

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО  
ОБРАЗОВАНИЯ РЕСПУБЛИКИ  
УЗБЕКИСТАН**

**ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ АБУ  
РАЙХАНА БЕРУНИЙ**

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**  
методические указания для выполнения практических  
работ для магистров

**2-ЧАСТЬ**

**ТАШКЕНТ – 2007**

**СОСТАВИТЕЛИ:** Сагатов М.В., Ирмухамедова Р.М., Каримова Д., Джураева Д.Ш.

Методические указания для выполнения практических работ по курсу «Информационные технологии» **СОСТАВИТЕЛИ:** Сагатов М.В., Каримова Д., Ирмухамедова Р.М., Джураева Д., ТГТУ, 2007, стр., - Ташкент.

Данная методическая работа включает в себя работы по следующим темам: автоматизация обработки документов с помощью программ сканирования, перевода, использование специализированных систем MathCad, Matlab, использование поисковых систем в научной практике, создание Web страниц с научными материалами с использованием языка HTML. Методическая разработка предназначена для проведения практических работ по курсу «Информационные технологии» для магистров

Кафедра «Общая информатика»

Напечатано по решению научно-методического совета Ташкентского Государственного Технического университета им. Абу Райхана Беруний(пр№4 от 04.06.2007).

Рецензенты:

1. Гаибназаров Т. - Ташкентский Университет Информационных технологий, к.т.н. ,доцент.
2. Закиров Т.З. – Ташкентский Государственный технический университет , к.т.н.,доц.

## **Практическое занятие № 1**

### **Средства автоматизации обработки данных.**

**Цель занятия:** Изучение методики автоматизации обработки данных, автоматизированного перевода документов.

#### **Теоретическая часть**

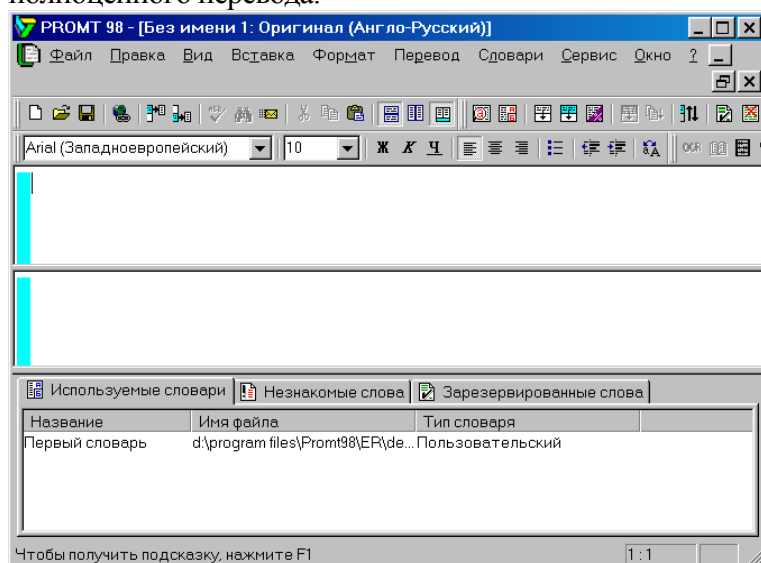
Программа автоматизированного перевода Promt 98 используется для перевода текста с английского языка на русский и с русского на английский.

Для перевода текста после запуска программы вводят текст, подлежащий переводу и, тогда в специальном окне программа выдает текст на другом языке. Надежный и качественный перевод документов с одного языка на другой пока остается недостижимым идеалом. Но современные средства автоматизации перевода позволяют эффективно использовать их на практике перевода именно технического текста, потому что в нем используется ограниченное число языковых конструкций происходит однозначная интерпретация. Большое число технических терминов имеют совершенно определенный смысл в рамках данной научной или технической дисциплины.

Программы автоматического перевода имеет смысл использовать технических текстов в следующих случаях:

- при абсолютном незнании иностранного языка;
- при необходимости получить перевод *быстро*, даже ценой снижения его качества(например, это относится к переводу Web-документов);

- для перевода на иностранный язык (умения читать иноязычные тексты недостаточно, чтобы научиться объясняться на иностранном языке);
- для быстрого создания первоначального черновика («подстрочника»), используемого в ходе подготовки полноценного перевода.



Рабочее окно программы Promt 98.

### Упражнение 1. Автоматический перевод текста.

1. Запустите программу Promt (используйте ярлык программы).
2. Командой **открыть** из меню **файл** выберите документ.
3. Выберите используемый формат файла в диалоговом окне **конвертировать файл** и щелкните на кнопке ОК.
4. Откройте вкладку **Используемые словари** на информационной панели и ознакомьтесь со

списком словарей. В контекстном меню выберите команду **изменить список словарей**.

5. Отключите все словарь кроме одного специализированного по теме документа.
6. Щелкните на кнопке **Весь текст** на панели инструментов **Перевод**.
7. Ознакомьтесь с переводом текста, обратив внимание на вид абзацев.
8. Оцените качество автоматического перевода.
9. Сохраните переведенный текст во внутреннем формате программы переводчика.
10. Сохраните текст в виде документа в **MS Word** и используя русский и английский варианты текста окончательно сформируйте правильный грамотный перевод исходного текста.

## **Упражнение 2. Редактирование словаря.**

1. Запустите программу Promt (используйте ярлык программы).
2. Командой **открыть** из меню **файл** выберите документ.
3. Выберите используемый формат файла в диалоговом окне **конвертировать файл** и щелкните на кнопке **ОК**.
4. Щелкните на кнопке **Весь текст** на панели инструментов **Перевод**.
5. Откройте вкладку **Незнакомые слова** на информационной панели.
6. Просмотрите список слов, незнакомых программе. Попробуйте разыскать их в англо-русском словаре, чтобы выяснить, какие из них действительно требуют перевода.
7. В списке на вкладке **Незнакомые слова** дважды щелкните на незнакомом слове, требующем перевода.

