

МИНИСТЕРСТВО ОБРАЗОВАНИЯ УКРАИНЫ
ОДЕССКИЙ НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра «Информационные системы в менеджменте»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению курсовой работы по дисциплине
«СЕТЕВЫЕ ТЕХНОЛОГИИ»
для студентов специальности 7.050102
«Экономическая кибернетика»
дневной формы обучения

Утверждено
на заседании кафедры ИСМ
протокол № от «__» ____2010г.

Одесса - 2010г.

Методические указания к выполнению курсовой работы по дисциплине
«СЕТЕВЫЕ ТЕХНОЛОГИИ» для студентов специальности 7.050102
«Экономическая кибернетика» дневной формы обучения / Сост. Е.А.
Арсирый. - Одесса: ОНПУ, 2010. - .44с

Составители:

Е.А.Арсирый доц., к.т.н

СОДЕРЖАНИЕ

	Стр.
ВВЕДЕНИЕ.....	4
1 ОБЩАЯ ХАРАКТЕРИСТИКА КУРСОВОЙ РАБОТЫ	6
2 ОСНОВЫ ЯЗЫКА ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ ДОКУМЕНТОВ (HTML).....	8
2.1 Разработка страниц WWW с помощью HTML.....	8
2.1.1 Принципы построения и интерпретации HTML.....	10
2.1.2 Элементы управления разметкой.....	13
2.1.3 Гипертекстовые ссылки.....	16
2.1.4 Средства описания графической информации.....	16
2.1.5 Средства описания таблиц.....	18
2.1.6 Выравнивание полей данных в формах.....	21
2.2 Некоторые дополнительные возможности языка HTML.....	23
2.2.1 Расширенное использование графики на страницах Web.....	23
2.2.2 Фоновый звук и видеоклипы.....	25
2.2.3 Текстовые эффекты.....	26
2.2.4 Фреймы.....	27
2.2.5 Карты изображения.....	29
3 СТРУКТУРА КУРСОВОЙ РАБОТЫ.....	33
3.1 Содержание раздела «Постановка задачи на проектирование ДИС и КС».....	34
3.2 Содержание раздела «Проектирование ДИС».....	36
3.3 Содержание раздела «Технический проект и расчет стоимости КС».....	37
4 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ.....	39
5 ТЕМАТИКА КУРСОВОЙ РАБОТЫ.....	40
6 ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.....	41
7 СПИСОК РЕКОМЕНДОВАННЫХ ИСТОЧНИКОВ.....	42
ПРИЛОЖЕНИЯ.....	44

ВВЕДЕНИЕ

Курсовая работа по дисциплине «СЕТЕВЫЕ ТЕХНОЛОГИИ» проводится согласно учебному плану в течении 7-го семестра. Тематика курсовых работ полностью соответствует образовательной квалификации по специальности 7.050102 «Экономическая кибернетика».

Выполнение курсовой работы позволяет студенту приобрести навыки по практическому использованию сетевых технологий (СТ) в работе какого-либо предприятия, а именно: по проектированию дистанционной информационной системы (ДИС), по выбору конфигурации и разработке компьютерной сети (КС) состоящей из сегментов различных типов сетевого и коммуникационного оборудования, а также более детального изучения основных факторов, которые следует учитывать при проектировании ДИС и КС:

- Техничко-экономический анализ деятельности выбранной бизнес-структуры;
- Возможности современных информационных технологий по представлению информации в Internet;
- Принципы работы пользователя с основными службами (сервисами) Internet такими как электронная почта, веб-сервер, ftp-сервер;
- Логику функционирования и ее основных информационных сервисов;
- Основы работы с растровой графикой;
- Основы Web-дизайна;
- Создание Web-страниц с помощью языка HTML;
- Основные информационные и технические характеристики Web-разработки;
- Перечень социальных, психологических, физиологических характеристик посетителя ДИС;
- Возможности по размещению ДИС в Internet (хостинг);
- Регистрация доменного имени ДИС;
- Требуемый размер КС (в ближайшем будущем и по прогнозу на перспективу);
- Требуемая структура КС, иерархия и основные части сети (по подразделениям предприятия, по комнатам, этажам, зданиям)
- Основные направления и интенсивность информационных потоков (в ближайшем будущем и по прогнозу на перспективу);
- Технические характеристики оборудования (компьютеров, адаптеров, кабелей, репитеров, концентраторов, коммутаторов) и его стоимость;
- Возможности прокладки кабельной системы в помещениях и между ними, а также меры обеспечения целостности кабеля;
- Обеспечение обслуживания сети и контроля за ее безотказностью и безопасностью, требования к сетевому администрированию;

- Требования к программным средствам по допустимому размеру сети, скорости, гибкости, разграничению прав доступа, стоимости, возможностям контроля за обменом информацией
- Необходимостью подключения к глобальным сетям или к другим локальным сетям;
- Требования к квалификации сотрудников, которые будут работать с спроектированными ДИС и КС.

1. ОБЩАЯ ХАРАКТЕРИСТИКА КУРСОВОЙ РАБОТЫ

Курсовая работа (КР) по дисциплине «СЕТЕВЫЕ ТЕХНОЛОГИИ» является одним из этапов в освоении будущими специалистами по специальности «Экономическая кибернетика» новейших информационных сетевых технологий. При выполнении курсовой работы предусматривается закрепление получаемых при изучении теоретического курса знаний по основам построения и функционирования технологий компьютерных сетей - локальных: Ethernet, Token Ring, FDDI и высокоскоростных Fast Ethernet 100VG-AnyLAN, Gigabit Ethernet, - глобальных: SNA, Frame Relay, ATM, TCP / IP, а также методов приема / передачи данных на физическом и канальном уровне, основных характеристик каналов связи, правил адресации и структур компьютерных сетей, методов создания, размещения, поиска, получения и пересылки информации в компьютерных сетях и др. Выполнение курсовой работы позволит также приобрести навыки по практическому использованию сетевых технологий в работе любого предприятия, а именно: по выбору конфигурации и проектированию КС, состоящей из сегментов различных типов оборудования и разработке ДИС (Web-сайта).

КР состоит из введения, трех разделов основного текста, вывода, списка использованных источников и приложений (см. подробнее разделы Структура КР и Требования к оформлению КР)

В первом разделе КР приводится постановка задачи на проектирование ДИС и КС для выбранной бизнес структуры. В данном разделе проводится анализ организационной структуры предприятия/подразделения и задач решаемых в рамках каждого структурного подразделения. Определяется роль информационных технологий в решении выделенных функций и задач, и провести анализ существующей технологии обработки информации в рамках выделенных функций и задач. Особенное внимание уделяется следующим направлениям: хранение данных на сервере, электронная почта, просмотр интернет страниц, передача файлов большого объема, использование офисных приложений работа с графическими файлами, IP телефония, базы данных, системы автоматизированного проектирования, издательские системы, проигрывание аудио и видео, приём радио и телепередач по сети, участие в аудио и видео конференциях (аналог селекторных совещаний). Проводится анализ преимуществ, которые может получить предприятие при размещении информации о себе в сети Интернет. Приводится обзор веб-сайтов конкурентов. Здесь же разрабатывается макет ДИС, включающий полосу логотипа, полосу навигации холст веб-сайта.

Во втором разделе КР приводится структурная схема веб-интерфейса пользователя ДИС, описание назначения и экранные формы всех составных частей разработанной системы со ссылками на их источник (в виде HTML), который помещается в приложении. Даются рекомендации о размещении

ДИС на хостинговой площадке и регистрации доменного имени. Производится расчет трудоемкости проектирования ДИС.

В третьем разделе КР решаются задачи определения архитектуры КС выбора типов компонент КС и расчета их количества, производится оценку показателей эффективности КС и стоимости ее проектирования. При этом должны учитываться правила соединения компонентов КС, основанные на стандартизации сетей, и их ограничения, специфицированные изготовителями компонент КС.

Таким образом, главной целью курсового проектирования является сформировать у студентов максимально полную и целостную картину функционирования сетевых технологий и выработать навыки по практическому использованию их возможностей для принятия эффективных управленческие решений.

2.ОСНОВЫ ЯЗЫКА ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ ДОКУМЕНТОВ

Тим Бернерс Ли (Tim Berners Lee) из Европейского центра ядерных исследований (CERN) в марте 1989 года предложил руководству концепцию новой распределенной информационной системы - World Wide Web. Свои соображения он изложил в проекте, который назвал «Гипертекст для ЦЕРН» («World Wide Web: Proposal for HyperText Project»). В 1990г. эти предложения были приняты и проект стартовал. Так началось развитие одной из наиболее популярных современных информационных технологий Internet.

Идея Тима Бернерса Ли заключалась в том, чтобы применить гипертекстовую модель к информационным ресурсам, распределенным в сети, и сделать это максимально простым способом. Он предложил три компоненты системы из четырех существующих ныне:

- язык гипертекстовой разметки документов HTML (HyperText Markup Language);
- универсальный способ адресации ресурсов в сети URL (Universal Resource Locator);
- протокол обмена гипертекстовой информацией HTTP (HyperText Transfer Protocol);

Позже группа в Национальном Центре Суперкомпьютерных Приложений (NCSA- National Center for Supercomputer Applications) добавила к этим трем компонентам четвертую:

- универсальный интерфейс шлюзов CGI (Common Gateway Interface).

Идея *гипертекстовой информационной системы* состоит в том, что пользователь имеет возможность просматривать документы (страницы текста) в произвольном порядке, т.е. в том, в котором ему это больше нравится, а не последовательном как это принято при чтении книг. Поэтому Нельсон еще в 1965 г. определил *гипертекст* как *нелинейный текст*. Достигается это путем создания специального механизма связи различных страниц текста при помощи гипертекстовых ссылок, т.е. у обычного текста есть ссылки типа «следующий - предыдущий», а у гипертекста можно построить еще сколько угодно других ссылок.

2.1 Разработка страниц WWW с помощью HTML.

Как вы уже знаете язык гипертекстовой разметки HTML был предложен Тимом Бернерсом Ли в 1989 г. в качестве одного из компонентов технологии разработки распределенной гипертекстовой системы World Wide Web.

