

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ АБУ
РАЙХАНА БЕРУНИЙ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

методические указания для выполнения
лабораторных работ для магистров

2-ЧАСТЬ

ТАШКЕНТ – 2007

СОСТАВИТЕЛИ: Сагатов М.В., Ирмухамедова Р.М., Каримова Д., Юлдашева М.

Методические указания для выполнения лабораторных работ по курсу «Информационные технологии»
СОСТАВИТЕЛИ: Сагатов М.В., Ирмухамедова Р.М., Каримова Д., Юлдашева М., ТГТУ, 2007, 62стр., - 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 лабораторных работ по курсу «Информационные технологии» для магистров

Кафедра «Общая информатика»

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

Рецензенты:

1. Гаибназаров Т. - Ташкентский Университет Информационных технологий, доцент, к.т.н.

2. Закиров Т.З. – Ташкенский Государственный технический университет , к.т.н.,доц.

Лабораторная работа № 1

Работа с программой Fine Reader 7.0.

Цель работы: Изучение основных принципов сканирования, сегментации и распознавания данных.

Задание:

Освоить приемы работы с программой сканирования ABBYY Fine Reader 7.0. :

1. Запустить программу Fine Reader 7.0.
2. Отсканировать тексты:
 - текст обычный
 - текст с рисунком
 - текст, разбитый на несколько колонок
 - сведения в табличном виде
3. Отсегментировать и распознать отсканированные данные.
4. Проверить орфографию распознанных данных.
5. Скопировать и сохранить данные в TP Word.
6. Аналогичным образом отсканировать текст с рисунком; текст, разбитый на несколько колонок; сведения в табличном виде, затем проверить орфографию и сохранить

результаты сканирования в отдельном пакете и оформить отчет.

Теоретическая часть

1. Система оптического распознавания **FineReader 7.0**

Чтобы отсканировать текст, картинку или фотографию потребуются программное обеспечение (ПО) - графический редактор и сканер, который нужно включить до запуска программы. Применение сканера возможно также для ввода в компьютер различных карт и чертежей.

– **Сканер**- это устройство, предназначенное для ввода информации в компьютер, которое позволяет получить «цифровую» (электронную) копию различных изображений. Процесс создания электронной копии напоминает процесс фотографирования изображения бумажного документа.

Все операции, необходимые в ходе преобразования бумажного документа в электронную форму могут быть выполнены с помощью программы **FineReader 7.0** которая разработана фирмой ABBYY.

Принцип действия программы **FineReader 7.0** следующий: сканируется любой текст, затем картинка текста преобразуется в "обычный электронный текст", такой, как если бы вы его напечатали с клавиатуры. Пользователю остается только сохранить текст на диске или скопировать его через буфер обмена в любой текстовый редактор.

Программа **FineReader 7.0** предназначена для распознавания текстов на русском, английском, немецком, украинском, французском и многих других языках, может распознавать смешанные двуязычные тексты.

С помощью программы **FineReader 7.0** можно выполнять пакетную обработку многострочных документов, а также программу можно “обучать” для повышения качества распознавания неудачно напечатанных текстов или сложных шрифтов, она позволяет редактировать

распознанный текст, проверять его орфографию и сохранять результаты в текстовом редакторе MS Word, MS Excel. Программа позволяет объединять сканирование и распознавание в одну операцию.

FineReader 7.0 автоматически распознает разные участки текста: текст как таковой, картинку (рисунок), таблицу и так называемые "нераспознаваемые" блоки в тексте.

Запустить программу можно через кнопку Пуск/ Программы/ ABBYY Fine Reader/ ABBYY Fine Reader 7.0 Handprint.

2. Главное окно программы FineReader 7.0

Вся работа в программе **FineReader 7.0** ведется в пакетном режиме.

– **Пакет** - это папка, в которой хранятся изображения и рабочие файлы программы. Каждое отсканированное изображение записывается как отдельная страница пакета.

Экран программы **FineReader 7.0** (рисунок 1) включает в себя строку меню, панели инструментов, окна **Пакет**, **Текст**, **Изображение**.

Окно **Пакет** отражает страницы пакета.

Окно **Текст** – распознанный текст активной страницы окна **Пакет**.

Окно **Изображение** – графическое изображение (картинку)

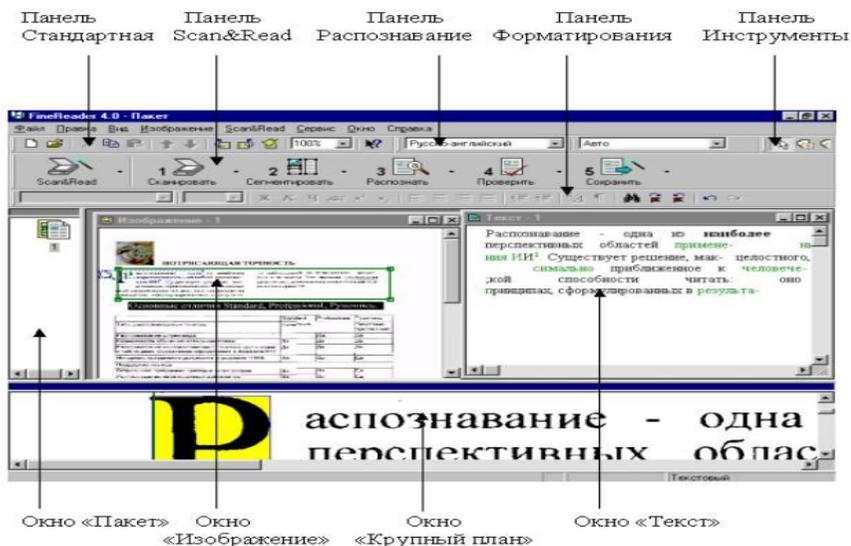


Рисунок 1. Вид окна программы Fine Reader 7.0

В левой рабочей области располагается окно **Пакет**, содержащее список графических документов, которые должны быть преобразованы в текст. Эти графические файлы рассматриваются как части одного документа. Результаты их обработки в дальнейшем объединяются в текстовый файл. Форма значка, отмечающего исходные файлы, указывает, было ли произведено распознавание. Окно **«Крупный план»** содержит фрагмент графического документа в увеличенном виде. С ее помощью можно оценить качество распознавания. Эту панель используют также при «обучении» программы в ходе распознавания текста. В окне **«Крупный план»** по умолчанию показывается черно-белое изображение независимо от того, какое именно (цветное, серое или черно-белое) изображение открывали. Остальную часть рабочей области занимают окна документов (окно **«Изображение»**, окно **«Текст»**). Здесь располагается окно графического документа, подлежащего распознаванию, а также окно текстового документа, полученного после распознавания. В верхней части окна приложения под строкой меню располагаются панели инструментов. На приведенном рисунке включено отображение всех панелей, которые могут использоваться в программе Fine Reader.

Панель инструментов **Стандартная** содержит кнопки для открытия документов и для операций с буфером обмена. Прочие кнопки этой панели служат для изменения представления документа.

Панель **Scan&Read** содержит кнопки, соответствующие всем этапам превращения бумажного документа в электронный текст. Первая кнопка позволяет выполнить такое преобразование в рамках единой операции. Остальные кнопки соответствуют отдельным этапам работы и содержат раскрывающиеся меню, служащие для управления соответствующих операций.

Панель **Распознавание** позволяет указать язык документа и вид шрифта. Последнее требуется делать только в тех случаях, когда документ имеет недостаточное качество печати.

Панель **Инструменты** используют при работе с исходным изображением. В частности, она позволяет управлять сегментацией документа (разбиением страницы на блоки текста). С помощью элементов управления этой панели задают последовательность фрагментов текста в итоговом документе.

Элементы управления панели **Форматирование** используют для изменения представления готового текста или при его редактировании. Внизу окна расположена информационная панель (еще ее называют строкой состояния), где можно получить сведения о состоянии системы и производимых ею операциях, а также краткую справку о выбираемых пунктах меню и кнопках.

3. Работа с программой Fine Reader 7.0 и порядок распознавания документов

Основные операции обработки бумажного документа в программе **Fine Reader 7.0** выполняются с помощью панели инструментов **Scan&Read**. На панели инструментов программы находятся соответствующие кнопки “Сканировать”, “Сегментировать”, “Распознать”, “Проверить”, “Сохранить”. Можно выполнять указанные операции и через меню **Scan&Read** в строке команд.