8.В диалоговом окне **Открыть словарную статью** уточните начальную форму слова и щелкните на кнопке ОК.

9. В диалоговом окне **Словарная статья** выберите вкладку с частью речи, которой принадлежит добавляемое слово.

10.Установите переключатели, описывающие свойства слова, и щелкните на кнопке ОК.

11.В диалоговом окне **Перевод** введите начальную форму слова или словосочетания и щелкните на кнопке ОК.

12.При появлении запросов, касающихся дополнительных грамматических форм исходного слова, введите необходимые данные.

13. При необходимости внесите другие слова в пользовательский словарь.(повторите п. 8-12).

14.Повторно щелкните на кнопке **Весь текст** на панели инструментов **Перевод**. Изучите, как изменился перевод в связи с пополнением словаря.

#### Контрольные вопросы:

1. Какие средства автоматизации перевода вы знаете?
2. В чем удобство электронных словарей?
3. Почему так распространен перевод технического текста с помощью электронных словарей?
4. Какие методики используются для перевода текста?

**Практическое занятие №2**  
**Использование спец. систем**  
**MathCad, Matlab**

**в решениях технических задач**

***Цель работы:** ознакомиться с программой MathCAD, изучить ее возможности при решении конкретной задачи, Ознакомиться с возможностями системы Matlab при моделировании технологических процессов, машинной реализации вычисления математических моделей.*

**Теоретическая часть.**

Mathcad является математическим редактором, позволяющим проводить разнообразные научные и инженерные расчеты, начиная от элементарной арифметики, и заканчивая сложными реализациями численных методов. С точки зрения классификации программного обеспечения, пакет Mathcad — типичный представитель класса PSE-приложений. Пользователи Mathcad — это студенты, ученые, инженеры, разнообразные технические специалисты и все, кому приходится проводить математические расчеты. Благодаря простоте применения, наглядности математических действий, обширной библиотеке встроенных функций и численных методов, возможности символьных вычислений, а также превосходному аппарату представления результатов (графики самых разных типов, мощных средств подготовки печатных документов и Web-страниц) Mathcad стал наиболее популярным математическим приложением.

Следует хорошо представлять себе, что в состав Mathcad входят несколько интегрированных между собой компонентов:

- мощный текстовый редактор, позволяющий вводить, редактировать и форматировать как текст, так и математические выражения;
- вычислительный процессор, умеющий проводить расчеты по введенным формулам, используя встроенные численные методы;
- символьный процессор, позволяющий проводить аналитические вычисления и являющийся, фактически, системой искусственного интеллекта;
- огромное хранилище справочной информации, как математической, так и инженерной, оформленной в качестве интерактивной электронной книги.

Отличительной чертой Mathcad от большинства других современных математических приложений является его построение по принципу WYSIWYG ("What You See Is What You Get" — "что вы видите, то и получите"). Поэтому он очень прост *и* использовании, в частности, из-за отсутствия необходимости сначала писать программу, реализующую те или иные математические расчеты, а потом запускать ее на исполнение. Вместо этого достаточно просто вводить математические выражения с помощью встроенного редактора формул, причем в виде, максимально приближенном к общепринятому, и тут же получать результат. Кроме того, можно изготовить на принтере печатную копию документа или создать страницу в Интернете именно в том виде, который этот документ имеет на экране компьютера при работе с Mathcad, либо можно включить документ в структуру электронной книги Mathcad.



## Задачи математического анализа в среде пакета Mathcad

### *Упражнение 1. Определение функций и построение графиков*

Для определения функции одной переменной нужно ввести с клавиатуры имя функции с аргументом в круглых скобках, знак присваивания (для ввода знака присваивания нужно нажать на клавиатуре комбинацию клавиш <Shift>+<:=> или щелкнуть по кнопке <:=> панели **Evaluation**) и справа от него - выражение для вычисления функции.

В записи выражения для функции можно использовать знаки (имена) элементарных функций, вводя их с клавиатуры или вставляя в рабочий документ функцию, выбранную из списка в пункте **Function** меню **Insert**.

Выражение можно вводить с помощью кнопок панели инструментов **Calculator Toolbar**.

Вставить в выражение букву греческого алфавита можно с помощью панели **Greek Symbol Toolbar**

Для вычисления значения функции в точке нужно ввести в рабочий документ с клавиатуры имя функции, указать в скобках значение аргумента, выделить выражение, ввести знак равенства (с помощью соответствующей кнопки панели **Evaluation**) и щелкнуть по свободному месту в рабочем документе.

Инструменты для построения графиков в Mathcad доступны в панели инструментов **Graph Toolbar**, которая открывается щелчком по соответствующей кнопке в панели математических инструментов или

через пункт **Graph** меню **Insert**. Для построения графика функции, заданной в **декартовых координатах**, нужно: щелкнуть по рабочему документу, по пункту по строке **X-Y Plot** в пункте **Graph** меню **Insert** (или по соответствующей кнопке в панели **Graph**); в рабочем документе откроется окно построения графиков; ввести в помеченной позиции возле оси абсцисс имя аргумента, а в позиции возле оси ординат – имя функции и щелкнуть по рабочему документу вне окна графиков. Если нужно построить одновременно графики нескольких функций, нужно ввести их имена в позиции возле оси координат разделяя запятой.

