

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
ГОУ ВПО
«Амурский государственный университет»

УТВЕРЖДАЮ
Зав. кафедрой ОМиИ
_____ Г.В. Литовка
« ____ » _____ 2010 г.

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО ДИСЦИПЛИНЕ
«ПРОГРАММНЫЕ СРЕДСТВА РАЗРАБОТКИ WEB-СТРАНИЦ»**

для специальности

080502.65 – экономика и управление на предприятии (в машиностроении)

Составители: Чалкина Н.А., канд. пед. наук, доцент

Лебедь О.А., старший преподаватель

Благовещенск, 2010

*Печатается по решению
редакционно-издательского совета
факультета математики и информатики
Амурского государственного университета*

Чалкина Н.А., Лебедь О.А.

Учебно-методический комплекс дисциплины «Программные средства разработки Web-страниц» для специальности 080502.65. – Благовещенск: АмГУ, 2010.

© Амурский государственный университет, 2010
© Кафедра общей математики и информатики, 2010

СОДЕРЖАНИЕ

I.	Рабочая	програм-	
ма.....			4
II.	Краткое изложение программного материала.....		
III.	Методические	указания	(рекоменда-
ции).....			11
1.	Методические	указания	по изучению дисципли-
ны.....			12
2.	Методические указания к практическим занятиям.....		
			12
3.	Методические указания по самостоятельной работе студентов.....		
			31
IV.	Контроль	зна-	
ний.....			32
1.	Текущий	контроль	зна-
ний.....			32
2.	Итоговый	контроль	зна-
ний.....			32
V.	Интерактивные технологии и инновационные методы, используемые в образовательном про-		
цессе.....			32

1. РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины: воспитание достаточно высокой культуры владения компьютером в работе с сетью Интернет; формирование необходимых знаний и навыков использования современных базовых компьютерных технологий в качестве инструмента для решения практических задач в своей предметной области; привитие навыков поиска и обработки информации в глобальной сети.

Задачи дисциплины:

- дать студентам теоретические знания о глобальной компьютерной сети Internet и способах представления информации в ней;
- научить использовать формат представления данных сети для создания собственной Web – страницы;
- формирование теоретического знания о глобальной компьютерной сети Internet и способах представления информации в ней.

После изучения дисциплины студент должен знать и уметь использовать:

- владеть основными методами, способами и средствами получения, хранения, переработки информации;
- знать историю и тенденции развития глобальной сети Internet и систему адресации;
- работать с информацией в глобальных компьютерных сетях;
- уметь работать с программными средствами поискового назначения;
- уметь использовать возможности сети Интернет для поиска и обработки данных и организации информационного обмена;
- знать проблемы информационной безопасности компьютерных систем и методы защиты информации;
- уметь создавать Web - страницы.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

Предлагаемая дисциплина относится к математическому и естественнонаучному циклу ООП, дисциплина по выбору. Индекс дисциплины согласно учебному плану – ЕН.В2.2.

Для успешного освоения данной дисциплины необходимы базовые знания курса «Информатика» в объеме государственного образовательного стандарта высшего профессионального образования.

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 39 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Самостоятельная работа	
1	Назначение и классификация компьютерных сетей	2	1-4	4	-	6	реферат
2	Основные службы Интернета	2	5-7	-	2	2	тест
3	Система адресации в Internet	2	8-10	-	2	3	тест

4	Поиск информации в Интернет	2	11-13	2	-	2	тест
5	Создание Web-страницы с помощью языка HTML	2	14-18	2	6	8	тест, контрольная работа
	ИТОГО			8	10	21	зачет

4. СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

4.1. Лекции

Наименование темы	Содержание темы
Назначение и классификация компьютерных сетей	Понятие компьютерной сети. Устройства сети: сервер, рабочая станция, коммуникационные узлы. Характеристики сети. Классификация сетей по территориальному признаку: LAN, MAN, WAN сети. Эталонная модель OSI. Топология компьютерных сетей. Стек протоколов TCP/IP. Программное обеспечение компьютерных сетей: одно-ранговые и с выделенным сервером. Сетевые ОС
Поиск информации в Интернет	Сравнение поисковых каталогов и указателей. Основные зарубежные поисковые системы. Отечественные поисковые системы Aport, Rambler, Яндекс, Mail
Создание Web-страницы с помощью языка HTML	Теги и атрибуты. Работа с текстом: заголовки, списки (нумерованные, маркированные, определений), комментарии. Таблицы. Графические объекты, создание ссылок при помощи ipg и gif-рисунков

4.2. Практические занятия

Наименование темы	Содержание темы
Основные службы Интернета	Служба World Wide Web и Web – пространства. Служба имен доменов (DNS). Служба Telnet. Электронная почта. Основы работы с FTP
Система адресации в Internet	Представление о структуре и системе адресации. Структура Internet. Цифровой IP-адрес. Доменный адрес
Создание Web-страницы с помощью языка HTML	Теги и атрибуты. Работа с текстом: заголовки, списки (нумерованные, маркированные, определений), комментарии. Таблицы. Графические объекты, создание ссылок при помощи ipg и gif-рисунков. Гиперссылки для вызова других Web-страниц. Фреймы. Формы

5. САМОСТОЯТЕЛЬНАЯ РАБОТА

№ п/п	№ раздела (темы) дисциплины	Форма (вид) самостоятельной работы	Трудоемкость в часах
1	1	Реферат	6
2	2	Подготовка к лабораторному занятию с использованием обучающего теста	2
3	3	Подготовка к лабораторному занятию с использованием обучающего теста	3
4	4	Подготовка к лабораторному занятию с использованием обучающего теста	2
5	5	Подготовка к лабораторному занятию с использованием обучающего теста. Подготовка к контрольной работе	8

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Образовательный процесс по дисциплине строится на основе комбинации следующих образовательных технологий.

Интегральную модель образовательного процесса по дисциплине формируют технологии методологического уровня: технология поэтапного формирования умственных действий, технология развивающего обучения, элементы технологии развития критического мышления.

Реализация данной модели предполагает использование следующих технологий стратегического уровня, осуществляемых с использованием определенных тактических процедур:

- практические (углубление знаний, полученных на теоретических занятиях, решение задач);
- тренинговые (формирование определенных умений и навыков, формирование алгоритмического мышления);
- активизации познавательной деятельности (приемы технологии развития критического мышления через чтение и письмо, работа с литературой, подготовка презентаций по темам домашних работ);
- самоуправления (самостоятельная работа студентов, самостоятельное изучение материала).

Рекомендуется использование информационных технологий при организации коммуникации со студентами для представления информации, выдачи рекомендаций и консультирования по оперативным вопросам (электронная почта), использование мультимедиа-средств при проведении лабораторных занятий.

7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

В качестве основных средств текущего контроля используется тестирование. В качестве дополнительной формы текущего контроля предлагаются аудиторские и внеаудиторские письменные задания (контрольные работы).

Для самостоятельной работы используется учебно-методическое обеспечение на бумажных и электронных носителях. Тематика самостоятельной работы соответствует содержанию разделов дисциплины. Освоение материала контролируется в процессе проведения лабораторных занятий.

Контрольные вопросы и задания для проведения текущего контроля выбираются из содержания разделов дисциплины. Выполнение домашнего задания обеспечивает непрерывный контроль за процессом освоения учебного материала каждого обучающегося, своевременное выявление и устранение отставаний и ошибок.

Промежуточная аттестация по итогам освоения дисциплины: зачет.

Итоговый контроль выставляется при 100% выполнении лабораторных работ по курсу.

Вопросы к зачету

1. Интернет. Адресация в Интернет.
2. Службы Интернет: электронная почта, списки рассылки, телеконференции, всемирная паутина WWW, служба передачи файлов(FTP), ICQ.
3. Стек протоколов TCP/IP.
4. Поиск информации в Интернет.
5. Популярные браузеры Интернета. Поисковые системы.
6. Основные зарубежные поисковые системы.
7. Основные протоколы сети Интернет: http, telnet, SMTP, HTTP, FTP, POP. Электронная почта.
8. Особенности отображения HTML-документа браузерами.
9. Язык HTML, основные понятия.
10. Сохранение Web-страниц. Особенности сохранения страниц, содержащих фреймы.
11. Интерпретация тегов HTML. Основные структурные элементы HTML.
12. Заголовки. Абзацы. Логические разделы. Отображение предварительно отформатированного текста.
13. Списки: маркированные, нумерованные.
14. Создание гиперссылок.
15. Добавление графических элементов.
16. Таблицы. Атрибуты тегов таблицы.
17. Организация гиперссылок средствами HTML.
18. Создание ссылок, действующих внутри страницы.
19. Использование форм на Web-страницах.
20. Использование фреймовых структур.
21. Понятие компьютерной сети. Устройства сети: сервер, рабочая станция, коммуникационные узлы.

22. Классификация компьютерных сетей по территориальному признаку: LAN, MAN, WAN сети.
23. Типы линий связи. Типы передающей среды в компьютерных сетях: кабельный вид связи, радиосвязь.
24. Одноранговые сети, сети с выделенным сервером.
25. Эталонная модель OSI. Топология компьютерных сетей.

Вариант заданий для промежуточного контроля

1. HTTP – это ...
 - a) протокол передачи гипертекста;
 - b) язык разметки гипертекста;
 - c) группа сообщений, посвященных определенной теме.
2. Для создания нумерованного списка используют тег:
 - a) ;
 - b) ;
 - c) <DL>.
3. Укажите правильный путь к файлу:
 - a) ;
 - b) ;
 - c) .
4. Тег TR задает: a) строку таблицы; b) столбец таблицы; c) границу таблицы.
5. Гиперссылки на Web – странице могут обеспечить переход...
 - a) только в пределах данной Web – страницы;
 - b) только на Web – страницы данного сервера;
 - c) на любую Web – страницу данного региона;
 - d) на любую Web – страницу любого сервера Интернет.
6. Правильным является следующий порядок расположения тегов:
 - a) <тэг1><тэг2><тэг3> ... </тэг1></тэг2></тэг3>;
 - b) <тэг1>...</тэг1><тэг3><тэг1>...</тэг3></тэг1>;
 - c) <тэг1><тэг2><тэг3> ... </тэг3></тэг2></тэг1>;
 - d) <тэг1>...</тэг2><тэг3> ... </тэг3><тэг2>...</тэг1>.
7. Что будет отображено на Web – странице при написании следующего текста на странице: <TABLE WIDTH=50%> <TR> <TD> AAA </TD> </TR> <TR> <TD> BBB </TD></TR> </TABLE>?
 - a) таблица из 1 строки и 1 столбца;
 - b) таблица из 1 строки и 2-х столбцов;
 - c) таблица из 2-х строк и 1 столбца.
8. Гиперссылка задается тегом:
 - a) текст ;
 - b) <a="http://www.da.ru"> текст ;
 - c) текст .
9. Что обозначает атрибут <ALT> тега ?
 - a) ширину рамки;
 - b) источник картинки;
 - c) высоту картинки.
10. Атрибут ALIGN используется для:
 - a) выравнивания объектов на экране;
 - b) указания размеров рисунка;
 - c) для задания толщины обрамления рисунка.