Процесс обработки документа состоит из следующих шагов:

- сканирование документа;
- сегментация документа;
- распознавание документа;
- редактирование и проверка результата;
- сохранение документа.

Первый этап работы – сканирование. На этом этапе используют сканер. Чтобы начать сканирование, нужно включить сканер, положить оригинал документа (обычно левой стороной вниз, если только вы не используете ручной сканер) и щелкнуть кнопку “Сканировать” или в раскрывающемся списке подкоманд данной кнопки использовать пункт **Сканировать** (рисунок 2):

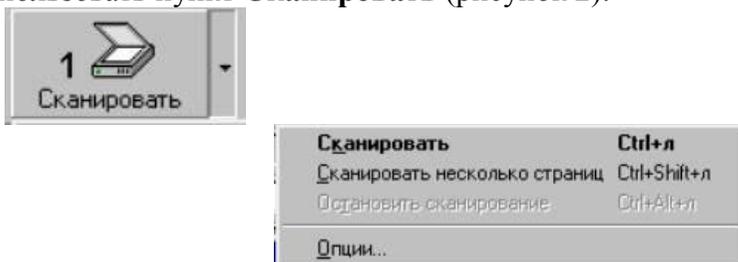


Рисунок 2 Вид кнопки “Сканировать”

Программа способна работать со сканером как непосредственно, так и через драйвер TWAIN. При непосредственном взаимодействии со сканером возможность сканирования цветных изображений не используется, так как предполагается, что текст в любом случае является одноцветным.

Программа использует для сканирования устройство, которое задано по умолчанию. Для того, чтобы выбрать такое устройство или изменить его настройку, нужно выбрать пункт **Опции...** в раскрывающемся списке кнопки “Сканировать” – откроется диалоговое окно **Опции** (рисунок 3).

Если к компьютеру подключено несколько сканеров или был подключен после установки программы **Fine Reader 7.0**, следует щелкнуть на кнопке **Выбрать сканер**. В этом случае проведет поиск подключенных к компьютеру сканеров и позволит выбрать нужный.

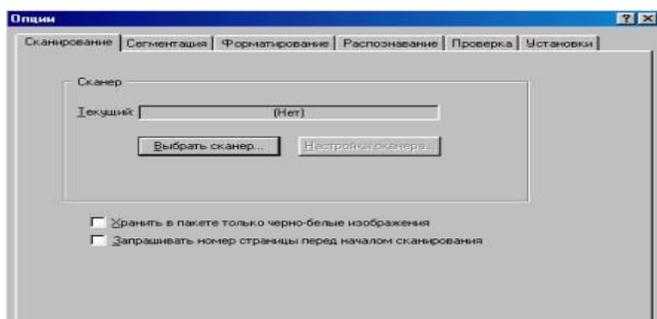


Рисунок 3 Вид диалогового окна

Для изменения настроек сканера используют кнопку **Настройка сканера**.

–Диалоговое окно интерфейса TWAIN может выглядеть по-разному – его вид определяется драйвером сканера. В нем с помощью пункта Scan Mode можно задать различные режимы сканирования:

Black&White (черно-белый) – предназначен для сканирования текстов и контурных рисунков.

Gray (серый) – 256 оттенков серого цвета, используют данный режим для сканирования черно-белых фотографий или для черно-белых принтеров.

Color (цветной) – для получения цветных картинок и фотографий.

Использовать протокол TWAIN имеет смысл только в том случае, когда работа напрямую невозможна или дает некачественные результаты.

–Кнопка "**Scan**" (сканировать, начать) запускает основной процесс сканирования, в результате которого мы видим на экране ПК получившуюся фотографию. Остается сохранить ее на диске для дальнейшего использования.

–Кнопка "**Prescan**" предназначена для пробного (предварительного) сканирования. Позволяет увидеть положение картинки и задать с помощью специального "пунктирного" прямоугольника (рамки) область сканирования.

Флажок Показывать опции перед началом сканирования из диалогового окна Опции (рисунок 3) применяют только в том случае, когда бумажные страницы документа существенно отличаются друг от друга по качеству (разные страницы печатались в разное время и разными средствами). В этом случае перед сканированием каждой страницы открывается диалоговое окно настройки сканера, чтобы пользователь мог отрегулировать качество процесса.

Сам процесс сканирования происходит в автоматическом режиме. Если требуется обработать много страниц, то лучше сначала все их отсканировать, а затем приступить к распознаванию. Это связано с тем, что сканирование требует присутствия пользователя из-за необходимости управления сканером (например, для смены страниц), а процесс распознавания может происходить в автоматическом режиме.

Когда процесс сканирования завершается, появляется окно с изображением текста.

Второй этап работы – сегментация текста. Дело в том, что в бумажном документе, например на странице книги или журнала, текст не всегда располагается в фиксированном порядке. Он может размещаться в нескольких колонках (столбцах), содержать иллюстрации

(и подписи к ним), другие элементы форматирования. Дополнительные врезки и данные, представленные в таблицах, также могут запутать естественный порядок текста. Поэтому, прежде чем включать текст в документ, его разбивают на *блоки*, содержащие цельные фрагменты. Блоки распознают последовательно. Полученный текст включается в документ в порядке нумерации блоков.

◆ Сегментация – разбиение документа на блоки.

Ручную установку блоков (рамка с помощью левой кнопки мыши) можно применить, если нужно распознать только часть текста.

Для создания блока непрямоугольной формы или для изменения последовательности блоков используют кнопки, выведенные на панели (рисунок 4.) инструментов «Инструменты»:

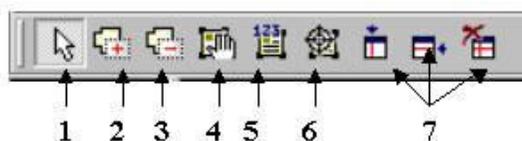


Рисунок 4. Вид кнопок на панели «Инструменты»

Все кнопки этой панели используются именно на этапе сегментации и позволяют:

- создать блок (кнопка **1** – Создать блок);
- объединить два блока в один (кнопка **2** – Добавить часть блока);
- удалить фрагмент блока (кнопка **3** – Удалить часть блока);
- изменить положение блоков (кнопка **4** – Переместить блоки);
- изменить порядок нумерации блоков (кнопка **5** – Перенумеровать блоки);
- удалить блоки (кнопка **6** – Удалить блок);
- изменить разбиение таблицы на ячейки (кнопки **7**).

Разные типы блоков обрабатываются программой по-разному. Программа **Fine Reader 7.0** поддерживает следующие типы блоков (рисунок 5).

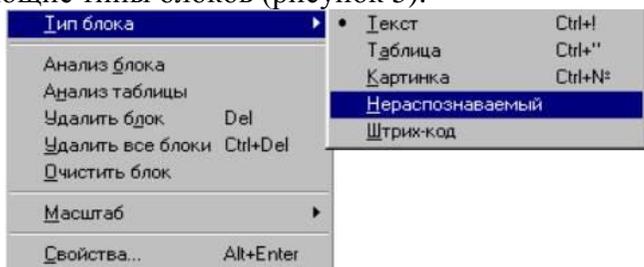
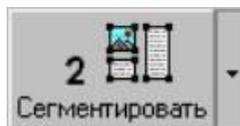


Рисунок 5. Вид контекстно-зависимого меню при работе с блоками

- текстовый (Текст) – на этапе распознавания преобразуется в текст;
- табличный (Таблица) – представляет собой набор ячеек, каждая из которых преобразуется в текст по отдельности;
- изображение (Картинка) – включается в документ без изменений как графическая иллюстрация, если формат сохранения преобразованного документа допускает вставные объекты;
- лишний (Нераспознаваемый) – игнорируется;
- содержащий штрих-код (Штрих-код) – распознается как штрих-код.

Для автоматической сегментации текста используют кнопку “Сегментировать” или ее раскрывающийся список подкоманд (рисунок 6):



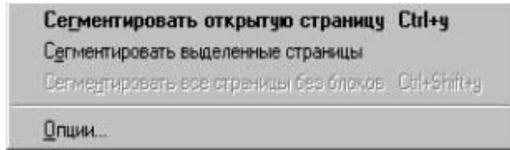


Рисунок 6 Вид кнопки “Сегментировать”

При автоматической сегментации определение границ блока определяется автоматически. При этом учитываются поля документа, просветы между колонками, рамки.