Вместо имени функции можно ввести выражение для ее вычисления. Параметры изображения (цвет и толщина линий, координатная сетка, разметка осей, надписи на графиках и т.д.) можно изменить, щелкнув дважды по полю графика и установив настройки в соответствующих появившихся окнах диалога.

Графики функций, заданных в **параметрической форме**, строятся аналогично, с учетом того, что в позициях аргумента и функции вводятся выражения или имена соответствующих функций параметра.

Для построения графика функции, заданной в **полярных координатах**, нужно: щелкнуть по рабочему документу, по строке **Polar Plot** в пункте **Graph** меню **Insert** (или по соответствующей кнопке в панели **Graph**); в рабочем документе откроется окно построения графиков; ввести в помеченной позиции возле оси абсцисс имя аргумента, а в позиции возле оси ординат - имя функции и щелкнуть по рабочему документу вне окна графиков.

## Упражнение 1. Построение графиков.

$x = 0, 0.5, 2\pi$   $y = 0, 1, 10$

$x =$

0  
0.5  
1  
1.5  
2  
2.5  
3  
3.5  
4  
4.5  
5  
5.5  
6

$y =$

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10

$x = 0, 0.5, 2\pi$

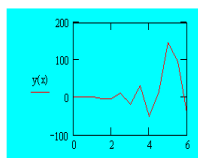
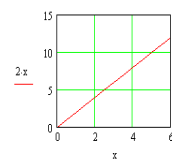
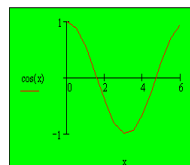
$x =$

0  
0.5  
1  
1.5  
2  
2.5  
3  
3.5  
4  
4.5  
5  
5.5  
6

$y(x) = \cos(x)^2 \exp(x)$

$y(x) =$

1  
1.597  
1.489  
-2.815  
-4.83  
12.176  
-18.301  
31.472  
-52.286  
15.261  
147.108  
96.384  
-51.624



## Упражнение 2. Численное решение уравнений

chislennoe reshenie uravneniy

**x** := 1

**root**(2·sin(**x**) - **x**, **x** = 1.895

**x** := 0      **y** := 0

**Given**

**x** + **y** = 1

**x**<sup>2</sup> + **y**<sup>2</sup> = 4

**Find**(**x**, **y**) =  $\begin{pmatrix} 1.823 \\ -0.823 \end{pmatrix}$

reshenie system **uravneniy**

## Система MatLab

Моделирование электрических и электронных схем в настоящее время невозможно представить без использования вычислительной техники. Существует и разрабатывается множество программных средств для этой цели такие как, MultiSim, MicroCap, SIMetrix, CircuitMaker, ASIMEC, в том числе и программа Simulink программного комплекса Matlab. Использование программы Simulink по сравнению с большинством других программ позволяет моделировать как физические воздействия, так и сами схемные решения с использованием мощных средств моделирования самих приборов. Однако библиотека электрических и электронных компонентов практически

не содержит полупроводниковых приборов, таких как диоды и биполярные транзисторы.

В настоящее время существуют математические модели, достаточно точно описывающие работу полупроводниковых диодов [1].

**Упражнение 1.** Исследование параметров конкретной системы( схемы устройства) с помощью процедур системы MatLab.

Расчёт параметров и исследование функциональных зависимостей средствами MathCad.

Выберем в Simulink блок "Matlab function" и в параметрах пропишем путь к нашему файлу с уравнениями (файл называется Mathfuncs\_Diode). Входным параметром будет напряжение, которое измеряется вольтметром, а выходным ток.

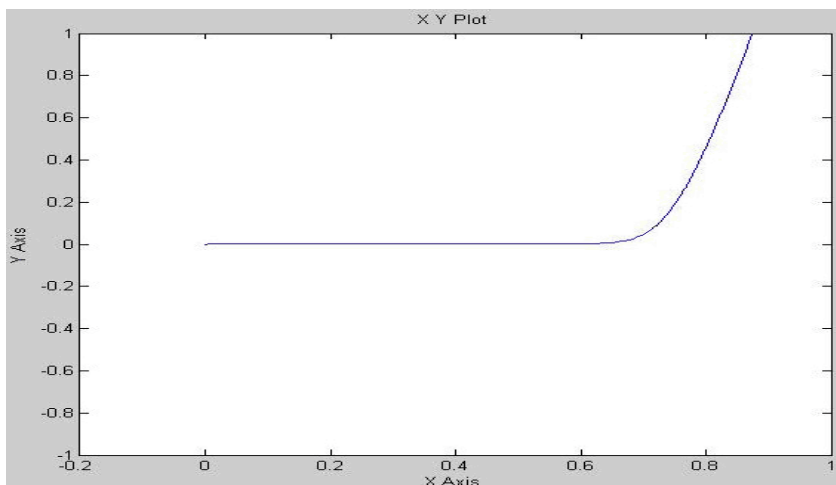
Всю систему поместим в один блок - Subsystem. Назовем наш блок с диодом DiodeModel. Создадим маску диода

Для тестирования модели диода соберем схему Блок XY Graph строит ВАХ диода, Multimeter - графики напряжения и тока на диоде и резисторе.

Полученные зависимости, полностью удовлетворяют работе диода при постоянном источнике тока.

Переходим к завершающей части нашей работы.

