

Федеральное агентство по образованию Российской Федерации (РФ)

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра Электронных приборов (ЭП)

УТВЕРЖДАЮ

Заведующий кафедрой ЭП

_____ С.М. Шандаров

Теория информации и кодирование

Учебно-методическое пособие

Методические указания к лабораторным работам

Разработчик:

ст. преподаватель каф. ЭП

_____ Е.С. Шандаров

Томск 2010

Шандаров Е.С.

Теория информации и кодирование: Учебно-методическое пособие. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2010. - 40 с.

Оглавление

Введение.....	4
Инструменты необходимые для выполнения лабораторных работ.....	5
Лабораторная работа №1. Исследование статистических характеристик текстового документа.....	7
Лабораторная работа №2. Работа с кодовыми таблицами русского языка.....	9
Лабораторная работа №3. Основы криптографии.....	12
Лабораторная работа №4. Изучение языка гипертекстовой разметки HTML.....	14

Введение

Данный курс лабораторных работ посвящен теме «Теория информации и кодирование» и включает в себя описание 4 лабораторных работ.

Лабораторные работы по курсу проводятся с использованием программного обеспечения бесплатно распространяемого пакета OpenOffice.org и программного продукта FreePascal, также бесплатно-распространяемого.

В рамках данного курса студенты создают программы на языке Pascal для реализации базовых методов теории информации и кодирования.

Инструменты необходимые для выполнения лабораторных работ

Free Pascal (FPC) компилятор с открытыми исходными кодами с двумя важными особенностями: высокая совместимость с Delphi и доступность для большого количества платформ, включая Windows, Mac OS X, и Linux.

Совместимость Free Pascal с Delphi включает не только поддержку того же языка программирования Object Pascal который использует Delphi, но также библиотеки конструкции и классы, которые "понимает" Delphi, например System, SysUtils, StrUtils, DateUtils, Classes, Variants, Math, IniFiles and Registry, которые включены в Free Pascal и поддерживаются всеми платформами. Free Pascal также включает модули Windows, ShellAPI, BaseUnix, Unix и DynLibs для доступа к специфичным для конкретной ОС функциям. Около дюжины модулей составляют ядро обычно упоминающееся как Free Pascal run-time library (RTL).

Консольные приложения не имеют графического интерфейса (GUI). Обычно они запускаются в консоли, читают информацию с консоли и выводят результаты на консоль. В Windows консоли обычно соответствует окно командной строки. В OS X и Linux консоли соответствует окно терминала. Консольные приложения это, как правило, небольшие утилиты, такие как программа FC (file compare - сравнение файлов) в Windows или команды cd и cp в Unix. Консольные приложения могут быть также мощными программами для тестирования, моделирования или управления данными, которые не нуждаются в причудливом GUI, потому как они запускаются другими программами или из командных файлов. Компилятор Free Pascal и прилагающийся набор утилит являются консольными приложениями, то есть могут быть запущены в консоли, из командных файлов или из IDE Lazarus.

Для создания консольного приложения необходим только текстовый редактор и компилятор FreePascal. Вам не нужно использовать Lazarus для создания консольных приложений. Однако, если вы предпочитаете работать в инте-

гированном окружении, то вы можете создать проект для консольного приложения, редактировать и компилировать код в IDE Lazarus.

Лабораторная работа №1. Исследование статистических характеристик текстового документа

Цель работы

Исследовать с точки зрения статистики некий осмысленный текстовый документ. Освоение инструментов Pascal для работы с текстовыми файлами, сортировка массивов.

Задание на лабораторную работу

Исследовать статистические характеристики текстового документа. Получить таблицу частоты использования символов алфавита отсортированную по убыванию частоты.

Порядок выполнения работы

Определитесь с текстовым документом, характеристики которого вы будете исследовать. Требования такие:

- документ на английском языке
- размер текстового файла - 20-60кБ
- документ должен быть осмысленным

В программной оболочке FreePascal создайте программу со следующим функционалом:

- на вход программы поступает текстовый файл
- в результате обработки программа формирует массив из двух элементов: символ или его ASCII-код и количество появлений данного символа в тексте

- производится сортировка массива любым из доступных вам методов
- на экран выводится таблица состоящая из следующих столбцов: код символа, сам символ в кавычках, количество появлений в тексте, частота нормированная к 1

В отчет необходимо поместить таблицу с частотой появления символов в тексте.

Содержание отчета

Отчет по проделанной работе готовится в текстовом редакторе OpenOffice.org Write и предоставляется для проверки в электронном виде в формате электронных документов PDF.

Отчет должен состоять из следующих частей:

- введение
- постановка задачи
- основная часть
- заключение
- приложение

В приложении приводится исходный код программы на языке Pascal.