Вариант тестовых вопросов к зачету

1. Протокол SMTP предназначен для ...
 - a) получения электронной почты;
 - b) отправки электронной почты;
 - c) просмотра Web – страниц.
2. Служба FTP в Интернете предназначена:
 - a) для создания, приёма и передачи Web-страниц;
 - b) для удалённого управления техническими системами;
 - c) для приёма и передачи файлов любого формата;
 - d) для обеспечения работы телеконференций.
3. Пропускная способность канала передачи информации измеряется в (несколько вариантов ответа):
 - a) бит/с;
 - b) Мбит/с;
 - c) Мбит;
 - d) Кбайт/с;
 - e) байт;
 - f) Мбайт
4. Протокол маршрутизации (IP) обеспечивает:
 - a) управление аппаратурой передачи данных и каналов связи;
 - b) сохранение механических, функциональных параметров физической связи в компьютерной сети;
 - c) интерпретацию данных и подготов-

ку их для пользовательского уровня; d) доставку информации от компьютера – отправителя к компьютеру получателю; e) разбиение файлов на IP-пакеты в процессе передачи и сборку файлов в процессе получения.

5. Время основными требованиями к вычислительным сетям являются ...

a) производительность; b) емкость; c) параллельность; d) надежность; e) пропускная способность; f) интегрируемость

6. Домен – это ...

a) единица измерения информации; b) название программы, для осуществления связи между компьютерами; c) часть адреса, определяющая адрес компьютера пользователя в сети; d) название устройства, осуществляющего связь между компьютерами.

7. Телеконференция – это ...

a) обмен письмами в глобальных сетях; b) информационная система с гиперсвязями; c) система обмена информацией между абонентами компьютерной сети; d) служба приема и передачи файлов любого формата.

8. Компьютер, предоставляющий свои ресурсы другим компьютерам при совместной работе, называется:

a) адаптером; b) коммутатором; c) рабочей станцией; d) сервером.

9. К службам сети Интернет не относят...

a) электронную почту (e-mail); b) службу передачи данных (FTP); c) HTML (Hyper Text Markup Language); d) World Wide Web.

10. Сжатый образ исходного текста обычно используется ...

a) в качестве ключа для шифрования текста; b) как открытый ключ в симметричных алгоритмах; c) как результат шифрования текста для его отправки по незащищенному каналу; d) для создания электронно-цифровой подписи

11. Какой тег служит для создания гипертекста?

a) `...`; b) `
... ;` c) `<A>...`; d) `<TD>...</TD>`

12. Какой атрибут устанавливает цвет фона документа?

a) bgcolor; b) background; c) color; d) нет правильного ответа

13. Как разделить текст горизонтальной чертой?

a) `<HR>`; b) `
`; c) `<A>`; d) ``

14. Как в HTML- документе выделить текст курсивом?

a) при помощи тега ``; b) используя тег `<I>`; c) при помощи тега `<U>`

15. Определение заголовка должно содержаться внутри тега:

a) `<Title>...</Title>`; b) `<Head>...</Head>`; c) `
`; d) `...`

16. К службам сети Интернет не относят...

a) Электронную почту (e-mail); b) Службу передачи данных (FTP); c) HTML (Hyper Text Markup Language); d) World Wide Web.

17. Код программы на языке HTML – это ...

a) обычный текстовый файл, созданный в текстовом редакторе; b) программа, написанная на специальном языке программирования; c) программа, написанная в Internet Explorer.

18. Что определяет атрибут BORDER у элемента разметки TABLE?

a) расстояние между ячейками; b) ширину ячейки; c) ширину границы.

19. Тег – это:

a) инструкция браузеру, указывающая способ отображения текста; b) текст, в котором используются спецсимволы; c) указатель на другой файл или объект; d) управляющая последовательность символов для написания Web-документов на языке разметки гипертекста.

20. Для вставки изображения в документ HTML используется команда:

a) `` ; b) `` ; c) ``

21. Что будет отображено на Web – странице при написании следующего текста на странице: `<BODY BGCOLOR=" BLACK" TEXT="YELLOW"> AAA BBB</BODY>?`

a) черный фон, «AAA, BBB» - желтый текст; b) черный фон, «AAA, BBB» - красный текст; c) черный фон, «AAA» - желтый, «BBB» - красный.

22. HTML – это ...

а) протокол передачи гипертекста; б) язык разметки гипертекста; с) группа сообщений, посвященных определенной теме.

23. Приложение Internet Explorer позволяет...

а) загружать новостные группы по протоколу NNTP; б) загружать Web - страницы по протоколу HTTP и файлы по протоколу FTP; с) общаться в чате по протоколу IRC; д) передавать файлы по протоколу FTP

24. Одной из поисковых систем в сети Интернет является...

а) Gov.ru; б) Lycos; с) THE BAT; д) File Search.

25. Формой написания IP – адреса является запись вида: xxx.xxx.xxx, где xxx – это...

а) десятичные числа от 0 до 256; б) десятичные числа от 0 до 255; с) двоичный код; д) буквы латинского алфавита.

26. Компьютерная сеть – это:

а) группа компьютеров, размещенных в одном помещении; б) объединение нескольких ЭВМ для совместного решения задач; с) мультимедийный компьютер с принтером, модемом и факсом.

27. Протокол – это:

а) пакет данных; б) правила организации передачи данных в сети; с) правила хранения данных в сети; д) структуризация данных в сети.

28. Какие линии связи используются для построения локальных сетей:

а) только витая пара; б) только оптоволокно; с) только толстый и тонкий коаксиальный кабель; д) витая пара, коаксиальный кабель, оптоволокно и беспроводные линии связи.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература

1. Советов Б.Я. Информационные технологии / Б.Я. Советов, В.В. Цехановский. 6-е изд. – М.: Издательство Юрайт, 2012. – 263 с. (ЭБС Университетская библиотека – online)

2. Диков А.В. Веб-технологии HTML и CSS. Учебное пособие. – 2-е изд. – М.: Директ-Медиа, 2012. – 78 с. (ЭБС Университетская библиотека – online)

б) дополнительная литература

1. Глушаков С. В. Компьютеры, программы, сети / С.В. Глушаков, А.С. Сурядный. - М.: АСТ: АСТ МОСКВА; Владимир: ВКТ, 2009. - 512 с.

2. Диков А.В. Интернет и Веб 2.0. Учебное пособие. – 2-е изд. - М.: Директ-Медиа, 2012. - 62 с. (ЭБС Университетская библиотека – online)

3. Исаев Г.Н. Информационные технологии: учеб. пособие / Г.Н. Исаев. – М.: Издательство «Омега-Л», 2012. – 464 с. (ЭБС Университетская библиотека – online)

4. Матросов А.В. HTML 4.0 / А.В. Матросов, А.С. Сергеев, М.П. Чаунин. – СПб.: БХВ – Санкт-Петербург, 2000. – 671 с.

5. Соловцова Л.А. Создание Web – страниц: лаб.практикум / Л.А. Соловцова, Н.В. Назаренко, Н.П. Семичевская. – Благовещенск, Изд-во Амур. гос. ун-та, 2002. – 44 с.

6. Шафран Э. Создание Web-страниц / Э. Шафран; пер. с англ. М. Федорова.–СПб.: Питер, 2000. – 310 с.

в) перечень журналов

1. Информационные технологии.

2. Информационные технологии и вычислительные системы.

г) Интернет-ресурсы

№	Наименование ресурса	Краткая характеристика
1	http://informatka.ru/	Содержит справочный материал по различным разделам информатики
2	http://www.iqlib.ru	Интернет-библиотека образовательных изданий, в которой собраны электронные учебники, справочные и учебные пособия. Удобный поиск по ключевым словам отдельным темам и отраслям знаний
3	http://elibrary.ru	Научная электронная библиотека журналов

№	Наименование ресурса	Краткая характеристика
4	http://www.biblioclub.ru	Электронная библиотечная система «Университетская библиотека – online»
5	«Интернет - технологии» (Учебно-методическое пособие)	\\10.4.1.254\DigitalLibrary\AmurSU_Edition\110.pdf

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Компьютерный класс, оснащенный ПЭВМ, с установленным программным обеспечением.

II. КРАТКОЕ ИЗЛОЖЕНИЕ ПРОГРАММНОГО МАТЕРИАЛА

Тема 1. Назначение и классификация компьютерных сетей.

План лекции:

1. Понятие компьютерной сети.
2. Устройства сети: сервер, рабочая станция, коммуникационные узлы.
3. Характеристики сети.
4. Классификация сетей по территориальному признаку: LAN, MAN, WAN сети.
5. Эталонная модель OSI.
6. Топология компьютерных сетей.
7. Стек протоколов TCP/IP.
8. Программное обеспечение компьютерных сетей: одноранговые и с выделенным сервером.
9. Сетевые ОС.

Цель: приобретение теоретических знаний и практических навыков выбора и использования вычислительных систем и сетей для решения экономических и информационных задач.

Задачи:

- ознакомить с общими сведениями о локальных и глобальных сетях ЭВМ;
- сформировать представление об основных конфигурациях сетей, типах подключения к сети.

Ключевые вопросы:

1. Дать определение компьютерной сети и её назначение.
2. По какому принципу строится архитектура сетей?
3. Как классифицируются компьютерные сети по территориальному признаку?
4. Какие существуют разновидности корпоративных сетей.
5. Дайте определение понятие «клиент», «сервер».
6. Какие ресурсы рабочих станций могут быть доступным для работы в локальной сети?
7. Какие задачи решаются рабочими станциями, а какие сервером?
8. Какая основная файловая операция используется для передачи документов по локальной сети?
9. Какие ресурсы сервера предоставляются рабочим станциям?
10. какие компоненты входят в локальную сеть учебных классов?
11. На каком уровне модели ISO/OSI работают протоколы TCP/IP, какие задачи они решают?

Рекомендуемая литература:

1. Советов Б.Я. Информационные технологии / Б.Я. Советов, В.В. Цехановский. 6-е изд. – М.: Издательство Юрайт, 2012. – 263 с. (ЭБС Университетская библиотека – online)
2. Глушаков С. В. Компьютеры, программы, сети / С.В. Глушаков, А.С. Сурядный. - М.: АСТ: АСТ МОСКВА; Владимир: ВКТ, 2009. - 512 с.

Тема 2. Поиск информации в Интернет.

План лекции:

1. Сравнение поисковых каталогов и указателей.
2. Основные зарубежные поисковые системы.
3. Отечественные поисковые системы Aport, Rambler, Яндекс, Mail

Цель: приобретение теоретических знаний и практических навыков выбора и использования вычислительных систем и сетей для решения экономических и информационных задач.

Задача:

- научить работе с сетью с целью поиска и сохранения информации.