Разработчики HTML пытались решить две задачи:

- дать дизайнерам гипертекстовых баз данных простое средство создания документов;

- сделать это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

В качестве элемента гипертекстовой базы данных был выбран обычный текстовый файл, который создается и хранится средствами файловой системы операционной среды компьютера. Такой выбор был сделан под влиянием следующих факторов:

- такой файл можно было создавать в любом текстовом редакторе на любой аппаратной платформе в среде любой операционной системы;
- к моменту разработки HTML существовал американский стандарт для разработки сетевых информационных систем Z39.50, в котором в качестве единицы хранения указывался простой текстовый файл в кодировке LATIN1, что соответствует US ASCII.

Таким образом, гипертекстовая база данных в концепции WWW - это набор текстовых файлов написанных на языке HTML, который определяет форму представления информации (разметка) и структуру связей этих файлов (гипертекстовые ссылки).

Такой подход предполагает наличие еще одной компоненты технологии - интерпретатора языка. В WWW функции интерпретатора разделены между сервером гипертекстовой базы данных и интерфейсом пользователя.

Сервер, кроме доступа к документам и обработки гипертекстовых ссылок осуществляет также предпроцессорную обработку документов, в то время как интерфейс пользователя осуществляет интерпретацию конструкций языка, связанных с представлением информации.

Первая версия HTML 0 - была разработана в CERN в 1990 г. и сейчас практически не используется. Следующая версия HTML 1.0 уже включала возможность встроенных изображений и текстовые стили (выделение текста). Стандарт HTML 2.0 уже являлся стандартом, на основе которого было построено большинство страниц WWW, был одобрен как стандарт 22 сентября 1995 года комитетом RFC 1866. В январе 1997 г. консорциум по World Wide Web (W³C) выпустил новую версию HTML 3.2, включающую тэги для создания таблиц, апплеты Java, карты-изображения располагающиеся на машине клиента и новые типы форматирования текста, далее 18 декабря 1997 года появилась версия 4.0. HTML версии 4.0 содержит много элементов, специфичных для отдельных браузеров, но в то же время произошла некоторая «очистка» стандарта. Многие элементы были отмечены как устаревшие и nereкомендованные (англ. deprecated). В частности, элемент font, используемый для изменения свойств шрифта, был помечен как устаревший (вместо него рекомендуется использовать таблицы стилей CSS). Следует отметить так называемый ISO HTML, основанный на HTML 4.01 Strict увидел свет 15 мая 2000 года.

Начиная с 2004 года сообщество WHATWG (Web Hypertext Application Technology Working Group) ведётся разработка HTML версии 5. Сейчас

Консорциум всемирной паутины разрабатывает HTML версии 5. Черновой вариант спецификации языка появился в Интернете 20 ноября 2007 года. Параллельно ведётся работа по дальнейшему развитию HTML под названием XHTML (англ. Extensible Hypertext Markup Language — «расширяемый язык разметки гипертекста»). Пока XHTML по своим возможностям сопоставим с HTML, однако предъявляет более строгие требования к синтаксису. Как и HTML, XHTML является подмножеством языка SGML, однако XHTML, в отличие от предшественника, основан на XML.

2.1.1. Принципы построения и интерпретации HTML

Тэгловая модель - описывает документ как совокупность элементов несущих информационную нагрузку, каждый из которых окружен тэгами (тагами) (tags). Тэги HTML - это последовательность символов начинающаяся знаком «<» и заканчивающаяся знаком «>». Тэги по своему значению близки к понятию скобок {} в С и «begin/end» в Паскале. Тэги определяют область действия правил интерпретации текстовых элементов. Следует заметить, что большинство тэгов специфицируют логические, а не физические стили. Например, тэги заголовков указывают обычно больший размер букв, а не какой именно размер нужно использовать. Это свойство HTML обеспечивает ему *независимость от платформы*. Однако существенным недостатком такого подхода является невозможность точно контролировать внешний вид документа HTML.

Общая схема построения элемента текста в формате HTML может быть записана в следующем виде:

элемент текста:=

<имя элемента список атрибутов> - тэг начала элемента

содержание элемента

</имя элемента> - тэг конца элемента

Посмотреть коды языка HTML можно следующим образом в браузере набрать (Меню → Вид → Источник). Следует еще раз обратить внимание на то, что код HTML в данном случае всего лишь текст. Интерпретировать тэги, и представлять информацию с соответствии с ними - задача браузера.

Браузеры Web обязательно следуют трем основным правилам при синтаксическом анализе HTML:

- пробелы и другие «невидимые символы» игнорируются;
- тэги форматирования могут быть написаны строчными и/или прописными буквами;
- большинство тэгов пишутся парами. Пара тэгов называется контейнером. Исключение составляют тэги:

<BASE>

<HR>

Структура гипертекстовой сети - задается гипертекстовыми ссылками. Гипертекстовая ссылка - это адрес другого HTML-документа, который тематически, логически или каким-либо другим способом связан с документом, в котором эта ссылка определена. Запись гипертекстовых ссылок происходит с помощью URL-адреса. URL состоит из 4 частей:

протокол://имя_сервера: порт/имя_файла.

Следовательно, примером гипертекстовой ссылки может служить следующая запись:

Этот текст содержит ссылку на сервер

<A HREF= «http://www.yahoo.com/index.html» Yahoo! .

В приведенном примере элемент A, который в HTML обычно называют якорем (anchor), использует атрибут HREF, который обозначает гипертекстовую ссылку (Hypertext REFerence). Данная ссылка указывает на документ index.html на сервере www.yahoo.com, доступ к которому осуществляется по протоколу http. Следует заметить, что имя файла в данном примере указывать было не обязательно, т.к. браузеры сконфигурированы таким образом, чтобы пересылать файл index.html, если другого имени файла не указано.

Гипертекстовые ссылки в HTML делятся на два класса:

- контекстные
- общие

Контекстные ссылки вмонтированы в тело документа. Общие ссылки связаны со всем документом в целом и могут быть использованы для просмотра любого фрагмента документа.

Структура HTML документа. Все, что расположено между <HTML> и </HTML> - это документ являющийся кодом HTML, который включает в себе все другие тэги HTML и все информационное содержание документа.

Документ обычно состоит из заголовка и тела документа.

1. Заголовок документа

Заголовок документа содержит информацию о самом документе. Заголовок идет сразу же после тэга <HTML>, начинается с тэга <HEAD> и заканчивается тэгом </HEAD>. В заголовке обязательно должны присутствовать два элемента название документа и его базовый URL.

1.1 Название документа. Название документа располагается между тэгами <TITLE> </TITLE>. Название документа появляется в строке заголовка Internet Explorer и в списках журнала (History) и папки Избранное (Favorites). Когда вы выбираете название, вы должны сделать его достаточно полным для того, чтобы описать содержание вашего документа, однако не слишком длинным, чтобы оно поместилось в строке заголовка браузера. Как правило название не должно превышать 40 символов. В отсутствие названия, в строке заголовка IE, в списках Журнала и в папке Избранное указывается URL документа.

1.2. Базовый URL. Базовый адрес URL указывается в тэге <BASE HREF= «базовый URL»>. Базовый URL приравнивается к абсолютному URL документа. После того как вы указали базовый URL, все другие URL могут указываться по отношению к нему и называются относительными или частично определенными. Например, мы указали в качестве базового URL - <http://www.microsoft.com/office/word/d1.html>, а далее в тексте хотим указать адрес документа d2.html, расположенного в каталоге office/excel/. В этом случае достаточно написать - ../excel/d2.html.

Содержание элемента BASE интерфейсом пользователя не отображается.

Указание базового URL сокращает объем вводимой информации, а также облегчает возможность переноса документов с сервера на сервер.

2. Тело документа. Тело документа HTML содержит весь текст, несущий информацию и все тэги HTML, используемые для форматирования текста. Тело документа размещается сразу же после заголовка в тэгах <BODY> </BODY>. Элемент BODY имеет атрибуты:

BACKGROUND= «» - определяет фон, на котором отображается текст документа. Например,

<BODY BACKGROUND= «file:///c:» >

BGCOLOR= «#XXXXXX» - цвет фона

TEXT= «#XXXXXX» - цвет текста

VLINK= «#XXXXXX» - цвет пройденных гипертекстовых ссылок

LINK= «#XXXXXX» - цвет гипертекстовых ссылок

ALINK= «#XXXXXX» - цвет ссылки, которая в данный момент активизируется

«#XXXXXX» - определяет цвет в терминах RGB - красный, зеленый, синий.

Если один или более параметров цвета текста и ссылок не определены, то по умолчанию цвета для них задаются следующим образом: LINK=blue, ALINK=red, VLINK=purple.

Например,

<BODY BGCOLOR= «#FFFFFF» TEXT= «#0000FF» VLINK=

белый синий

«#FF0000» LINK=«#00FF00» ALINK= «#FF00FF»>

красный зеленый сиреневый.

Приведем восемь основных цветовых комбинаций в шестнадцатичном коде. Шестнадцатеричные значения RGB

белый #FFFFFF

красный #FF0000

зеленый #00FF00

синий #FFFF00

голубой #00FFFF

сиреневый #FF00FF

желтый #FFFF00

черный #000000

Возможность, введенная в уже HTML 3.2 - можно использовать вместо шестнадцатеричного представления RGB следующие английские названия:

AQUA - аквамарин NAVY - темно-синий
 BLACK - черный OLIVE - оливковый
 BLUE - голубой PURPLE - пурпурный
 FUSCHIA - RED - красный
 GRAY - серый SILVER - серебряный
 GREEN - зеленый TEAL
 LIME - зеленоватый WHITE - белый
 MAROON - каштановый YELLOW - желтый

Пользоваться этими атрибутами следует осторожно, т.к. у пользователя может быть другой интерфейс, который эти параметры не интерпретирует.

2.1.2 Элементы управления разметкой.

Абзац (Тэг <P>)

Используется для того, чтобы обозначить начало нового абзаца. Хорошим стилем программирования считается заканчивать абзац тэгом </P>. Тэг <P> имеет атрибуты ALIGN и CLEAR.

ALIGN, используемый для изменения выравнивания внутри абзаца, может принимать следующие значения:

ALIGN=LEFT - выравнивание по левому краю

ALIGN=RIGHT - - выравнивание по правому краю

ALIGN=JUSTIFY - выравнивание по левому и правому краям

ALIGN=CENTER - центрирование.