Лабораторная работа №2. Работа с кодовыми таблицами русского языка

Введение. Кодировка букв русского алфавита

В настоящее время наиболее широко используются пять (!) различных таблиц кодировки для формального представления русских букв:

- I. ISO 8859-5 — международный стандарт;
- II. Кодовая страница 866 (Microsoft CP866) — используется в MS-DOS;
- III. Кодовая страница 1251 (Microsoft CP1251) для Microsoft Windows;
- IV. На базе ГОСТ КОИ-8, koi8-r — применяется в мире Unix;
- V. Unicode — используется в Microsoft Windows, Unix и клонах Unix.

Основная кодировка ГОСТ (государственный стандарт СССР) от 1987 года создана на основе рекомендаций ISO и в дальнейшем стала основой для представления знаков русских букв в Unicode. В ней и в кодировках II, III и V все буквы кроме ё и Ё расположены в алфавитном порядке. На практике эту кодировку можно встретить только на старых IBM PC совместимых компьютерах ЕС-1840 и в некоторых принтерах. Internet браузеры обычно поддерживают ее наряду с кодировками II–IV.

Кодировка CP866, разработанная на основе альтернативной кодировки ГОСТ, создана специально для ОС MS-DOS, в которой часто используются символы псевдографики. В этой кодировке эти символы имеют те же коды, что и в стандартном IBM PC совместимом компьютере.

Альтернативная кодировка ГОСТ, которая имеет два варианта, совпадает с CP866 по позициям для букв русского алфавита и знакам псевдографики. Основная кодировка ГОСТ совпадает с ISO 8859-5 только по всем знакам рус-

ских букв, кроме заглавной буквы Ё. Использование CP1251 обусловлено почти исключительно влиянием на компьютерные технологии разработок фирмы Microsoft. В ней наиболее полно по сравнению с I, II, IV представлены такие символы как (с), №, различные виды кавычек и тире и т. п.

Кодировка koі8-r основана на стандартах по обмену информацией, используемых на компьютерах под управлением ОС Unix, CP/M и некоторых других с середины 1970-х. В 1993 она стандартизирована в Internet документом RFC1489.

Кодировка Unicode опирается на каталог символов UCS (Universal Character Set) стандарта ISO 10646. UCS может содержать до 231 различных знаков. Коды UCS-2 — 2-байтные, UCS-4 — 4-байтные. Используются также коды переменной длины UTF-8 (Unicode Transfer Format) — 1–6-байтные, наиболее совместимые с ASCII, и UTF-16 — 2 или 4-байтные. Unicode в прикладных программах реализуется лишь частично, и в полном объеме пока нигде не поддерживается. В Linux используется UTF-8.

Достаточно широко используется кодирование на основе ASCII: VI. На базе КОИ-7 — можно использовать при отсутствии кириллических шрифтов, код получается вычитанием 128 от соответствующего кода в koі8-r, что, как правило, дает код латинской буквы, близкой фонетически к русской. В кодировке VI нет видимого символа для для Ђ.

Кроме перечисленных можно встретить еще используемую до введения кодировок ГОСТ болгарскую кодировку, называемую также MIC, Interprog или “старый вариант ВЦ АН СССР”. На компьютерах под управлением Macintosh OS используется также своя собственная таблица кодировки для русских букв, по своему набору знаков почти совпадающая с CP1251.

Цель работы

Познакомиться с разнообразием кодовых таблиц для русских символов, научиться производить конвертацию текста из одной кодировки в другую.

Задание на лабораторную работу

Исходный текст - файл с русским текстом. Кодировка файла - CP1251. Произвести кодирование текста в кодировки KOI8-R, ISO 8859-5 , Microsoft CP866 , Unicode .

Порядок выполнения работы

Выберите исходный файл на русском языке, может содержать цифры, знаки и буквы латинского алфавита. Кодировка файла CP1251. Размер файла - от 200 байт до 1 килобайта. Имя файла: <filename>.cp1251

На языке Pascal написать программу которая производит преобразование файла в другие кодировки. Выбор кодировки производится с помощью меню. Возможен также выбор кодировки с помощью ключей командной строки, например:

```
transcode -koi8r <filename>
```

```
transcode -cp866 <filename>
```

Результат работы программы записывается в файлы с именами: <filename>.cp866, <filename>.koi8r и т.д.

Проверку правильности преобразования можно произвести с помощью программы kwrite.

Кодовые таблицы можно найти либо в учебном пособии, в разделе приложений, либо в интернет.

По окончании работы необходимо подготовить отчет.

Лабораторная работа №3. Основы криптографии

Цель работы

Изучить базовые методы криптографической защиты документов. Создать программы для шифрования документов несколькими способами.

Задание на лабораторную работу

Исходный текст - текстовый файл на английском или русском языках (только однобайтовая кодировка, например CP1251). Необходимо на языке Pascal создать следующие программы:

- шифрование/дешифрование простой заменой или подстановкой
- кодирование/декодирование шифром-перестановкой
- шифрование/дешифрование с ключевым словом.

Все программы должны получать имя файла с исходным текстом в командной строке. Для работы с командной строкой воспользуйтесь функцией ParamStr() языка Pascal.