Ключевые вопросы:

1. Как называются документы, опубликованные в Интернете в формате службы Word Wide Web?
2. Для чего предназначен Браузер?
3. Назовите средства поиска информации в Интернет.

Рекомендуемая литература:

1. Советов Б.Я. Информационные технологии / Б.Я. Советов, В.В. Цехановский. 6-е изд. – М.: Издательство Юрайт, 2012. – 263 с. (ЭБС Университетская библиотека – online)
2. Глушаков С. В. Компьютеры, программы, сети / С.В. Глушаков, А.С. Сурядный. - М.: АСТ: АСТ МОСКВА; Владимир: ВКТ, 2009. - 512 с.

Тема 3. Создание Web-страницы с помощью языка HTML.

План лекции:

1. Теги и атрибуты.
2. Работа с текстом: заголовки, списки (нумерованные, маркированные, определений), комментарии.
3. Таблицы.
4. Графические объекты, создание ссылок при помощи ipg и gif-рисунков

Цель: приобретение теоретических знаний и практических навыков выбора и использования вычислительных систем и сетей для решения экономических и информационных задач.

Задача:

– научить создавать Web-страницы.

Ключевые вопросы:

1. Как называются документы, опубликованные в Интернете в формате службы Word Wide Web?
2. Для чего предназначен Браузер?
3. Назовите средства поиска информации в Интернет.

Рекомендуемая литература:

1. Диков А.В. Веб-технологии HTML и CSS. Учебное пособие. – 2-е изд. – М.: Директ-Медиа, 2012. – 78 с. (ЭБС Университетская библиотека – online)
2. Диков А.В. Интернет и Веб 2.0. Учебное пособие. – 2-е изд. - М.: Директ-Медиа, 2012. - 62 с. (ЭБС Университетская библиотека – online)
3. Матросов А.В. HTML 4.0 / А.В. Матросов, А.С. Сергеев, М.П. Чаунин. – СПб.: БХВ – Санкт-Петербург, 2000. – 671 с.
4. Соловцова Л.А. Создание Web – страниц: лаб.практикум / Л.А. Соловцова, Н.В. Назаренко, Н.П. Семичевская. – Благовещенск, Изд-во Амур. гос. ун-та, 2002. – 44 с.
5. Шафран Э. Создание Web-страниц / Э. Шафран; пер. с англ. М. Федорова.–СПб.: Питер, 2000. – 310 с.

III. МЕТОДИЧЕСКИЕ УКАЗАНИЯ (РЕКОМЕНДАЦИИ)

1. Методические указания по изучению дисциплины

Комплексное изучение предлагаемой студентам учебной дисциплины «Информационные технологии» предполагает приобретение практических навыков работы на ПК в глобальной сети Интернет, творческую работу при выполнении индивидуальных самостоятельных заданий.

Процесс по освоению всей совокупности теоретического и практического материала по дисциплине должен быть реализован в течение одного семестра и, проходить в соответствии с предложенным выше планом.

Лабораторные занятия проводятся в компьютерных классах с применением специально разработанных учебно-методических пособий, в которых изложены подробные методические рекомендации по изучению каждой темы и выполнению заданий. Наличие таких учебно-методических и дидактических материалов позволяет каждому студенту работать в своем индивидуальном темпе, а также дополнительно прорабатывать изучаемый материал во время самостоятельных занятий, в т.ч. дома. Все эти материалы имеются на кафедре в печатном виде и доступны в электронном виде во всех компьютерных классах.

Вместе с тем, каждая новая тема сначала объясняется преподавателем, рассматривается на примерах, затем для закрепления полученных на занятии знаний студенты выполняют соответствующие упражнения и получают домашние задания. Полученные оценки за выполненные упражнения и домашние задания являются основой для выставления промежуточной и итоговой аттестации. Итоговой аттестацией в пятом семестре является зачет. Зачет проводится с использованием тестов.

Для закрепления полученных теоретических и практических знаний студентам в течение всего семестра предлагаются индивидуальные задания для самостоятельной работы. Особенности выполнения самостоятельной работы и тематика индивидуальных заданий подробно изложены в методических указаниях по их выполнению. Консультирование по выполнению индивидуальных заданий проводится как непосредственно в компьютерных классах (во время консультаций). Контроль выполненных заданий осуществляется либо непосредственно на занятиях, либо на консультациях.

Наличие методических рекомендаций по изучению каждой темы, большого набора заданий для самостоятельной работы по закреплению изучаемого материала (как в виде электронных заданий, так и в виде печатного сборника), компьютерных тестов для контроля знаний по каждой теме позволяет повысить эффективность учебного процесса.

Для подготовки к зачету студентам рекомендуются подготовленные преподавателями кафедры учебник и практикум, включающий терминологическую часть, вопросы для самоконтроля и тесты.

2. Методические указания к практическим занятиям

Задачей преподавателя при проведении практических занятий является грамотное и доступное разъяснение принципов и правил проведения работ, побуждение студентов к самостоятельной работе, определения места изучаемой дисциплины в дальнейшей профессиональной работе будущего специалиста.

Цель практического занятия – научить студентов самостоятельно производить необходимые действия для достижения желаемого результата.

Прежде чем приступить к выполнению заданий практического занятия, студенту необходимо ознакомиться с теоретическим материалом, соответствующим данной теме.

Выполнение заданий целесообразно разделить на несколько этапов:

- формулировка и обоснование цели работы;
- определение теоретического аппарата, применительно к данной теме;
- выполнение заданий;
- анализ результата;
- выводы.

Индивидуальные задания для практических занятий должны быть представлены конкретно-практическими и творческими задачами.

На первой ступени изучения темы выполняются конкретно-практические задачи, при решении которых формируется минимальный набор умений. Преподаватель опосредованно руководит познавательной деятельностью студентов, консультирует и подробно разбирает со студентами возникшие затруднения в ходе решения задачи, обращает внимание группы на возможные ошибки.

Вторая ступень изучения темы дифференцируется в зависимости от степени усвоения его обязательного уровня. Студенты, усвоив содержание типовых методов и приемов решения задач, приступают к решению творческих задач. Если уровень знаний и умений, демонстрируемых студентом при контрольном обследовании, не соответствует установленным требованиям, студент вновь возвращается к стандартным упражнениям, но под более пристальным наблюдением преподавателя.

После изучения отдельной темы курса дисциплины, каждый студент получает оценку по результатам выполнения заданий.

Студенты, пропустившие практические занятия, должны их выполнить во внеаудиторное время и отчитаться до начала зачетно-экзаменационной сессии.

Рекомендации для организации рабочего места студента: для проведения лабораторных работ требуется компьютерный класс с установленным программным обеспечением.

Правила техники безопасности в компьютерном классе:

1. Находиться в компьютерном классе без разрешения преподавателя.
2. Включать без разрешения оборудование.
3. Трогать разъемы соединительных кабелей и проводов.
4. Прикасаться к питающим проводам и устройствам заземления.
5. Включать и выключать аппаратуру без указания преподавателя.
6. Работать в верхней одежде и влажными руками.
7. Класть диски, книги, тетради и другие предметы на монитор и клавиатуру.
8. При появлении запаха гари немедленно прекратите работу, выключите аппаратуру и сообщите об этом преподавателю.

Методические указания к практическим занятиям

Тема: Основные службы Интернета. Система адресации в Интернет.

План:

1. Служба World Wide Web и Web – пространства.
2. Служба имен доменов (DNS).

3. Служба Telnet.
4. Электронная почта.
5. Основы работы с FTP.
6. Система адресации в Интернет.

Объем аудиторных часов: 4 ч.

Объем часов для самостоятельной работы: 5 ч.

Указания к практической работе:

Работа с электронной почтой на почтовых WWW-серверах

1. Запустите программу Internet Explorer через кнопку Пуск - Программы - Internet Explorer или с помощью значка на Рабочем столе (Панели задач).
 2. В адресной строке браузера введите адрес сайта www.yandex.ru.
 3. Выберите ссылку Почта - Зарегистрироваться или Завести почтовый ящик.
 4. Заполните форму регистрации.
 5. Подтвердите данные, нажав кнопку Зарегистрировать.
 6. После успешной регистрации появляется ваш личный адрес.
 7. Подтвердите согласие, нажав кнопку Сохранить.
 8. Откройте свой почтовый ящик.
 9. Откройте свой почтовый ящик на бесплатном почтовом сервере введя логин и пароль в соответствующую форму.
 10. Создайте сообщение с темой «Распоряжение»:
 - Щелкните по кнопке панели инструментов написать автору;
 - заполните заголовки сообщения: Кому, Копия, Тема следующим образом: в заголовке Кому укажите адрес преподавателя, Копия – адрес «соседа слева». В качестве Темы укажите «Распоряжение»;
 - впишите текст сообщения.
 11. Отправьте сообщение с помощью кнопки Отправить или воспользовавшись соответствующей гиперссылкой.
 12. Перейдите в папку Входящие. Для того, чтобы прочитать полученное сообщение, необходимо нажать на ссылку в поле От кого
 13. В появившемся окне нажать на кнопку Ответить . Напишите ответ на это письмо и нажмите на кнопку Отправить.
 14. Создайте новое сообщение и вложите в него файл:
 - в редакторе Microsoft Word создайте файл-проект приказа по персоналу с именем r1kaz_N.doc и сохраните его в своем каталоге (D:\Рабочая);
 - вернитесь в свой электронный ящик;
 - щелкните по кнопке панели инструментов Написать
 - заполните заголовки сообщения: Кому, Копия, Тема следующим образом: в заголовке Кому укажите адрес преподавателя, Копия – адрес «соседа справа». В качестве Темы укажите «Приказ по персоналу»;
 - нажмите на кнопку Обзор, укажите местонахождение файла (D:\Рабочая\);
 - напишите текст сообщения.
 15. Отправьте сообщение, нажав на соответствующую кнопку.
 16. Перейдите в папку Входящие. В списке сообщений найдите электронное письмо с темой «Приказ по персоналу», отправленное «соседом слева». Значок в виде скрепки свидетельствует о наличии в полученном письме вложения. Сохраните вложенный файл в папке D:\Рабочая\Ваша фамилия:
 - откройте полученное сообщение;
 - щелкните по значку вложенного файла левой кнопкой мыши;
 - в появившемся окне нажмите на кнопку Сохранить;
 - укажите путь сохранения D:\Рабочая\Ваша фамилия.
 17. Сообщение с темой «Приказ по персоналу» перешлите преподавателю.
 - откройте нужное письмо и нажмите на кнопку Переслать ;
 - заполните поле Кому, впишите электронный адрес преподавателя и отправьте сообщение.
- Тема: Создание Web-страницы с помощью языка HTML.

План:

1. Теги и атрибуты.
2. Работа с текстом: заголовки, списки (нумерованные, маркированные, определенных), комментарии.
3. Таблицы.
4. Графические объекты, создание ссылок при помощи `img` и `gif`-рисунков.
5. Гиперссылки для вызова других Web-страниц.
6. Фреймы.
7. Формы.

Объем аудиторных часов: 6 ч.