В Simulink преобразуем всю нашу схему, в одну модель. Таким образом, получаем статическую модель полупроводникового диода, к которой все начальные параметры пользователь сможет задавать самостоятельно



Вольтамперная характеристика моделируемого диода (стабилитрона)

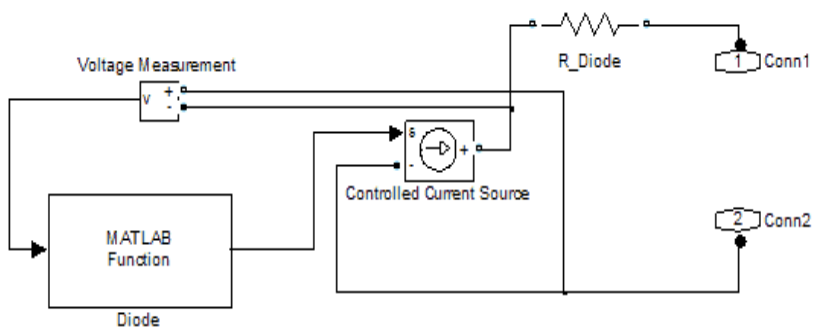


Схема стабилитрона(моделируемого диода)

## Практическое занятие № 4

### Алгоритмы поиска информации в поисковых системах.

**Цель занятия:** *Ознакомиться поисковыми серверами Интернет, научиться правильно сформировать алгоритм правильного и эффективного поиска необходимой информации в сети Интернет.*

#### Теоретическая часть

В Интернет существуют специальные сервера, которые занимаются исключительно поиском информации и ее сортировкой.

Каждый такой сервер ведет базу данных, которая содержит информацию о местоположении обнаруженной информации, краткое ее описание, информацию о доступе к ней и т.д. Базы данных заполняются программами роботами – поисковыми машинами, которые запускаются сервером и непрерывно просматривают содержимое Сети. Поисковые машины - *поисковая система с формируемой программным "роботом" базой данных, содержащей информацию об информационнх ресурсах.*

Существуют два вида поисковых серверов. Первый представляет собой каталог, т.е. вся информация в нем сгруппирована по темам, которые подразделяются на другие разделы и т.д.. Типичный пример такого каталога Yahoo.

Второй вид поисковых серверов – это серверы, которые могут искать информацию согласно вашему запросу, составленному из ключевых слов. Запросы состоят из списка ключевых слов и символов – модификаторов. Знак + означает, что слово обязательно

должно быть в искомом документе, а знак – означает, что слово не должно быть. Выражение. Которое нужно найти заключается в кавычки. Ответы на запрос представляются в виде списка ссылок с небольшими комментариями.

В последнее время многие из серверов реализуют одновременно обе системы поиска.

Кстати, не пренебрегайте возможностью проверить наличие искомого адреса прямой подстановкой ключевого слова в URL. Например, очевидно, что при поиске сайта компании Intel, было бы разумным проверить адреса [www.intel.com](http://www.intel.com) или [www.intel.org](http://www.intel.org).

В случае если ваш интерес имеет весьма размытые границы и определяется лишь некой областью человеческих знаний, - обратитесь к тематическому каталогу .

Поиск интересующих вас ссылок ведется в постоянно обновляемой собственной информационной базе (колоссального размера индексные файлы) поисковой машины, и в ответ на ваш запрос выдаются подходящие URL-адреса во всех концах сети Интернет. Поисковая машина самостоятельно постоянно ведет циклический опрос заданных адресов в Сети, пополняя собственную базу данных. В значительной степени, доступность документа для поисковой системы зависит от его автора. В его власти использовать в гипертексте наиболее запрашиваемые ключевые слова, поместить документ на доступном для основных поисковых машин сервере и не забыть заявить о себе поисковым службам международной сети Интернет.

Многие поисковые системы позволяют проводить поиск в найденных документах, причем вы можете уточнить ваш запрос введением дополнительных терминов. Если интеллектуальность системы высока, вам могут предложить услугу поиска похожих документов.



Некоторые поисковые машины позволяют провести пересортировку результатов. Для экономии вашего времени можно сохранить результаты поиска в виде файла на локальном диске для последующего изучения в автономном режиме.

Главной задачей любой поисковой машина является поиск информации, соответствующей информационным потребностям пользователя. Очень важно в результате проведенного поиска ничего не потерять, то есть найти все документы, относящиеся к запросу, и не найти ничего лишнего. Поэтому вводится качественная характеристика процедуры поиска - релевантность.

Релевантность - *это соответствие результатов поиска сформулированному запросу.*

Наиболее известные и популярные системы для поиска информации:

- **Anopm** (русскаяязычная) - <http://www.aport.ru>
- **Yandex** (русскаяязычная) - <http://www.yandex.ru>
- **Rambler** (русскаяязычная) - <http://www.rambler.ru>
- **Yahoo!** (англюязычная) - <http://www.yahoo.com>
- **AltaVista** (англюязычная) - <http://www.altavista.com>
- **InfoSeek** (англюязычная) - <http://www.infoseek.com>

Кроме рассмотренных, существуют также системы для поиска файлов (**files.ru**), людей (**whowhere.ru**) и т. д. Список ссылок на различные поисковые системы размещен на Web-странице **monk.newmail.ru**.

Рассмотрим работу поисковой машины на примере поисковой системы Яндекс.

Для вызова поисковой системы, пользователь вводит в адресной строке обозревателя Интернет ее адрес, в нашем примере мы используем поисковую систему Яндекс: [www.yandex.ru](http://www.yandex.ru) или [www.ya.ru](http://www.ya.ru) После загрузки поисковой системы в строке для поиска вводите

запрос (ключевую фразу), который представляет собой строку текста и нажимайте кнопку *Найти*.

**Упражнение1.** Проработать конкретную (по специальности) тему с использованием следующих поисковых машин:  
<http://www.aport.ru>, <http://www.yandex.ru>,  
<http://www.rambler.ru>.

Для выполнения задания целесообразно учесть следующие рекомендации.