Порядок выполнения работы

Выберите исходный файл на русском или английском языках, может содержать цифры, знаки и буквы латинского алфавита. Кодировка файла CP1251. Размер файла - от 200 байт до 1 килобайта. Имя файла: <filename>.src

На языке Pascal написать программы которая производит шифрование/дешифрование файла. Выбор режима работы программы производится с помощью меню. Возможен также выбор режима работы с помощью ключей командной строки, например:

```
cyfer -encode <filename>
```

```
cyfer -decode <filename>
```

Результат работы программ записывается в файлы с именами: <filename>.dst.

По окончании работы необходимо подготовить отчет. В отчете привести тексты как исходного сообщения, так и закодированного шифром.

Лабораторная работа №4. Изучение языка гипертекстовой разметки HTML

Введение

HTML был изобретён в 1990 году учёным Тимом Бёрнсом-Ли (Tim Berners-Lee) и предназначался для облегчения обмена документами между учёными различных университетов. Проект имел больший успех, чем Tim Berners-Lee мог ожидать. Этим изобретением HTML он заложил основы современной сети Internet.

HTML это язык, который позволяет представлять информацию (например, научные исследования) в Internet. То, что вы видите при просмотре страницы в Internet, это интерпретация вашим браузером HTML-текста. Чтобы увидеть HTML-коды страницы в Internet, щёлкните "View" в линейке меню вашего браузера и выберите "Source".

Структура HTML-документа

Для того, чтобы текстовый файл превратился в HTML-файл, поменять его расширение с ".txt" на ".html" недостаточно. Надо соблюсти "правило первой строки":

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, обязан начинаться со строки декларации версии HTML !DOCTYPE, которая обычно выглядит так:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

Эта строка поможет браузеру определить, как правильно интерпретировать данный документ. В данном случае мы говорим браузеру, что HTML соответствует международной спецификации версии 3.2, которая хоть и не отличается новизной, но, в отличие от более поздних версий, является полноценным,

широко распространенным стандартом без каких-либо неопределенностей.

После объявления версии и типа документа необходимо обозначить его начало и конец. Это делается с помощью тега-контейнера `<html>`. Необходимо отметить, что любой HTML-документ открывается тегом `<html>` и им же закрывается.

Затем, между тегами `<html>` и `</html>` следует разместить заголовок и тело документа. Вот как должен выглядеть ваш базовый HTML-файл перед началом работы:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

```
<html>
```

```
<head>
```

```
<title>Заголовок документа</title>
```

```
</head>
```

```
<body>
```

```
Текст документа
```

```
</body>
```

```
</html>
```

Из схемы видно, что документ состоит из двух основных блоков – "заголовка" и "тела документа". Заголовок определяется с помощью элемента HEAD, а тело – элементом BODY.

Заголовок содержит "техническую" информацию о документе, хотя чаще всего используется только для обозначения его названия (см. элемент TITLE).

Тело документа – святая святых. Именно в нем находится все то, что отображается на странице: текст, картинки, таблицы. Соответственно, делаем вывод: большинство ваших HTML-экспериментов будет проводиться в пространстве между тегами `<BODY>` и `</BODY>`.

Заголовок HTML-документа

Заголовок документа создается с помощью элемента HEAD, между тегами которого размещаются элементы, содержащие техническую информацию о документе. Заголовок обычно располагается до тела документа (см. структуру HTML-документа).

Элементы, относящиеся к заголовку документа:

- HEAD Определяет начало и конец заголовка документа
- TITLE Определяет имя всего документа, которое отображается в заголовке окна браузера
- BASE Определяет базовый адрес, от которого отсчитываются относительные ссылки внутри документа
- STYLE Используется для вставки в документ таблицы стилей CSS
- LINK Описывает взаимосвязь документа с другими объектами
- META Используется для вставки метаданных

HEAD

Определяет начало и конец заголовка документа. Является контейнером для элементов, содержащих техническую информацию о документе. (TITLE,BASE,STYLE,LINK,META).

Пример:

```
<HTML>
<!-- Начинаем заголовок... -->
<HEAD>
<title>Справочник по HTML</title>
</HEAD>
<!-- ...кончили. Дальше пошло тело документа -->
<BODY>
```

Текст документа

</BODY>

</HTML>

TITLE

Определяет имя всего документа. Имя, как правило, отображается в заголовке окна браузера. Данный элемент обязателен для любого HTML-документа и может быть указан не более одного раза.

Пример:

...

<HEAD>

<TITLE>Руководство по эксплуатации</TITLE>

</HEAD>

BODY

Указывает начало и конец тела HTML-документа. Между начальным и конечным тегами содержится текст документа, изображения и таблицы. Одним словом, все HTML-элементы, отвечающие за отображение документа, управление им и гипертекстовые ссылки. Элемент BODY должен встречаться в документе не более одного раза.