Объем часов для самостоятельной работы: 8 ч.

Указания к практической работе:

Сценарий и структура Web-документа

Для того чтобы понять структуру и сценарий Web-документа, мы должны рассмотреть несколько Web-страниц и выявить общие элементы.

Любой Web-документ состоит из тегов, причем обычно начальные теги пишутся большими буквами, а конечные – маленькими.

1. **<HTML></html>**.

Отличительный признак HTML-документа. Одним из принципов языка является многоуровневое вложение элементов. HTML является самым внешним, так как между его стартовым и конечным тегами должна находиться вся Web-страница.

2. **<HEAD></head>**.

Область заголовка Web-страницы. Иными словами, ее первая часть. Так же, как и HTML, HEAD служит только для формирования общей структуры документа.

3. **<BODY></body>**.

Элемент включает в себе гипертекст, который определяет собственно Web-страницу. Эта та часть документа, которую разрабатывает автор страницы и которая отображается Браузером. Соответственно, конечный тег этого элемента надо искать в конце HTML-файла. Внутри BODY можно использовать все элементы, предназначенные для дизайна Web-страницы. Внутри стартового тега элемента BODY можно расположить ряд атрибутов, обеспечивающих установки для всей страницы целиком.

Структура текста Web-документа

1. **<COMMENT></comment>**.

Текст комментария. В любом языке программирования есть конструкции, позволяющие создавать произвольные ремарки. HTML в этом смысле – не исключение. Текст, помещенный внутри COMMENT, игнорируется Браузером. COMMENT может располагаться в любом месте кода Web-страницы. Без конечного тега здесь, по-видимому, не обойтись: комментарий должен быть отделен от основного текста.

Существует, правда, одно ограничение: внутри комментария не должны располагаться другие элементы. Так должно быть, разумеется, только в том случае, когда необходимо, чтобы все содержимое элемента COMMENT не отображалось на экране монитора. Если в комментарии будет присутствовать другой элемент, то его содержимое будет выведено на экран, отформатированное соответствующим образом. С этой проблемой связана другая, похожая: обычный текст не может содержать фрагменты, имеющие вид тегов. Для того чтобы включать в текст подобные конструкции, существует элемент PLAINTEXT, о котором речь пойдет ниже.

Существует и другой способ обозначения комментария. Он заключается в использовании восклицательного знака и обрамлении текста комментария двойным тире. Например:

```
<!--Строка комментария--> <!--Комментарий- Не комментарий- Снова комментарий-->
```

Внутри подобной конструкции можно помещать и теги: они не будут восприниматься Браузером.

2. **<HTML></html>**.

Отличительный признак HTML-документа. Одним из принципов языка является многоуровневое вложение элементов. HTML является самым внешним, так как между его стартовым и конечным тегами должна находиться вся Web-страница. В принципе этот элемент можно рассматривать как формальность. Он имеет атрибуты `version`, `lang` и `dir`, которыми в данном случае мало кто пользуется и допуска-

ет вложение элементов HEAD, BODY и PLAINTEXT, определяющих общую структуру Web-страницы. Естественно, что конечным тегом </html> заканчиваются все гипертекстовые документы.

3. <HEAD></head>.

Область заголовка Web-страницы. Иными словами, ее первая часть. Так же, как и HTML, HEAD служит только для формирования общей структуры документа. Этот элемент может иметь атрибуты lang и dir, и допускается вложение элементов TITLE, ISINDEX, BASE, META, LINK, NEXTID.

4. <TITLE></title>.

Элемент для размещения заголовка Web-страницы. Строка текста, расположенная внутри, отображается не в документе, а в заголовке окна Браузера. Эта особенность часто используется для организации поиска в WWW. Поэтому авторы, создающие Web-страницы, должны позаботиться о том, чтобы строка внутри TITLE, не будучи слишком длинной, достаточно отображала назначение документа.

5. <STYLE></style>.

Описание стиля некоторых элементов Web-страницы. В приведенном выше примере назначены шрифты для элементов H2 и CODE. Естественно, что для каждого элемента существует стилевое оформление по умолчанию, поэтому употребление STYLE не обязательно.

6. <META>.

Этот элемент содержит служебную информацию, которая не отображается при просмотре Web-страницы. Внутри него нет текста в обычном понимании, поэтому нет и конечного тега. Каждый элемент META содержит два основных атрибута, первый из которых определяет тип данных, а второй – содержание. Далее приведены несколько примеров meta-данных.

7. <BODY></body>.

Данный элемент заключает в себе гипертекст, который определяет собственно Web-страницу. Это та часть документа, которую разрабатывает автор страницы и которая отображается Браузером. Соответственно, конечный тег этого элемента надо искать в конце HTML-файла. Внутри BODY можно использовать все элементы, предназначенные для дизайна Web-страницы. Внутри стартового тега элемента BODY можно расположить ряд атрибутов, обеспечивающих установки для всей страницы целиком. Рассмотрим их по порядку.

Один из самых полезных для дизайна атрибут, определяющий фон страницы. Его появление можно уподобить маленькой революции в WWW, так как одинаковые серые Web-страницы благодаря ему расцвели яркими цветными узорами:

background= ''Путь к файлу фона''.

Более простое оформление фона сводится к заданию его цвета:

bcolor= ''#RRGGBB''.

Цвет фона задается тремя двухразрядными шестнадцатеричными числами, которые определяют интенсивность красного, зеленого и синего цветов соответственно.

Оба названных атрибута не являются альтернативными и часто используются совместно.

Поскольку фон страницы может изменяться, необходимо иметь возможность подбирать соответствующий цвет текста. Для этого существует следующий атрибут:

text= ''#RRGGBB''.

Линейки. Горизонтальные линейки задаются одиночным тегом <HR>. Атрибут ALIGN задает способ выравнивания линейки:

LEFT – слева,

RIGHT – справа,

CENTER – по центру.

Заголовки. В примере для заголовка используются теги <H1></h1>.

Существует шесть уровней заголовков, которые обозначаются H1... H6. Заголовок уровня 1 самый крупный, а уровень 6 обеспечивает самый маленький заголовок.

Для заголовков можно использовать атрибут, задающий выравнивание влево, по центру или вправо:

align= ''left'' align= ''center'' align= ''right''.

Заголовки, создаваемые при помощи элементов H1... H2. Элемент FONT может с успехом заменять элементы заголовка H1... H6. Для последних, например, не предусмотрена возможность указания цвета букв. Чтобы заголовок, созданный на основе элемента FONT, хорошо смотрелся, этот элемент необходимо комбинировать с другими: CENTER, B, I, P и т.д.

Абзацы. Текст – единственный объект Web-страницы, который не требует специального определения. Иными словами, произвольные символы интерпретируются по умолчанию как текстовые данные. Но для форматирования текста существует большое количество элементов.

8. **<P></p>**.

Элемент абзаца (paragraph) – один из самых полезных. Он позволяет использовать только начальный тег, так как следующий элемент P обозначает конец предыдущего и начало следующего абзаца. Конечный тег удобно использовать в тех случаях, когда по смыслу необходимо обозначить конец абзаца. Вместе с элементом P можно применять атрибут выравнивания align.

Логические разделы.

1. **
**.

Элемент, обеспечивающий принудительный переход на новую строку. Он имеет только стартовый тег. В месте его размещения строка заканчивается, а оставшийся текст печатается с новой строки.

2. **<NOBR></nobr>**.

Этот элемент по своему действию является прямой противоположностью предыдущему. Текст, заключенный между его тегам, будет выведен в одну строку. Длинная строка не уместится на экране, и для ее просмотра придется использовать горизонтальную полосу прокрутки.

3. **<PRE></pre>**.

Элемент для обозначения текста, отформатированного заранее (preformatted). Подразумевается, что текст будет выведен в том виде, в каком он был подготовлен пользователем. Например, учитываются символы конца строки, появившиеся при наборе текста в редакторе. Во всех других случаях Браузер игнорирует эти символы. Этот элемент удобно использовать для демонстрации листингов программ или для вывода текстовых документов, переформатирование которых может привести к искажению их смысла.

4. **<BLOCKQUOTE></blockquote>**.

Обозначение цитаты. Элемент требует наличия конечного тега. Текст не претерпевает никаких изменений, но абзац располагается с отступом. К кавычкам этот элемент тоже не имеет никакого отношения: если в цитате используются кавычки, то они должны быть проставлены явным образом. В настоящее время используют и другое написание этого элемента: BQ.

5. **<CENTER></center>**.

Элемент используется для центрирования текста, а точнее, любого содержимого. Не является общеупотребительным.

6. **<DIV></div>**.

Элемент, похожий на предыдущий. Он позволяет выравнивать содержимое по левому краю, по центру или по правому краю. Для этого стартовый тег должен содержать соответствующий атрибут:

align="left" align="center" align="right".

Задание 1. На диске C: создать свою папку, а в ней – текстовый файл с именем *first.htm*.

1. Запустить редактор *Блокнот*, ввести в него текст:

Приветствую вас на моей первой web-страничке.

2. Сохранить файл в созданной папке. При сохранении в окне диалога *Сохранить как...* в строке *Тип файла:* выбрать вариант *Все файлы*, а строке *Имя файла* задать имя с расширением *htm*, – например, *first.htm*.

3. Закрыть документ, найти его пиктограмму в окне *Проводника*.

4. Открыть файл непосредственно из окна *Проводника*. Проанализировать, с помощью какого приложения отображается файл и как выглядит введенная фраза.

5. Сделать выводы о том, что HTML-документ – это всего лишь текст.

Задание 2. Ввести теги, определяющие структуру HTML-документа.

1. С помощью меню Браузера *Вид* → *В виде HTML* вызвать документ для его редактирования. Ввести приведенные ниже теги, в разделе *Заголовка* (TITLE) указать свою фамилию.

<HTML>

<HEAD><TITLE>Фамилия</TITLE>

</HEAD>

<BODY>

Приветствую вас на моей первой web-страничке

</BODY>

</HTML>

2. Сохранить документ под тем же именем. Обновить его отображение в Браузере. Проанализировать произошедшие изменения в отображении документа.

Задание 3. Отредактировать документ.

1. Вызвать меню Браузера *Вид* → *В виде HTML* и добавить текст подписи, например:

Студент группы XX факультета XXX Имя Фамилия

2. Сохранить документ и обновить его просмотр в Браузере.

3. Отредактировать документ так, чтобы подпись начиналась с новой строки, для этого использовать тег
. Просмотреть в Браузере новый вариант.

Задание 4. Выполнить оформление текста стилем *Заголовок*.

1. Оформить первую строку документа стилем Заголовок-1 с помощью парного тега <H1>...</H1>.

2. Оформить строку с подписью стилем Заголовок-3.

3. Изменить стиль оформления первой строки на Заголовок-2, а подпись – на Заголовок-4.

Задание 5. Выполнить оформление абзацев.

1. После заголовка ввести текст монолога Гамлета:

Быть или не быть – вот в чем вопрос.