В качестве ключевого слово задайте более чем одно слово. Рекомендуем также сужать область вашего вопроса.

Если список найденных страниц слишком мал или не содержит полезных страниц, попробуйте изменить слово - используйте синонимы. Попробуйте задать для поиска три-четыре слова-синонима сразу. Для этого перечислите их через вертикальную черту (|).

Используйте знаки "+" и "-". Чтобы исключить документы, где встречается определенное слово, поставьте перед ним знак минуса. И наоборот, чтобы определенное слово обязательно присутствовало в документе, поставьте перед ним плюс. Обратите внимание, что между словом и знаком плюс или минус не должно быть пробела.

Ключевое слово писать прописными буквами. Начиная слово с большой буквы, вы не найдете слов, написанных с маленькой буквы, если это слово не первое в предложении.

Если один из найденных документов ближе к искомой теме, чем остальные, нажмите на ссылку "найти похожие документы". Ссылка расположена под краткими описаниями найденных документов. Yandex анализирует страницу и найдет документы, похожие на тот, что вы указали.

**Упражнение 2.** Выполнение более глубокого поиска по заданной теме на основе технологии расширенного поиска.

Для выполнения этого упражнения используйте следующие рекомендации:

Используйте язык запросов. С помощью специальных знаков вы сможете сделать запрос более точным. Например, укажите, каких слов не должно быть в документе, или что два слова должны идти подряд, а не просто оба встречаться в документе.

Искать без морфологии. Вы можете заставить Яндекс не учитывать формы слов из запроса при поиске. Например, запрос !иванов найдет только страницы с упоминанием этой фамилии, а не города "Иваново".

**Упражнение 3.** Проработать заданную тему по разделам каталогов указанных выше поисковых машин и отобрать, сохранить на дискете результаты поиска по заданной теме.

Для выполнения данного задания используйте расширенный поиск, для этого используйте страницу "расширенный поиск", где большая часть настроек Яндекса задается простым образом.

Система предоставляет следующие объекты:

*Искомые слова.* Здесь вы можете указать, какие слова обязательно должны встретиться в документе, каких быть не должно, а какие желательны (то есть могут быть, а могут не быть). Зоной поиска слова может быть как текст документа (слова находятся в одном предложении или всем документе), так и его заголовок, аннотация (тэг description), ссылка (подпись URL) или адрес (сам URL).

*Страницы.* Можно задать язык документа, где надо провести поиск: русский (кириллица) или не русский. В базе Яндекса находятся только документы русскоязычного Интернета (по умолчанию в поисковую

машину вносятся сервера в доменах su, ru, am, az, by, ge, kg, kz, md, tj, ua, uz), а также зарубежные сайты, представляющие интерес для русскоязычного поиска. Так же вы можете указать ограничение выдачи документов по дате и формат документа. Внизу страницы вы можете посмотреть ваш запрос и все указанные параметры.

Каталоги. Каталог - поисковая система с классифицированным по темам списком аннотаций со ссылками на web-ресурсы. Такая классификация, как правило, проводится людьми. Поиск в каталоге проводится посредством последовательного уточнения тем, разделов, подразделов и т.д. Кроме этого, *каталоги* поддерживают возможность быстрого поиска определенной категории или страницы по ключевым словам с помощью локальной поисковой машины. Некоторые каталоги используют автоматическое обновление индекса. Результат поиска в каталоге представляется в виде списка, состоящего из краткого описания (аннотации) документов с гипертекстовой ссылкой на первоисточник.

Адреса популярных каталогов:

- **@Rus** (русскоязычный) - [www.atrus.ru](http://www.atrus.ru)
- **Weblist** (русскоязычный) - [www.weblist.ru](http://www.weblist.ru)
- **Созвездие интернет** (русскоязычный) - [www.stars.ru](http://www.stars.ru)
- **Yahoo** (англоязычный) - [www.yahoo.com](http://www.yahoo.com)
- **Magellan** (англоязычный) - [www.mckinley.com](http://www.mckinley.com).

В результате поиска вы получите следующую страницу результатов:

1. Поисковая форма. Главный ее элемент - строка запроса. При желании можно искать только в результатах предыдущего поиска ("искать в найденном"). Можно также задать запрос по любому из

источников информации ("Везде", "Каталог", "Новости", "Маркет", "Энциклопедии", "Картинки").

2. Статистика поиска: число найденных документов и частота заданных в запросе слов.

3. Рубрики каталога. По нажатию на ссылку происходит переход в соответствующий раздел каталога (выводится только при точном соответствии запросу).

4. Результаты поиска по вебу. - ссылка на ресурс, краткое описание ресурса, информация о размере, кодировке и, дате документа, его URL. Найденные слова - переход на "подсвеченный" документ, в котором контекст запроса выделен стрелочками, поиск похожего документа и, если найденный сайт описан в каталоге, переход в рубрику каталога.

5. Переход на следующие страницы результата. Сортировка по убыванию дат или релевантности.

6. Результат поиска в новостных лентах информационных агентств (выводится только при точном соответствии запросу). При высоком соответствии и актуальности эта секция может оказаться над результатами поиска по вебу.

7. Результат поиска в базе товарных предложений магазинов (выводится только при точном соответствии запросу).

### **Контрольные вопросы**

1. Принципы поиска информации в Интернете.
2. Типы поисковых систем в Интернете.
3. Принцип работы поисковой машины.
4. Правила составления запроса из ключевых слов.
5. Дополнительные возможности расширенного поиска.
6. Возможности поиска по каталогам.