Третий этап работы программы – непосредственно *распознавание*. Этот этап обычно не требует вмешательства пользователя, за исключением тех случаев, когда распознавание сопровождается «обучением».

Для большинства случаев сразу нужно нажать на кнопку инструмента “Распознать” и подтвердить автоматическое определение блоков (рисунок 7):

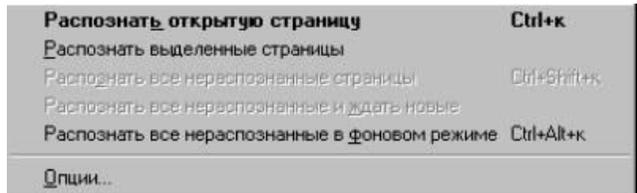


Рисунок 7 Вид кнопки “Распознать”

Процесс распознавания будет “иллюстрироваться” закраской участков текста.

Точность распознавания **FineReader 7.0** около 92- 97 %, так как при печати книг и газет на бумаге иногда не прилипает или осыпается краска. Это будет хорошо видно в окне “Крупный план” с увеличенным масштабом.

Результаты распознавания будут отражены в окне “Текст”.

На четвертом этапе, средствами программы в полученном тексте, можно провести *проверку грамматики* с помощью соответствующей кнопки “Проверить” на панели инструментов или пунктом **Проверить** из раскрывающегося списка подкоманд (рисунки 8):

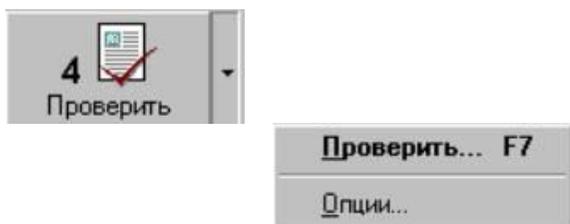


Рисунок 8 Вид кнопки “Проверить”

В результате появится следующее окно диалога (рисунок 9):

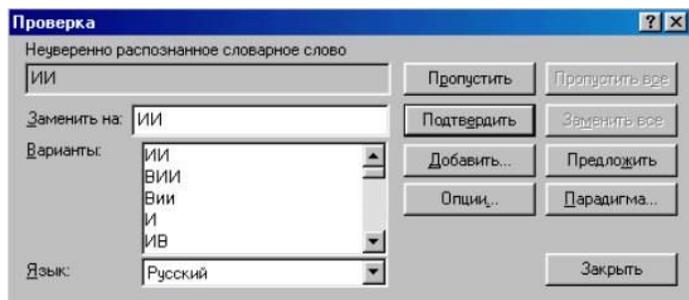


Рисунок 9. Вид окна диалога проверка орфографии

Распознанный текст отображается в отдельном окне “Текст”, в виде форматированного текстового документа. Он «теряет связь» с исходным изображением и может редактироваться и форматироваться независимо от него. Программа выделяет цветом те символы, которые она рассматривает как неоднозначно опознанные. Это упрощает поиск ошибок.

Пятый этап: по щелчку на кнопке “Сохранить” на панели инструментов **Scan&Read** (рисунок 10) запускается Мастер сохранения результатов (рисунок 11). Он позволяет сохранить распознанный текст или передать его в другую программу (например, MS Word) с помощью выбора соответствующего пункта из раскрывающегося списка подкоманд:

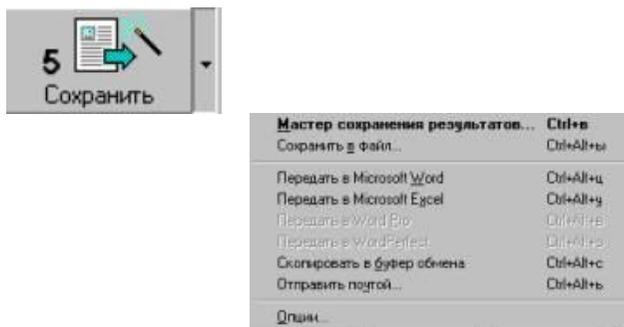
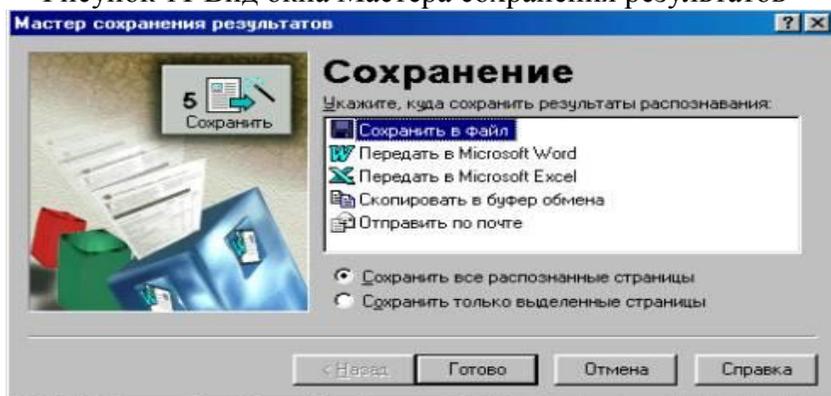


Рисунок 10 Вид кнопки “Сохранить”

Рисунок 11 Вид окна Мастера сохранения результатов



Контрольные вопросы

1. Перечислите основные элементы окна программы **Fine Reader 7.0**.
2. Из каких основных этапов состоит процесс сканирования?
3. Какой стороной нужно помещать “оригинал” в сканер?
4. Какие типы распознаваемых блоков вы знаете?
5. Как распознать только часть текста сканируемого документа?
6. Назначение и сфера применения программ OCR.
7. Что такое “разрешение”, в каких единицах оно измеряется?
8. Какие типы сканеров Вы знаете?
9. Можно ли распознать фотографию текста, записанную в виде файла?
10. С каким разрешением лучше сканировать большую фотографию, которую в дальнейшем нужно показать на экране компьютера?

Лабораторная работа №2

Решение научно-технических задач с помощью систем MathCad, Matlab

Цель работы: ознакомиться с программой MathCAD, изучить ее возможности при решении конкретной задачи.

Задание:

1. Изучить особенности и основные возможности программы MathCAD.
2. Выбрать задачу по специальности для решения.
3. Произвести необходимые операции для решении данной задачи.

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

Определение функций и построение графиков

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

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].

Задачей данной работы было исследование и разработка модели полупроводникового диода, отражающей физическую сущность прибора с возможностью задания физических параметров.

Реализация.

1. Диод полупроводниковый прибор, имеющий два вывода для включения в электрическую цепь. Для реализации модели полупроводникового диода использовано уравнение Шокли, дополненное

зависимостями, включающими различные физические эффекты [1].

$V_t = k \cdot T / q$ температурный потенциал перехода

Где $k = 1.38 \cdot 10^{-23}$ Дж/град - постоянная Больцмана;

T - температура (передается в функцию);

$q = 1.6 \cdot 10^{-19}$ Кл - заряд электрона;

$E_{GT} = E_{G0} - a \cdot T^2 / (b + T)$ температурная зависимость ширины запрещенной зоны

Где $E_{G0} = 1,16$ эВ;

$a = 0,000702$;

$b = 1108$;

$I_S = I_{S0} \cdot \exp(E_{GT} / (N \cdot V_t) \cdot (T / T_0 - 1)) \cdot (T / T_0)^{X_{TI} / N}$

температурная зависимость тока насыщения

Где I_S - ток насыщения при температуре 27°C (по умолчанию 10^{-14} А);

N - коэффициент неидеальности (1);

T_0 - номинальная температура (27°C);

X_{TI} - температурный коэффициент тока насыщения (3 для диодов с р-п переходом и 2 для диодов с барьером Шоттки);

$I_{SR} = I_{SR0} \cdot \exp(E_{GT} / (N \cdot V_t) \cdot (T / T_0 - 1)) \cdot (T / T_0)^{X_{TI} / N}$

температурная зависимость тока рекомбинации

Где I_{SR} параметр тока рекомбинации (0 А);

$I_{KF} = I_{KF0} \cdot (1 + T_{IKF} \cdot (T - T_0))$ температурная зависимость тока перегиба при высоком уровне инжекции

Где I_{KF} - тока перегиба при высоком уровне инжекции (inf А);

T_{IKF} - линейный температурный коэффициент I_{KF} (0°C^{-1});

$V_B = V_{B0} \cdot (1 + T_{BV1} \cdot (T - T_0) + T_{BV2} \cdot (T - T_0)^2)$

температурная зависимость напряжения обратного пробоя

Где BV - напряжение обратного пробоя ($\inf B$);
 $TBV1$ - линейный температурный коэффициент BV ($0\text{ }^\circ\text{C}^{-1}$);
 $TBV2$ - квадратичный температурный коэффициент BV ($0\text{ }^\circ\text{C}^{-2}$);

Система уравнений, описывающая токи в диоде.

