстрая система памяти.
- 11 Утилита проверки и ремонта таблицы (isamchk).
- 12 Все данные хранятся в формате ISO8859_1.
- 13 Все операции работы со строками не обращают внимания на регистр символов в обрабатываемых строках.
- 14 Псевдонимы применимы как к таблицам, так и к отдельным колонкам в таблице.
- 15 Все поля имеют значение по умолчанию. INSERT можно использовать на любом подмножестве полей.
- 16 Легкость управления таблицей, включая добавление и удаление ключей и полей [8].

4.1.2 Основные команды

Для создания баз данных и таблиц в них, для настройки прав доступа к данным и манипулирования ими в СУБД MySQL применяется программа-монитор mysql. Запустив ее из командной строки, вы подключаетесь к СУБД с именем текущего пользователя операционной системы, если он имеет право на подключение. Для выбора базы данных применяется команда use dbname, где dbname – имя базы данных. Можно указать базу данных, имя пользователя и пароль при вызове монитора:

```
Mysql dbname -h hostname -u username -p password.
```

В командной строке оставить ключ -p, тогда пароль будет запрошен дополнительно. Для выхода из монитора применяется команда exit или quit.

Для создания базы данных применяется команда create database dbname, а для создания таблиц – create table tablename (columns). В скобках перечисляются названия, типы данных и другие атрибуты полей, например:

```
Create table books (id int unsigned not null auto_increment primary key,
```

author char(30), title char(60), price float(5,2));

Для просмотра списка созданных таблиц применяют команду show tables, а для просмотра структуры таблицы – команду describe tablename.

Для ввода, извлечения, изменения и удаления данных из таблиц применяют операторы структурированного языка запросов SQL: insert, select, update и delete. Для отбора данных по условию, уточнения изменяемых и удаляемых записей применяется конструкция where, для группирования – group by, для упорядочивания – order by. Для удаления таблиц служит команда drop table tablename [8].

4.1.3 Лабораторная работа №8

«Динамическая поддержка выбора значений из предлагаемого списка»

Цель: получить теоретические знания и практические навыки в использовании технологии AJAX.

Используемые приемы и технологии: JavaScript, CSS, DOM(Document Object Model), DHTML (Dynamic HTML), объект XMLHttpRequest.

Ключевые термины: AJAX-приложение, JavaScript-код, каскадные таблицы стилей, веб-приложение, одностороннее и двустороннее взаимодействие с сервером, библиотеки стандартных компонентов AJAX-приложений, кодирование параметров, потоковая обработка, файлы cookie, серверная часть, клиентская часть приложения, динамически изменяемое содержание.

Варианты

Разработать web-приложение, реализующее интерактивный выбор из базы данных организации.

1 Предприятие по учету платы за потребленную электроэнергию. Плата взимается с каждой квартиры в зависимости от количества потребленной энергии или от числа проживающих, если счетчик отсутствует: адрес, район города, балансодержатель, количество подъездов в доме, количество квартир в доме, фамилия квартиросъемщика, имя квартиросъемщика, отчество квартиросъемщика, номер паспорта.

2 Касса авиакомпании занимается продажей билетов на предстоящие рейсы: номер рейса, название рейса, дата вылета, время вылета, дата прибытия, время прибытия, промежуточные посадки (есть/нет).

3 Приемная комиссия университета. Каждый год университет зачисляет новых абитуриентов для возможного их поступления в университет после сдачи вступительных экзаменов: регистрационный номер абитуриента, ФИО абитуриента.

4 Институт селекции растений занимается сбором, выведением и продажей различных сортов семян: уникальный номер сорта, название сорта, год, в котором выведен сорт, адаптация к местным условиям (да/нет),

морозоустойчивый (да/нет), описание характеристик сорта, способ посадки, номер партии, годен до.

5 В гостинице существует множество вариантов заселения гостей: регистрационный номер гостя, ФИО гостя, дата регистрации, адрес гостя, город, из которого приехал гость, цель приезда, номер паспорта, дата выдачи паспорта, кем выдан паспорт, место работы или учебы.

6 Открытое акционерное общество, занимающееся междугородними пассажирскими перевозками: фамилия водителя, имя водителя, отчество водителя, стаж работы, год рождения, категория водителя (D,E).

7 Мелкооптовый книжный магазин. Менеджер магазина, изучив спрос на книжную продукцию в городе, принимает решение о закупке партии книг в издательстве: поставщик книг, ИНН поставщика книг.