Атрибуты:

MARGINHEIGHT – определяет ширину (в пикселах) верхнего и нижнего полей документа. Работает только в браузерах Netscape.

TOPMARGIN – определяет ширину (в пикселах) верхнего и нижнего полей документа. Работает только в браузерах Internet Explorer.

MARGINWIDTH – определяет ширину (в пикселах) левого и правого полей документа. Работает только в браузерах Netscape.

LEFTMARGIN – определяет ширину (в пикселах) левого и правого полей

документа. Работает только в браузерах Internet Explorer.

BACKGROUND – определяет изображение для "заливки" фона. Значение задается в виде полного URL или имени файла с картинкой в формате GIF или JPG.

BGCOLOR – определяет цвет фона документа.

TEXT – определяет цвет текста в документе.

LINK – определяет цвет гиперссылок в документе.

ALINK – определяет цвет подсветки гиперссылок в момент нажатия.

VLINK – определяет цвет гиперссылок на документы, которые вы уже просмотрели.

Значения атрибутов BGCOLOR, TEXT, LINK, ALINK и VLINK задаются либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

Замечания: несмотря на то что с помощью атрибутов тега BODY можно задать много полезных параметров HTML документа, в настоящий момент целесообразнее использовать CSS – каскадные таблицы стилей.

Пример:

```
<HTML>
```

```
<BODY      BACKGROUND="images/bricks.jpg"      BGCOLOR="#202020"  
TEXT="#FFFFFF" LINK="#FF0000" VLINK="#505050" MARGINHEIGHT="30"  
TOPMARGIN="30" LEFTMARGIN="40" MARGINWIDTH="40">
```

...

Текст документа.

...

```
</BODY>
```

```
</HTML>
```

Гиперссылки

Ссылки на другие документы в HTML создаются либо с помощью элемента А, либо с помощью навигационных карт.

Элемент А применяется, если ссылкой планируется сделать часть текста или целое изображение. Навигационные карты имеет смысл применять, если ссылкой будет часть изображения.

А

Самый необходимый элемент, без которого Интернет просто немислим. Используется для создания и использования гипертекстовых ссылок.

Атрибуты:

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 латиницей 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`, а линк `В конец документа` – в раздел "bottom" текущего документа. (см. Пример 1)

`TARGET` – определяет окно (фрейм), на которое указывает гипертекстовая ссылка. Этот атрибут используется только совместно с атрибутом `HREF`. В качестве значения необходимо задать либо имя одного из существующих фреймов, либо одно из следующих зарезервированных имен:

`_self` – указывает, что определенный в атрибуте `HREF` документ должен отображаться в текущем фрейме;

`_parent` – указывает, что документ должен отображаться во фрейме-родителе текущего фрейма. Иначе говоря, `_parent` ссылается на окно, содержащее `FRAMESET`, включающий текущий фрейм;

`_top` – указывает, что документ должен отображаться в окне-родителе всей текущей фреймовой структуры;

`_blank` – указывает, что документ должен отображаться в новом окне.

Пример 1:

```
<!-- Использование атрибута NAME: -->
```

```
<A NAME="history">История бодибилдинга</A>
```

```
...
```

```
<A NAME="now">Спорт глазами современника</A>
```

...

Вернуться к разделуистории

Пример 2:

<!-- Использование атрибута HREF: -->

FTP-site

Русский проект по OpenGL

...

Пример 3:

<!-- Создадим ссылку для письма с указанием кучи атрибутов -->

<A HREF="mailto:green@igf.ru?subject=Приглашение
&cc=bg@microsoft.com&body=Приезжай на вечеринку.">

Отправить приглашение .

<!-- или просто письмо : -->

авторам.

Текстовые блоки

В этом разделе описаны элементы, разбивающие текст документа на блоки тем или иным способом. Типичными примерами текстовых блоков являются параграфы, абзацы и главы. Для отделения одной части текста от другой также используются разделительные горизонтальные линии и символы возврата каретки.

Элементы:

H1,H2,...H6 Используются для создания заголовков текста

P Используется для разметки параграфов.

DIV Отделяет блок HTML-документа от остальной его части

ADDRESS Оформляет текст как почтовый адрес

BLOCKQUOTE Оформляет текст как цитату

BR Осуществляет перевод строки

HR Вставляет в текст горизонтальную разделительную линию.

PRE Включает в документ (моноширинным шрифтом) блок предварительно отформатированного текста

H1,H2,...H6

Используются для создания заголовков текста. Существует шесть уровней заголовков, различающихся величиной шрифта. С их помощью можно разбивать текст на смысловые уровни – разделы и подразделы.

Атрибуты:

ALIGN – определяет способ выравнивания заголовка по горизонтали.

Возможные значения: left, right, center.

Пример:

<H1 ALIGN="center">Самый большой заголовок посередине</H1>

<H2>Заголовок поменьше</H2>

...