$I_n = I_{ST} * (\exp(\text{sym}(u/(N * V_t), 'd')) - 1)$ нормальная составляющая тока

Где u - напряжения B ;

$I_{rec} = I_{SRT} * (\exp(\text{sym}(u/(NR * V_t), 'd')) - 1)$ ток рекомбинации

Где NR - коэффициент неидеальности для тока ISR (2);

$K_{inj} = (IKFT / (IKFT + I_n))$ коэффициент инжекции

$K_{gen} = (((1 - u/VJ)^2) + 0.005)^{M/2}$ коэффициент генерации

Где VJ - контактная разность потенциалов (1 В);

M - коэффициент, учитывающий плавность перехода $p-n$ (0,5);

$I_{revhigh} = IBV * \exp(\text{sym}(-(u + BVT)/(NBV * V_t), 'd'))$

Где IBV - начальный ток пробоя (10-10 А);

NBV - коэффициент неидеальности при участке пробоя (1);

$I_{revlow} = IBVL * \exp(\text{sym}(-(u + BVT)/(NBVL * V_t), 'd'))$

Где $IBVL$ - начальный ток пробоя низкого уровня (0 А);

$NBVL$ - коэффициент неидеальности на участке пробоя низкого уровня;

$I_{fwd} = (I_n * K_{inj}) + (I_{rec} * K_{gen})$ ток диода при положительном напряжении

$I_{rev} = I_{revhigh} + I_{revlow}$ ток диода при отрицательном напряжении на переходе

$I = I_{fwd} - I_{rev}$ ток диода

Практическая часть.

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.

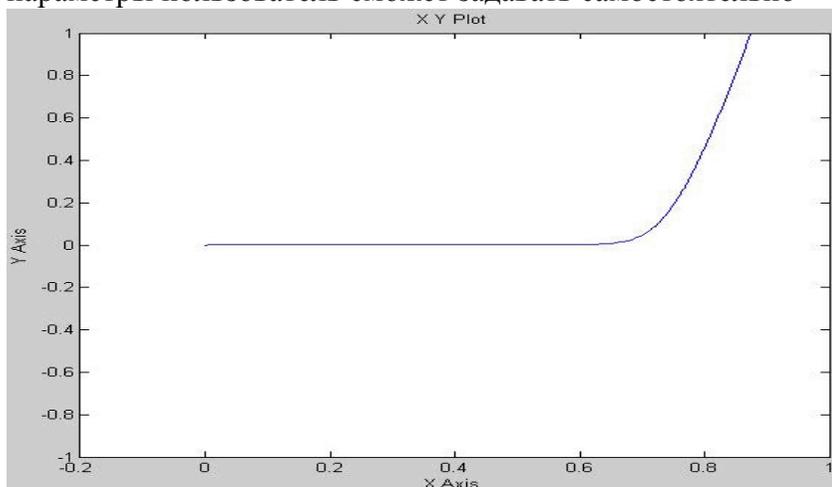
Всю систему поместим в один блок - Subsystem. Назовем наш блок с диодом DiodeModel. Создадим маску диода

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)

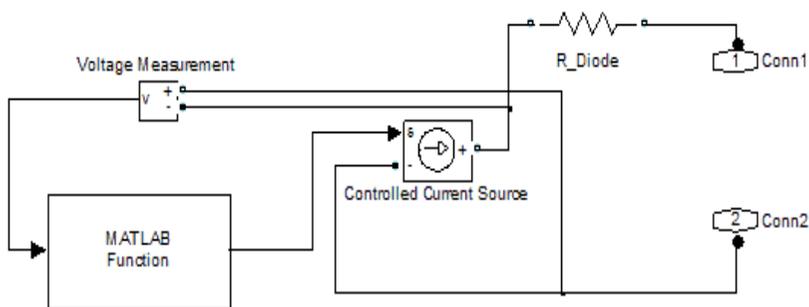


Схема стабилитрона(моделируемого диода)

Лабораторная работа № 3

Работа в поисковой системе.

Цель работы: Изучить основные принципы поиска, информации в сети Интернет. Научиться находить требуемую информацию.

Задание:

1. Получите у преподавателя или согласуйте с ним свое индивидуальное задание - тему для поиска.

2. Проработать тему с использованием следующих поисковых машин:

<http://www.aport.ru>,

<http://www.yandex.ru>,

<http://www.rambler.ru>.

3. Сделать более глубокий поиск по заданной теме на основе технологии расширенного поиска.

4. Проработать заданную тему по разделам каталогов указанных выше поисковых машин.

5. Отобрать и сохранить на дискете результаты поиска по заданной теме.

Теоретическая часть

Отправляясь в увлекательное путешествие по Сети, включающей в себя колоссальный объем информации, вы рискуете заблудиться и не сумеете отыскать нужной информации. Интернет не только является крупнейшей базой данных, но и, безусловно, активизирует полёт творческой мысли, в отличие от других средств массовой информации, которые оставляют вам лишь роль пассивного наблюдателя.

Не имеет смысла перечислять различные объекты поиска в Интернете. Они могут быть самыми разнообразными. Отличия в схемах поиска научной статьи и детской игрушки могут оказаться совсем незначительными.

Представьте себе, что вы - посетитель огромной библиотеки, хранящей многие тысячи томов. Вы оказываетесь на распутье, от которого (по традиции) ведут три дороги:

- Отправиться на поиск по известным вам ключевым словам;
- Полистать предлагаемые тематические каталоги,
- Обратиться в специальную службу по поиску информации.

Прежде, чем решиться на окончательный выбор пути, попытайтесь зримо представить себе объект поиска.

Помните, - чтобы грамотно задать вопрос, надо знать большую часть ответа.

Если Вам удастся определить искомый объект несколькими признаками (словами или числами), - решайтесь на самостоятельный поиск с помощью специальных поисковых машин. 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 или www.intel.org.

В случае если ваш интерес имеет весьма размытые границы и определяется лишь некой областью человеческих знаний, - обратитесь к тематическому каталогу (смотрите раздел 2.3).

Поисковые машины

Поисковая машина - *поисковая система с формируемой программным "роботом" базой данных, содержащей информацию об информационных ресурсах.*

Все поисковые машины, предназначенные для сети Интернет, имеют более или менее схожие принципы работы. Поиск в такой системе проводится по запросу, составляемому пользователем, состоящему из набора ключевых слов или фразы, заключенной в кавычки. Индекс формируется и поддерживается в актуальном состоянии роботами-индексировщиками.

Невероятно распространено заблуждение, что во время поиска по вашему запросу избранная поисковая машина обходит миллионы известных ей адресов и сверяет соответствие ключевых слов содержимому "свежих" гипертекстов. Не так! 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.

Что можно делать с полученными результатами? Если название и описание документа соответствует вашим требованиям, можно немедленно перейти к его первоисточнику по ссылке. Это удобнее делать в новом окне, чтобы сохранить окно с результатами запроса к поисковой машине.

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. Для этого вы выбираете особо понравившийся документ и указываете его системе в качестве образца для подражания. Однако, автоматизация определения схожести - весьма нетривиальная задача, и зачастую эта функция работает неадекватно вашим ожиданиям. 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.

Поскольку поисковые машины существуют в Интернете, в основном, за счёт публикуемой рекламы, как правило, самые популярные системы поиска могут предоставить вам наилучшие возможности. Для рядового пользователя услуги поисковых серверов, как правило, предоставляются бесплатно. Достаточно лишь указать адрес поисковой

системы в рабочей строке браузера или обратиться к ней через каталог закладок.

Главной задачей любой поисковой машины 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* (русскоязычная) - <http://www.aport.ru>
- *Yandex* (русскоязычная) - <http://www.yandex.ru>
- *Rambler* (русскоязычная) - <http://www.rambler.ru>
- *Yahoo!* (англоязычная) - <http://www.yahoo.com>
- *AltaVista* (англоязычная) - <http://www.altavista.com>
- *InfoSeek* (англоязычная) - <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**.