Что благороднее: сносить удары

Неистойвой судьбы – или против моря

Невзгод вооружиться, в бой вступить

И все покончить разом...

2. Оформить монолог как отдельный абзац с помощью тега <P>.

Задание 6. Оформление абзацев с помощью парного тега <PRE>.

1. Скопировать в конец документа монолог, поместить его вовнутрь тега PRE и оформить по приведенному образцу «лесенкой»:

Быть или не быть – вот в чем вопрос.

 Что благороднее: сносить удары

 Неистойвой судьбы – или против моря

 Невзгод вооружиться, в бой вступить

 И все покончить разом...

2. Сохранить документ *first.htm*, скопировать его на диске A: для дальнейшей работы.

Списки

Язык HTML поддерживает 5 видов списков: нумерованный; маркированный; список определенных; список каталогов; список меню.

Все списки оформляются однотипным образом. Начало и конец списка помечаются парным тегом, описывающим тип списка. Элементы списка – это блочные элементы HTML. Внутри каждого элемента списка можно использовать любые блочные или тестовые элементы HTML.

Маркированный список задается тегом и может включать атрибут TYPE, описывающий тип маркера. Например:

TYPE= «DISC» – круглый черный маркер (используется по умолчанию);

TYPE= «SQUAR» – квадратный четный маркер;

TYPE= «CIRCLE» – маркер в виде колечка.

Нумерованный список – задается с помощью тега и может иметь значения атрибута TYPE=:

TYPE= «I» – арабские списки (используются по умолчанию);

TYPE= «a» – латинские буквы нижнего регистра (a, b, c...);

TYPE= «A» – латинские буквы верхнего регистра (A, B, C...);

TYPE= «i» – римские цифры с использованием нижнего регистра (i, ii, iii...);

TYPE= «I» – римские цифры с использованием верхнего регистра (I, II, III...).

Элементы списка, задаваемые внутри маркированного и нумерованного списка, описываются парным тегом .

Списки определения описываются парным тегом <DL>. Предполагается, что данный список представляет собой «словарь», в котором чередуются термины и определения. Термины выделяются обычно полужирным шрифтом, а определения отображаются с отступом от левого поля.

Форматирование текста

Для задания размера, цвета и начертания шрифта служит парный тег ****, влияющий на весь текст, заключенный между открывающим и закрывающим тегами. Тег **** должен иметь хотя бы один из трех возможных атрибутов:

SIZE – задает размер шрифта. Возможны семь заранее заданных размеров шрифтов (от 1 до 7). Чем больше значение, тем крупнее шрифт. По умолчанию используется значение 3;

COLOR – задает цвет шрифта, может быть задан либо ключевым словом (например, **RED** – красный), либо шестнадцатеричным значением в системе RGB (см. приложение 1);

FACE – задает вид шрифта. Этот атрибут называет один из шрифтов, установленных на компьютере.

Чтобы задать значения этих параметров для всего документа в целом, используют одиночный тег **<BASEFONT>**, содержащий аналогичные атрибуты и задающий значения вида, цвета и размера шрифта, используемые по умолчанию.

Специальная группа тегов служит для изменения начертания шрифтов:

**** – выделение текста полужирным шрифтом. Очень популярный элемент. Использование полужирного шрифта – прием, позаимствованный из текстовых редакторов.

<BIG></big> – увеличенный размер шрифта.

<SMALL></small> – уменьшенный размер шрифта.

<I></i> – выделение текста курсивом.

**** и **<DFN></dfn>** – элементы, означающие выразительность фрагмента текста и определение чего-либо. Оба элемента аналогичны по своему действию элементу **I**, т. е. в большинстве случаев позволяют выделить текст курсивом.

Они могут пригодиться только для того, чтобы единообразно выделить одинаковые по назначению (или смыслу) фрагменты текста, находящиеся в разных частях документа или даже на разных страницах. Разработчик в этом случае не может точно знать, какой именно шрифт будет использован: это определяется каждым Браузером по-своему. Но он может быть точно уверен, что все фрагменты текста будут отформатированы одинаково. В языке можно найти еще несколько элементов, которым можно дать такую же характеристику:

<U></u> – подчеркнутый текст;

<S></s> – вычеркивание текста;

**** – подстрочный текст;

**** – надстрочный текст;

<TT></tt> – тег фиксированной ширины букв (машинописный режим).

Задание 7. Выполнить шрифтовое оформление документа.

1. Создать документ *second.htm* в вашей папке на жестком диске. За основу документа взять файл *first.htm*, вставить в нем заголовок, монолог и подпись.

2. 1-ю и 3-ю строки оформить полужирным шрифтом, 2-ю и 4-ю – курсивом, 5-ю – подчеркнутым.

Задание 8. Оформление списков.

1. Дополнить текст документа – ввести после заголовка еще три строки:

Я знаю, как оформлять:

Заголовки,

Абзацы

2. Оформить две последние строки как список нумерованный.

3. Дополнить список своих знаний. Например, между пунктами «Заголовки» и «Абзацы» добавить пункт «Текст». Проанализировать, как изменилась нумерация элементов списка.

4. Создать вложенный список. Добавить уточнение видов оформления шрифтов и абзацев и оформить список по следующему образцу:

Я знаю, как оформлять:

1. Шрифты

○ Размер

○ Цвет

○ Гарнитура

2. Заголовки

- От 1-го до 6-го уровня

3. Абзацы

- Выравнивание
- Разрыв строк внутри абзаца
- Использовать переформатирование

Задание 9. Создать новый документ HTML, состоящий из нескольких абзацев. Для первого задайте тег: `<BASEFONT SIZE="5" COLOR="0000FF">`, для второго – ``. Ввести текст абзаца, закончив его тегом ``.

1. В следующем абзаце использовать по своему умолчанию парные теги: ``, `<I>`, `<U>`, `<S>`, `<SUB>`, `<SUP>`. Можете использовать вложенное форматирование.

2. В четвертом абзаце использовать по своему усмотрению парные теги: `<CITE>`, ``, ``, `<CODE>`, `<SAMP>`, `<VAR>`.

3. Сохранить файл под именем *format.html*.

4. Запустить Internet Explorer и посмотреть свой файл. Обратит внимание на способ отражение вашего текста.

Списки

Язык HTML поддерживает 5 видов списков:

- нумерованный;
- маркированный;
- список определений;
- список каталогов;
- список меню.

Все списки оформляются однотипным образом. Начало и конец списка помечаются парным тегом, описывающим тип списка. Элементы списка – это блочные элементы HTML. Внутри каждого элемента списка можно использовать любые блочные или тестовые элементы HTML.

Маркированный список задается тегом `` и может включать атрибут `TYPE`, описывающий тип маркера. Например:

`TYPE= «DISC»` – круглый черный маркер (используется по умолчанию);

`TYPE= «SQUAR»` – квадратный четный маркер;

`TYPE= «CIRCLE»` – маркер в виде колечка.

Нумерованный список – задается с помощью тега `` и может иметь значения атрибута `TYPE=`:

`TYPE= «I»` – арабские списки (используются по умолчанию);

`TYPE= «a»` – латинские буквы нижнего регистра (a, b, c...);

`TYPE= «A»` – латинские буквы верхнего регистра (A, B, C...);

`TYPE= «i»` – римские цифры с использованием нижнего регистра (i, ii, iii...);

`TYPE= «I»` – римские цифры с использованием верхнего регистра (I, II, III...).

Элементы списка, задаваемые внутри маркированного и нумерованного списка, описываются парным тегом ``.

Списки определения описываются парным тегом `<DL>`. Предполагается, что данный список представляет собой «словарь», в котором чередуются термины и определения. Термины выделяются обычно полужирным шрифтом, а определения отображаются с отступом от левого поля.

Форматирование текста

Для задания размера, цвета и начертания шрифта служит парный тег ``, влияющий на весь текст, заключенный между открывающим и закрывающим тегами. Тег `` должен иметь хотя бы один из трех возможных атрибутов:

`SIZE` – задает размер шрифта. Возможны семь заранее заданных размеров шрифтов (от 1 до 7). Чем больше значение, тем крупнее шрифт. По умолчанию используется значение 3;

`COLOR` – задает цвет шрифта, может быть задан либо ключевым словом (например, RED – красный), либо шестнадцатеричным значением в системе RGB (см. приложение 1);

`FACE` – задает вид шрифта. Этот атрибут называет один из шрифтов, установленных на компьютере.

Чтобы задать значения этих параметров для всего документа в целом, используют одиночный тег `<BASEFONT>`, содержащий аналогичные атрибуты и задающий значения вида, цвета и размера шрифта, используемые по умолчанию.

Специальная группа тегов служит для изменения начертания шрифтов:

`` – выделение текста полужирным шрифтом. Очень популярный элемент. Использование полужирного шрифта – прием, позаимствованный из текстовых редакторов.

`<BIG></big>` – увеличенный размер шрифта.

`<SMALL></small>` – уменьшенный размер шрифта.

`<I></i>` – выделение текста курсивом.

`` и `<DFN></dfn>` – элементы, означающие выразительность фрагмента текста и определение чего-либо. Оба элемента аналогичны по своему действию элементу `I`, т. е. в большинстве случаев позволяют выделить текст курсивом.

Они могут пригодиться только для того, чтобы единообразно выделить одинаковые по назначению (или смыслу) фрагменты текста, находящиеся в разных частях документа или даже на разных страницах. Разработчик в этом случае не может точно знать, какой именно шрифт будет использован: это определяется каждым Браузером по-своему. Но он может быть точно уверен, что все фрагменты текста будут отформатированы одинаково. В языке можно найти еще несколько элементов, которым можно дать такую же характеристику:

`<U></u>` – подчеркнутый текст;

`<S></s>` – вычеркивание текста;

`` – подстрочный текст;

`` – надстрочный текст;

`<TT></tt>` – тег фиксированной ширины букв (машинописный режим).

Задание 10. Выполнить шрифтовое оформление документа.

1. Создать документ *second.htm* в вашей папке на жестком диске. За основу документа взять файл *first.htm*, вставить в нем заголовок, монолог и подпись.

2. 1-ю и 3-ю строки оформить полужирным шрифтом, 2-ю и 4-ю – курсивом, 5-ю – подчеркнутым.

Задание 11. Оформление списков.

1. Дополнить текст документа – ввести после заголовка еще три строки:

Я знаю, как оформлять:

Заголовки,

Абзацы

2. Оформить две последние строки как список нумерованный.

3. Дополнить список своих знаний. Например, между пунктами «Заголовки» и «Абзацы» добавить пункт «Текст». Проанализировать, как изменилась нумерация элементов списка.

4. Создать вложенный список. Добавить уточнение видов оформления шрифтов и абзацев и оформить список по следующему образцу:

Я знаю, как оформлять:

1. Шрифты

○ Размер

○ Цвет

○ Гарнитура

2. Заголовки

○ От 1-го до 6-го уровня

3. Абзацы

○ Выравнивание

○ Разрыв строк внутри абзаца

○ Использовать переформатирование

Задание 12. Создать новый документ HTML, состоящий из нескольких абзацев. Для первого задайте тег: `<BASEFONT SIZE=“5” COLOR=“0000FF”>`, для второго – ``. Ввести текст абзаца, закончив его тегом ``.