8 Телефонная компания. Отслеживание абонентской платы за телефоны. Клиентами компании могут быть физические лица и организации: идентификатор клиента компании, тип клиента (физ./юр. лицо), ФИО клиента (для физ. лица), название клиента (для юр. лица), ИНН клиента (для юр. лица), руководитель (для юр. лица), телефон для связи (для юр. лица).

9 Организация, осуществляющая сдачу в аренду площадей: идентификатор арендатора, тип арендатора (физ./юр. лицо), ФИО арендатора (для физ. лица), название арендатора (для юр. лица), ИНН арендатора (для юр. лица), руководитель (для юр. лица), телефон арендатора, юридический адрес арендатора.

10 «Бюро технической инвентаризации» по изготовлению и выдаче технических паспортов на объекты недвижимости. Перед регистрацией сделки с объектом недвижимости собственник объекта должен получить в БТИ на него технический паспорт: номер заявки на изготовление документа.

11 Предприятие по учету платы за пользование газом и газовыми приборами: адрес здания, район города, количество подъездов в доме, количество квартир в доме, фамилия квартиросъемщика, имя квартиросъемщика, отчество квартиросъемщика.

12 Отдел приватизации жилья администрации города: адрес здания, район города, балансодержатель, год постройки здания, материал стен здания, материал фундамента, износ в процентах.

13 Организация, осуществляющая продажу и покупку квартир: регистрационный номер клиента, адрес клиента, ФИО клиента, телефон для связи с клиентом, регистрационный номер договора, адрес квартиры, район города, этажей в доме, этаж, тип дома.

14 Регистратура ведомственной поликлиники: идентификационный номер врача, фамилия врача, имя врача, отчество врача, номер кабинета, образование (университет), специализация (терапевт, лор...), стаж работы, номер рабочего телефона, год рождения, ФИО пациента, номер карточки пациента, адрес пациента.

15 Туристическая компания: идентификатор клиента, фамилия клиента, имя клиента, отчество клиента, документ, удостоверяющий личность, серия

документа, номер документа, дата выдачи, кем выдан документ, дата рождения, наличие заграничного паспорта.

16 Отдел ведения реестра имущества университетского городка. В состав имущества входит несколько зданий. В зданиях располагаются аудитории, кафедры, лаборатории, вычислительные центры, деканаты и т. д.: регистрационный номер здания, название здания (корпуса) университета, площадь земельного участка, адрес здания, год постройки.

17 Государственная автомобильная инспекция по безопасности дорожного движения города. База данных ГИБДД содержит сведения обо всех транспортных средствах города и их владельцах: идентификатор владельца, тип владельца (физ./юр. лицо), ФИО владельца (для физ. лица), название организации.

18 Отдел заселения муниципальных общежитий администрации города. В ведении администрации города находится несколько десятков общежитий. Раньше они принадлежали предприятиям города, а теперь, после банкротства предприятий, все эти общежития переданы муниципальным властям: номер общежития, адрес общежития.

19 Отдел учета налогообложения физических лиц городской налоговой инспекции: идентификационный номер, фамилия налогоплательщика, имя налогоплательщика, отчество налогоплательщика, документ, удостоверяющий личность, серия документа.

20 Отдел учета нежилых помещений: кадастровый номер здания, адрес здания, район города, площадь земельного участка, год постройки здания, материал стен здания.

Методические указания

1 Продумайте структуру сайта. Выполните поиск графического и текстового материала для информационного контента сайта.

2 Спроектировать структуру базы данных согласно варианту задания.

3 Разработать интерактивную web-форму выбора в базе данных. Включает два поля: текстовое и поле SELECT. В текстовое поле вводится значение, отсылаемое на сервер. Сервер использует значение как критерий поиска по базе данных и возвращает содержимое полей. Полученные от сервера данные без перезагрузки документа помещаются в поле SELECT формы, в котором пользователь сделает окончательный выбор. В результате при отправке на сервер полностью заполненной формы требуемые пользователем данные определяются значением, выбранным в поле SELECT [8].

4.2 Язык PHP

PHP – Язык программирования, разработанный для написания web-приложений (сценариев), исполняющихся на web-сервере.

Аббревиатура PHP означает «Hypertext Preprocessor (Препроцессор Гипертекста)». Синтаксис языка берет начало из C, Java и Perl. PHP достаточно

прост для изучения. Преимуществом PHP является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц [9].