## **Практическое занятие № 5.**

### **Создание Web-страниц с помощью языка HTML**

**Цель занятия:** Изучение основных тегов языка разметки гипертекста HTML, приобретение навыков создания Web-страниц.

#### **Теоретическая часть**

Любой гипертекстовый документ представляет собой обычный текстовый файл в формате ANSI ASCII, содержащий собственно текст и специальные теги для его разметки, а также ссылки на другие подобные документы, графические изображения и любые иные файлы. Когда браузер – программа просмотра гипертекста – загружает подобный файл, все теги обрабатываются браузером в соответствии с правилами специального языка – HTML (Hyper Text Markup Language – Язык Разметки Гипертекста).

HTML – это язык разметки, т.е. с его помощью можно оформлять документы, создавать ссылки, но ни как не писать программы. Даже специальные эффекты, которые можно увидеть на страницах Web, создаются не с помощью HTML, а с использованием дополнительных средств – например, встроенных в документ программ на языке JavaScript.

Страницы HTML представляют собой обычные текстовые файлы, поэтому можно использовать любой текстовый редактор для создания страницы

Документ HTML представляет собой текстовый файл и содержит элементы, называемые тегами, которые могут занимать одну или несколько строк. Теги легко отличить от остальных элементов: они ограничены угловыми скобками « < » и « > ». Большинство тегов

используются попарно – открывающий и закрывающий. Название тегов в языке HTML разрешено записывать как строчными, так и прописными буквами – браузеру все равно.

Тег <HTML> указывает, что этот документ является HTML документом.

Тело документа HTML, ограничено элементами <HTML> и </HTML>, как это показано ниже:

<HTML>

... (документ HTML)

</HTML>

Документ HTML состоит из двух основных частей – заголовка документа и тела документа:

<HTML>

<HEAD>

... (заголовок документа)

</HEAD>

<BODY>

... (тело документа)

<BODY>

</HTML>

В заголовок документа входит следующая информация:

- Название страницы. <title> ... </title> позволяют определить название страницы. Оно не появляется в теле HTML страницы, а выводится в строке заголовка браузера;
- Сценарии. Любой сценарий, выполняемый на странице, например, написанный на JavaScript;
- Стили. Можно добавлять элементы управления и стили;
- Метаинформация.

*Пример: заголовок.*

```
<html>
<head>
<title> First Page </title>
</head>
</html>
```

Следует отметить, что форматирование и оформление – не одно и то же. Форматирование задает структуру, а оформление – внешний вид.

Следующие теги являются тегами заголовка:

- **<h1> ... </h1> - <h6> ... </h6>**. Диапазон тэгов заголовков применяется для заголовков и подзаголовков содержимого.

*Пример : Типы заголовков.*

```
<HTML>
  <HEAD>
    <TITLE>First page</TITLE>
  </HEAD>
  <BODY>
    <H1>Заголовок 1</H1>
    <H2>Заголовок 2</H2>
    <H3>Заголовок 3</H3>
    <H4>Заголовок 4</H4>
    <H5>Заголовок 5</H5>
    <H6>Заголовок 6</H6>
  </BODY>
</HTML>
```

Определены следующие теги форматирования:

- **<BR>**. Тэг разрыва, который равнозначен одному возврату каретки.
- **<P>**. Тэг абзаца, используемый для обозначения абзаца.
- **<P> ... </P>**. Способ обозначения абзаца с помощью открытия/закрытия.



- **<PRE> ... </PRE>**. Тэг предварительного форматирования текста, используется для сохранения количества пробелов в тексте.

- **<NOBR> ... </NOBR>**. Если вы хотите использовать строку без естественного разрыва.

При подготовке документа используются ряд шрифтов.

**Элемент Font.** Синтаксис **<font> --text-- </font>** этот элемент позволяет задавать тексту нужный размер шрифта, цвет, стиль.

Добиться этого можно, используя следующие атрибуты:

- **size** – этот атрибут помогает определить высоту знаков текста;
- **color** – устанавливает цвет шрифта;
- **face** – задаёт название требуемого шрифта.

### **Упражнение1. Использование атрибутов <font>**

```
<html>
<head>
<title> Использование атрибутов </title>
</head>
<body>
<font size="5" color="#555555" >
<p><font face="Arial">Arial</font><br>
<font face="Arial Black">Arial black</font><br>
<font face="Arial Narrow">Arial narrow</font><br>
<font face="Arial Unicode MS">Arial Unicode
MS</font><br>
<font face="Batang">Batang</font><br>
<font face="Book Antiqua">Book Antiqua</font><br>
<font face="Bookman Old Style">Bookman Old
Style</font><br>
<font face="Century">Century</font><br>
```

```

<font face="Century Gothic">Century Gothic</font><br>
<font face="Comic Sans MS">Comic Sans MS</font><br>
<font face="Courier New">Courier New </font><br>
<font face="Fixedsys">Fixedsys</font><br>
<font face="Garamond">Garamond</font>
</font>
</body>
</html>

```

В языке HTML определены теги, предназначены для так называемого физического форматирования символов текста. Эти теги определяют внешний вид символов явным образом:

```

<HTML>
  <HEAD>
    <TITLE>First page</TITLE>
  </HEAD>
  <BODY>
    <B>Жирный текст</B><BR>
    <I>Наклонный текст</I><BR>
    <U>Подчеркнутый текст</U><BR>
    <STRIKE>Перечеркнутый текст
  </STRIKE><BR>
    <BIG>Увеличенный текст </BIG><BR>
    <SMALL>Уменьшенный текст
  </SMALL><BR>
  <BODY>
</HTML>

```

Выравнивание играет большую роль, определяя, каким образом текст располагается по отношению к другим объектам.