Организация поиска

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 или www.ya.ru (в варианте упрощенной загрузки). После загрузки поисковой системы в строке для поиска вводите запрос (ключевую фразу), который представляет собой строку текста (на русском, английском или любом другом языке) и нажимайте кнопку *Найти*.

Советы по поиску

Ищите больше, чем по одному слову. Слово "психология" или "продукты" дадут при поиске поодиночке большое число бессмысленных ссылок. Добавьте одно или два ключевых слова, связанных с искомой темой. Например, "психология Юнга" или "продажа и покупка продовольствия". Рекомендуем также сужать область вашего вопроса. Если вы интересуетесь автомобилями ГАЗ, то запросы "автомобиль Волга" или "автомобиль ГАЗ" выдадут более подходящие документы, чем просто "легковые автомобили".

Проверяйте орфографию. Если поисковая машина не нашла ни одного документа, то вы, возможно, допустили орфографическую ошибку в написании слова. Проверьте правильность написания. Если вы использовали при поиске несколько слов, то посмотрите на количество каждого из слов в найденных документах (перед их списком после фразы "Результат поиска"). Какое-то из слов не встречается ни разу? Скорее всего, его вы и написали неверно.

Используйте синонимы. Если список найденных страниц слишком мал или не содержит полезных страниц, попробуйте изменить слово. Например, вместо "рефераты" возможно больше подойдет "курсовые работы" или "сочинения". Попробуйте задать для поиска три-четыре слова-синонима сразу. Для этого перечислите их через вертикальную черту (|). Тогда будут найдены страницы, где встречается хотя бы одно из них. Например, вместо "фотографии" попробуйте "фотографии|фото|фотоснимки".

Не пишите большими буквами. Начиная слово с большой буквы, вы не найдете слов, написанных с маленькой буквы, если это слово не первое в предложении. Поэтому не набирайте обычные слова с Большой Буквы, даже если с них начинается ваш запрос Яндекс. Заглавные буквы в запросе рекомендуется использовать только в именах собственных. Например, "группа Черный кофе", "телепередача Здоровье".

Найти похожие документы. Если один из найденных документов ближе к искомой теме, чем остальные, нажмите на ссылку "найти похожие документы". Ссылка расположена под краткими описаниями найденных документов. Яндекс проанализирует страницу и найдет документы, похожие на тот, что вы указали. Но если эта страница была стерта с сервера, а Яндекс еще не успел удалить ее из базы, то вы получите сообщение "Запрошенный документ не найден".

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. Например, если вам нужно описание Парижа, а не предложения многочисленных турагентств, имеет смысл задать такой запрос "путеводитель по Парижу -агентство -тур". Плюс стоит использовать в том случае, когда нужно найти так называемые стоп-слова (наиболее частотные слова русского языка, в основном это местоимения, предлоги, частицы). Чтобы найти цитату из Гамлета, надо задать запрос "+быть или +не быть".

Попробуйте использовать 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".

Результаты поиска

После того, как вы задали запрос, Яндекс выведет список ссылок на документы, наиболее точно ему

соответствующие. На этой странице результатов вы обнаружите некоторые специальные обозначения и ссылки, предназначенные облегчить просмотр и сортировку найденных страниц. На рисунке приведен общий вид страницы результатов:

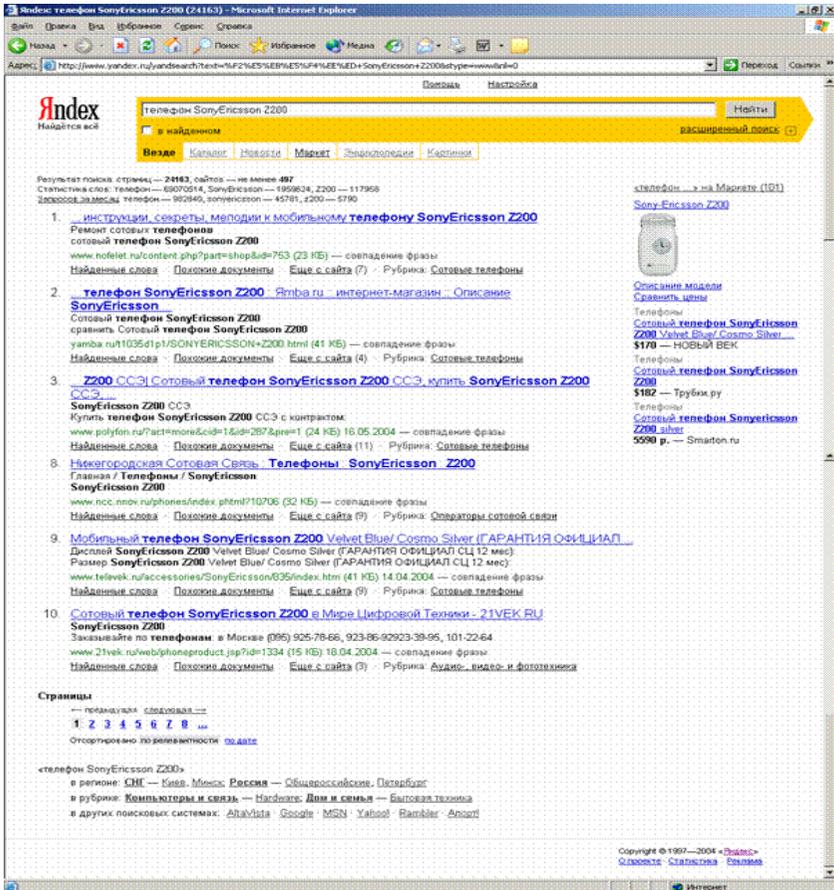
1. Поисковая форма. Главный ее элемент - строка запроса. При желании можно искать только в результатах предыдущего поиска ("искать в найденном"). Можно также задать запрос по любому из источников информации ("Везде", "Каталог", "Новости", "Маркет", "Энциклопедии", "Картинки").

2. Статистика поиска: число найденных документов и частота заданных в запросе слов. По ссылке "страниц" можно перейти на несгруппированную по серверам выдачу.

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

4. Результаты поиска по вебу. Для каждого документа выдается следующая информация: заголовок - 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. Возможность поискать в других русскоязычных поисковых машинах.

7. 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.

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

Расширенный поиск

Яндекс обладает развитым языком запросов, позволяющим осуществлять тонкий поиск. Для того, чтобы воспользоваться широким спектром возможностей, используйте страницу "расширенный поиск", где большая часть настроек Яндекса задается простым образом.

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. Яндекс умеет определять язык документа. Вы можете 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

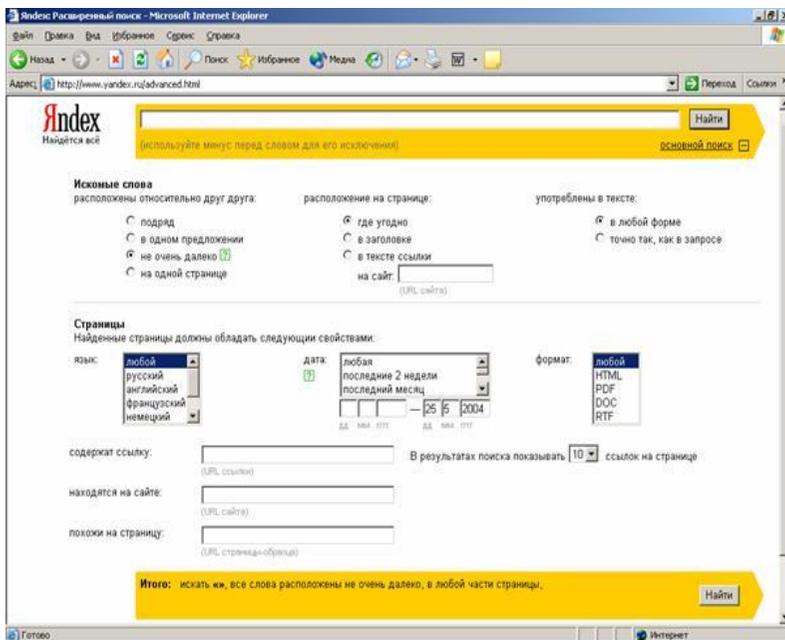
Каталог - *поисковая система с классифицированным по темам списком аннотаций со ссылками на web-ресурсы. Такая классификация, как правило, проводится людьми.*

Поиск в каталоге очень удобен и 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. База данных ссылок (индекс) каталога обычно

имеет ограниченный объем, заполняется вручную специалистами создающими каталог. 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
- **Weblist** (русскоязычный) - www.weblist.ru
- **Созвездие интернет** (русскоязычный) - www.stars.ru
- **Yahoo** (англоязычный) - www.yahoo.com
- **Magellan** (англоязычный) - www.mckinley.com.