<H6>Малюююсенький такой заголовочек</H6>

P

Используется для разметки параграфов.

Атрибуты:

ALIGN – определяет способ горизонтального выравнивания параграфа.

Возможные значения: left, center, right. По умолчанию имеет значение left.

Пример:

```
<P ALIGN="center">Это центрированный параграф.<BR>
```

Текст располагается в центре окна браузера</P>

```
<P ALIGN="right">А это параграф, выровненный по правому краю.</P>
```

DIV

Используется для логического выделения блока HTML-документа. Элемент группировки, как и элемент SPAN. В современном сайтостроении используется как удобный контейнер для объектов страницы, которым легко динамически манипулировать – перемещать, включать/выключать, создавать слои, регулировать отступы и т.п.

В браузеронезависимой вёрстке обычно используется для выравнивания блока html-кода в окне браузера.

Находящиеся между начальным и конечным тегами текст или HTML-элементы по умолчанию оформляются как отдельный параграф.

Атрибуты:

ALIGN – определяет выравнивание содержимого элемента DIV. Атрибут может принимать значения: left, right, center.

Пример:

...Текст документа...

```
<DIV ALIGN="center">
```

...Текст, таблицы, изображения. Выравнивание по центру.

```
</DIV>
```

...Текст документа...

ADDRESS

Находящийся между начальным и конечным тегами текст оформляется как почтовый адрес. Чаще всего оформление выражается в выделении строки адреса курсивом.

Пример:

Пишите по следующему адресу:

<ADDRESS>

Москва. ул. Академика Королева, 13

Мурзилке

</ADDRESS>

BLOCKQUOTE

Оформляет находящийся между начальным и конечным тегами текст как цитату. Используется для длинных цитат (в отличие от элемента CITE). Цитируемый текст отображается отдельным абзацем с увеличенным отступом.

Пример:

Редакция журнала "Домосед" выражает благодарность

Бухаресту Магарычу Шницелю за замечательное стихотворение:

<BLOCKQUOTE>

Синели красные ромашки,

Желтели в небе облака,

Синицы в теплый край летели,

К истоку двигалась река.

...

</BLOCKQUOTE>

BR

Данный элемент осуществляет перевод строки, то есть практически аналогичен нажатию Enter в текстовом редакторе. После того, как в браузерах появилась возможность обтекания изображения текстом (см. атрибут ALIGN элемента IMG), понадобился дополнительный атрибут CLEAR. Элемент не имеет конечного тега.

Атрибуты:

CLEAR – указывает на необходимость завершения обтекания изображения текстом. Может принимать следующие значения :

- all – завершить обтекание изображения текстом.
- left – завершить обтекание текстом изображения, выровненного по левому краю.
- right – завершить обтекание текстом изображения, выровненного по правому краю.

Пример:

Первое предложение
Второе предложение на следующей строке

HR

Вставляет в текст горизонтальную разделительную линию.

Атрибуты:

WIDTH – определяет длину линии в пикселах или процентах от ширины окна браузера.

SIZE – определяет толщину линии в пикселах.

ALIGN – определяет выравнивание горизонтальной линии. Атрибут может принимать следующие значения:

- left – выравнивание по левому краю документа.
- right – выравнивание по правому краю документа.
- center – выравнивание по центру документа (используется по умолчанию).

NOSHADE – определяет способ закрашки линии как сплошной. Атрибут является флагом и не требует указания значения. Без данного атрибута линия отображается объемной.

COLOR – определяет цвет линии. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов. Атрибут работает только в Internet Explorer.

Пример:

Текст, разделенный `<HR NOSHADE WIDTH="50%">` сплошной горизонтальной линией.

Форматирование текста

В этом разделе описаны элементы для оформления и смыслового выделения текста – подчеркивания, изменения шрифта, выделения курсивом, цитирования и т.д.

Элементы форматирования текста:

- BASEFONT Определяет основной шрифт, которым должен отображаться текст документа
- FONT Позволяет изменять цвет, размер и тип шрифта текста
- I Выделяет текст курсивом
- EM Используется для смыслового выделения текста (курсивом)

- B Выделяет текст жирным шрифтом
- STRONG Усиленное выделение текста (жирным)
- U Выделяет текст подчеркнутым
- S, STRIKE Выделяет текст перечеркнутым
- BIG Отображает текст увеличенным шрифтом (относительно текущего)
- SMALL Отображает текст уменьшенным шрифтом (относительно текущего)
- SUP Отображает текст со сдвигом вверх (верхний индекс)
- SUB Отображает текст со сдвигом вниз (нижний индекс)
- CODE, SAMP Оформляют текст как формулу или программный код
- TT Отображает текст моноширинным шрифтом
- KBD Выделяет текст, который предлагается набрать на клавиатуре
- VAR Используется для обозначения в тексте переменных
- CITE Оформляет текст как цитату или ссылку на источник

FONT

Позволяет изменять цвет, размер и тип шрифта текста, находящегося между начальным и конечным тегами. Вне тегов и используется шрифт, указанный в элементе BASEFONT.

