

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**Методические рекомендации для
самостоятельной работы обучающихся по дисциплине**

Б1.Б.19 Информационные технологии

Направление подготовки (специальность) 09.03.01 Информатика и вычислительная техника

Профиль образовательной программы “Автоматизированные системы обработки информации и управления”

Форма обучения очная

СОДЕРЖАНИЕ

1. Организация самостоятельной работы.....	3
2. Методические указания по самостоятельному изучению вопросов	4
3. Методические рекомендации по подготовке к занятиям	10
3.1 Общие сведения об информационных технологиях.....	10
3.2 Классификация информационных технологий.	10
3.3 Использование Visual Basic for Applications.	10
3.4 Понятие базы данных.....	11
3.5 Архитектура СУБД.....	11
3.6 СУБД ACCESS.....	11
3.7 Работа с MathCAD.....	12
3.8 Решение систем линейных алгебраических уравнений в MathCAD.	12
3.9 Аппаратные средства связи.	12
3.10 Локальные и глобальные вычислительные сети.....	12
3.11 Организация беспроводных сетей.	12
3.12 Работа в HTML.....	12

1. ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1.1 Организационно-методические данные дисциплины

№ п.п . .	Наименование темы	Общий объем часов по видам самостоятельной работы <i>(из табл. 5.1 РПД)</i>				
		Подготовка курсовой работы (работы)	подготовка реферата/эссе	индивидуаль- ные домашние задания (ИДЗ)	самостоятель- ное изучение вопросов (СИВ)	подготовка к занятиям (ПкЗ)
1	2	3	4	5	6	7
1	Раздел 1 Введение в информационные технологии				12	12
2	Тема 1 Общие сведения об информационных технологиях					4
3	Тема 2 Классификация информационных технологий					4
4	Тема 3 Использование Visual Basic for Applications				12	4
5	Раздел 2 Базы данных и информационные технологии				12	12
6	Тема 4 Понятие базы данных					4
7	Тема 5 Архитектура СУБД					4
8	Тема 6 СУБД ACCESS				12	4
9	Раздел 3 Основы MathCAD				10	12
10	Тема 7 Работа с MathCAD					6
11	Тема 8 Решение систем линейных алгебраических уравнений в				10	6

	MathCAD					
12	Раздел 4 Информационные технологии передачи данных				10	10
13	Тема 9 Аппаратные средства связи.					4
14	Тема 10 Локальные и глобальные вычислительные сети					2
15	Тема 11 Организация беспроводных сетей.					2
16	Тема 12 Работа в HTML				10	2

2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ ВОПРОСОВ

2.1 Использование Visual Basic for Applications

Объектно-ориентированные языки программирования

1. Среда VBA.

VBA (Visual Basic for Applications) — это диалект языка Visual Basic, расширяющий возможности Visual Basic и предназначенный для работы с приложениями Microsoft Office и другими приложениями от Microsoft и третьих фирм.

2. Создание макросов.

Алгоритм создания макроса для поставленной задачи:

1. Выберите Сервис/Макрос, Начать запись.

2. В поле Имя макроса введите имя для макроса. Первым символом имени макроса должна быть буква. В имени макроса не допускаются пробелы; в качестве разделителей слов можно использовать знаки подчеркивания.

3. Для того чтобы запускать макрос с помощью сочетания клавиш, введите букву в поле Сочетание клавиш. Допускается использование сочетаний CTRL+ буква (для строчных букв) или CTRL+SHIFT+ буква (для прописных букв), где буква — любая буквенная клавиша на клавиатуре. Не выбирайте стандартного сочетания клавиш, так как выбранное сочетание клавиш подавляет стандартные сочетания клавиш Microsoft Excel на то время с данной книгой.

4. В поле "Сохранить" выберите книгу, в которой требуется сохранить макрос. Сохраните макрос в «Эта книга». Для создания краткого описания макроса, введите необходимый текст в поле Описание. На скриншоте представлен пример заполнения окна диалога «Запись макроса».

2.2 СУБД ACCESS

FoxPro

1. Создание таблиц и форм.

В Access 2007 можно вводить данные непосредственно в таблицу в режиме таблица. Но обычно для ввода данных в БД Access 2007 используют формы (forms). Form ускоряет работу с базой данных. Form в БД - это структурированное интерактивное окно с элементами управления, в котором отображаются поля одной или нескольких таблиц или запросов.