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.

Лабораторная работа № 4.

Использование услуг электронной почты.

Цель работы: Изучение программ для электронной почты, приобретение навыков использования электронной почты в научных целях.

Задание:

1. Ознакомьтесь с программами предназначенных для электронной почты
2. Запустите программу MAIL в сети Интернет.
3. Отправьте сообщение почте специализированной кафедры.
4. Дайте название темы отправляемой почте.

Теоретическая часть.

Электронная почта – самая эффективная из служб Интернет в смысле затрат. Обычно она достигает любой точки земного шара за несколько минут. Электронная почта не использует географическую адресацию, и ее адреса состоят из двух частей:

имя_абонента@адрес_компьютера_в_Интернет

Например, shavkat@tgtu.sarcor.uz. Прописные и строчные буквы в почтовом адресе не различаются.

Электронную почту чаще всего используют следующим образом. Соединившись с провайдером, запускают почтовую программу и получают накопившуюся почту. Подготовив ответы и новые письма, отправляют их. Если какое-то из писем не может быть сразу доставлено адресату, например, если в данный момент не работает сервер, на котором находится его почтовый ящик, то письмо будет поставлено в очередь на отправку. Каждые 10-15 минут будут производиться новые попытки. Если через несколько часов сообщение все еще не отправлено, то отправитель получает первое (предварительное) уведомление, к которому приложена копия отправленного сообщения. Это уведомление носит информационный характер и не требует от отправителя никакой реакции. Попытки отправить сообщение будут продолжаться еще несколько дней. Если и они будут безуспешны, то отправитель получает второе (окончательное) уведомление с копией отправленного сообщения. Таким образом, любое отправленное сообщение не может затеряться и исчезнуть бесследно.

Почтовые программы для персональных компьютеров используют разные протоколы для приема и отправки почты. При отправке почты программа взаимодействует с сервером исходящей почты, или SMTP-сервером, по протоколу SMTP. Прием почты осуществляется по протоколу POP3. Используется также протокол IMAP, который позволяет выборочно копировать письма, пришедшие на адрес получателя, с почтового сервера на компьютер получателя.

Существует очень много почтовых программ, значительная часть из которых бесплатна. Наиболее распространены следующие программы:

- **Microsoft Internet Mail** – поставляется в составе операционной системы, начиная с Windows 95; более свежая версия может быть скопирована с сервера www.microsoft.com. Эта программа наиболее корректно работает с письмами на русском языке, однако не поддерживает IMAP-протокол.

- **Outlook Express**. Для работы с электронной почтой в Internet Explorer имеется компонент Outlook Express. Для его запуска надо выбрать команду **Сервис►Почта** и новости и далее команду в зависимости от вида работы.

Для создания нового сообщения выбрать команду **Создать сообщение**. В открывшемся окне, выбрав команду **Формат►Вид кодировки**, определить кодировку символов сообщения, ввести адрес получателя, указать тему сообщения и ввести текст сообщения. Для вставки в текст рисунка выбрать команду **Вставка►Рисунок**, открыть окно **Рисунок**, выбрать файл, указать параметры его расположения в бланке сообщения и щелкнуть **ОК**.

Выделив фрагмент текста в сообщении и выбрав команду **Вставка►Гиперссылка**, можно определить адрес Web-документа в Интернет, который будет раскрываться при щелчке мышкой по этому тексту. Командой **Формат►Фон** можно определить фоновый рисунок и звук, цвет фона сообщения.

Для отправки с сообщением какого-нибудь файла щелкнуть кнопку **Вложить** на стандартной панели

инструментов Outlook Express, в диалоговом окне **Вставка** выбрать папку и файл и нажать кнопку Вложить. После этого на бланке сообщения появится поле **Присоединить.**, в котором будет имя вложенного файла.

Для отправки подготовленного сообщения щелкнуть кнопку **Отправить**. Можно подготовить в папке **Исходящие** несколько сообщений, завершая их создание выбором команды **Файл►Отправить позже**. В этом случае все подготовленные сообщения будут отправлены при первом подключении к сети.

При подключении к сети Outlook Express загрузит сообщения с почтового сервера. Для получения сообщений выбрать команду **Сервис►Почта и новости►Чтение почты**. Для просмотра полученных сообщений щелкнуть значок папки **Входящие** на панели Outlook Express или в списке папок. Для просмотра сообщений в отдельном окне дважды щелкнуть по нему в списке сообщений. Для ответа на сообщение можно щелкнуть кнопку **Ответить** на стандартной панели инструментов.

Outlook Express поддерживает IMAP-протокол.

- Netscape Messenger – бесплатная программа фирмы Netscape, входит в состав Netscape Communicator. Обладает достоинствами Outlook Express.
- Eudora – почтовая программы фирмы Qualcomm. Полная версия программы продается за деньги. Не рекомендуется для переписки по-русски.

При первом запуске почтовой программы автоматически запускается мастер по настройке, который запрашивает минимальную необходимую информацию: адрес и название серверов входящей и исходящей почты. Может отдельно запрашиваться имя пользователя – эта та часть почтового адреса, которая находится слева от знака «@». При первой проверке почты и всех последующих, если не пометить галочкой «Запомнить пароль», программа приглашает ввести

пароль. Если программа поддерживает как POP3, так и IMAP, при настройке следует выбрать POP3.

Можно воспользоваться почтой и не имея почтовой программы. Существует большое количество серверов, которые предлагают завести бесплатный почтовый ящик и позволяют работать с почтой, используя только браузер. Это службы mail.ru, www.tomcat.ru, hotmail.com, mail.yahoo.com и другие. Такое использование почты имеет определенные достоинства. Можно менять провайдеров, не меняя свой адрес электронной почты. Можно просматривать свою почту с любого компьютера, подключенного к Интернет, и не тратить время на конфигурирование почтовой программы. Недостатком такой почты является то, что нельзя минимизировать время подключения к Интернет в той мере, в какой позволяют почтовые программы. К тому же общедоступные почтовые сервера зачастую перегружены.

Лабораторная работа № 5.

Создание Web-сайта в языке HTML.

Цель работы: Изучить основные принципы работы языка разметки гипертекста HTML и получить навыки создания web-страниц.

Задание:

1. Изучить основы работы с языком разметки гипертекста HTML
2. Ознакомиться с процессом создания web-страниц
3. Создать web-страницы по заданной теме

Теоретическая часть

Сеть Интернета представляет собой гигантский набор информации, основанной на гипертекстовых документах. 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).

Неудачный, так и не реализованный стандарт языка HTML версии 3.0 сменил стандарт 3.2. В его разработке участвовали такие компании, как Netscape Communications Corporation, Microsoft, IBM, Novell, SoftQuad, Sun Microsystems и Spyglass.

На сегодняшний день выпущен еще один стандарт HTML версии 4.0, в который добавлены многие полезные расширения.

Следует отметить, что HTML, вопреки распространенному заблуждению, ни в коей мере не является языком программирования. HTML – это типичный 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.

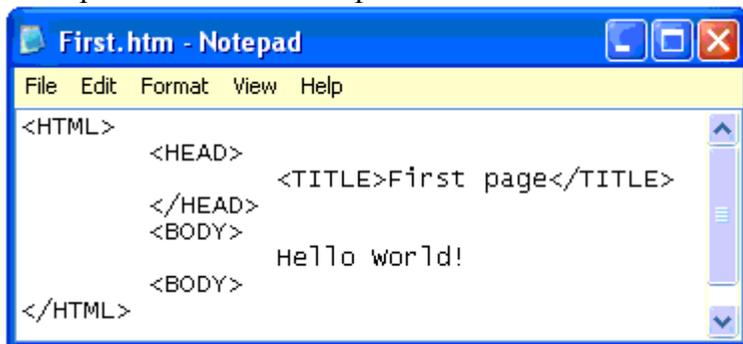
Создание страницы HTML.