4.2.1 Особенности языка PHP

Преимущество языка PHP перед языками Perl и C заключается в возможности создания HTML-документов с внедренными командами PHP.

Отличием PHP от кода, выполняющегося на стороне клиента, например, JavaScript, является выполнение PHP-скриптов на стороне сервера. Вы можете сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP и клиенты не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

PHP позволяет создавать качественные web-приложения за короткие сроки. PHP прост для освоения и удовлетворяет запросы профессиональных программистов.

4.2.2 Язык PHP и объектно-ориентированное программирование

Класс – абстрактный тип данных, разрабатываемый пользователем. Экземпляр класса – объект. Объект – совокупность данных (свойств) и функций (методов) для их обработки. Свойства и методы называются членами класса. Если класс можно рассматривать как тип данных, то объект – как переменную. Скрипт может одновременно работать с несколькими объектами одного класса, как с несколькими переменными. Внутри объекта данные и код (члены класса) могут быть либо открыты, либо нет. Открытые данные и члены класса являются доступными для других частей программы, которые не являются частью объекта. Закрытые данные и члены класса доступны внутри объекта.

Описание классов в PHP начинаются служебным словом `class`:

```
class Имя_класса {  
    // описание членов класса – свойств и методов для их обработки  
}
```

Для объявления объекта необходимо использовать оператор `new`:

```
Объект = new Имя_класса;
```

Данные описываются с помощью служебного слова `var`. Метод описывается как и обыкновенная функция. Методу можно передавать параметры.

В листинге 5.1 приведен пример класса на PHP.

Листинг 5.1 Класс

```
<?php  
// Создаем новый класс Coor:  
class Coor { // данные (свойства):  
    var $name; var $addr;  
    // методы:  
    function Name() { echo "<h3>John</h3>"; } }
```

```
// Создаем объект класса Coor:
```

```
$object = new Coor;?>
```

В листинге 5.2 приведен пример доступа к членам класса на PHP

Листинг 5.2 Доступ к членам класса

```
<?php
```

```
// Создаем новый класс Coor:
```

```
class Coor { // данные (свойства):
```

```
var $name;
```

```
// методы:
```

```
function Getname() { echo "<h3>John</h3>"; }}
```

```
// Создаем объект класса Coor:
```

```
$object = new Coor;
```

```
// Получаем доступ к членам класса:
```

```
$object->name = "Alex"; echo $object->name;
```

```
// Выводит 'Alex'
```

```
// А теперь получим доступ к методу класса (фактически, к функции внутри класса):
```

```
$object->Getname();
```

```
// Выводит 'John' заглавными буквами
```

```
?>
```

Указатель `$this` используется для доступа к членам класса.

Конструктор – метод, инициализирующий поля класса.

В листинге 5.3 приведен пример конструктора.

Листинг 5.3 Конструктор

```
<?
```

```
class Webpage {
```

```
var $bgcolor; function Webpage($color) {
```

```
    $this->bgcolor = $color; }}
```

```
// Вызвать конструктор класса Webpage
```

```
$page = new Webpage("brown");?>
```

В PHP отсутствует непосредственная поддержка деструкторов. Имитировать работу деструктора можно, вызывая функцию PHP `unset()`.

Необходимость в вызове деструкторов возникает лишь при работе с объектами, использующими большой объем ресурсов, поскольку все переменные и объекты автоматически уничтожаются по завершении сценария.

Обращение к элементам классов осуществляется с помощью оператора :: «двойное двоеточие». Используя «двойное двоеточие», можно обращаться к методам классов [10].

4.2.3 Лабораторная работа №9 «Проектирование сайта электронной коммерции»

Цель: получить теоретические знания и практические навыки в использовании PHP и СУБД MySQL.

Используемые приемы и технологии: СУБД MySQL.

Ключевые термины: база данных, запись, поле, СУБД, протокол, учетная запись.

Задание: разработать сайт фирмы, посетители которого смогут посмотреть:

- предложения фирмы по ассортименту товаров (услуг), изготовителям и программам продаж;

- выбрать товар (услугу) и программу продаж;

- оставить заказ на приобретение товара (услуги).

При оформлении заказа должна подсчитываться его стоимость.

На сайте должна быть представлена информация о номенклатуре товара (услуги), включающая в себя описания и фотографии.

Информация о сделанных заказах и списке заказчиков должна быть доступна администратору. Он должен иметь возможность внести исправления в данные о заказах и заказчиках, добавить или изменить сведения о товаре (услуге) и программам продаж сайта электронной коммерции.