Существуют следующие типы выравнивания:

- по умолчанию(по левому краю);
- по левому краю;
- по правому краю;

- по центру;
- по ширине.

Тэг-контейнер `<DIV>` является элементом уровня блока, служащим для выделения фрагмента документа. Целью этого выделения является управление параметрами данного фрагмента, которое обычно выполняется с помощью назначения стилей. Приведем пример:

```
<DIV ALIGN=CENTER>
```

(Фрагмент документа)

```
</DIV>
```

В этом примере фрагмент HTML-документа обрамляется тэгами `<DIV>` и `</DIV>` для задания некоторых его свойств. В данном случае все текстовые элементы выделенного фрагмента будут отображаться зеленым (green) цветом. Аналогом тэга `<DIV>` уровня текста является элемент `<SPAN>`.

- `<DIV align="right">` - по правому краю,
- `"left">` - по левому краю,
- `"center">` - по центру,
- `"justify">` - по ширине.

**Упражнение 2.** Создание страницы с вложенной элементом DIV.

```
<html>
```

```
<head>
```

```
<title> Div </title>
```

```
<body>
```

```
  <DIV align="right">
```

```
Hello world!
```

```
</DIV>
```

```
  <DIV align="center">
```

```
Hello world!
```

```
</DIV>
```

```
<DIV align="left">
Информационные технологии!
</DIV>
</body>
</html>
```

Выравнивание абзаца требует для каждого фрагмента наличия открывающего и закрывающего дескрипторов. Используются те же значения что и с тэгом <div>:

```
<P align="center"> ... some text ... </p>
```

## Списки.

Списки в HTML позволяют разделять информацию на логические последовательности элементов.

Благодаря спискам содержимое документов можно представить таким образом, чтобы быстро и в нужной последовательности ознакомить читателей с основным их содержанием.

Существуют следующие тэги списков:

- <UL> ... </UL>. Неупорядоченный, или маркированный список.
- <OL> ... </OL>. Упорядоченный, или нумерованный список.
- <LI>. Тэг элемента списка.
- <DL> ... </DL>. Тэг списка определений.
- <DT> ... </DT>. Определяемый термин, является частью списка определений.
- <DD> ... </DD>. Определение термина.

```
<html>
<head>
<title>Списки</title>
```

</head>

<body>

Чтобы изменить вид символа, который используется для выделения строки в списке, Вы можете использовать атрибут TYPE. Синтаксис для маркированного списка:

<UL TYPE="символ">

Где *символ* это:

- DISK (default) – диск(устанавливается по умолчанию)
- SQUARE - квадрат
- CICLE - круг

Чтобы изменить вид символа, который используется для выделения строки в списке, Вы можете использовать атрибут TYPE. Синтаксис для нумерованного списка:

<OL TYPE="символ">

Где *символ* это: арабские цифры (по умолчанию) 1, 2, 3, ....

латинский алфавит, прописные a, b, c, ....;

латинский алфавит, заглавные A, B, C, ....;

римские цифры, прописными i, ii, iii, ....;

римские цифры, заглавными I, II, III, ....

Сущность Web заключается в связывании. Связывание позволяет выйти за пределы одного документа и получить доступ не только к другим, относящимся к нему документам, но и соединиться с другим сайтом.

Сегодня в качестве ссылок используют множество объектов и средств мультимедиа. Чтобы отобразить этот аспект, был введён термин: гипермедиа.

В HTML существует элемент привязки <A>, с помощью которого происходит связывание страниц. <A> ... </A>. - Синтаксис элемента привязки.

Элемент привязки должен иметь атрибуты и значения. Самым главным, простым и распространённым атрибутом является href или гипертекстовая ссылка. После него вводят значение, чаще всего представляющее собой URL.

Универсальный локатор ресурсов URL – это адрес Web узла. Он состоит из префикса (обычно http://), который представляет собой протокол соединения, имени домена (чаще всего начинается с www) и указания ресурса (обычно представляющего собой имя файла). Элементы привязки создают ссылки двух видов: абсолютные и относительные.

Абсолютной называется ссылка, где в качестве значения используется полный URL. Это означает, что вы используете абсолютно весь Web-адрес, а не его часть. В ссылку включена информация и о протоколе и о домене. Эти данные позволяют вам перейти на начальную страницу по умолчанию этого Web-узла.

Относительные ссылки позволяют связываться с файлами, находящимися по тому же адресу, т.е. на том же сервере. При создании ссылки с одной страницы на другую в пределах одного и того же узла, когда обе страницы находятся в одном и том же каталоге, всё, что нужно сделать, – это присвоить в качестве значения гипертекстовой ссылки имя файла.

**Упражнение 3:** Использование ссылок.

default.htm связывается абсолютной ссылкой с:

Page1– <A HREF=  
"http:\\Server\\FirstSite\\page1.htm">Page1</A>

Dog1 – <A HREF=  
"http:\\Server\\FirstSite\\Dogs\\Dog1.htm">Dog1</A>

Cat1 – <A HREF=  
"http:\\Server\\FirstSite\\Cats\\Cat1.htm">Cat1</A>

Mouse1 – <A HREF=  
"http:\\Server\\FirstSite\\Cats\\Mouse\\Mouse1.htm">Mouse1</  
A>

default.htm связывается относительной ссылкой с:

Page1 – <A HREF= "page1.htm">Page1</A>

Dog1 – <A HREF="Dogs\\Dog1.htm">Dog1</A>

Cat1 – <A HREF= "Cats\\Cat1.htm">Cat1</A>

Mouse1 – <A HREF=  
"Cats\\Mouse\\Mouse1.htm">Mouse1</A>

Внутренние ссылки используются для перемещений в пределах одной страницы. Если вы в верхней части страницы перечисляете содержание, а текст к содержанию находится ниже на этой же странице, можно использовать внутренние ссылки для быстрого перехода к нужному разделу.