Форму можно использовать для ввода, изменения или отображения данных из таблицы или запроса. В Microsoft Office Access 2007 предусмотрены новые средства, помогающие быстро создавать forms, а также новые типы форм и функциональные возможности.

Формы в БД Access можно создавать с помощью различных средств:

- инструмента Form;
- инструмента Разделенная form;
- инструмента Несколько элементов;
- инструмента Пустая form;
- Мастера form;
- Конструктора form.

2. Создание запросов и отчетов.

В СУБД Access применяются различные типы запросов: на выборку, на обновление, на добавление, на удаление, перекрестный query, выполнение вычислений, создание таблиц. Наиболее распространенным является query на выборку. Применяются два типа запросов: query по образцу (QBE) и query на основе структурированного языка запросов (SQL).

2.3 Решение систем линейных алгебраических уравнений в MathCAD MatLAB

1. Формулы и текст.

Формульный редактор MathCAD позволяет быстро и эффективно вводить и изменять математические выражения. Тем не менее, некоторые аспекты его применения не совсем интуитивны, что связано с необходимостью избежать ошибок при расчетах по этим формулам. Поэтому не пожалейте немного времени на знакомство с особенностями формульного редактора, и впоследствии при реальной работе вы сэкономите гораздо больше.

Перечислим элементы интерфейса редактора MathCAD:

- указатель мыши (mouse pointer) – играет обычную для приложений Windows роль, следя за движениями мыши;
- курсор – обязательно находится внутри документа в одном из трех видов:
 - курсор ввода (crosshair) - крестик красного цвета, который отмечает пустое место в документе, куда можно вводить текст или формулу;
 - линии ввода (editing lines) – горизонтальная (underline) и вертикальная (insertion line) линии синего цвета, выделяющие в тексте или формуле определенную часть;
 - линия ввода текста (text insertion point) – вертикальная линия, аналог линий ввода для текстовых областей;
- местозаполнители (placeholders) – появляются внутри незавершенных формул в местах, которые должны быть заполнены символом или оператором:
 - местозаполнитель символа – черный прямоугольник;
 - местозаполнитель оператора – черная прямоугольная рамка.

2. Символьные преобразования.

Символьный знак равенства позволяет Mathcad выйти за рамки численного вычисления выражений. Можно подумать, что это обычный знак =. В отличие от

обычного знака равенства, который всегда возвращает число, символный знак равенства может возвращать выражение.

2.4 Работа в HTML

Содержание урока:

Заголовок документа - тег HEAD

Тело документа - тег BODY

Структура html-документа состоит из трех пар тегов:

```
<html>
    <head>
        Заголовок документа
    </head>

    <body>
        Тело документа
    </body>

</html>
```

Теги `<html> </html>` являются контейнером для всех остальных, т.е в них помещаются все остальные. Таким образом, ваш документ должен начинаться с тега `<html>`, а заканчиваться тегом `</html>`.

Сам документ условно разделен на две части - заголовок документа (теги `<head> </head>`) и тело документа (теги `<body> </body>`).

Заголовок документа - тег HEAD и его элементы

Заголовок документа содержит служебную информацию и не влияет на внешний вид документа. Его задачей является предоставление браузеру пользователя или серверу информации о том, как отобразить ваш документ.

Title

Единственным обязательным элементом заголовка документа являются теги `<title> </title>`. Они необходимы, чтобы дать документу название, оно отражается в заголовке окна браузера. Например, если написать следующий код:

```
<html>
    <head>
        <title>Заголовок документа</title>
    </head>

    <body>
        Тело документа
    </body>

</html>
```

В окне браузера он будет выглядеть так:

* Если вы не знаете, как создать html-страницу, то ознакомьтесь сначала с серией уроков [Делаем сайт - первые шаги](#) *

Теоретически название документа может иметь неограниченную длину, на практике рекомендуется ограничиться 60 символами.

Не давайте своим документам безликие названия, типа "Первая страница", во-первых название документа должно характеризовать его содержимое, а во-вторых, содержимое тегов `<title> </title>` играет не последнюю роль при оптимизации и раскрутке сайта.