Например,

<P ALIGN=CENTER>

Этот текст будет располагаться по центру

</P>

CLEAR применяется в случаях обтекания текстом графики или таблиц. Если автор не хочет управлять обтеканием он применяет CLEAR. CLEAR может принимать следующие значения:

CLEAR=LEFT - пропустить картинку, расположенную у левого края листа

CLEAR=RIGHT - пропустить картинку, расположенную у правого края листа

CLEAR=ALL - прервать отображение текста перед картинкой или таблицей и продолжить ниже.

Следует сказать, что атрибуты ALIGN и CLEAR являются общими для всех блочных элементов (заголовки, абзацы и т.п.)

Горизонтальное подчеркивание (Тэг <HR>) применяется для разделения документа на части

*Конец строки (Тэг
)* - принудительный перевод строки. В обычном режиме интерпретации программа интерфейса пользователя отображает текст в рабочем окне автоматически разбивая его на строки. В

этом режиме существующие в тексте концы строк игнорируются. По этому когда нужно, чтобы начинался с новой строчки следует ставить принудительный конец строки.

Стили заголовков. Язык HTML поддерживает шесть стилей заголовков, которые используются для того, чтобы в различной степени выделять текст. Стили нумеруются от одного до шести, причем стиль номер один является самым крупным по размеру шрифта. Для того, чтобы разметить текст каким-либо из стилей заголовков, следует заключить его в тэги

<Hn> </Hn>, где n- номер стиля

Стандарт языка HTML насчитывает 11 атрибутов у элемента заголовка. Но чаще всего применяется только ALIGN.

Элементы управления отображением символов. Все эти элементы можно разбить на два класса: элементы, управляющие формой отображения (font style) или физические стили, и элементы характеризующие тип информации (information type) или логические стили. Логические стили не связаны с атрибутами шрифта, они отмечают значение текста помеченного ими.

Для того, чтобы применить управление отображением достаточно просто расположить текст между соответствующими тэгами.

Таблица 1 Элементы стиля

Физические стили	
<I>.....</I>	Курсив (Italic)
.....	Усиление (Bold)
<U>.....</U>	Подчеркивание
<TT>.....</TT>	Телетайп (фиксированная ширина)
<BIG>.....</BIG>	Перечеркнутый
<SMALL>.....</SMALL>	Увеличенный
_{.....}	Уменьшенный
^{.....}	Подстрочный символы
Логические стили	
<ADDRESS> </ADDRESS>	Адрес (Курсив)
<BLOCKQUOTE> </BLOCKQUOTE>	Цитируемый абзац (Отступы справа и слева)
<CITE>.....</CITE>	Цитата (Курсив)
<CODE>.....</CODE>	Код (Шрифт с фиксированной шириной)
<DFN>.....</DFN>	Определение (Жирный или комбинация жирного и курсива)
.....	Эмфаза, выделение в тексте (Курсив)
<KBD>.....</KBD>	Клавиатура (Шрифт с фиксированной шириной)
<SAMP>.....</SAMP>	Пример (Шрифт с фиксированной шириной)

<code>.....</code> <code></code>	Выделение (Жирный шрифт)
<code><VAR>.....</VAR></code>	Переменная (Курсив)

Преформатированный текст (Тэги `<PRE>` `</PRE>`). представляется при помощи шрифтов с фиксированной шириной и не игнорирует лишние пробелы и другие «невидимые» символы, т.е. дополнительные пробелы, знаки табуляции и возврата каретки не игнорируются если они расположены между тэгами `<PRE>`.

Списки. Списки являются важным средством структурирования текста и применяются во всех языках разметки. Все списки требуют пару тэгов, указывающих тип списка и тэг, отмечающий начало каждого элемента.

Таблица 2 Списки HTML

Тип	Тэги списка	Тэги элемента
Нумерованный	<code>.....</code>	<code>....</code>
Маркированный	<code>.....</code>	<code>....</code>
Список описаний	<code><DL>.....</DL></code>	<code><DD>...</DD></code> <code><DT>...</DT></code>
Меню	<code><MENU>.....</MENU></code>	<code>....</code>
Директория	<code><DIR>.....</DIR></code>	<code>....</code>

Элементы в нумерованном списке автоматически нумеруются браузером, начиная с номера 1. Маркированные списки имеют символы маркера, а не номера. Список описаний позволяет представить термин, за которым на строку ниже за отступом следует его описание. Меню и каталоги подобны маркированным спискам, но задуманы как имеющие более короткие элементы. Можно помещать один список во внутрь другого, образуя вложенные списки. Для пропуска строки между элементами списка используют тэг `<P>` без закрывающего тэга.

Специальные символы. Поскольку многие символы имеют в HTML специфическое значение, необходимо использовать специальные последовательности, чтобы эти символы были показаны на экране

Таблица 3 Специальные последовательности символов для зарезервированных символов HTML

Последовательность	Вид	Значение
<code>&lt;</code>	<code><</code>	меньше чем
<code>&gt;</code>	<code>></code>	больше чем
<code>&amp;</code>	<code>&</code>	амперсant
<code>&quot;</code>	<code>«</code>	кавычка

Вы можете указывать в HTML документе любой символ ASCII включая знаки «&» «#» затем следует десятичный код из таблицы и «;». Например, для того, чтобы включить символ авторского права © следует написать - `©`;

Комментарии. В HTML можно включать комментарии, которые не будут выводиться на экран браузера. Комментарии имеют следующий формат
 <!-- Комментарий -->

2.1.3 Гипертекстовые ссылки.

Все рассмотренные выше средства управления отображением являются безусловно важными, но только дополнительными к основному элементу документа гипертекстовой ссылке.

Для записи гипертекстовой ссылки используется элемент <A>....., который называется якорь (anchor). Якорь имеет несколько атрибутов, главным из которых является HREF (HyperText REference). Вид простой ссылки я вам уже записывала - вспомним

```
<A HREF= «http://www.yahoo.com/index.html» Yahoo! </A>
```

Значением атрибута HREF является адрес документа index.html на машине www.yahoo.com, доступ к которой осуществляется по протоколу HTTP. Это все еще называют адресной частью гиперссылки. Содержание элемента A заключенное между тэгом начала и окончания, в данном случае это Yahoo! выделяется в тексте цветом, определенным для контекстных гипертекстовых ссылок (LINK).

Другой формой использования элемента A является определение точек внутри текста на которые можно сослаться:

```
<A NAME= «point»>
```

Для ссылки на такую точку используют следующую форму URL:

```
<A HREF= «http://www.yahoo.com/index.html#point» Ссылка на точку внутри документа Yahoo! </A>.
```

Следует заметить, что A имеет еще один интересный атрибут SHAPE. Этот атрибут описывает форму участка графического образа, который можно использовать как гипертекстовую ссылку, но эту возможность мы обсудим позже, когда будем рассматривать графический элемент MAP.

2.1.4. Средства описания графической информации.

Вы уже имеете представление об некоторых тэгах языка HTML. В принципе этих знаний достаточно чтобы начать действовать. При помощи тех тэгов, которые мы уже изучили можно расположить любое количество информации, причем в логической и *достаточно* привлекательной форме. Недостатком этих документов будет то, что они полностью текстовые. Документы с большим количеством текста обычно быстро теряют аудиторию, если ничто не нарушает монотонность больших блоков слов. Графические элементы, нарушающие текстовое однообразие Web-страниц, могут быть очень разнообразными. Наиболее важным в графике Web является не ее сложность или простота, а насколько она эффективна в вопросе передачи той информации, которую вы хотите сообщить посетителям вашего Web-узла.

Графические форматы GIF и JPEG.

В WWW наиболее часто встречаются *GIF* и *JPEG* форматы

Формат *GIF* (Graphics Interchange Format - формат обмена графическими данными). Графика сохраняемая в *GIF*, ограничивается 256 цветами. Поэтому вы скорее не будете хранить полноцветные фотографии в этом формате. *GIF* лучше всего использовать для значков эмблем и других искусственно созданных изображений. Сейчас используется формат *GIF89a*, который поддерживает следующие эффекты:

- чересстрочный показ (Interlacing). Это означает, что части изображения не являющиеся соседними хранятся вместе. По мере того как браузер считывает изображение оно постепенно прорисовывается в течении нескольких проходов. Это может оказаться полезным, т.к. пользователь может понять, что изображено на рисунке не дожидаясь его полной загрузки.
- прозрачность (Transparency). В прозрачном *GIF*-изображении один из цветов обозначается как прозрачный, что позволяет фону документа проступать сквозь него.
- анимация. Анимированные *GIF* создаются при помощи сохранения в одном файле последовательности изображений, необходимых для анимации. Браузер будет знать, что необходимо показывать изображения одно за другим для того, чтобы создать анимацию. Программы, которые позволяют вам сохранять несколько изображений в *GIF* файле также позволяют вам указать промежуток времени до начала анимации и сколько раз она должна быть повторена.

Photoshop, позволяет создавать как прозрачные так и чересстрочные *GIF*.

Формат *JPEG* (Joint Photographic Experts Group - Объединенная группа экспертов по изображениям) является именем для набора форматов, которые поддерживают полноцветные изображения и сохраняют их в сжатой форме. Недавно появившийся формат прогрессивного *JPEG* (p- *JPEG*) изображению эффект «прорисовки», так же как и чересстрочный *GIF*. Прозрачность в *JPEG*-изображениях невозможна.

Вы должны сохранять изображения в отдельных файлах, несмотря на то, что они демонстрируются внутри документа HTML.

Тэг **

Для того, чтобы поместить изображение внутрь страницы используется тэг ** с обязательным параметром *SCR= «URL»*
**, где URL-адрес файла, содержащего графические данные. Атрибут *SCR* является обязательным, а URL может указывать куда угодно.

Тэг ** может иметь следующие атрибуты:

ALIGN= «расположение» - расположение текста по отношению к изображению

TOP - наверху

BOTTOM - внизу

MIDDLE - посередине

ALT= «текст» - текст, который нужно показывать вместо изображения для текстовых браузеров.

ISMAP - атрибут, указывающий, что изображение будет использоваться в виде карты.

WIDTH - ширина может указываться в пикселях или %

HEIGHT - высота

Рекомендуется указывать ширину и высоту, т.к. они позволяют браузеру оставить достаточно места для изображения при загрузке страницы. При этом процесс создания страницы заканчивается быстро без ожидания полной загрузки изображения для того чтобы определить его размеры.