Страницы HTML представляют собой обычные текстовые файлы, поэтому на первом этапе Вам достаточно простейшего текстового редактора Microsoft Notepad.

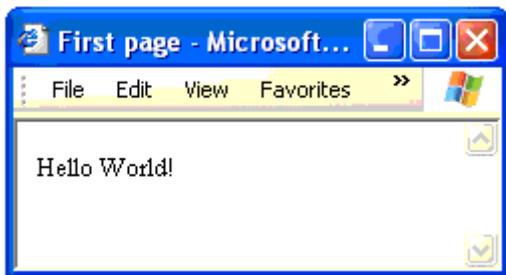
Запустите программу Notepad и наберите текст, приведенный ниже:

```
<HTML>
  <HEAD>
    <TITLE>First page</TITLE>
  </HEAD>
  <BODY>
    Hello World!
  </BODY>
</HTML>
```

Сохраните этот текст в файл с именем First.htm.



Откройте папку, в которую Вы сохранили файл First.htm и дважды щелкните левой кнопкой мыши значок файла – автоматически запустится браузер, в окне которого отобразится содержимое вашего документа.



Структура документа HTML.

Любой 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>
```

Структура заголовка.

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.

Example: заголовок.

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

Форматирование и выравнивание текста.

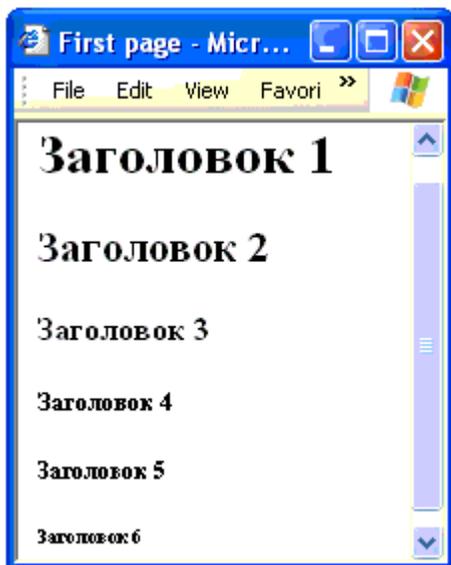
Форматирование текста является самой простой и одновременно самой мощной частью HTML. Поговорим о том, как форматировать документ. Следует отметить, что форматирование и оформление – не одно и то же. Форматирование задает структуру, а оформление – внешний вид. Для оформления документа в HTML 4.0 используется CSS (Cascading Style Sheet – Каскадные стили). О CSS мы поговорим позднее.

Тэги заголовка.

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

Example: 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>
```



Основные тэги форматирования.

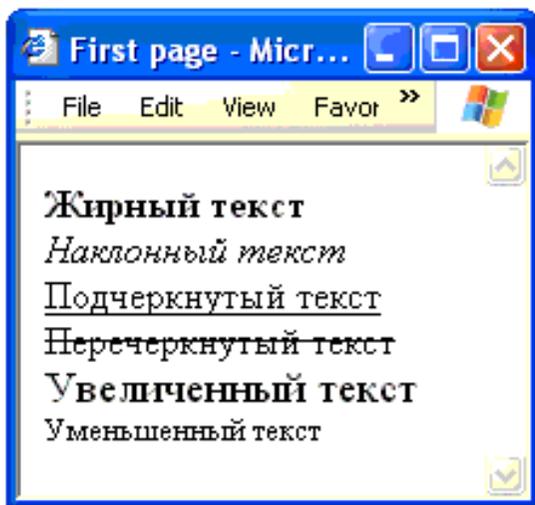
- **
**. Тэг разрыва, который равнозначен одному возврату каретки.
- **<P>**. Тэг абзаца, используемый для обозначения абзаца.
- **<P> ... </P>**. Способ обозначения абзаца с помощью открытия/закрытия.
- **<PRE> ... </PRE>**. Тэг предварительного форматирования текста, используется для сохранения количества пробелов в тексте.
- **<NOBR> ... </NOBR>**. Если вы хотите использовать строку без естественного разрыва.

Начертания шрифта.

Очень часто для выделения отдельных предложений, слов или символов используют шрифт, вид которого отличен от вида основного текста.

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>
```



Знакомство с Атрибутами.

Множество тегов HTML также принимают параметры, называемые атрибутами. Атрибуты используются для указания дополнительной информации Web-браузеру о том, как применять этот тег элемента. Атрибуты - это имена, которым через знак равенства (=) присваиваются определенные значения. Например:

```
<BODY>
```

```
  <P align="center" title="Indian">Remain close to the  
  great spirit.</P>
```

```
  <P title="Indian" align="center">Remain close to the  
  great spirit.</P>
```

```
  <P align="center" title="Indian" >Remain close to the  
  great spirit.</P>
```

```
<BODY>
```

Результат этого кода одинаков для каждой строки.

Атрибут title отвечает за текст подсказки, всплывающей при наведении на объект мыши.

Выравнивание текста.

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 ``.

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

Example: Элемент DIV может быть вложенным.

```
<html>
```

```
<head>
```

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

```
<body>
```

```
  <DIV align="right">
```

```
Hello world!
```

```
</DIV>
```

```
  <DIV align="center">
```

```
Hello world!
```

```
  </DIV>
```

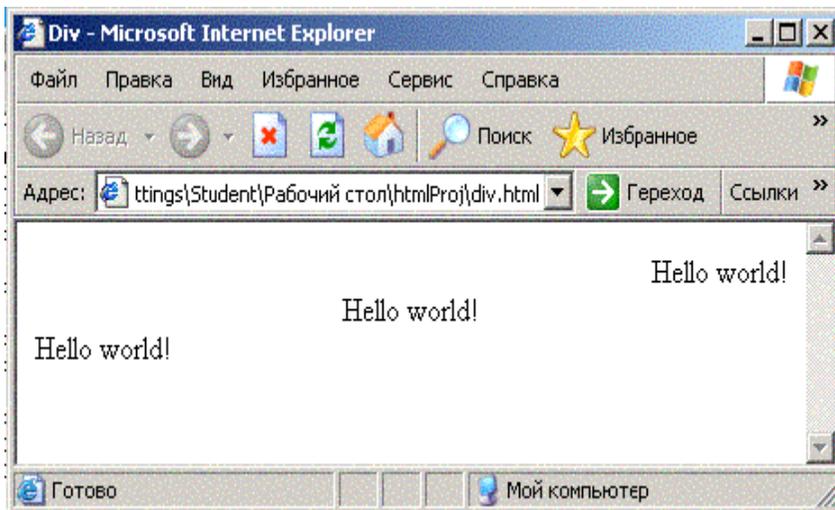
```
  <DIV align="left">
```

```
Hello world!
```

```
  </DIV>
```

```
</body>
```

```
</html>
```



Выравнивание текста с помощью элемента абзаца.

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:

- ` ... `. Неупорядоченный, или маркированный список.

- ** ... **. Упорядоченный, или нумерованный список.
- ****. 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>
  Неупорядоченные списки:
  <ul>
    <li>Стол
    <li>Стул
  </ul>
```

```
Упорядоченные списки:
<ol>
  <li>Стол
  <li>Стул
</ol>
```

```
Списки определений:
<dl>
  <dt>windows
  <dd> Operation system of the Microsoft
  <dt>OS/2
  <dd> Operation system of the IBM
</dl>
```

```
Вложенные списки:
<ul>
```

```

    <li>Стол
      <ol>
        <li>Деревянный.
        <li>Пластмассовый.
      </ol>
    </li>
  </ul>
</body>
</html>

```

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
- CIRCLE - 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:

1	арабские цифры (по умолчанию)	1, 2, 3,
a	латинский алфавит, прописные	a, b, c,
A	латинский алфавит, заглавные	A, B, C,
i	римские цифры, прописными	i, ii, iii,
I	римские цифры, заглавными	I, II, III,

Связывание страниц.

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.

Первоначально названный методом гиперссылок, технический приём связи с другими документами позволяет работать не только с текстовыми ссылками. 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> ... `. - 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.

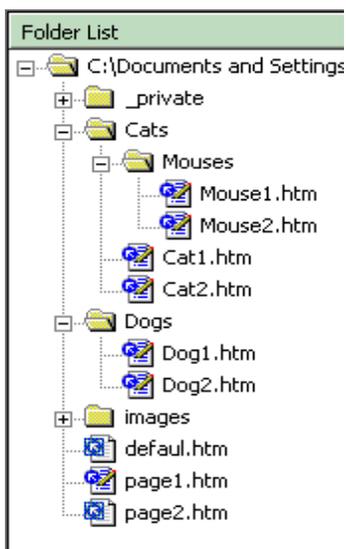
Абсолютное связывание.