Base

Одиночный тег `<base>` служит для указания полного URL-адреса документа. Зачем это нужно? Представьте, что блуждая по интернету, вы сохранили какую-нибудь html-страницу себе на компьютер, с тем, чтобы просмотреть ее позже. Все картинки на этой страницы превратятся в красные крестики. Но если вы не отключены от сети, а на странице присутствует тег `<base>`, то браузер будет знать, где искать необходимый файл, найдет его и загрузит картинки.

У этого тега один атрибут `href`, значением которого является адрес страницы. Пример кода:

```
<html>
  <head>
    <title>Структура html</title>
    <base href="http://www.my_site.ru/">
  </head>

  <body>
    Тело документа
  </body>
</html>
```

Link

Одиночный тег `<link>` необходим для подключения внешних файлов. Например, если вы будете использовать каскадную таблицу стилей, то ее удобнее хранить в отдельном файле и подключать этот файл ко всем страницам сайта.

У тега `<link>` несколько атрибутов:

`href` - указывает URL-адрес подключаемого файла.

`rel` - указывает на тип отношения данного документа к внешнему (например: `rel="stylesheet"` указывает, что внешний файл определяет стиль текущего документа).

`type` - указывает тип и параметры присоединенной таблицы стилей.

Пример кода:

```
<html>
  <head>
    <title>Структура html</title>
    <base href="http://www.my_site.ru/">
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>

  <body>
    Тело документа
  </body>
</html>
```

Meta

Информация в этом теге не имеет никакого отношения к HTML, однако ее использование очень удобно для решения ряда задач:

- указание кодировки страницы, например, для русского текста в кодировке Windows:

```
<meta http-equiv="Content-Type"  
      content="text/html; charset=windows-1251">
```

- указание ключевых слов страницы (используется при оптимизации страниц):

```
<meta http-equiv="KEYWORDS"  
      content="тег, структура html, заголовок страницы">
```

- указание краткого описания страницы:

```
<meta http-equiv="DESCRIPTION"  
      content="Описание структура html и элементов заголовка">
```

Возможны и другие варианты. Рассмотрим параметры тега *<meta>*:

http-equiv - определяет свойство тега (тип контента, ключевые слова, описание и т.д.)

name - используется для дополнительного описания тега, если отсутствует, то считается эквивалентным параметру *http-equiv*.

content - значение параметра *http-equiv*.

Пример кода:

```
<html>  
  
  <head>  
    <title>Структура html</title>  
    <base href="http://www.my_site.ru/">  
    <link rel="stylesheet" type="text/css" href="style.css">  
    <meta http-equiv="Content-Type"  
          content="text/html; charset=windows-1251">  
    <meta http-equiv="KEYWORDS"  
          content="тег, html, заголовок страницы">  
    <meta http-equiv="DESCRIPTION"  
          content="Описание элементов заголовка">  
  </head>  
  
  <body>  
    Тело документа  
  </body>  
  
</html>
```

* Пишите теги в одну строчку, здесь они разбиты на две по причине ограничения ширины страницы. *

Script

Теги *<script></script>* используются для подключения внешних файлов скриптов. Это позволяет оптимизировать код страниц. Например, если вы используете функции java script для большинства своих страниц, то поместив эти функции на отдельную страницу - function.js, с помощью тега *<script>* можно указать путь к этой страницы.

Это повышает читабельность кода и ускоряет загрузку страниц. У этого тега несколько

параметров:

language - указывает язык написания скрипта, в спецификации HTML 4.0 данный параметр не рекомендуется к употреблению. Вместо него следует указывать параметр *type*.

type - указывает тип MIME для языка.

src - указывает путь к внешнему файлу со скриптами.

Пример кода с относительным адресом скрипта:

```
<html>
  <head>
    <title>Структура html</title>
    <base href="http://www.my_site.ru/">
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta http-equiv="Content-Type"
          content="text/html; charset=windows-1251">
    <meta http-equiv="KEYWORDS"
          content="тег, html, заголовок страницы">
    <meta http-equiv="DESCRIPTION"
          content="Описание элементов заголовка">
    <script type="text/javascript" src="function.js">
    </script>
  </head>

  <body>
    Тело документа
  </body>
</html>
```

Тело документа - тег BODY

Все, что отображается на web-странице, находится в тегах *<body></body>*. Это текст, картинки и исполняющиеся скрипты, а также теги для оформления всего этого.