1. В следующем абзаце использовать по своему умолчанию парные теги: ``, `<I>`, `<U>`, `<S>`, `<SUB>`, `<SUP>`. Можете использовать вложенное форматирование.

2. В четвертом абзаце использовать по своему усмотрению парные теги: <CITE>, , , <CODE>, <SAMP>, <VAR>.

3. Сохранить файл под именем *format.html*.

4. Запустить Internet Explorer и посмотреть свой файл. Обратит внимание на способ отражение вашего текста.

Средства описания таблиц в HTML

По мере роста системы WWW стало ясно, что средств, которые заложены в HTML, недостаточно для качественного отображения различного типа документов. Недостатком HTML было отсутствие в его составе средств отображения таблиц. Для этой цели обычно использовался предформатированный текст (тег <PRE>), в котором таблица обрисовывалась символами ASCII. Но такая форма представления таблиц была недостаточно высокого качества и выделялась из общего стиля документа.

Тег <TABLE>

Для описания таблиц служит тег <TABLE>. Тег <TABLE>, как и многие другие, автоматически переводит строку до и после таблицы.

Тег <TR>

Тег <TR> (сокращение от Table Row – строка таблицы) создает строку таблицы. Если в таблице содержится два набора тегов <TR></TR>, в ней будут две строки. Весь текст, другие теги и атрибуты, которые вы хотите поместить в одну строку, должны быть помещены между тегами <TR></TR> .

Тег <TD>

Внутри строки таблицы обычно размещаются ячейки с данными. Каждая ячейка, содержащая текст или изображение, должна быть окружена тегами <TD></TD>. Число тегов <TD></TD> в строке определяет число ячеек. Строка с пятью парами тегов <TD></TD> будет состоять из пяти ячеек.

Тег <TH>

При задании заголовков для столбцов и строк таблицы используются тег заголовка <TH></TH> (Table Header, заголовок таблицы). Эти теги аналогичны <TD></TD>. Отличие состоит в том, что текст, заключенный между тегами <TH></TH>, автоматически записывается жирным шрифтом и по умолчанию располагается посередине ячейки. Центрирование можно отменить и выровнять текст по левому или правому краю. Если воспользоваться <TD></TD> с тегом и атрибутом <ALIGN=CENTER>, текст тоже будет выглядеть как заголовок. Однако следует иметь в виду, что не все Браузеры поддерживают жирный шрифт в таблицах, поэтому лучше задавать заголовки таблиц с помощью <TH>.

Тег <CAPTION>

<CAPTION> позволяет создавать заголовки таблицы. По умолчанию заголовки центрируются и размещаются либо над (<CAPTION ALIGN=TOP>), либо под таблицей (<CAPTION ALIGN=BOTTOM>). Заголовок может состоять из любого текста и изображений. Текст будет разбит на строки, соответствующие ширине таблицы. Иногда тег <CAPTION> используется для подписи под рисунком. Для этого достаточно описать таблицу без границ.

Заголовок может состоять из любого текста и изображений. Текст будет разбит на строки, соответствующие ширине таблицы. Иногда тег <CAPTION> используется для подписи под рисунком. Для этого достаточно описать таблицу без границ.

Атрибут NOWRAP

Обычно любой текст в таблице, не помещающийся в одну строку ячейки, переходит на следующую строку. Однако при использовании атрибута NOWARP с тегами <TH> или <TD> длина ячейки расширяется настолько, чтобы заключенный в ней текст поместился в одну строку.

Атрибут COLSPAN=

Теги <TD> и <TH> модифицируются с помощью атрибута COLSPAN= (Column Span, соединение столбцов). Если вы хотите сделать какую-нибудь ячейку шире, чем верхняя или нижняя, можно воспользоваться атрибутом COLSPAN=, чтобы растянуть ее над любым количеством обычных ячеек.

Атрибут ROWSPAN=

Атрибут ROWSPAN=, используемый в тегах <TD> и <TH>, аналогичен атрибуту COLSPAN=, только он задает число строк, на которые растягивается ячейка. Если вы указали в атрибуте ROWSPAN= число, большее единицы, то соответствующее количество строк должно находиться под растягиваемой ячейкой. Нельзя поместить ее внизу таблицы.

Гипертекстовые ссылки и якоря

Способность Web-страниц содержать ссылки на другие Web-страницы – одна из наиболее привлекательных особенностей. Для этого используются теги `<A>` и ``.

При создании гиперссылки открывающий тег должен содержать атрибут HREF.

HREF – определяет находящийся между начальным и конечным тегами текст или изображение как гипертекстовую ссылку (URL, или линк) на документ (и/или область документа), указанный в значении данного атрибута. Возможные значения:

`http://...` – создает ссылку на `www`-документ;

`ftp://...` – создает ссылку на `ftp`-сайт или расположенный на нем файл;

`mailto:...` – запускает почтовую программу-клиент с заполненным полем имени получателя. Если после адреса поставить знак вопроса, то можно указать дополнительные атрибуты, разделенные знаком "&"

`news:..` – создает ссылку на конференцию сервера новостей;

`telnet://...` – создает ссылку на `telnet`-сессию с удаленной машиной;

`wais://...` – создает ссылку на `WAIS`-сервер;

`gopher://...` – создает ссылку на `Gopher`-сервер;

Если тип соединения и адрес машины не указаны, в качестве отправной точки используется адрес текущего документа. Это позволяет использовать относительные ссылки. Например, линк

`Документация`

будет ссылаться на файл `title.html` в подкаталоге `docs` (относительно текущего).

NAME – помечает находящуюся между начальным и конечным тегами область документа как возможный объект для ссылки. В качестве значения нужно написать любое слово-указатель, уникальное для данного документа. Например: `Раздел1`. Теперь вы можете ссылаться на помеченную область простым указанием ее имени после имени документа. Например, линк `Раздел1` отправит вас в раздел `"part"` файла `document.html`, линк `` – в конец документа, линк `` – в раздел `"bottom"` текущего документа.

Задание 13. Создать таблицу по приведенному образцу, сохранить документ под именем `tabl_name.htm`. Сверху над таблицей разместить заголовок **Таблица 1**.

	*	

При отображении таблицы в Браузере должны удовлетворяться следующие условия: в центральной ячейке поместить символ «*», остальные ячейки должны быть пустыми.

Примечание. Для отображения пустых ячеек в них нужно поместить символьный примитив пробела ` `.

Задание 14. В этом же документе создать копию таблицы 1, ввести заголовок **Таблица 2** и модифицировать ее.

1. В центральной ячейке разместить рисунок.

2. «Раскрасить» все остальные ячейки в различные цвета.

3. Создать еще одну копию таблицы – **Таблица 3** и отредактировать теги таблицы так, чтобы она соответствовала приведенному образцу.

	рисунок	

4. В верхнюю строку поместить заголовок

Студент Фамилия И.О.

Примечания. Для объединения ячеек в тегах `<TD>` необходимо использовать параметры `colspan=` и `rowspan=`. Для отображения пустых ячеек в них нужно поместить символьный примитив пробела ` `.

5. Показать результат преподавателю.

Задание 15. Запустить текстовый редактор *Блокнот*.

1. Создать документ *second1.htm* в вашей папке на жестком диске.
2. В середине документа создать гипертекстовую ссылку на документ *tabl_name.htm*.
3. Просмотреть свой документ в окне обозревателя. Убедиться, что текст между тегами `<A>` и `` выделен как ссылка.
4. Щелкнуть мышью по документу и убедиться, что при этом загружается документ, на который указывает ссылка.
5. Закрыть окно *Обозревателя* и *Блокнот*.

Фреймы

Задание 1. Для начала мы должны представить себе общий вид страницы – где и какого размера будут фреймы. Затем можно подумать об их содержании. Ниже приводится код простой фреймовой структуры с использованием тега `<FRAMESET>`. Обратите внимание: страница с фреймовой структурой не содержит тега `<BODY>` (рис. 53).

Пример 1.

```
<HTML>
<HEAD>
<TITLE>Пример фреймов</TITLE>
</HEAD>
<FRAMESET COLS="25%, 75%">
<FRAME SRC="a.html">
<FRAME SRC="b.html" NAME="main">
</FRAMESET>
<NOFRAMES>
Вы видите эту страницу Браузером, не поддерживающим фреймы.
</NOFRAMES>
</HTML>
```

Вот и весь код, необходимый для задания фреймовой структуры. Обратите внимание на тег `<NOFRAMES>`. Через несколько минут мы к нему вернемся. В результате мы получили экран, разделенный на два окна. Левое окно занимает 25 процентов экрана и содержит страницу с названием *a.html*. Окно справа займет 75 процентов и вначале покажет файл *b.html*. Пока у нас их нет, так что вы увидите страницу с двумя пустыми фреймами. Прежде чем она появится, нам придется пару раз щелкнуть мышью в ответ на сообщения об ошибках, потому что Браузер будет пытаться найти несуществующие страницы. Заметьте, что правую страницу мы назвали `<main >` (`<главная>`) с помощью строки:

```
<FRAME SRC="b.html" NAME="main">
```

Это означает, что фрейм под именем *main* будет содержать страницу *b.html*. Заметим, что поскольку мы не собираемся показывать в левом фрейме никаких страниц, кроме *menu.html*, нам не нужно его называть.

Задание 2. Теперь давайте загрузим фреймы с содержимым. Зададим страницу *menu.html* в левом фрейме, где мы собираемся щелкать мышью, переключаясь между двумя страницами в правом фрейме. Файл *menu.html* – это обычная HTML-страница, построенная как оглавление. На самом деле мы можем взять готовую страницу с оглавлением и использовать ее. Имейте в виду, что этот фрейм узкий и высокий, так что страница, которая будет в него загружаться, должна быть соответствующим образом спроектирована. Теперь мы должны определить, где будут появляться другие страницы при щелчке мышью на ссылке. Поскольку мы хотим, чтобы они отображались в правом фрейме, добавим атрибут `TARGET="main"` в тег ссылки. Это означает, что когда пользователь щелкает на ссылке, вызываемая страница появляется в фрейме *main*. Мы отображаем все страницы в фрейме *main*, поэтому давайте добавим атрибут `TARGET="main"` во все теги ссылок в оглавлении. Если мы не определим атрибут `TARGET`, то страница появится там, где мы щелкнули мышью, – в левом фрейме, что нас не устраивает, хотя в какой-нибудь другой ситуации подобное поведение было бы очень кстати. Например, вы можете добавить ссылку `<Другие пункты оглавления>`, которая будет просто выводить следующие ссылки. Имеет смысл сделать оглавление подлиннее, чтобы читатели видели как можно больше ссылок. Но сейчас давайте ограничимся простым примером. Ниже приведен код для левого фрейма *menu.html* (рис. 54).