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.

Абсолютные ссылки важны при задании в элементах привязки адресов не на вашем узле, а на узлах, находящихся на других серверах.

Example: Абсолютная ссылка.



Если вы хотите default.htm связать абсолютной ссылкой с:

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

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

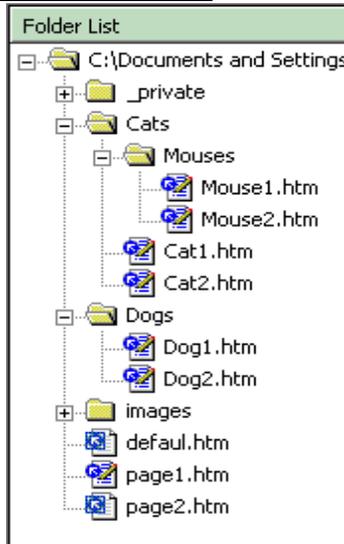
Cat1 – <A HREF=
"http:\\Server\FirstSite\Cats\Cat1.htm">Cat1

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

Относительное связывание.

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.

Example: Относительная ссылка.



**Если вы хотите default.htm связать
относительной ссылкой с:**

Page1 – Page1

Dog1 – Dog1

Cat1 – Cat1

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

**Если вы хотите Mouse1.htm связать
относительной ссылкой с:**

Page1 – Page1

Dog1 – Dog1

Cat1 – Cat1

Mouse2 – Mouse2

**Если вы хотите Cat1.htm связать
относительной ссылкой с:**

Page1 – Page1

Dog1 – Dog1

Cat2 – Cat2

Mouse1 – Mouse1

Внутренние ссылки.

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 `#`.

Example: Использование внутренних ссылок.

```
<p>здесь Вы найдёте описания к следующим моделям<br>
<a href="#OpelOmega">Opel Omega</a>
<!------- some text ----->>
<a name="OpelOmega"></a>
```

Почтовые ссылки.

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

Example: Использование почтовой ссылки.

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

Изображения.

Для размещения на странице изображений используется тэг ``. Этот тэг работает самостоятельно, т.е. закрывать его (``) не нужно.

Для тэга `` 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:

```

```

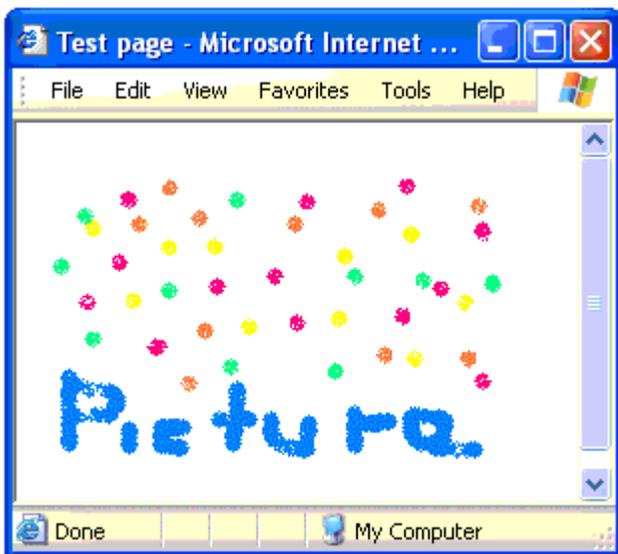
Изображения всегда помещают внутри раздела `body` документа HTML.

Example: Вставка изображения.

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

<body>

</body>
</html>
```



Работа со шрифтами.

Элемент Font.

Синтаксис ` --text-- ` 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.

Example: Использование атрибутов

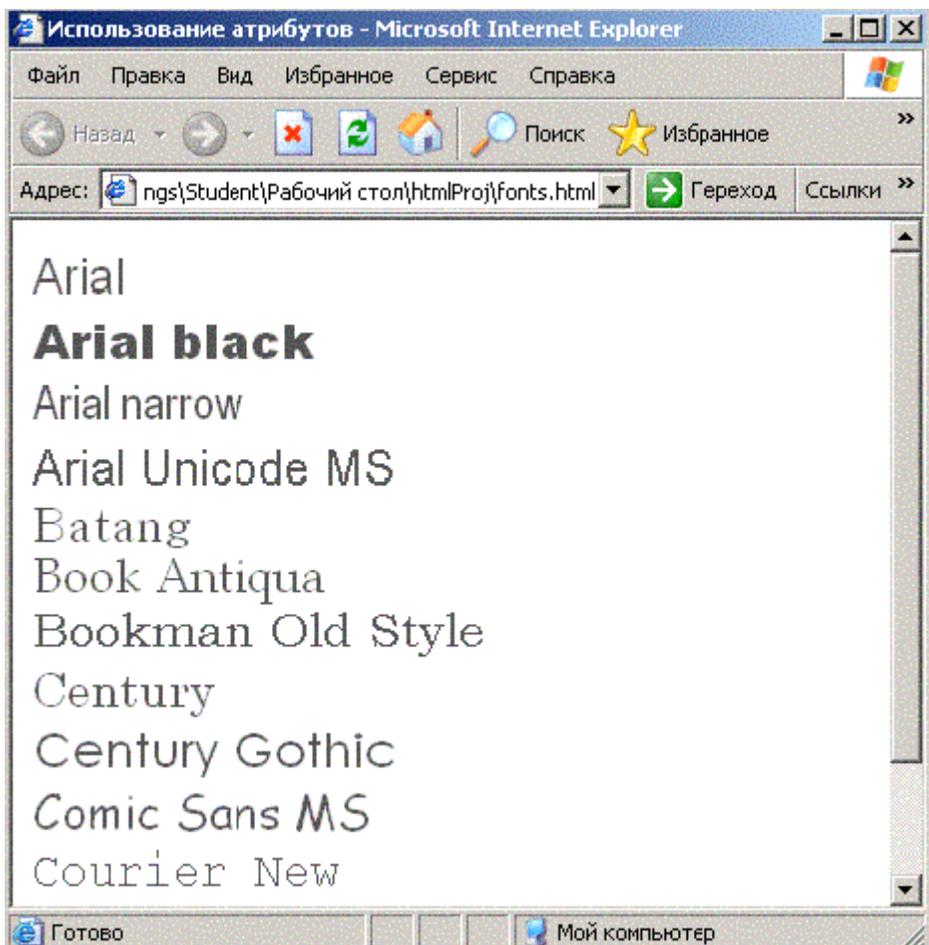
```
<html>
```

```
<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>
```



Таблицы.

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.

Example: Размещение таблицы.

```
<html>
<head>
<title> Применение свойств стилей</title>
</head>
<body>
<table>
<tr>
<td>
Первая ячейка таблицы.
</td>
<td>
Вторая ячейка таблицы.
</td>
</tr>
</table>
</body>
</html>
```

Атрибуты элементов 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).

Example: использование атрибутов таблицы.

```
<html>
<head>
<title>Использование атрибутов таблицы</title>
</head>
<body>
<table border="1" align="right" cellspacing="20"
cellspadding="15">
<tr>
<td>
Первая ячейка таблицы.
</td>
<td>
Вторая ячейка таблицы.
</td>
</tr>
</table>
</body>
</html>
```

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;

bgcolor="x" – цвет фона ячейки;

background="url" – фоновый рисунок в ячейке.

Контрольные вопросы

1. Что такое HTML?
2. Для чего используется HTML?
3. Какая текущая версия HTML?
4. Что такое тег?
5. Каковы требования к оформлению тегов?
6. Как выделить теги из общего текста описания страницы?
7. Поясните термины открывающий тег и закрывающий тег?
8. Назовите основные теги языка HTML и их назначение?

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. Яндекс - <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

Лабораторная работа №1	
Работа с программой Fine Reader 7.0.	3
Лабораторная работа №2	
Решение научно-технических задач с помощью систем MathCad, Matlab	16
Лабораторная работа №3	
Работа в поисковой системе.	24
Лабораторная работа №4	
Использование услуг электронной почты.	35
Лабораторная работа №5	
Создание Web-сайта в языке HTML.	38
Литература	63