Использование графики в качестве указателя ссылок.

Формат графической привязки такой же как и текстовой. Однако вместо того, чтобы разместить текст между тэгами следует расположить тэг
 Yahoo!
 <IMG SRC= «ulg_ie40.jpg»
 ALIGN= «MIDDLE»> Это изображение является ссылкой.

Горизонтальные линии. Тэг <HR>

Говоря о графике следует вспомнить тэг <HR>. Internet Explorer поддерживает различные атрибуты, позволяющие управлять размером толщиной, выравниваем, цветом и видом тени от линеек. Тэг <HR> может иметь следующие атрибуты:

WIDTH=pixels | percent - позволяет изменить ширину линейки задав размер в пикселях или процентах (рекомендуется в %, т.к. пользователь не знает ширину экрана)

ALIGN=LEFT | RIGHT | CENTER - задает выравнивание куска линейки по умолчанию центрируется

SIZE=n - высота линейки по умолчанию 1

COLOR=RGB триплет | название цвета

NOSHADA - отменяет тени от линеек

2.1.5 Средства описания таблиц. (Тэг <TABLE>.....</TABLE>)

Для того, чтобы лучше понимать табличные тэги необходимо бы рассмотреть как структурированы таблицы HTML. Основным строительным блоком являются ячейки, которые содержат элемент данных таблицы или заголовков столбца данных. Логически связанные ячейки сгруппированы вместе в строку таблицы. Строки в свою очередь составляют саму таблицу. (Ячейки, строки, таблица). Итак, тэги <TABLE>.....</TABLE> являются контейнером для других элементов описания таблицы.

Тэг <TABLE> может иметь атрибут рамки вида BORDER=n, который создает рамку шириной n пикселей вокруг таблицы. По умолчанию таблица не заключена в рамку.

<TABLE BORDER=3>

</TABLE>

В стандарте языка HTML тэг <TABLE> может иметь еще один атрибут BORDERCOLOR, который должен быть установлен равным шестнадцатеричному RGB триплету или английскому названию требуемого цвета и задает цвет рамки.

Тэги <TR>.....</TR> (Table Row) определяют строку таблицы и заключают в себя все тэги необходимые для создания строки таблицы.

Тэг <TR> может иметь атрибуты ALIGN и VALIGN

ALIGN - контролирует горизонтальное расположение содержимого в ячейках строки. Он может принимать значения
ALIGN=LEFT - выравнивание по левому краю
ALIGN=RIGHT - - выравнивание по правому краю
ALIGN=CENTER - центрирование.

VALIGN - контролирует вертикальное расположение содержимого в ячейках строки. Он может принимать значения
VALIGN=TOP - выравнивание по верхнему краю
VALIGN=BOTTOM - - выравнивание по нижнему краю
VALIGN=MIDDLE - посередине.

По умолчанию ячейки заголовки центрируются, а ячейки данные выравниваются к левому верхнему краю.

Когда вы определили строку, вы готовы к тому чтобы заполнить ячейки.

Тэги <TH>.....</TH> (Table Header) используется для описания ячейки содержащей заголовков. Текст между тэгами заголовка является тем, что появится в ячейке заголовка.

Тэги <TD>.....</TD> (Table Data) используется для описания ячейки содержащей данные таблицы. Текст между тэгами элемента данных является тем, что появится в ячейке данных таблицы.

В тэгах <TH> <TD> могут также применяться атрибуты вертикального и горизонтального выравнивания.

Выравнивание элементов данных и заголовков в таблицах происходит по следующим правилам:

- параметры выравнивания, которые указаны в тэгах <TH> <TD> имеют преимущество перед всеми другими параметрами, но применяются только для данной ячейки;
- параметры выравнивания, указанные в тэге <TR> используются вместо значений по умолчанию и во всех ячейках строки, если только их действие не отменяется указанием параметров выравнивания в тэгах <TH> <TD>;
- в отсутствии атрибутов выравнивания в тэгах <TR> <TH> <TD> используются их значения по умолчанию.

Уже в стандарте языка HTML 3.2 в тэге <TD> имеется возможность использовать атрибут BGCOLOR который должен быть установлен равным

шестнадцатеричному RGB триплету или английскому названию требуемого цвета и задает цвет фона ячейки

Заголовок таблицы

Текст заголовка таблицы помещается между тэгами

<CAPTION>.....</CAPTION> и размещается в табличном контейнере.

<CAPTION><H3>заголовок таблицы</H3></CAPTION> - заголовок появится над таблицей. Если нужно поместить заголовок под таблицей следует использовать в тэге <CAPTION> атрибут ALIGN=BOTTOM
<CAPTION ALIGN=BOTTOM > <H3> заголовок таблицы </H3>
</CAPTION>

Объединение строк или столбцов

По умолчанию ячейка занимает участок ограниченный одной строкой и одним столбцом. В тэгах <TH> <TD> можно указать атрибуты:

ROWSPAN=n - устанавливает количество строк, которые должна занимать ячейка, n- количество строк

COLSPAN=n устанавливает количество столбцов, которые должна занимать ячейка, n- количество столбцов

Пример таблицы:

```
<HTML>
<HEAD>
<TITLE>Таблица</TITLE>
</HEAD>
<BODY>
<H2 ALIGN="CENTER">Пример построения таблицы</H2>
<P ALIGN=CENTER>
<TABLE BORDER=2>
<CAPTION> <H3>Заголовок тавлицы</H3> </CAPTION>
<TR ALIGN=CENTER VALIGN=MIDDLE>
<TH ROWSPAN=2>Название элементов</TH>
<TH COLSPAN=3>Значения элементов</TH>
</TR>
<TR ALIGN=CENTER VALIGN=MIDDLE>
<TH>Значение 1</TH>
<TH>Значение 2</TH>
<TH>Сумма</TH>
</TR>
<TR ALIGN=CENTER VALIGN=MIDDLE>
<TD>Элемент 1</TD>
<TD>10</TD>
<TD>20</TD>
<TD>30</TD>
</TR>
<TR ALIGN=CENTER VALIGN=MIDDLE>
<TD>Элемент 2</TD>
```

```

<TD>50</TD>
<TD>40</TD>
<TD>90</TD>
</TR>
</TABLE>
</P>
<HR>
</BODY>
</HTML>

```

Таблица, HTML-код которой приведен в примере, показана на рисунке 1.

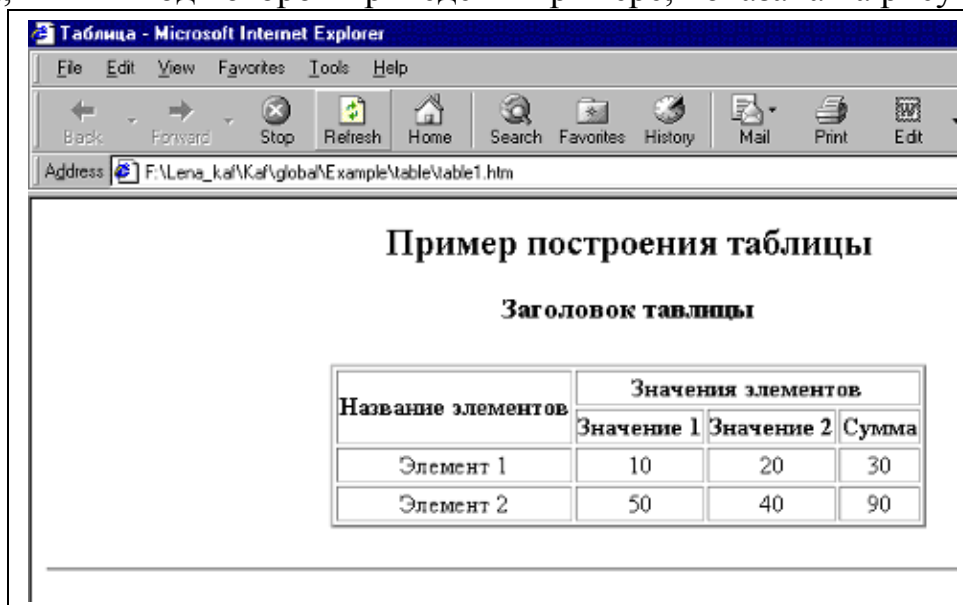


Рис. 1. Пример построения таблицы

В заключение следует сказать, что таблицы нашли широкое применение. Они используются для размещения графической, текстовой информации, гиперссылок, цитат и т.д. В конечном итоге таблицы предназначены для выравнивания информации.

2.1.6 Выравнивание полей данных в формах

Еще одно полезное применение таблиц - формы. Формы HTML используются для того, чтобы собирать информацию от тех, кто посещает страницы Web. Вы можете использовать формы для того, чтобы запросить имя пользователя и электронный адрес посетителя или для того, чтобы провести исследование рынка. Данные собираются в формах и посылаются на ваш Web-сервер для обработки и хранения. Формы создавать легко, т.к. для создания полей ввода в форму вам потребуется лишь несколько тэгов HTML. Например тэг `<INPUT>`, предназначен для создания поля текстового ввода. Однако от вас зависит какой описательный текст вы поместите рядом с полем, что бы пользователь знал, что он должен ввести. С точки зрения дизайнера форма выглядит более привлекательной если ее поля выровнены. Помещая текст и поля в соответствующие ячейки таблицы, вы добьетесь