Пример 2.

```
<HTML>
```



```

<HEAD>
<TITLE> Меню</TITLE>
</HEAD>
<BODY>
<H3 ALIGN=CENTER>Компания.</H3>
<HR>
<UL><font color="#400040" size="4">
<LI><a href="html-pr2-3.html" TARGET="main">Главная</a>
<LI><a href="news.shtml" TARGET="main">Новости</a>
<LI><a href="products.shtml" TARGET="main">Товары</a>
<LI><a href="servis.shtml" TARGET="main">Услуги</a>
<LI><a href="contact.shtml" TARGET="main">Контакты</a>
<LI><a href="search.shtml" TARGET="main">Поиск</a>
</UL>
</font>
</BODY>
</HTML>

```

Заметим, что здесь ничего не говорится о фреймах. О них все сказано в фреймовой структуре. Единственное, о чем нужно позаботиться на каждой HTML-странице, появляющейся в фрейме, так это о том, где поместить ссылки, активирующие те или иные действия. В нашем примере, в левом фрейме, где находится оглавление, будет располагаться только одна страница. Мы хотим, чтобы при щелчке мышкой в левой странице ссылочный документ появлялся бы в правом фрейме main.

Задание 3. Правый фрейм main будет содержать сами HTML-страницы. Ваша задача так их спроектировать, чтобы они хорошо смотрелись в меньшем чем обычно окне, потому что часть экрана будет занята левым кадром оглавления. Но больше эти страницы ничем не примечательны. Ниже приводится код для страницы, упомянутой первой в оглавлении (html-pr2-3.html).

Пример 3.

```

<HTML>
<HEAD>
<TITLE> Главная страница</TITLE>
</HEAD>
<BODY>
<H1 ALIGN=CENTER>Компания.</H1>
<HR>
<H3 ALIGN=LEFT><I>Открытое акционерное общество «Компания», основанная в 1996 г.,
является одним из ведущих поставщиков бытовой электроники в России.</I></H3>
<HR>
<H3>Основными направлениями деятельности
«Компании» являются:
<OL>
<LI>реализация бытовой электроники ведущих фирм мира через сеть магазинов;
<LI>создание сервисных центров по обслуживанию бытовой электроники.</H3>
</OL>
<HR>
<H3><U><FONT COLOR=RED> Компания ищет агентов по сбыту бытовой электрони-
ки.</FONT></U></H3>
<H5 ALIGN=CENTER>Контактная информация</H5>
<UL>
<LI>Telephone (123) 12-34-56
<LI>FAX (123) 12-34-56
<LI>Почтовый адрес 123456 г. Город, ул Лесная, 106
<UL TIPE=CIRCLE>Электронная почта<BR>
<LI>Общая информация: abc@abc.su
<LI>Продажи: abc@abc.abc.su

```

```
</UL>
</UL>
<BR><BLINK>Copyright</BLINK> © 1997 «Компания»
</BODY>
</HTML>
```

Обратите внимание: на приведенных страницах нигде тайно не запрограммирован фрейм. Это значит, что можно без особых проблем конвертировать все документы вашего Web-сервера для использования в фреймовой структуре.

Задание 4. У многих ваших посетителей еще остались Браузеры, не умеющие обращаться с фреймами. По этим причинам разумно предоставить доступ к версии ваших основных страниц без фреймов. Если читатель с устаревшим Браузером окажется на вашей странице с фреймовой структурой, все, что находится на ней между тегами <NOFRAMES> и </NOFRAMES>, будет выглядеть отлично – Браузер просто проигнорирует фреймы. Вот почему обязательно нужно использовать теги <BODY> и </BODY>. Возможно, вам придется иначе организовать экран без фреймов.

Можно поместить на страницу с фреймами кнопку No Frames (Без фреймов). Ее назначение очевидно. Такой вариант достаточно разумен и легко осуществим.

На тот случай, если вы неуверенно чувствуете себя при использовании <NOFRAMES >, ниже приведен пример страницы с фреймовой структурой с добавленным в конце разделом <NOFRAMES> (рис. 55).

Пример 4.

```
<HTML>
<HEAD>
<TITLE> Пример фрейма</TITLE>
</HEAD>
<FRAMESET COLS="25%, 75%"
<FRAME SRC="html-pr5-2.html">
<FRAME SRC="html-pr2-3.html" NAME="main">
</FRAMESET>
<NOFRAMES>
```

Вы видите эту страницу Браузером, не поддерживающим фреймы. Браузер, поддерживающий фреймы, не видит этот текст.

```
</NOFRAMES>
</HTML>
```

Имейте в виду, что поддерживающий фреймы Браузер проигнорирует все, что находится между тегами <NOFRAMES> и </NOFRAMES>. И наоборот, не поддерживающий фреймы Браузер проигнорирует все, что находится между тегами <FRAMESET> и </FRAMESET>. Код без фреймов можно поместить и в начало, и в конец страницы.

Формы HTML

Форма HTML представляет собой документ, созданный с использованием элементов HTML. Назначением формы является сбор информации от пользователей. После того как пользователь заполнит форму и запускает процесс ее обработки, информация из нее попадает в программу, работающую на сервере. Другая программа под названием Common Gateway Interface (CGI) обрабатывает ее. Таким образом, пользователь может интерактивно взаимодействовать с сервером Web через Internet.

Элементы формы (элемент <FORM>)

Элемент <FORM> используется для обозначения документа как *формы*. Данный элемент определяет границы использования других элементов, размещаемых в форме. <FORM> определяется последовательностью элементов <INPUT>, размещенных внутри пары <FORM> и </FORM>. Элемент формы использует как *метод*, так и *действие* для описания обработки формой данных, вводимых пользователем:

метод (GET или POST) – определяет, как программист должен обрабатывать входные данные из формы;

действие – указывает на URI программы, ответственной за обработку данных.

Этот элемент поддерживает атрибуты [ACTION](#), [ENCTYPE](#), [METHOD](#).

Сбор данных при помощи форм (элемент <INPUT>)

Элемент <INPUT> используется для определения области внутри формы, где собираются данные. Данный элемент представляет собой поле для ввода информации пользователем (обычно одна строка текста). В таком случае требуется наличие атрибута **NAME** для определения наименования переменной поля. Можно использовать следующие атрибуты:

MAXLENGTH – ограничивает число вводимых символов (по умолчанию ограничений нет).

SIZE – размер видимой на экране области, занимаемой текущим полем. Если **MAXLENGTH** > **SIZE**, Браузер будет прокручивать данные в окне.

VALUE – определяет начальное значение поля ввода.

А также атрибуты: **ALIGN**, **CHECKED**, **NAME**, **SRC**, **TYPE**.

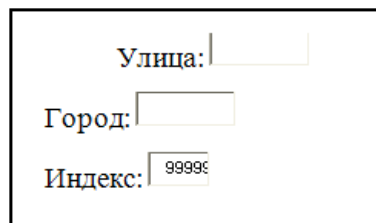
Пример 1. Простая форма для ввода.

<P> Улица: <INPUT NAME= "street">

Город: <INPUT NAME= "city" SIZE= "20" MAXLENGTH= "20">

Индекс: <INPUT NAME= "zip" SIZE= "5" MAXLENGTH= "5" VALUE= "99999">

Окно Браузера примет вид.



Атрибут CHECKBOX

При создании форм часто бывает необходимо получить ответ пользователя на вопрос типа (Да/Нет) или (Правда/Ложь). Например, нужно выбрать из списка несколько значений. Для создания независимых кнопок в формах можно использовать атрибут **CHECKBOX**. В зависимости от содержания можно отметить несколько флагов. Вместе с атрибутом **CHECKBOX** должны использоваться следующие атрибуты:

CHECKED – инициализировать данный флаг как отмеченный;

NAME – наименование поля ввода формы;

VALUE – значение поля ввода.

Пример 2. Выбор блюд.

<P> Выберите ваше любимое блюдо:

<FORM>

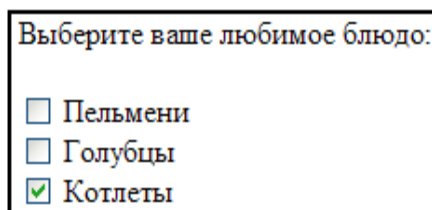
<INPUT TYPE="checkbox" NAME="food" VALUE="Пельмени"> Пельмени

<INPUT TYPE="checkbox" NAME="food" VALUE="Голубцы"> Голубцы

<INPUT TYPE="checkbox" NAME="food" VALUE="Котлеты" CHECKED> Котлеты

</FORM>

Окно Браузера примет вид:



Атрибут IMAGE

В зависимости от содержимого формы может случиться так, что пользователю потребуется щелкнуть мышью на изображении, чтобы завершить работу с формой. Для организации этого используется атрибут **IMAGE**. После щелчка пользователя по изображению Браузер сохраняет координаты соответствующей точки экрана и принимает всю форму. Вместе с атрибутом **IMAGE** должны использоваться следующие атрибуты:

ALIGN – необязательный атрибут, указывает на положение изображения на экране (аналогично элементу **IMAGE**).

NAME – наименование поля ввода формы.

SRC – URI файла-источника изображения.

Пример 3.

<P> Выберите точку на изображении:
<INPUT TYPE= "image" NAME="point" SRC="globe.gif">

Атрибут PASSWORD

Данный атрибут используется для организации ввода пароля без вывода на экран составляющих его символов (вместо символов выводятся звездочки).

Пример 4.

<FORM>

<P> Введите имя:

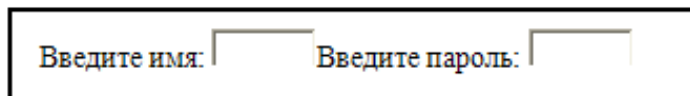
<INPUT NAME="login">

<P> Введите пароль:

<INPUT TYPE="password" NAME="p_word">

</FORM>

В окне Браузера это будет выглядеть так:



Атрибут RADIO

Данный атрибут используется для организации выбора одного-единственного варианта их нескольких возможных. Вместе с атрибутом RADIO должны использоваться следующие атрибуты:

CHECKED – инициализировать данный флаг как отмеченный.

NAME – наименование поля ввода формы.

VALUE – значение поля ввода.

Пример 5.

<P> Укажите сорт вашего любимого сока:

<FORM>

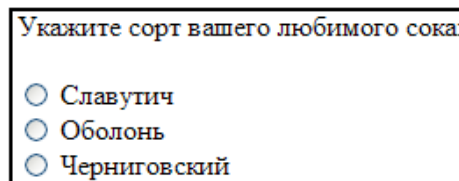
<INPUT TYPE="radio" NAME="beer" VALUE="slav"> Славутич

<INPUT TYPE="radio" NAME="beer" VALUE="obol"> Оболонь

<INPUT TYPE="radio" NAME="beer" VALUE="chern"> Черниговский

</FORM>

Окно Браузера примет вид.