Атрибуты:

SIZE – определяет размер шрифта. Возможные значения:

- целое число от 1 до 7;
- относительный размер с указанием знака, вычисляется путем сложения с базовым размером, определенным с помощью атрибута SIZE элемента BASEFONT.

COLOR – определяет цвет текста. Задается либо RGB-значением в шест-

надцатиричной системе, либо одним из 16 базовых цветов.

FACE – определяет используемый шрифт.

Пример:

Увеличенный красный шрифт

Моноширинный фиолетовый текст 3 размера

I

Текст, заключенный между начальным и конечным тегами, будет выделен курсивом.

Пример:

Текст с <I>курсивом</I>

EM

Логическое ударение. Используется для смыслового выделения текста, стоящего между начальным и конечным тегами. Результат обычно отображается курсивом. То есть элемент EM по действию практически аналогичен элементу I.

Пример:

Порой в России встречаются действительно талантливые веб-мастера. Но только не друг с другом.

B

Текст, заключенный между начальным и конечным тегами, будет выделен жирным шрифтом.

Пример:

Текст с выделенным словом

Списки

Списки в HTML бывают двух видов: упорядоченные (пронумерованные) и неупорядоченные (непронумерованные). Отличаются они лишь способом оформления. Перед пунктами неупорядоченных списков обычно ставятся символы-буллеты (bullets), например, точки, ромбики и т.п., в то время как пунктам упорядоченных списков предшествуют их номера.

Элементы:

- UL Создает неупорядоченный список
- OL Создает упорядоченный список
- LI Создает пункт меню внутри элементов OL или UL
- MENU, DIR Создает неупорядоченный список, подобный UL
- DL Открывает и закрывает список определений
- DT Создает термин в списке определений внутри элемента DL
- DD Создает определение термина внутри элемента DL

UL

Создает неупорядоченный список. Между начальным и конечным тегами должны присутствовать один или несколько элементов LI, обозначающих отдельные пункты списка.

Пример:

```
<UL>
<LI> Первый пункт списка </LI>
<LI> Второй пункт списка </LI>
<LI> Третий пункт списка </LI>
</UL>
```

OL

Создает упорядоченный список. Между начальным и конечным тегами должны присутствовать один или несколько элементов LI, обозначающих отдельные пункты списка.

Атрибуты:

START – определяет первое число, с которого начинается нумерация пунктов. (только целые числа)

TYPE – определяет стиль нумерации пунктов. Может иметь значения:

- "A" – заглавные буквы A, B, C ...
- "a" – строчные буквы a, b, c ...
- "I" – большие римские числа I, II, III ...
- "i" – маленькие римские числа i, ii, iii ...
- "1" – арабские числа 1, 2, 3 ...

По умолчанию <OL TYPE="1">.

Пример:

```
<OL TYPE="I" START="2">
<LI> Пункт два </LI>
<LI> Пункт три </LI>
<LI> Пункт четыре </LI>
</OL>
```

LI

Создает пункт в списке. Располагается внутри элементов OL или UL.

Атрибуты:

VALUE – изменяет порядок нумерации элементов списка. Используется только если элемент LI находится внутри элемента OL. В качестве значения указывается порядковый номер элемента.

Пример:

```
<OL TYPE="A" START="2">  
<LI> Пункт, озаглавленный буквой B. </LI>  
<LI VALUE="6"> Пункт, озаглавленный буквой F. </LI>  
<LI> Пункт, озаглавленный буквой G. </LI>  
</OL>
```

Объекты

Объекты – это графические и мультимедийные вставки в HTML-документ, такие как картинки, Flash-анимация, Java-апплеты, звуки, музыка, VRML.

Элементы:

- IMG Используется для вставки в HTML изображений
- EMBED Используется для вставки в HTML различных объектов
- NOEMBED Используется, если браузер не поддерживает элемент EMBED
- APPLET Используется для вставки в HTML Java-апплетов

- **PARAM** Используется для передачи параметров Java-программе (см. элемент **APPLET**)

IMG

Используется для вставки изображений в HTML-документ.

Это один из самых популярных элементов, незаменимый инструмент web-дизайнера. Элемент допускает вставку изображений в форматах JPEG (в том числе progressive jpeg) и CompuServe GIF (включая прозрачные и анимированные). Четвертые версии браузеров позволяют также использовать формат PNG, но до тех пор, пока они не устареют, от применения PNG лучше воздержаться.

Элемент **IMG** не имеет конечного тега.

Атрибуты:

SRC – обязательный атрибут. Указывает адрес (URL) файла с изображением.

HEIGHT и **WIDTH** – определяют ширину и высоту изображения соответственно. Если указанные значения не совпадают с реальным размером изображения, изображение масштабируется (порой с заметной потерей качества).