Варианты

- 1 Фирма, осуществляющая продажу строительных материалов.
- 2 Фирма, осуществляющая продажу продуктов питания.
- 3 Фирма, осуществляющая продажу средств связи.
- 4 Фирма, предоставляющая туристические услуги.
- 5 Фирма, осуществляющая продажу торгового оборудования.
- 6 Фирма, предоставляющая юридические услуги.
- 7 Фирма, осуществляющая продажу книгопечатной продукции.
- 8 Фирма, предоставляющая риэлторские услуги.
- 9 Фирма, осуществляющая продажу электротоваров.
- 10 Фирма, осуществляющая продажу лакокрасочной продукции.
- 11 Фирма, осуществляющая продажу легковых автомобилей.
- 12 Фирма, осуществляющая продажу сельскохозяйственной техники.
- 13 Фирма, осуществляющая продажу канцелярских принадлежностей.
- 14 Фирма, осуществляющая продажу спутникового оборудования.
- 15 Фирма, осуществляющая продажу вело-, мототехники.
- 16 Фирма, осуществляющая продажу кабельного оборудования.
- 17 Фирма, осуществляющая продажу косметики.
- 18 Фирма, осуществляющая продажу музыкальных инструментов.
- 19 Фирма, осуществляющая продажу вычислительной техники.
- 20 Фирма, осуществляющая продажу теле-, видеоаппаратуры.

Методические указания

- 1 Разработать структуру сайта согласно варианта задания.
- 2 Спроектировать базу данных.
- 3 Создать сценарии создания и заполнения базы данных.
- 4 Создать запросы к базе данных.

5 Применить объектно-ориентированный метод при создании приложения. Объектный подход помогает структурировать данные и упорядочить методы получения данных из базы.

6 Разработать классы, в которых определены методы, позволяющие оперировать заказами, сведениями о заказчиках, списках и описаниях товаров (услуг) и т.д.

7 Создать сценарии сайта (домашняя страница сайта, форма заказа, оформление заказа, администрирование сайта, запись в базу данных и т.д.).

4.2.4 Лабораторная работа №10 «Разработка интернет-магазина»

Цель: получить теоретические знания и практические навыки в использовании технологии AJAX, PHP, MySQL.

Используемые приемы и технологии: PHP, СУБД MySQL.

Ключевые термины: интернет-магазин, база данных, корзина, калькулятор.

Основные задачи, решаемые студентом в процессе выполнения лабораторной работы:

- проектирование структуры сайта электронного магазина;
- создание каталога товаров с поддержкой поиска по ключевым словам и выборки по критериям;
- реализация страниц администрирования каталога, позволяющих добавлять товары в каталог, изменять сведения о товарах, категориях и отделах и выполнять другие операции;
- создание корзины покупателя и механизма приема платежей с помощью PHP;
- реализация системы учетных записей посетителей;
- проектирование базы данных и составление SQL-запросов;
- использование возможностей поиска MySQL для организации поиска товаров в каталоге;
- использование системы шаблонов Smarty для создания структурированного и расширяемого кода на PHP;
- оптимизация сайта для работы с поисковыми системами;
- применение технологии AJAX для ускорения реакции сайта на действия пользователей;
- программная реализация интернет-магазина;
- документирование контрольной работы в соответствии с требованиями [11].

Варианты

- 1 Фирма, осуществляющая продажу средств связи.
- 2 Фирма, осуществляющая продажу легковых автомобилей.
- 3 Фирма, предоставляющая риэлторские услуги.
- 4 Фирма, предоставляющая туристические услуги.

- 5 Фирма, осуществляющая продажу строительных материалов.
- 6 Фирма, предоставляющая юридические услуги.
- 7 Фирма, осуществляющая продажу торгового оборудования.
- 8 Фирма, осуществляющая продажу книгопечатной продукции.
- 9 Фирма, осуществляющая продажу лакокрасочной продукции.
- 10 Фирма, осуществляющая продажу продуктов питания.
- 11 Фирма, осуществляющая продажу вычислительной техники.
- 12 Фирма, осуществляющая продажу теле-, видеоаппаратуры.
- 13 Фирма, осуществляющая продажу косметики.
- 14 Фирма, осуществляющая продажу музыкальных инструментов.
- 15 Фирма, осуществляющая продажу вело-, мототехники.
- 16 Фирма, осуществляющая продажу кабельного оборудования.
- 17 Фирма, осуществляющая продажу спутникового оборудования.
- 18 Фирма, осуществляющая продажу электротоваров.
- 19 Фирма, осуществляющая продажу сельскохозяйственной техники.
- 20 Фирма, осуществляющая продажу канцелярских принадлежностей.