того, что поля будут автоматически выровнены. Приведем следующий пример:

```
<HTML>
<HEAD>
<TITLE>Формы</TITLE>
</HEAD>
<BODY BGCOLOR="FUSCHIA"
TEXT="MAROON">
<FORM METHOD=rost
ACTION="mailto:user@mail.ru">
<H3>Использование таблиц для выравнивания
полей форм </H3>
<TABLE>
<TR>
<TD VALIGN=TOP>Текстовое поле
TEXT</TD>
<TD><INPUT TYPE=text NAME="text1"
VALUE="Sample of text1"
SIZE=30></TD>
</TR>
<TR>
<TD VALIGN=TOP>Текстовое поле
PASSWORD</TD>
<TD><INPUT TYPE=password NAME="pwd"
VALUE="Sample of password"
SIZE=32></TD>
</TR>
<TR>
<TD VALIGN=TOP>Текстовое поле
TEXTAREA</TD>
<TD><TEXTAREA NAME="text2" ROWS=4
COLS=30>Sample of text1
</TEXTAREA></TD>
</TR>
<TR>
<TD VALIGN=TOP>Переключатели
CHECKBOX</TD>
<TD><INPUT TYPE=CHECKBOX
NAME="chk1" VALUE="on" CHECKED>
Первый<BR>
<INPUT TYPE=CHECKBOX NAME="chk2"
VALUE="on">
Второй<BR>
<INPUT TYPE=CHECKBOX NAME="chk3"
VALUE="on" CHECKED>
Третий<BR>
</TD>
<TD>
<TR>
<TD VALIGN=TOP>Переключатели
RADIO</TD>
<TD><INPUT TYPE=RADIO NAME="pad"
VALUE="on1" CHECKED>
Первый<BR>
<INPUT TYPE=RADIO NAME="pad"
VALUE="on2">
Второй<BR>
```

```
<INPUT TYPE=RADIO NAME="pad" ALUE="on3" >
Третий<BR>
</TD>
</TR><TR>
<TD VALIGN=TOP>Список</TD>
<TD><SELECT NAME="sel" SIZE=1>
<OPTION VALUE="First Option">First Option</OPTION>
<OPTION VALUE="Second Option">Second
Option</OPTION>
<OPTION VALUE="None">None Selected</OPTION>
</SELECT>
</TD>
</TR>
<TR>
<TD VALIGN=TOP>Скрытый орган управления</TD>
<TD><INPUT TYPE=HIDDEN NAME="hid"
VALUE="Hidden"></TD>
</TR>
<TR>
<TD align= center VALIGN=middle><INPUT
TYPE=SUBMIT VALUE="Send"></TD>
<TD align= center VALIGN=middle><INPUT TYPE=RESET
VALUE="Reset"></TD>
</TR>
<TR>
<TD colspan=2 align= center VALIGN=middle><INPUT
TYPE=IMAGE SRC="ulg_ie40.jpg" BORDER=0></TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

Рис. 2. Экранная форма и исходный код страницы «Использование таблиц для выравнивания полей в форме»

2.2 Некоторые дополнительные возможности языка HTML

2.2.1 Расширенное использование графики на страницах Web

Основные принципы выбора размеров изображения.

Как мы уже говорили одним из способов уменьшения времени ожидания загрузки встроенных изображений является явное указание в HTML-файле размеров изображения. IE использует эту информацию для форматирования страницы. Зная точные размеры графического элемента IE может немедленно разместить текстовые данные с учетом окончательного расположения графических изображений. Что следует учитывать для определения размера изображения?

Поскольку большинство пользователей используют 14-дюймовый монитор с 256-палитрой, следует принять это в качестве стандарта де-факто при создании изображений. Типичное разрешение монитора - 640 пикселей в ширину и 480 в высоту. При разработке изображений следует учитывать, что окно просмотра будет занимать лишь часть этой области. Таким образом в качестве правила следует принять, что максимальная ширина любого изображения не должна превышать 500 пикселей, а максимальная высота - 300 пикселей. Следовательно тэг должен выглядеть примерно так:

```
<IMG SRC= «URL» HEIGHT=300 WIDTH=500>
```

Порядок включения параметров HEIGHT и WIDTH не играет роли.

Следует ответить здесь еще на два вопроса:

1. Каким образом определить ширину и высоту изображения?
2. Каким образом изменить ширину и высоту изображения?

Для того чтобы определить размер изображения достаточно загрузить его в любой графический редактор и просмотреть свойства. Обычно длина и ширина дается в дюймах, см. и пикселях.

Что касается изменения размеров изображения это можно сделать в том же графическом редакторе, либо с помощью IE

Чтобы пропорционально уменьшить размер изображения 200×200, например в два раза, следует указать

```
<IMG SRC= «URL» HEIGHT=100 WIDTH=100>
```

Чтобы пропорционально увеличить размер изображения 200×200, например в два раза следует указать

```
<IMG SRC= «URL» HEIGHT=400 WIDTH=400>
```

Также возможно увеличивать и уменьшать размер изображения относительно размеров окна IE

```
<IMG SRC= «URL» HEIGHT=50% WIDTH=50%>
```

При такой записи изображение займет 50% от высоты и ширины окна IE.

Совместное использование параметров BGCOLOR и BACKGROUND.

Как вы уже знаете параметр BACKGROUND тэга <BODY> позволяет указать изображение, которое можно загрузить в качестве фона для определенной страницы. Повторим еще раз как это должно выглядеть:

<BODY BACKGROUND = «URL»>

документ

</BODY>

Изображение, используемое в качестве фона многократно повторяется по принципу мозаики. Поэтому фоновые изображения должны быть такими, чтобы при их появлении в окне просмотра границы повторяющихся изображений были бы невидимыми. Существуют коллекции фоновых изображений, которые вы можете использовать при оформлении собственных Web-страниц. Такую коллекцию можно посмотреть по адресу: http://home netscape com assist net_sites/bg/background.html.

Фоновые изображения требуют, чтобы IE выбрал и загрузил фоновое изображение прежде, чем будет отображена остальная часть страницы. Чтобы уменьшить время загрузки, используйте достаточно мелкие изображения - например, 100×100 пикселей.

Параметр BGCOLOR тэга <BODY> изменяет только цвет фона окна просмотра. Цвет окна просмотра изменяется немедленно после интерпретации соответствующего тэга HTML. Если вы используете BACKGROUND и BGCOLOR вместе, то IE немедленно закрасит фон тем цветом, который указан в параметре BGCOLOR. Потом когда будет загружено фоновое изображение, определенное параметром BACKGROUND, оно заменит фон определенный BGCOLOR. Зная об этом авторы страниц стараются подобрать цвет фона таким, чтобы он был близок к цвету фона изображения загружаемого в параметре BACKGROUND.

На серверах WWW доступен целый ряд превосходных средств для предварительного просмотра цветов, заданных в шестнадцатеричном коде: http://home netscape com assist net_sites/bg/index.html.

Плавающие изображения

Новые значения атрибута ALIGN тэга позволяют создать плавающие изображения и облегчают размещение текста вокруг изображения.

ALIGN=LEFT | RIGHT, прижимают изображения влево или вправо и позволяют тексту его обтекать.

Поскольку допускается обтекание изображения текстом, необходимо иметь механизм, позволяющий оставлять некоторое пустое пространство вокруг изображения.

Атрибуты HSPACE VSPACE позволяют оставить некоторое количество пикселей справа, слева, сверху и снизу изображения.

HSPACE =n (слева и справа)

VSPACE=n (сверху и снизу)

Атрибут BORDER=n задает рамку вокруг изображения BORDER=0 - отменяет рамку, таким образом можно отменить рамку вокруг изображения являющегося гиперссылкой.

Размещение графики с помощью тэга <FIG>

В качестве альтернативы тэга `` предложены тэги `<FIG>.....</FIG>`, не вошедший, однако, в спецификацию HTML 3.2. Тэг `<FIG>` требует атрибутов:

`SRC= «URL»` - задает адрес изображения

`NOFLOW` - запрещает обтекание рисунка текстом

`ALIGN=LEFT | RIGHT | CENTER | JUSTIFY`

`BLEEDLEFT` - размещает рисунок у левой кромки браузера, не оставляя полей.

`BLEEDRIGHT` - размещает рисунок у правой кромки браузера, не оставляя полей.

`UNITS` - задает единицы измерения

`WIDTH`

`HEIGHT`

`IMAGEMAP` - определяет изображение как карту.

Между тэгами `<FIG>` могут располагаться текст, включая заголовки и подписи.

Заголовки располагаются между тэгами `<CAPTION>.....</CAPTION>`

Подписи располагаются между тэгами `<CREDIT>.....</CREDIT>`

Для тэгов `<FIG>.....</FIG>` предложена возможность перекрытия двух изображений, которая достигается за счет использования тэга `<OVERLAY>`

Пример:

2.2.2 Фоновый звук и видеоклипы

Используя в теле документа тэг `<BGSOUND>`, вы можете сопровождать ваши Web-страницы звуком. Тэг `<BGSOUND>` имеет обязательный атрибут `SRC`, в котором необходимо указать URL звукового файла. Звуковой файл может быть в одном из следующих форматов: wav, au, mid.

Тэг `<BGSOUND>` имеет также атрибут `LOOP=n`, где `n` позволяет задать число повторений звукового сигнала или же `LOOP=INFINITE`, что соответствует проигрыванию звука в течении всего времени, пока страница открыта. Например:

```
<BGSOUND SRC= «hello. wav» LOOP=3>
```

```
<BGSOUND SRC= «hello. wav» LOOP=INFINITE>
```

Компания Microsoft значительно расширила возможности тэга ``, обеспечив поддержку видеоклипов в формате Audio Video Interleave (AVI). Рассмотрим следующие атрибуты для тэга ``, необходимые для показа видеоклипов:

`DYBSRC= «URL»` - задает URL файла в формате AVI, содержащего клип

`CONTROLS` - размещает в окне браузера панель управления, что позволяет пользователю управлять воспроизведением клипа (остановить и перемотать клип)

START=FILEOPEN | MOUSEOVER - задает способ запуска видеоклипа - либо при открытии файла либо по щелчку мыши (могут присутствовать оба параметра через запятую)

LOOP=n | INFINITE - управляет числом повторений клипа

LOOPDELAY=n задает задержку в миллисекундах перед повторением клипа.

Пример:

2.2.3 Текстовые эффекты

*Задание текстовых эффектов с помощью тэгов *

Пара тэгов позволяет задавать цвет и размер и начертание текста.

Для изменения размера используется атрибут

SIZE=n, где n может быть равным 1-7 (по умолчанию 3)

Можно задавать SIZE также в относительных единицах, указав на сколько больше (+) или меньше (-) чем основной должен быть задаваемый текст.

Используя атрибут COLOR= «RGB триплету» можно изменять цвет текста. Пример:

```
<FONT SIZE=+2 COLOR= «RED»>I</FONT>nternet<FONT SIZE=+2
COLOR= «RED»>E </FONT>xplorer
```

Используя атрибут FASE можно выбрать гарнитуру шрифта. В атрибуту может быть указано несколько шрифтов как альтернатива