HSPACE и **VSPACE** – определяют отступ картинки (в пикселах) по горизонтали и вертикали от других объектов документа. Просто необходимо при обтекании изображения текстом.

ALIGN – обязательный атрибут. Указывает способ выравнивания изображения в документе. Может принимать следующие значения:

- **left** – выравнивает изображение по левому краю документа. Прилегающий текст обтекает изображение справа.
- **right** – выравнивает изображение по правому краю документа. Прилегающий текст обтекает изображение слева.

- `top` и `texttop` – выравнивают верхнюю кромку изображения с верхней линией текущей текстовой строки.
- `middle` – выравнивает базовую линию текущей текстовой строки с центром изображения.
- `absmiddle` – выравнивает центр текущей текстовой строки с центром изображения.
- `bottom` и `baseline` – выравнивает нижнюю кромку изображения с базовой линией текущей текстовой строки.
- `absbottom` – выравнивает нижнюю кромку изображения с нижней кромкой текущей текстовой строки.

`NAME` – определяет имя изображения, уникальное для данного документа. Вы можете указать любое имя без пробелов с использованием латинских символов и цифр. Имя необходимо, если вы планируете осуществлять доступ к изображению, например, из JavaScript-сценариев.

`ALT` – определяет текст, отображаемый браузером на месте изображения, если браузер не может найти файл с изображением или включен в текстовый режим. В качестве значения задается текст с описанием изображения.

`BORDER` – определяет ширину рамки вокруг изображения в пикселах. Рамка возникает, только если изображение является гипертекстовой ссылкой. В таких случаях значение `BORDER` обычно указывают равным нулю.

`LOWSRC` – указывает адрес (URL) файла с альтернативным изображением более низкого качества (и, соответственно, меньшего объема), чем изображение, указанное в атрибуте `SRC`. Браузеры Netscape, поддерживающие данный атрибут, сначала загрузят картинку из `LOWSRC`, а затем заменят ее картинкой из `SRC`.

`USEMAP` – применяет к изображению навигационную карту (`image map`), заданную элементом `MAP`. В качестве значения задается имя карты с предшествующей ему решеткой. Например, если имя карты – "map1", то ссылка на нее

будет выглядеть как "#map1" (см. Пример 4). Обратите внимание: прописные и строчные буквы в данном атрибуте трактуются браузером как разные.

ISMAP – определяет изображение как навигационную карту (image map), обрабатываемую сервером. Имеет смысл использовать только тогда, когда изображение является гиперссылкой. После клика мышкой на изображении серверу отправляются x,y-координаты нажатия. В зависимости от полученных координат, сервер (при наличии на нем соответствующего программного обеспечения) может показать вам тот или иной документ. Данный атрибут является флагом и не требует присвоения значения.

Пример 1:

```
<IMG src="/img/picture.gif" WIDTH="45" HEIGHT="53" ALT="Рысь"
HSPACE="10" ALIGN="left">
```

 Этот текст обтекает картинку справа и находится от нее на расстоянии 10 пикселей.

Пример 2. Использование изображения в качестве гиперссылки:

```
<A HREF="carlson.html">
<IMG src="/img/button.jpg" WIDTH="70" HEIGHT="30" ALIGN="right"
BORDER="0" ALT="Досье Карлсона">
</A>
```

Для просмотра досье нажмите на кнопку справа.

Пример 3. Использование ISMAP:

```
<A HREF="http://www.igf.ru/bin/imagemaps/map1">
<IMG SRC="map.gif" ISMAP></A>');
```

Пример 4. Использование USEMAP:

```
<IMG src="/img/buttons.jpg" WIDTH="170" HEIGHT="120" ALIGN="middle"
BORDER="0" USEMAP="#ButtonsMap">');
```

Примечания (особо важно):

Золотое правило web-мастера – всегда явно задавать размеры картинки в атрибутах HEIGHT и WIDTH, резервируя тем самым место в окне браузера еще до загрузки изображения. В противном случае документ при загрузке каждой картинки будет заново перерисовываться. А на медленных машинах (или при подключении к сети через модем) это смотрится просто отвратительно.

Второе золотое правило web-мастера: если на картинке изображено что-то разборчивое, нужно описать это словами в атрибуте ALT.

Всегда сразу после ставьте
! А то проблем не миновать – после картинки появится пустое пространство в несколько пикселей. Причём ставьте вплотную, без пробелов:
.

Для завершения обтекания изображения текстом используйте атрибут CLEAR элемента BR.

Значения top и texttop атрибута ALIGN не совсем идентичны, и их использование порой дает разный результат. Попробуйте поэкспериментировать.

Указывайте значения атрибутов HSPACE и VSPACE, даже если вы хотите оставить поля нулевой ширины. Бывает, что некоторые браузеры по умолчанию присваивают им какое-то небольшое значение, не равное нулю.

Таблицы

Элементы для создания таблиц:

Таблицы в HTML формируются нетрадиционным способом – построчно. Сначала с помощью элемента TR необходимо создать ряд таблицы, в который затем элементом TD помещаются ячейки.