Для этого нужно создать ссылку на элемент <a> (по атрибуту name) с #.

: Использование внутренних ссылок.

<p>здесь Вы найдёте описания к следующим  
моделям<br>

```
<a href="#OpelOmega">Opel Omega</a>
<!----- some text ----->
<a name="OpelOmega"></a>
```

Обычно посетители Web-узла могут связаться с вами через вашу Web-страницу, используя для этого ссылку на адрес электронной почты. Это осуществляется с помощью элемента привязки и ссылки `mailto:`:

#### Использование почтовой ссылки.

```
<a href="mailto:kdanil@netdec.uz">отправьте мне
сообщение</a>
```

Для размещения на странице изображений используется тэг **<img>**. Этот тэг работает самостоятельно, т.е. закрывать его (**</img>**) не нужно.

Для тэга `<img>` требуется указать атрибут `src`:

```

```

Обычно все рисунки хранят в отдельном каталоге, который в свою очередь хранится в корневом каталоге HTML. Если изображение находится в специальном каталоге, то необходимо указать источник:

```

```

Изображения всегда помещают внутри раздела `body` документа HTML.

#### **Упражнение 4:: Вставка изображения.**

1)

```
<html>
  <head>
    <title> Вставка изображения.</title>
  </head>

  <body>
    
```



```

</body>
</html>
2)
<html>

  <head>
    <title>New Page 1</title>
  </head>
  <body>

    <p align="left"><b>Вставка рисунка</b></p>
    <p><img
      src=file:///H:/magistr/рисунки/motive/6654.jpg
      width="467" height="375"></p>
  </body>

</html>

```

Таблицы предназначены для размещения всевозможных элементов – от графических до текстовых. Сначала ознакомимся с основными элементами, применяемыми для создания таблиц.

**<table> ... </table>** – основной элемент таблицы, объявляющий её начало и конец;

**<tr> ... </tr>** – элементы, открывающие и закрывающие строку таблицы;

**<td> ... </td>** – элементы, создающие единичную ячейку таблицы.

Атрибуты элемента **<table>**:

**align="x"** – выравнивание содержимого ячейки по горизонтали. Допустимые значения: left, right и middle;

**border="x"** – ширина границ обрамления таблицы;

**cellspacing="x"** – ширина внутреннего пробела между соседними ячейками таблицы;

**cellpadding="x"** – отступы от границ ячейки таблицы до данных, размещенных в ячейке;

**width="x"("%x")** – ширина таблицы в пикселах(процентах).

Элемент объявления строки таблицы **<tr>** дополняется двумя атрибутами **align** (выравнивание содержимого ячейки по горизонтали) и **valign** (выравнивание содержимого ячейки по вертикали). Эти атрибуты применимы и для элементов **<table>** и **<td>**.

Основные атрибуты ячейки:

**width="x"("%x")** – значение ширины ячейки в пикселах (в процентах);

**height="x"** – высота ячейки;

**align="x"** – способ выравнивания по горизонтали;

**valign="x"** – способ выравнивания по вертикали;

**bbgcolor="x"** – цвет фона ячейки;

**background="url"** – фоновый рисунок в ячейке.

```
<html>
```

```
<head>
```

```
<title>Вставка таблицы 1</title>
```

```
</head>
```

```
<body>
```

```
<p align="left"><b>Вставка таблицы</b></p>
```

```
<table border="1" width="36%">
```

```
<tr>
```

```
<td width="32%">Валиев Д.</td>
```

```
<td width="68%">г.Наманган</td>
```

```
</tr>
```

```
<tr>
```

```
<td width="32%">Икрамов К</td>
```

```
<td width="68%">г.Ташкент</td>
```

```
</tr>
```

```
<tr>
```

```
<td width="32%">Захидова С</td>
```

```
<td width="68%">г.Самарканд.</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

## **Контрольные вопросы**

1. Что такое HTML?
2. Для чего используется HTML?
3. Что такое тег?
4. Каковы требования к оформлению тегов?
5. Как вставить изображение в страницу?
6. Как вставить таблицу в страницу?
7. Назовите основные теги языка HTML и их назначение?
8. Какие еще средства позволяют создать web страницы.

## Литература

1. Алексей Гончаров "Самоучитель HTML" - Питер. 2005
2. А.Петюшкин. HTML Экспресскурс. Феникс. 2004
3. А.Матросова,М.Чаунин,А.Сергеев HTML 4.0 "Новый уровень создан". Питер. 2005
4. <http://www.ict.nsc.ru/rus/docs/html-gd/refrence.html> - Институт вычислительных технологий Сибирского отделения Российской академии наук.
5. Апорт - <http://www.aport.ru>
6. Yandex - <http://www.yandex.ru>
7. Rambler - <http://www.rambler.ru>
8. Симонович С.В., Мураховский В.И. Интернет у вас дома: Полное руководство начинающего пользователя. – М.: АСТ-ПРЕСС КНИГА, 2004. – 432 с.
9. Леонтьев В.П. Новейшая энциклопедия Интернет 2004, Олма-Пресс, 2004.-784 с.

## Содержание

### Практическое занятие №1

Средства автоматизации обработки  
данных..... ..3

### Практическое занятие №2

Использование спец.систем MathCad, Matlab  
в решениях технических  
задач .. 7

### Практическое занятие №3

Алгоритмы поиска информации в поисковых  
системах.....15

### Практическое занятие №4,5

Создание Web-сайта в языке HTML..... 21

Литература .....36