```
<FONT FASE= «Times, Helvetica»>text</FASE>
```

Бегающий текст - тэг <MARQUEE>.

Тэг <MARQUEE> имеет несколько атрибутов, позволяющих эффективно управлять появлением и движением текста:

BGCOLOR= «RGB триплет» - определяет цвет фона окна

BEHAVOIR=SCROLL | SLIDE | ALTERNATE - определяет способ появления текста в окне

DIRECTION=LEFT|RIGHT - указывает направление, в котором движется текст

SCROLLAMOUNT = n - указывает число пикселей между последовательностями бегущего текста

SCROLLDELAY = n - указывает временную задержку в миллисекундах между появлениями текста

HEIGHT = pixels | percent - определяет высоту окна бегущего текста в пикселях или процентах

WIDTH = pixels | percent - определяет ширину окна бегущего текста в пикселях или процентах

HSPASE = n - определяет отступы сверху и снизу

VSPASE = n - определяет отступы слева и справа

LOOP = n | INFINITE - задает число повторов

ALIGN = TOP | MIDDLE | BOTTOM - задает тип выравнивания

Атрибут BEHAVOIR требует отдельного пояснения. BEHAVOIR=SCROLL задает прокрутку текста в направлении заданном в DIRECTION.

BEHAVOIR= SLIDE - текст появляется в окне и остается там

BEHAVOIR= ALTERNATE - текст перемещается от одной границы окна до другой, но не покидает окна.

Пример:

Упорядоченные списки

IE полезное расширение к тэгу - атрибут TYPE. По умолчанию элементы упорядоченного списка нумеруются целыми числами начиная с 1. Атрибут TYPE можно установить равным «A» «a» «I» «i»

```
<OL START= «I»>
```

```
</OL>
```

Есть еще один атрибут для тэга - атрибут START, позволяющий задавать начальное значение для нумерации.

```
<OL TYPE=«i» START=3>
```

```
</OL>
```

2.2.4. Фреймы.

IE поддерживает современную концепцию - т.н. фреймов. Используя тэги фреймов можно разделить окно браузера на отдельные части, и в каждую из этих частей загрузить свой HTML-документ.

Первый этап в создании документа с фреймами - это деление экрана IE на требуемые вам части. Это достигается путем замены тэгов <BODY>.....</BODY> в HTML-документе на <FRAMESET>....</FRAMESET>.

Тэг <FRAMESET> может иметь один из двух атрибутов:

ROWS= «pixels | percent | *» для деления экрана на несколько полос

COLS= «pixels | percent | *» для деления экрана на несколько колонок.

Для создания действительно интересных компоновок можно группировать тэги <FRAMESET>

Пример деления экрана на 4 части:

```
<FRAMESET ROWS= «50%, 50%»>
```

```
<FRAMESET COLS= «50%, 50%»>
```

```
</FRAMESET>
```

```
<FRAMESET COLS= «50%, 50%»>
```

```
</FRAMESET>
```

```
</FRAMESET>
```

Если вы не знаете, начать ли вам с колонок или полос, попробуйте вооружиться карандашом и бумагой и набросайте предполагаемый вид экрана. Если у вас получаются неразрывные горизонтальные линии, начните с полос, если вертикальные - с колонок.

После разметки фреймов можно заняться их содержимым. Для этого используется тэг <FRAME>. Наиболее важным атрибутом этого тэга является атрибут SRC, который сообщает IE URL документа, который вы хотите загрузить в фрейм.

SRC= «URL» - задает URL документа, загружаемого в фрейм

Кроме этого атрибута у тэга <FRAME> могут быть следующие атрибуты:

MARGINHEIGHT = n - задает размер пустого пространства над и под фреймом

MARGINWIDTH = n - задает размер пустого пространства слева и справа фрейма

NAME = «name» - присваивает фрейму уникальное имя, на которое можно ссылаться из других документов

NORESIZE - не позволяет пользователю изменить размеры фрейма

SCROLLING=YES | NO | AUTO - управляет появлением в фрейме горизонтальных и вертикальных лифтов (SCROLLBARS)

FRAMEBORDER =n - задает ширину границы между фреймами допускается значение 0.

Пример:

В заключение следует сказать, что пользователи не имеющие IE 3.0 или Netscape 2.0 не смогут увидеть того, что вы хотели им показать, поскольку их браузеры не поддерживают тэги <FRAMESET>. Идя навстречу пользователям, имеющим «нефреймовые» браузеры, вы можете разместить альтернативный HTML-документ между тэгами

<NOFRAMES>

</NOFRAMES>

Эти тэги должны вставляться после начального тэга <FRAMESET>, но до любого группируемого тэга <FRAMESET>.

Любой альтернативный текст расположенный между этими тэгами будет понят и отображен «нефреймовыми» браузерами. «Фреймовые» браузеры, напротив, игнорируют все расположенное между этими тэгами, и работают с фрейм-ориентированным текстом.

Теперь несколько практических советов:

Большинство узлов, удачно использующих фреймы, имеют узкую полосу навигации в левой части экрана, а остальное пространство используют для изменяющегося материала:

<FRAMESET COLS= «25%, 75%»>

<FRAME NAME= «nav» SRC= «navigate.html»>

<FRAME NAME= «main» SRC= «main.html»>

</FRAMESET>

Эти команды создают область навигации в левой части экрана и загружают в нее документ navigate.html. Оставшиеся три четверти отводятся для меняющегося материала, и первым в этот фрейм загружается документ main.html.

Одной из приятных особенностей работы с фреймами является то, что разметив фреймы вы можете продолжать использовать ранее созданные HTML-документы. Пожалуй самое сложное в работе с фреймами - это заставить материал появляться там, где вы хотите, чтобы он появился. Здесь становятся существенными имена фреймов. Присвоив фрейму с меняющимся содержанием имя main.html, вы можете использовать атрибут TARGET во всех тэгах <A> для того чтобы загрузить во фрейм main все документы на которые есть гиперссылки. Пример ссылки в файле navigate.html может выглядеть так:

```
<A HREF= «press/index.html» TARGET= «main»>
Press Releases
</A>
```

Плавающие фреймы

Плавающий фрейм можно представить как окно браузера, которое вы можете открыть в основном окне. Это похоже на картинку в картинке телевизора.

Плавающий фрейм размещается на странице с помощью тэгов <IFRAME>.....</IFRAME> . Тэг <IFRAME> имеет три обязательных атрибута: WIDTH, HEIGHT и SRC. Первые два задают ширину и высоту в пикселях SRC задает URL-адрес документа загружаемого в фрейм.

В дополнение к трем обязательным атрибутам тэг <IFRAME> имеет несколько и других атрибутов:

FRAMEBORDER - плавающий фрейм имеет скошенную рамку, что создает эффект углубления в основном окне. Установка FRAMEBORDER=0 убирает скошенную рамку

SCROLLING - лифт для скролинга можно отменить задав SCROLLING=NO

HSPACE VSPACE - оставляет пустое пространство вокруг фрейма.

ALIGN

NAME - присвоение имени плавающему фрейму позволяет ссылаться на него в атрибуте TARGET тэга <A>

Если все ссылки в navigate.html должны попасть в фрейм main, в заголовке документа можно использовать тэг <BASE>, в котором значение TARGET будет сразу задано для всех ссылок:

```
<HEAD>
<TITLE>Site Navigator Options</TITLE>
<BASE TARGET= «main»>
</HEAD>
```

2.2.5. Карты изображения.

Мы уже знаем, что изображение можно использовать в качестве гиперссылки. Карта-изображение позволяет привязывать гипертекстовые ссылки к различным областям изображения. Когда карты-изображения только появились, существовал только один их вид: карты-изображения,

существующие только на сервере (server-side imagemaps). В зависимости от того под управлением какой операционной системы работал сервер, существовали некоторые различия в том как можно было создавать карты-изображения. Сегодня понятие карты-изображения стандартизовано в языке HTML и появились карты-изображения, работающие на стороне клиентской программы (браузера) IE.

Карта-изображение для клиента (client-side) позволяет разместить всю информацию о карте в HTML-файле, который включает изображение. Когда пользователь перемещает указатель в пределах изображения, реализующего клиентский вариант, соответствующий URL-адрес отображается в нижней части браузера.

Существуют две возможности для организации работы клиентских карт-изображений. Обе служат целям представления информации из файла описания карты для клиентской программы:

- первая возможность подразумевает использование парных тэгов `<FIG>.....</FIG>` с помещенными между ними с помощью тэгов `<A>.....` данными из файла описания карты.
- вторая возможность, отвечающая стандарту HTML 3.2 подразумевает использование парных тэгов `<MAP>.....</MAP>` с помещенными между ними с помощью тэга `<AREA>` данными из файла описания карты.

Рассмотрим первый вариант:

Атрибут `HREF` в тэге `<A>` содержит URL, который требуется загрузить, когда пользователь щелкнет мышью над активной областью. Форму этой активной области описывает атрибут `SHAPE`, и он предоставляет информацию о координатах характерных точек этой фигуры.

В качестве типов активных областей могут использоваться: прямоугольник, круг, многоугольник и точка. Эти области отражают геометрическую форму активных областей. Их координаты определены в пикселях относительно верхнего левого угла изображения (0,0). Каждая координата отделяется от соседней с помощью запятой. Следующий список определяет основные формы областей и правила задания их координат:

- `RECT` - определяет прямоугольник. Задаются координаты верхнего левого и нижнего правого углов в прямоугольнике. Активная область внутри прямоугольника.
- `CIRCLE` - определяет круг. Задаются координаты центра круга и одной из граничных точек. Активная область внутри круга.
- `POLY` - определяет многоугольник. Задается список координат вершин многоугольника. Многоугольник может содержать до 100 вершин. Активная область внутри многоугольника.
- `POINT` - определяет точку на изображении. Координаты точки задаются относительно верхнего левого угла изображения. Активная область находится в непосредственной близости от данной точки.

- DEFAULT - определяет все области изображения, которые не являются какой-либо другой активной областью. Не может использоваться совместно с POINT.

Пример иллюстрирующий первую возможность:

Рассмотрим второй вариант:

Тэги `<MAP>.....</MAP>` содержат информацию об активной области. Каждой карте, определенной с помощью этих тэгов необходимо дать уникальное имя, на которое можно сослаться из тэга ``, отвечающего за размещение изображения карты. Имя, указанное для тэга `<MAP>`, является уникальным идентификатором. Между тэгами `<MAP>.....</MAP>` располагаются тэги `<AREA>`, где указываются координаты активных областей и даются URL, на которые можно перейти щелкнув по данной области.

Атрибут HREF в тэге `<AREA>` содержит URL, который требуется загрузить, когда пользователь щелкнет мышью над активной областью. Форму этой активной области описывает атрибут SHAPE, и он предоставляет информацию о координатах характерных точек этой фигуры. Рассмотрим пример:

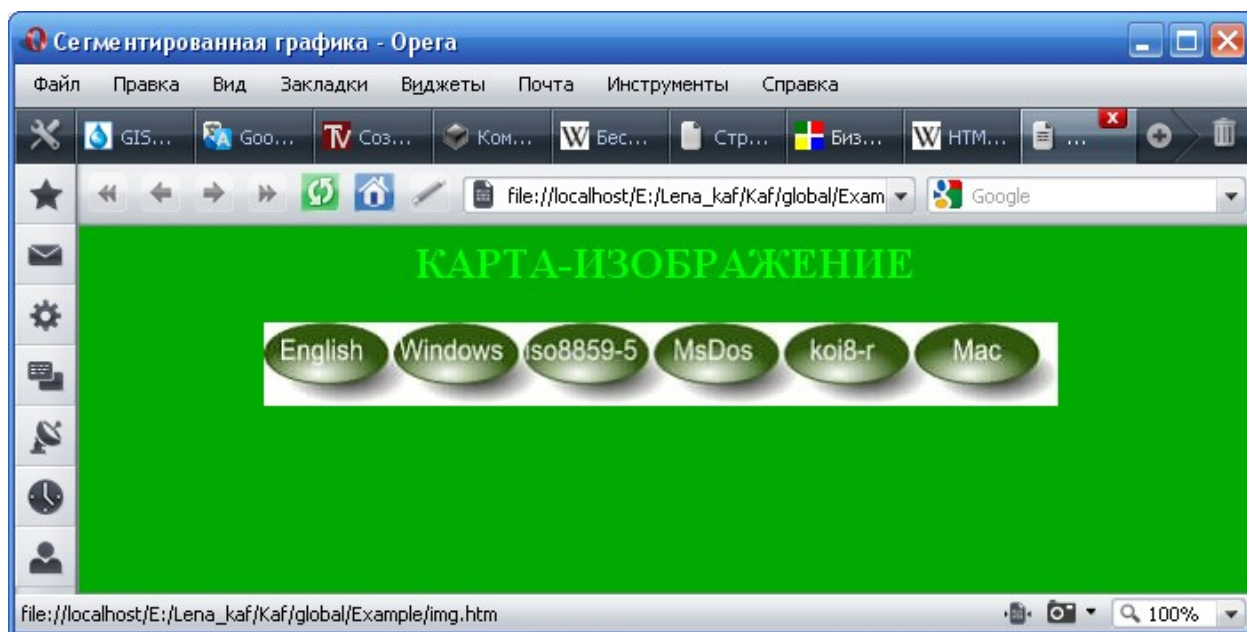


Рис.3. Карта-изображение

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Сегментированная графика</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR=#00AA00 TEXT=#00FF00 >
```

```
<H2 ALIGN="CENTER">КАРТА-ИЗОБРАЖЕНИЕ</H2>
```

```
<P ALIGN=CENTER>
```

```
<IMG SRC="cp.jpg" BORDER=0 usemap="#search" HEIGHT=47 WIDTH=447  
ISMAP>&nbsp;
```

```

<MAP NAME="search">
<AREA SHAPE="Rect" COORDS="1,1,70,47" HREF="img.htm"
TARGET="_blank">
<AREA SHAPE="Rect" COORDS="70,1,140,47" HREF="hr.htm"
TARGET="_self">
<AREA SHAPE="Rect" COORDS="140,1,210,47" HREF="MARQUEE.htm"
TARGET="_top">
<AREA SHAPE="Rect" COORDS="210,1,280,47" HREF="href.htm"
TARGET="_parent">
<AREA SHAPE="Rect" COORDS="280,1,350,47" HREF="text.htm">
<AREA SHAPE="Rect" COORDS="350,1,420,47" HREF="listm.htm">
</MAP>
</P>
</BODY>
</HTML>

```

На рис. 2. изображение `sr.jpg`, используется в карты с активными прямоугольными областями, которые отсылают пользователя на документы `img.htm`, `hr.htm`, `MARQUEE.htm`, `href.htm`, `text.htm` и `listm.htm`.

3. СТРУКТУРА КУРСОВОЙ РАБОТЫ

Курсовая работа по дисциплине «СЕТЕВЫЕ ТЕХНОЛОГИИ» состоит из пояснительной записки, содержащей текстовый, графическо-иллюстрационный материал и электронной версии (CD-диск) курсовой работы

Структура пояснительной записки курсовой работы такова:

- титульный лист (см. приложение);
- лист-содержание;
- введение;
- Раздел 1. Постановка задачи на проектирование ДИС и КС ...
- Раздел 2. Проектирование ДИС...
- Раздел 3. Технический проект и расчет стоимости КС...;
- вывод;
- список использованных источников;
- приложения.

Лист- содержание включает в себя перечень последовательно расположенных в тексте пояснительной записки частей с их нумерацией, названием и указанием страниц размещения.

Во введении необходимо показать актуальность выбранной темы, сформулировать цель и задачи, решаемые в рамках курсовой работы, объект и предмет исследований.

В выводе обобщаются выводы по разделам, оцениваются результаты выполненной работы, анализируется достижение поставленной цели и выполнение поставленных задач, предлагаются направления возможного использования полученных результатов для принятия эффективных управленческих решений.

Список использованных источников содержит список литературных источников, а также URL-адресов, использованной информации с комментариями о месте нахождения и принадлежности URL-адреса.

В приложения выносятся HTML-код основных страниц, прайс-лист, а также все, что может загромоздить основной текст.

Структура электронной версии такова:

1. Главная – досье предприятия (о фирме).
2. Информация о предлагаемых товарах и услугах..
3. Производственная деятельность (объемы продаж)
4. Финансовые показатели предприятия (план маркетинговых исследований)
5. Перспективы и планы развития
6. Анализ отрасли или ее сегментов

7. Список сайтов конкурентов
8. Карта сайта (или как добраться).
9. Возможное сотрудничество (форма заказа).

3.1 Содержание раздела «Постановка задачи на проектирование ДИС и КС»

В первом разделе **«Постановка задачи на проектирование ДИС и КС»** содержится анализ предметной области и обоснование необходимости проектирования ДИС и КС для выбранной бизнес структуры.

В данном разделе проводится анализ организационной структуры предприятия/подразделения. Приведите состав и характеристики структурных подразделений, например, цеха механообработки, администрации производственного объединения, бухгалтерии, отдела кадров и т.д. Нарисуйте схему взаимосвязей основных структурных подразделений.

КС полезно рассматривать как сложную систему, состоящую из нескольких взаимодействующих слоев. В основании пирамиды лежит слой компьютеров – центров хранения информации. Далее идет транспортная подсистема (рис. 4), обеспечивающая надежную передачу информационных пакетов между компьютерами.



Рис. 4. Иерархия слоев КС

Над транспортной системой работает слой сетевых операционных систем, который организует работу приложений в компьютерах и предоставляет через транспортную систему ресурсы своего компьютера в общее пользование. Над операционной системой работают различные приложения, но из-за особой роли систем управления базами данных, хранящих в упорядоченном виде основную корпоративную информацию и производящих над ней базовые операции поиска, этот класс системных

приложений обычно выделяют в отдельный слой КС. На следующем уровне работают системные сервисы, которые, пользуясь СУБД, как инструментом для поиска нужной информации, предоставляют конечным пользователям эту информацию в удобной для принятия решения форме, а также выполняют некоторые общие для предприятий всех типов процедуры обработки информации. К этим сервисам относится служба WorldWideWeb, система электронной почты, системы коллективной работы и многие другие.

И, наконец, верхний уровень корпоративной сети представляют специальные программные системы, которые выполняют задачи, специфические для данного предприятия или предприятий данного типа. Примерами таких систем могут служить системы автоматизации банка, организации бухгалтерского учета, автоматизированного проектирования, управления технологическими процессами и т.п.

Дайте характеристику задач решаемых в рамках каждого структурного подразделения. Определите роль информационных технологий в решении выделенных функций и задач, и проведите анализ существующей технологии обработки информации в рамках выделенных функций и задач. Определите направления для совершенствования технологии обработки информации в результате внедрения КС и разработки ДИС. Проведите анализ необходимости решения задач на выбранном предприятии таких как: хранение данных на сервере, электронная почта, просмотр интернет страниц, передача файлов большого объёма, использование офисных приложений работа с графическими файлами, IP телефония, базы данных, системы автоматизированного проектирования, издательские системы, проигрывание аудио и видео, приём радио и телепередач по сети, участие в аудио и видео конференциях (аналог селекторных совещаний). Необходимо также проанализировать преимущества, которые может получить предприятие при размещении информации о себе в сети Интернет.

Приведите план зданий и помещений с указанием метража и отмеченными на нем местоположениями существующих компьютеров и сделайте выводы и необходимости проектирования КС и разработке ДИС. Проведите аналитический обзор веб-сайтов конкурентов, на основании которого разработайте и приведите макет ДИС, содержащий информацию «Где я?» - полоса логотипа (1); «Куда я могу пойти?» - полоса навигации (2); «Что здесь?» - холст страницы (3). Один из возможных макетов показан на рисунке 5.