Важно:

В HTML таблицы используются не только для отображения таблиц как та-

ковых, но и для дизайна. С помощью таблиц можно создать невидимый "каркас" страницы, помогающий расположить текст и изображения так, как вам нравится.

Элементы для создания таблиц:

- TABLE Создает таблицу
- CAPTION Задает заголовок таблицы
- TR Создает новый ряд (строку) ячеек таблицы
- TD и TH Создает ячейку с данными в текущей строке

TABLE

Элемент для создания таблицы. Обязательно должен иметь начальный и конечный теги. По умолчанию таблица печатается без рамки, а разметка осуществляется автоматически в зависимости от объема содержащейся в ней информации. Ячейки внутри таблицы создаются с помощью элементов TR, TD, TH и CAPTION.

Атрибуты:

ALIGN – определяет способ горизонтального выравнивания таблицы. Возможные значения: left, center, right. Значение по умолчанию – left.

VALIGN – должен определять способ вертикального выравнивания таблицы. Возможные значения: top, bottom, middle.

BORDER – определяет ширину внешней рамки таблицы (в пикселах). При BORDER="0" или при отсутствии этого атрибута рамка отображаться не будет.

CELLPADDING – определяет расстояние (в пикселах) между рамкой каждой ячейки таблицы и содержащимся в ней материалом.

CELLSPACING – определяет расстояние (в пикселах) между границами соседних ячеек.

WIDTH – определяет ширину таблицы. Ширина задается либо в пикселах, либо в процентном отношении к ширине окна браузера. По умолчанию этот атрибут определяется автоматически в зависимости от объема содержащегося в таблице материала.

HEIGHT – определяет высоту таблицы. Высота задается либо в пикселах, либо в процентном отношении к высоте окна браузера. По умолчанию этот атрибут определяется автоматически в зависимости от объема содержащегося в таблице материала.

BGCOLOR – определяет цвет фона ячеек таблицы. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

BACKGROUND – позволяет заполнить фон таблицы рисунком. В качестве значения необходимо указать URL рисунка.

TR

Создает новый ряд (строку) ячеек таблицы. Ячейки в ряду создаются с помощью элементов TD и TH

Атрибуты:

ALIGN – определяет способ горизонтального выравнивания содержимого всех ячеек данного ряда. Возможные значения: left, center, right.

VALIGN – определяет способ вертикального выравнивания содержимого всех ячеек данного ряда. Возможные значения: top, bottom, middle.

BGCOLOR – определяет цвет фона для всех ячеек данного ряда. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

TD и TH

Элемент TD создает ячейку с данными в текущей строке. Элемент TH также создает ячейку, но определяет ее как ячейку-заголовок.

Такое разграничение позволяет браузерам оформлять содержимое ячейки-заголовка и ячеек с данными разными шрифтами. Кроме того, должна улучшиться работа браузеров, использующих речевой интерфейс. В качестве содержимого ячейки можно использовать другие таблицы.

Атрибуты:

ALIGN – определяет способ горизонтального выравнивания содержимого ячейки. Возможные значения: left, center, right. По умолчанию способ выравнивания определяется значением атрибута ALIGN элемента TR. Если же и он не задан, то для TD выполняется выравнивание по левому краю, а для TH – центрирование.

VALIGN – определяет способ вертикального выравнивания содержимого ячейки. Возможные значения: top, bottom, middle. По умолчанию происходит выравнивание по центру (VALIGN="middle"), если значение этого атрибута не было задано ранее в элементе TR.

WIDTH – определяет ширину ячейки. Ширина задается в пикселах или в процентном отношении к ширине таблицы.

HEIGHT – определяет высоту ячейки. Высота задается в пикселах или в процентном отношении к высоте таблицы.

COLSPAN – определяет количество столбцов, на которые простирается данная ячейка. По умолчанию имеет значение 1.

ROWSPAN – определяет количество рядов, на которые простирается данная ячейка. По умолчанию имеет значение 1.

NOWRAP – блокирует автоматический перенос слов в пределах текущей ячейки. Обратите внимания на примечание, касающееся использования данного

атрибута (далее, внизу страницы).

BGCOLOR – определяет цвет фона ячейки. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

BACKGROUND – заполняет ячейку фоновым рисунком. Необходимо указать URL рисунка. Данный атрибут не работает в старых версиях браузера Netscape (до 3.X включительно).

Цель работы

Познакомиться с языком гипертекстовой разметки HTML. Создать и оформить html-документ.

Задание на лабораторную работу

Порядок выполнения работы

Министерство науки и образования РФ

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра ЭЛЕКТРОННЫХ ПРИБОРОВ (ЭП)
дисциплина «Теория информации и кодирование»

ОТЧЕТ

по лабораторной работе

« _____ »

Выполнил студент

гр. 348

XXXXXXXXXXXXXX

Проверил преподаватель