Методические указания

- 1 Продумать структуру сайта. Выполните поиск графического и текстового материала для информационного контента сайта.
- 2 Спроектировать структуру базы данных согласно варианту.
- 3 Создать сайт с базовым набором функций: каталог товаров, поиск товаров, механизм приема платежей, механизм приема и передачи данных посетителя, калькулятор, автоматически подсчитывающий сумму заказа и отображающий ее на странице.
- 4 Разработать механизм динамических рекомендаций, повышающий привлекательность сайта для посетителей и побуждающий делать дополнительные покупки.
- 5 Разработать гостевую книгу, позволяющую делать отзывы о товарах.

5 Язык XML и PHP

XML (*eXtensible Markup Language*) – язык разметки, описывающий класс объектов данных, называемых XML- документами. Язык XML используется в качестве средства для описания грамматики других языков и контроля за правильностью составления документов [12].

В отличие от HTML язык XML позволяет:

- 1 Создавать собственные элементы.
- 2 Присваивать имена по своему выбору.
- 3 Представлять документы любого типа.
- 4 Сортировать, фильтровать и выполнять поиск информации.
- 5 Представлять информацию в структурированном виде [13].

5.1 Структура XML-документа

Структура XML документа включает заголовок, секцию DOCTYPE, тело XML документа. В заголовке описывается версия и кодировка. В секции DOCTYPE описываются сущности. Сущность – константа, используемая в теле XML-документа для сокращения записи и лёгкой сопровождаемости. В теле XML документа в отличие от HTML Вы сами придумываете названия элементов. В XML очень строгий синтаксис, то есть не должно быть незакрывающих дескрипторов или пропущенных кавычек в значениях атрибутов. В листинге 6.1 приведен пример XML документа.

Листинг 6.1 Пример XML документа.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE shop [<!ENTITY n "Ноутбук"><!ENTITY g "Игра">]>
<shop id="1"><notebooks><prod>&n; IBM Lenovo V570</prod>
<prod>&n; DELL Inspiron N5010</prod></notebooks>
<games><prod>&g; Гомика 2</prod><prod>&g; Might & Magic
6</prod></games></shop>
```

5.2 Лабораторная работа №11 «Разработка SAX-парсера новостной ленты»

Цель: получить теоретические знания и практические навыки в использовании технологии SAX и работе с новостной лентой RSS.

Используемые приемы и технологии: SAX, XML, PHP, CSS.

Ключевые термины: программа-парсер, XML-parser, RSS-лента, XSL-функция, DOM-функция.

Варианты

Разработайте SAX-парсер новостной ленты RSS.

Методические указания

1 Теоретические сведения

SAX (Simple Application Programming Interface for XML) – технология обработки XML-документов, основанная на анализе дерева документа и обработке событий.

RSS (Really Simple Syndication) – технология описания лент новостей, анонсов статей и т.п.

SAX основан на событийном программировании. Программа-парсер разбирает XML-документ, вызывая функции-обработчики определенных событий. Вначале создается парсер, затем функции обработки событий:

```
$xml_parser=xml_parser_create();
```

Функция возвращает указатель на парсер, который используется другими функциями, работающими с XML.

Функция обработки открывающего тега
function startElement(\$parser, \$name, \$attrs)

Функция обработки открывающего тега
function endElement(\$parser, \$name)

Функция задания обработчиков событий:

xml_set_element_handler() – XML-парсер встречает открывающий или закрывающий тег.

xml_set_character_data_handler()–XML-парсер встречает символьные данные в XML-документах.

2 Ход выполнения

2.1 Создать парсер.

2.2 Определить функцию обработки открывающего тега.

2.3 Определить функцию обработки закрывающего тега.

2.4 Определить функцию обработки символьного содержимого XML-элемента.

ЗАКЛЮЧЕНИЕ

Уровень проникновения интернета с каждым годом возрастает. Происходит переход к безбумажной технологии. Широко внедряются web-приложения. Дисциплина «Web-программирование» изучает методы и способы разработки web-приложений.

Интернет – всемирная система объединённых компьютерных сетей, построенная на базе протокола IP (Internet Protocol) и маршрутизации IP-пакетов. Интернет образует глобальное информационное пространство, служит физической основой для Всемирной паутины (World Wide Web, WWW) и множества других систем (протоколов) передачи данных.

В 90-х годах web-сайты разрабатывались вручную с использованием языка HTML (Hypertext Markup Language), CGI (Common Gateway Interface), DLL-библиотек ISAPI (Internet Server Application Programming Interface), Java и ASP – страниц (Active Server Pages). В июле 2000 г. компания Microsoft предложила платформу .NET, которая заменит перечисленное выше программное обеспечение для разработки web-сайтов.

Несколько лет назад для создания web-сайта нужно было лишь установить сервер, приобрести IP-адрес и выложить на сайт информацию. Сайт становился доступным всем – достаточно было знать URL-адрес. Коммерческие предприятия использовали WWW для размещения данных. Web-среда оказалась ценным исследовательским инструментом и средством распространения информации. В будущем значимость Web возрастет. Содержимое web-сайтов будет использоваться компьютерами и пользователями, т.е. доступ к содержимому web-сайтов станет возможным из программ благодаря web-сервисам. Согласно концепциям .NET за организацию многофункционального пользовательского интерфейса ответственность будет возложена на сервер. Принципы web-сервисов в .NET и поддерживаемые сервером многофункциональные пользовательские интерфейсы будут использоваться разработчиками. Полнофункциональные пользовательские интерфейсы уживутся с интерфейсами другого типа, в том числе поддерживаемыми сервером web-интерфейсы [20].

В методическом указании приводятся задания к выполнению лабораторных, практических и контрольной работ по темам, посвященным HTML, CSS, JavaScript, MySQL, PHP. Темы сопровождаются кратким теоретическим обоснованием. Выполнение лабораторных, практических и контрольной работ позволят приобрести практические навыки разработки web-приложений и закрепить теоретические знания, полученные студентами на лекциях по дисциплине web-программирование.

СПИСОК ЛИТЕРАТУРЫ

- 1 Гиз Ксения, Холмс Александра. Основы Web-дизайна: вспомогательное руководство / пер. с англ. – М. : Издательский дом «Вильямс», 2002. – 640 с.
- 2 Комолова Н. В. HTML. – СПб. : Питер, 2008. – 268 с.
- 3 Мальчук Е. В. HTML и CSS. – М. : Издательский дом «Вильямс», 2007. – 416 с.
- 4 Днепров А. Г. JavaScript. – СПб. : Питер, 2008. – 304 с.
- 5 Слепцова Л. Д., Бидасюк Ю. М. JavaScript. – М. : Издательский дом «Вильямс», 2007. – 448 с.
- 6 Бенкен Е. С. AJAX: программирование для Интернета. – СПб. : БХВ-Петербург, 2009. – 464 с.
- 7 Веллинг Люк, Томсон Лора. MySQL : учебное пособие. – М. : Издательский дом «Вильямс», 2005. – 304 с.
- 8 Гольцман В. MySQL 5.0. Библиотека программиста. – СПб. : Питер, 2010. – 256 с.
- 9 Котеров Д. В. PHP 5. - 2-е изд., перераб. и доп. – СПб. : Петербург, 2008. – 1104 с.
- 10 Коггэол Джон. PHP 5. Полное руководство / пер. с англ. – М. : Издательский дом «Вильямс», 2006 – 752 с.
- 11 Дари К., Баланеску Э. PHP и MySQL: создание интернет-магазина. - 2-е изд. ; пер. с англ. – М. : ООО «И. Д. Вильямс», 2010. – 640 с.
- 12 Хантер Дэвид, Рафтер Джеф, Фаусет Джо, Ван дер Влиет Эрик. XML. Базовый курс. – М. : ООО «И.Д. Вильямс», 2009. – 1344 с.
- 13 Бенкен Е. С. PHP, MySQL, XML: программирование для Интернета. – СПб. : БХВ-Петербург, 2011. – 304 с.

Семахин Андрей Михайлович

WEB-ПРОГРАММИРОВАНИЕ

Методические указания
к выполнению лабораторных работ
для студентов направления подготовки 09.03.04
«Программная инженерия»

Редактор Н.М. Быкова

Подписано в печать	Формат 60x84 1/16	Бумага 65 г/м ²
Печать цифровая	Усл. печ. л. 2,5	Уч.-изд. л. 2,5
Заказ	Тираж 25	Не для продажи

РИЦ Курганского государственного университета.
640000, г. Курган, ул. Советская, 63/4.
Курганский государственный университет.