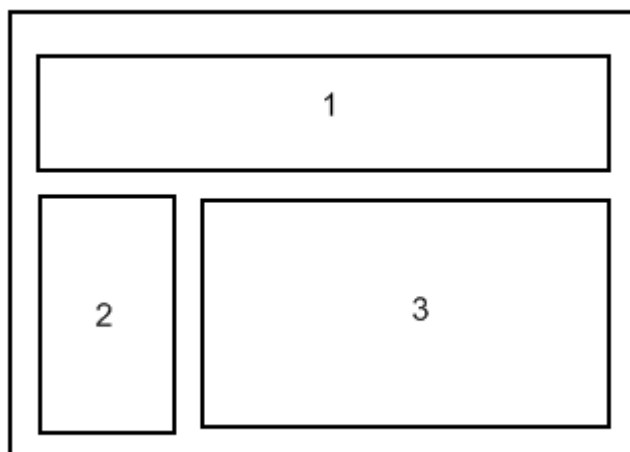


Рис .5. Макет ДИС

3.2 Содержание раздела «Проектирование ДИС»

В втором разделе **«Проектирование дистанционной информационной системы»** приводится структурная схема веб-интерфейса пользователя ДИС с описанием последовательности действий и/или логики решения задачи. Далее приводятся экранные формы отдельных страниц и описание основных тэгов HTML, которые использовались при их разработке со ссылками на полный текст источника (в виде HTML), который помещается в приложение. Анализируются следующие характеристики веб-разработки:

- основные технические (скорость соединения с Интернет, характеристики монитора, возможности просмотра графики, версия и настройки браузера, версия операционной системы, местоположение принимающего компьютера, модификация "мышки" и т.д.),
- социальные (образование, должность, материальное положение, вероисповедание посетителя),
- психологические (мотивация, задачи и ожидания, личностные способности, психологическое состояние, самочувствие, степень усталости посетителя, а также соответствие контента сайта поисковым ключевым словам), физиологические (зрение, возраст, пол посетителя),
- другие дополнительные факторы (опыт, время суток, дни недели, режимы просмотра сайта, кто оплачивает Интернет, групповой или индивидуальный просмотр)

Даются рекомендации о размещении ДИС на хостинговой площадке и приводится оценка стоимости такого размещения.

Приводятся рекомендации по регистрации доменного имени, оценивается стоимость покупки предлагаемого доменного имени.

В заключении производится расчет трудоемкости проектирования ДИС.

Таблица 2 "Трудоемкость этапов разработки ДИС"

№ этапа	Наименование этапа	Содержание работ	Т (календарные дни)
1	2	3	6
1	Технико-экономическое обоснование необходимости разработки.		
2	Анализ веб-сайтов конкурентов		
3	Эскизный проект (разработка макетов)		
4	Технический проект (ТП)		
6	Рабочий проект (РП)		
7	Внедрение проекта (ВП)		
8	Итого:		

3.3 Содержание раздела «Технический проект и расчет стоимости КС»

В третьем разделе «**Технический проект и расчет стоимости КС**» необходимо проанализировать характеристики и факторы, влияющие на выбор комплекса программно-аппаратных средств КС и проектирование соответствующей конфигурации:

1. характеристики среды передачи информации или кабельной системы, такие как: помехозащищенность, защита от климатических воздействий, протяженность без промежуточного усиления сигнала, стоимость приобретения и установки;
2. максимальная протяженность сети;
3. предполагаемое количество конечных систем;
4. основная сфера применения (на производственном предприятии, в учреждении или в учебной сфере);
5. функциональное назначение, то есть классы решаемых задач (научная деятельность, образование, резервирование мест, удаленный ввод/вывод,

"распределенная обработка данных, управление и учет, финансовые операции);

6. тип передаваемой информации (данные, изображения, речь);
7. оценка пропускной способности сети;
8. сетевое программное обеспечение;
9. интерсетевое обеспечение (необходима ли связь с другими сетями ЭВМ);
10. показатель надежности сети в целом и отдельных ее частей;

Все перечисленные характеристики должны быть связаны с перечисленными в первом разделе задачами и планом зданий и помещений. с отмеченными на нем местоположениями существующих компьютеров

Далее необходимо:

1. Выбрать топологию КС (и обосновать выбор).
2. Нарисовать функциональную схему ЛВС и составить перечень аппаратных средств.
3. Выбрать оптимальную конфигурацию ЛВС.
4. Произвести ориентировочную трассировку кабельной сети и выполнить расчет длины кабельного соединения для выбранной топологии с учетом переходов между этажами. Поскольку существуют ограничения на максимальную длину одного сегмента локальной сети для определенного типа кабеля и заданного количества рабочих станции, требуется установить необходимость использования повторителей.
5. Рассчитать надежность ЛВС.
6. Рассчитать стоимостную оценку предлагаемого проектного решения по конфигурации КС. Для расчетов следует воспользоваться любым доступным прайс-листом, который следует привести в Приложении.

4. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

Требования к оформлению пояснительной записки

При оформлении пояснительной записки курсовой работы необходимо соблюдать требования, изложенные в методических указаниях по стандартизации оформления дипломного и курсового проектирования для студентов специальности «экономическая кибернетика».

Изменение структуры основных разделов возможно только по согласованию с преподавателем.

Требования к оформлению электронной версии

Электронная версия курсовой работы чтобы подтвердить умение студента разрабатывать и размещать коммерческую информацию в сети Internet.

1. Все Web-страницы должны быть разработаны согласно структуре и связаны между собой гиперссылками;
2. На всех страницах должен присутствовать фон в виде фонового изображения.
3. Название фирмы должно быть оформлено в виде рисунка.
4. Макет всех страниц должен содержать информацию «Где я?» - полоса логотипа; «Куда я могу пойти?» - полоса навигации; «Что здесь?» - холст страницы.
5. Информация о товарах и услугах должна быть оформлена с помощью табличных средств.
6. Информацию о возможном сотрудничестве следует заносить в подготовленную форму.
7. Одна из страниц должна содержать карту-изображение.

5. ТЕМАТИКА КУРСОВОЙ РАБОТЫ

Общая часть темы: «Использование сетевых технологий при проектировании дистанционной информационной системы и компьютерной сети ...»

Варианты заданий:

1. Финансовое предприятие (банк)
2. Коммерческое предприятие (магазин).
3. Консалтинговая фирма
4. Косметическая фирма.
5. Косметический салон
6. Фитнес клуб
7. Рекламное агентство.
8. Предприятие, оказывающее транспортные и складские услуги
9. Строительное предприятие
10. Популярная газета.
11. Популярный журнал
12. Популярный телеканал.
13. Учебное заведение
14. Лечебный центр
15. Городская библиотека
16. Городской музей.
17. Туристическая фирма
18. Агентство по продаже недвижимости.
19. Торгово-закупочная фирма
20. Полиграфическая фирма.
21. Книжное издательство
22. Интернет-провайдер.
23. Научно-исследовательское учреждение.

6. ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

№ п/п	Этапы работы	Срок выполнения (номер учебной недели)
1.	Закрепление студентов за руководителями, выдача и согласование индивидуального задания	1
2.	Согласование плана работы, предварительный поиск информации и оформление введения	2
3.	Разработка основных разделов курсовой работы оформление выводов и списка использованных источников:	3-14
3.1.	Разработка и оформление 1-го раздела	3-5
3.2.	Разработка и оформление 2-го раздела	6-9
3.3.	Разработка и оформление 3-го раздела	10-14
4.	Сдача курсовой работы на проверку	15
5.	Защита курсовой работы	16

7. СПИСОК РЕКОМЕНДОВАННЫХ ИСТОЧНИКОВ

1. Олифер В.Г., Олифер Н.А. Компьютерные сети Принципы, технологии, протоколы. [текст]/ /СПб: Питер,2001.-672с.:ил.
2. Олифер В.Г., Олифер Н.А. Компьютерные сети Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. [текст]/ /СПб: Питер,2010.-944с.:ил.
3. Примеры технических заданий на разработку веб-сайтов/ Профессиональная веб-студия Antula [Электронный ресурс] – Режим доступа: <http://www.antula.ru/>
4. Стратегическое планирование сетей масштаба предприятия. Центр Информационных Технологий, 2001/ Учебное пособие под ред Н.А. Олифер, В.Г. Олифер, П.Б. Храмцов, В.И. Артемьев, С.Д. Кузнецов [Электронный ресурс] – Режим доступа: <http://citforum.ru/nets/spsmp/index.shtml>
5. Компьютерные сети и технологии. Сетевые службы и протоколы. [Электронный ресурс] – Режим доступа: <http://www.xnets.ru/plugins/content/content.php?content.103>.
6. Википедия. Беспроводные компьютерные сети [Электронный ресурс] – Режим доступа: [http://ru.wikipedia.org/wiki/ Беспроводные_компьютерные_сети](http://ru.wikipedia.org/wiki/Беспроводные_компьютерные_сети)
7. Владимир Ткаченко Создание веб-страниц и сайта. Типы сайтов: статические, динамические, Flash и комбинированные сайты. [Электронный ресурс] – Режим доступа <http://www.lessons-tva.info/>
8. Обучение в интернет. Бесплатное дистанционное обучение информатике, телекоммуникациям, основам электронного бизнеса. Модуль 1: Локальные вычислительные сети [Электронный ресурс] – Режим доступа: http://www.lessons-tva.info/edu/telecom-loc/m1t2_2loc.html

9. Исаев Г.Н. Информационные системы в экономике: учебное пособие. – [текст] М.: Омега-Л, 2006.
10. Полак-Брагинский А.В. Сеть своими руками. [текст] – СПб.: БХВ-Петербург, 2003.- 320с.:ил.
11. Система автоматизированного проектирования компьютерных сетей NETWIZARD [Электронный ресурс] – Режим доступа: www.netwizard.ru.
12. Музей средств связи . История развития сети Интернет. [Электронный ресурс] – Режим доступа: <http://www.nntu.ru/RUS/fakyl/KAFEDR/TCIT/musey/inte.htm#9>
13. Вин Дж. Искусство web-дизайна. Самоучитель. [текст]– СПб.: Питер, 2002.-224с.:ил.
14. Фролов А.В., Фролов Г.В. Сервер Web своими руками. Язык HTML, приложения CGI и ISAPI, установка серверов Web. [текст] – М.: ДИАЛОГ-МИФИ, 1997.-288с.- (Библиотека системного программиста; Т.29).

МИНИСТЕРСТВО ОБРАЗОВАНИЯ УКРАИНЫ
ОДЕССКИЙ НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра «Информационные системы в менеджменте»

КУРСОВАЯ РАБОТА

по дисциплине «СЕТЕВЫЕ ТЕХНОЛОГИИ»

Тема: «Использование сетевых технологий при проектировании
дистанционной информационной системы и компьютерной сети ...»

ИСПОЛНИТЕЛЬ: ст. гр. «_____»

«_____»

(Ф.И.О.)

РУКОВОДИТЕЛЬ: доц. каф

«_____»

(Ф.И.О.)

Одесса -2010г.