Атрибут RESET

Данный атрибут используется для создания кнопки *Reset*. При нажатии на эту кнопку форма восстанавливает первоначальные значения полей всех элементов <INPUT>, в которых присутствует атрибут RESET. Вместе с атрибутом RESET может использоваться атрибут VALUE – значение поля ввода по умолчанию.

Пример 6.

<P>

<FORM>

Код: <INPUT NAME="cod">

Телефон: <INPUT NAME="phone" SIZE="6" MAXLENGTH="6" >

<P>

<INPUT TYPE=RESET>

</FORM>

Окно Браузера примет вид.

Атрибут *SELECT*

Для организации списков с прокруткой и выпадающим меню можно использовать атрибут `<SELECT>`. Для определения списка пунктов используются элементы `<OPTION>` внутри `<SELECT>`. Вместе с атрибутом `SELECT` можно использовать следующие атрибуты:

[NAME](#) – наименование объекта.

[MULTIPLE](#) – позволяет выбрать более чем одно наименование.

[SIZE](#) – определяет число пунктов, видимых для пользователя. Если `SIZE=1`, то Браузер выводит список на экран в виде выпадающего меню (видно одно наименование). Если `SIZE>1`, то Браузер представляет на экране обычный список (число – количество видимых наименований)

С элементом `OPTION` можно использовать следующие атрибуты:

[SELECTED](#) – для первоначального выбора значения элемента по умолчанию.

[VALUE](#) – значение, возвращаемое формой после выбора пользователем данного пункта. По умолчанию значение поля равно элементу `<OPTION>`.

Когда пользователь заполняет форму, атрибут `NAME` – элемента `<SELECT>` состыковывается с атрибутом `VALUE`-элемента для формирования наименования, выбранного пользователем.

Пример 7.

```
<FORM>
<SELECT NAME="Фрукты" SIZE=3 >
<OPTION> Сливы
<OPTION> Груши
<OPTION value="Lemon_and_orange">Лимоны и апельсины
<OPTION selected> Яблоки
</SELECT>
</FORM>
```

Данный атрибут используется при окончании ввода пользователем данных. Браузер, в свою очередь выводит данный элемент как кнопку, на которой пользователь может щелкнуть, чтобы завершить процесс редактирования. Вместе с атрибутом `SUBMIT` можно использовать следующие атрибуты:

[NAME](#) – наименование кнопки `SUBMIT`.

[VALUE](#) – значение переменной поля в вашей форме.

Пример 8.

```
<P>
<FORM>
Код: <INPUT NAME="cod"> <BR>
Телефон: <INPUT NAME="phone" SIZE="6" MAXLENGTH="6" ><BR>
<P>
<INPUT TYPE=RESET> <INPUT TYPE=SUBMIT>
</FORM>
```

Окно Браузера примет вид.

Атрибут *TEXTAREA*

Данный атрибут используется для ввода большого количества текстовой информации (несколько строк). Вместе с атрибутом `TEXTAREA` можно использовать следующие атрибуты:

`NAME` – наименование поля.

COLS – число колонок (символов) в текстовой области.

ROWS – число видимых строк в текстовой области.

Пример 9.

<FORM>

<TEXTAREA NAME="adress" COLS=64 ROWS=6>

Украина

Мариуполь

пр.Нахимова, 99

</TEXTAREA>

</FORM>

Задание 1. Описать анкету пользователя.

Пример содержит большинство описанных выше элементов.

<P> <H3 ALIGN=CENTER> Анкета пользователя </H3>

<P> Ответьте, пожалуйста, на вопросы:

<FORM METHOD="POST" ACTION="http://ibutov.narod.ru ok">

<P> Ваше имя: <INPUT NAME="name" SIZE="48">

<P> Мужчина <INPUT NAME="gender" TYPE=RADIO VALUE="male">

<P> Женщина <INPUT NAME="gender" TYPE=RADIO VALUE="female">

<P> Детей <INPUT NAME="family" TYPE=TEXT>

<P> Страна <INPUT NAME="family" TYPE=TEXT>

 Украина <INPUT NAME="country" TYPE="checkbox" VALUE="Ukraine">

 Россия <INPUT NAME="country" TYPE="checkbox" VALUE="Russian">

 Другая <TEXTAREA NAME="other" COLS=48 ROWS=3></TEXTAREA>

<P> Электронный адрес: <INPUT NAME="email" SIZE="42">

<P> Для подтверждения ввода нажмите на кнопку SUBMIT.

<P> Для сброса введенных данных нажмите на кнопку RESET.

<P>

<INPUT TYPE=SUBMIT><INPUT TYPE=RESET>

</FORM>

Когда пользователь щелкает на кнопке SUBMIT, Браузер передает сообщение (метод POST) по адресу <http://ibutov.narod.ru/ok>, а тело сообщения будет выглядеть примерно так:

name=Iliya+Butov&gender=male&family=2&country=Russian&other=None&email=k116@yandex.ru

Далее специальная программа размещенная на сервере <http://ibutov.narod.ru/ok> начнет обработку ответов пользователя.

3. Методические указания по самостоятельной работе студентов

Для закрепления полученных теоретических и практических знаний студентам в течение всего учебного года предлагаются индивидуальные задания для самостоятельной работы. Консультирование по выполнению индивидуальных заданий проводится как непосредственно в компьютерных классах (во время консультаций), так и через электронный обмен сообщениями, посредством Интернет. Защита индивидуальных заданий по темам может проводиться в виде Круглого стола, когда каждый студент выступает с презентацией выполненной работы, а преподаватель вместе с остальными студентами оценивает работу. Задания по темам также могут быть выданы студентам в качестве домашних заданий в виде электронных файлов. Контроль выполненных заданий осуществляется либо непосредственно на занятиях, либо на консультациях.

В рабочей программе п.5 представлены виды самостоятельной работы по каждой теме дисциплины и трудоемкость в часах.

Индивидуальная самостоятельная работа включает две части: текстовую (реферат), подготовленную с использованием текстового процессора MS Word, и электронную презентацию, выполненную средствами MS Power Point. Текстовая часть должна быть представлена в виде твердой копии на бумажном носителе и в электронном виде (на дискете или CD, в формате RTF).

Защита индивидуальных заданий будет происходить на практическом занятии в аудитории. Каждое выступление должно сопровождаться презентацией.

Примерная тематика домашних заданий

Темы рефератов

1. Автоматизированные системы обработки информации.
2. Базы данных в Internet.
3. Данные и информация.
4. Информационные технологии и их роль в обществе.
5. Информационное обеспечение как необходимая услуга для функционирования экономики в современных условиях.
6. Локальные и глобальные сети. Электронная почта.
7. Система автоматизированной обработки статистической информации.
8. Экспертные системы. Классификация экспертных систем. Разработка простейшей экспертной системы.
9. Электронная почта.
10. Технологии создания сетей ЭВМ.
11. Электронная коммерция.
12. Экспертные системы.
13. Технология World Wide Web.
14. Теория фреймов.
15. Современные текстовые редакторы.
16. Сетевые операционные системы.
17. Интеллектуальные и экспертные системы.

Требования к оформлению реферата

1. Поля страницы: слева – 3, сверху и снизу – 2, справа – 1.
2. Размер шрифта – 14.
3. Межстрочный интервал – 1,5.
4. Шрифт – Times New Roman.
5. Нумерация страниц – внизу, по центру.
6. В конце – список использованной литературы.
7. Объем реферата – не менее 15 листов.

IV. КОНТРОЛЬ ЗНАНИЙ

В течении семестра знания студентов оцениваются с использованием рейтинговой системы, которая складывается из оценки за работу в семестре – максимально 60 баллов и экзаменационной (зачетной) оценки – максимально 40 баллов. Максимально возможное количество равно 100. Баллы по разным модулям приведены в рабочей программе п.12.

Минимальное количество баллов в семестре, необходимое для получения студентом допуска на зачет, равно 40 баллов.

Минимальное количество баллов за выполнение экзаменационной (зачетной) работы, необходимое для получения оценки: «удовлетворительно» - 15 баллов, «хорошо» - 20 баллов, «отлично» - 30 баллов.

Соответствие итогового рейтинга студента и традиционных оценок устанавливается по следующей шкале: «неудовлетворительно» - 0-50 баллов, «удовлетворительно» - 51-75 баллов, «хорошо» - 76-90 баллов, «отлично» - 91-100 баллов.

В качестве основных средств текущего контроля используется тестирование. В качестве дополнительной формы текущего контроля предлагаются аудиторные и внеаудиторные письменные задания (контрольные работы).

1. Текущий контроль знаний

В рабочей программе п.7 представлен вариант теста для текущего контроля знаний.

2. Итоговый контроль знаний.

В рабочей программе п.7 представлены вопросы к зачету и вариант тестовых вопросов к зачету.

V. ИНТЕРАКТИВНЫЕ ТЕХНОЛОГИИ И ИННОВАЦИОННЫЕ МЕТОДЫ, ИСПОЛЬЗУЕМЫЕ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ.

Образовательный процесс по дисциплине строится на основе комбинации следующих образовательных технологий.

Интегральную модель образовательного процесса по дисциплине формируют технологии методологического уровня: модульно-рейтинговое обучение, технология поэтапного формирования умственных действий, технология развивающего обучения, элементы технологии развития критического мышления.

Рекомендуется использование информационных технологий при организации коммуникации со студентами для представления информации, выдачи рекомендаций и консультирования по оперативным вопросам (электронная почта), использование мультимедиа-средств при проведении лекционных и лабораторных занятий.

Игровые имитационные методы:

Мозговой штурм – наиболее свободная форма дискуссии, позволяющей быстро включить в работу всех членов учебной группы. Используется там, где требуется генерация разнообразных идей, их отбор и критическая оценка. Этапы продуцирования идей и их анализа намеренно разделены: во время выдвижения идей запрещается их критика.. Тема: «Теория вероятностей» (2 часа).

Круглый стол – это метод активного обучения, одна из организационных форм познавательной деятельности учащихся, позволяющая закрепить полученные ранее знания, восполнить недостающую информацию, сформировать умения решать проблемы, укрепить позиции, научить культуре ведения дискуссии.

Дискуссия – это всестороннее обсуждение спорного вопроса в публичном собрании, в частной беседе, споре.

Деловая игра – форма воссоздания предметного и социального содержания профессиональной деятельности, моделирования систем отношений, разнообразных условий профессиональной деятельности, характерных для данного вида практики.

Метод анализа конкретной ситуации (ситуационный анализ, анализ конкретных ситуаций, case-study) – это педагогическая технология, основанная на моделировании ситуации или использования реальной ситуации в целях анализа данного случая, выявления проблем, поиска альтернативных решений и принятия оптимального решения проблем.

Мастер-класс – это главное средство передачи концептуальной новой идеи своей (авторской) педагогической системы. Преподаватель как профессионал на протяжении ряда лет вырабатывает индивидуальную (авторскую) методическую систему, включающую целеполагание, проектирование, использование последовательности ряда известных дидактических и воспитательных методик, занятий, мероприятий, собственные «ноу-хау», учитывает реальные условия работы с различными категориями учащихся и т.п.