Обязательных параметров у тега *<body>* нет, да и применение необязательных параметров тоже не приветствуется. Тем не менее, большинство параметров до сих пор поддерживается разными браузерами. Рассмотрим те, которые пока поддерживаются всеми браузерами:

alink - устанавливает цвет активной ссылки. Текущий цвет ссылки меняется на активный при нажатии на нее.

vlink - устанавливает цвет посещенной ссылки, т.е. той, по которой уже щелкали.

background - указывает на изображение, которое будет использоваться в качестве фонового рисунка. Этот рисунок заполняет собой все видимое пространство окна. Если рисунок меньше окна браузера, то он повторяется, образуя мозаику из одинаковых картинок. На стыке этих картинок возникают видимые переходы. Поэтому к подбору фоновых рисунков следует подходить с большим вниманием.

bgcolor - указывает фоновый цвет документа.

leftmargin - определяет отступ от левого края окна браузера до контента страницы.

rightmargin - определяет отступ от правого края окна браузера до контента страницы.

topmargin - определяет отступ от верхнего края окна браузера до контента страницы.

bottommargin - определяет отступ от нижнего края окна браузера до контента страницы.

text - устанавливает цвет текста для всего документа.

Пример кода:

```
<html>
  <head>
    <title>Тег body в html</title>
    <base href="http://www.my_site.ru/">
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta http-equiv="Content-Type"
          content="text/html; charset=windows-1251">
    <meta http-equiv="KEYWORDS"
          content="тег, html, заголовок страницы">
    <meta http-equiv="DESCRIPTION"
          content="Описание элементов заголовка">
    <script type="text/javascript" src="function.js">
  </script>
  </head>

  <body bgcolor="khaki" leftmargin="100" topmargin="50"
        rightmargin="50" bottommargin="50" text="gray"
        alink="red" vlink="green">
    Просто текст
    <br>
    <a href="index.html">Ссылка на страницу index.html</a>
    <br>
    <a href="map.html">Ссылка на страницу map.html</a>
  </body>

</html>
```

В окне браузера он будет выглядеть так:

На этом второй урок закончен, мы рассмотрели основную структуру html-документа.

Запомните, любой ваш html-документ должен содержать следующие теги и именно в том порядке, как они указаны:

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

  <body>  </body>

</html>
```

Все остальные теги рассмотренные в этом уроке на этом этапе вам не нужны, будете добавлять их позже по мере необходимости.

Все элементы, которые мы будем рассматривать на следующих уроках, будут помещаться внутрь тегов **<body></body>** и их порядок уже не будет иметь принципиального значения.

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЗАНЯТИЯМ

3.1 Практическое занятие 1,2 Общие сведения об информационных технологиях

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Сущность информационных технологий
2. Характеристика основных информационных технологий

3.2 Практическое занятие 3,4 Классификация информационных технологий.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Классификация информационных технологий по преобладания обрабатываемой информации
2. По классу реализуемых технологических операций
3. По ориентации на компоненты информационных систем

3.3 Практическое занятие 5 Использование Visual Basic for Applications.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Основы VBA
2. Алфавит языка VBA
3. Математические функции
4. Линейный процесс

3.4 Практическое занятие 6,7 Понятие базы данных.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. База знаний (БЗ)
2. Банк данных
3. База данных

3.5 Практическое занятие 8 Архитектура СУБД.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. телеобработка;
2. файловый сервер;
3. технология «клиент/сервер».

3.6 Практическое занятие 9 СУБД ACCESS.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Объекты Microsoft Access
2. Работа с таблицами

3.6 Практическое занятие 10,11 Работа с MathCAD.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Структура программы
2. Структура программы
3. Ввод и вычисление математических выражений

3.7 Практическое занятие 12,13 Решение систем линейных алгебраических уравнений в MathCAD.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Системы линейных уравнений.
2. Системы нелинейных уравнений.

3.9 Практическое занятие 14,15 Аппаратные средства связи

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Программно-аппаратные средства

3.10 Практическое занятие 16 Локальные и глобальные вычислительные сети.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Локальные компьютерные сети.
 2. Глобальные вычислительные сети
- 3.11 Практическое занятие 17 Организация беспроводных сетей.

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Что такое Wi-Fi?
2. Основные элементы сети

3.12 Практическое занятие 18 Работа в HTML

При подготовки к занятию необходимо обратить внимание на следующие моменты:

1. Создание гиперссылок.
2. Теги HTML.