

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

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБУЧАЮЩИХСЯ
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.Б.09 Информационные технологии

Направление подготовки (специальность) 27.03.04 Управление в технических системах

Профиль подготовки (специализация) Интеллектуальные системы обработки информации и управления

Квалификация выпускника бакалавр

Форма обучения очная

СОДЕРЖАНИЕ

1. Конспект лекций.....
1.1 Лекция № 1 Общие сведения об информационных технологиях
1.2 Лекция № 2,3 Классификация информационных технологий
1.3 Лекция № 4,5 Работа в HTML
1.4 Лекция № 6 Работа с MathCAD
1.5 Лекция № 7 Решение систем линейных алгебраических уравнений в MathCAD
1.6 Лекция № 8,9 Графические построения в MathCAD.....
1.7 Лекция № 10 Понятие базы данных.....
1.8 Лекция № 11 Архитектура СУБД.....
1.9 Лекция № 12-13 СУБД ACCESS
1.10. Лекция № 14 Аппаратные средства связи
1.11 Лекция № 15 Локальные и глобальные вычислительные сети.
1.12 Лекция № 16 Организация беспроводных сетей
1.13 Лекция № 17,18 Использование Visual Basic for Applications
2.Методические материалы по выполнению лабораторных работ
2.1.Лабораторная работа № ЛР-1-5 Работа в HTML
2.2.Лабораторная работа № ЛР-6-9 Работа с MathCAD
2.3.Лабораторная работа № ЛР-10-13 Решение систем линейных алгебраических уравнений в MathCAD
2.4.Лабораторная работа № ЛР-14-17 Графические построения в MathCAD
3. Методические материалы по проведению практических занятий
3.1 Практическое занятие № ПЗ-1 Общие сведения об информационных технологиях
3.2 Практическое занятие № ПЗ-2 Классификация информационных технологий.....
3.3 Практическое занятие № ПЗ-3 Работа в HTML
3.4 Практическое занятие № ПЗ-4 Работа с MathCAD
3.5 Практическое занятие № ПЗ-5 Решение систем линейных алгебраических уравнений в MathCAD
3.6 Практическое занятие № ПЗ-6 Графические построения в MathCAD
3.7 Практическое занятие № ПЗ-7 Понятие базы данных
3.8 Практическое занятие № ПЗ-8 Архитектура СУБД
3.9 Практическое занятие № ПЗ-9 СУБД ACCESS
3.10 Практическое занятие № ПЗ-10 Аппаратные средства связи
3.11 Практическое занятие № ПЗ-11 Локальные и глобальные вычислительные сети
3.12 Практическое занятие № ПЗ-12 Организация беспроводных сетей.....
3.13 Практическое занятие № ПЗ-13 Использование Visual Basic for Applicatins

1. КОНСПЕКТ ЛЕКЦИЙ

1.1 Лекция № 1 (2 часа)

Тема: «Общие сведения об информационных технологиях»

1.1.1 Вопросы лекции:

1. Сущность информационных технологий.
2. Характеристика информационных технологий.

1.1.2 Краткое содержание вопросов:

1. Сущность информационных технологий.

Информационная технология — это представленное в проектной форме (т. е. в формализованном виде, пригодном для практического использования) концентрированное выражение научных знаний, сведений и практического опыта, позволяющее рациональным образом организовать тот или иной достаточно часто повторяющийся информационный процесс. При этом достигается экономия затрат труда, энергии или материальных ресурсов, необходимых для реализации данного процесса.

1. Характеристика информационных технологий.

Информационные технологии позволяют оптимизировать и во многих случаях автоматизировать информационные процессы, которые в последние годы занимают все большее место в жизнедеятельности человеческого общества. Общеизвестно, что развитие цивилизации происходит в направлении становления информационного общества, в котором объектами и результатами труда большинства занятого населения становятся уже не материальные ценности, а главным образом, информация и научные знания. В настоящее время в большинстве развитых стран большая часть занятого населения в своей деятельности в той или иной мере связана с процессами подготовки, хранения, обработки и передачи информации и поэтому вынуждена осваивать и практически использовать соответствующие этим процессам информационные технологии.

1.2 Лекция № 2,3 (4 часа)

Тема: «Классификация информационных технологий»

1.2.1 Вопросы лекции:

1. Современные информационные технологии.
2. Сетевые информационные технологии.

1.2.2 Краткое содержание вопросов:

1. Современные информационные технологии.

Информационные технологии сегодня играют исключительно важную роль в обеспечении информационного взаимодействия между людьми, а также в системах подготовки и распространения массовой информации. Эти средства быстро ассимилируются культурой нашего общества, так как они не только создают большие удобства, но снимают многие производственные, социальные и бытовые проблемы, вызываемые процессами глобализации и интеграции мирового сообщества, расширением внутренних и международных экономических и культурных связей, миграцией населения и его все более динамичным перемещением по планете. В дополнение ставшим уже традиционными средствам связи (телефон, телеграф радио и телевидение) в социально-культурном сервисе и туризме все более широко используются системы электронных телекоммуникаций, электронная почта, факсимильная передача информации и другие виды связи.

2. Сетевые информационные технологии.

Сейчас в мире зарегистрировано более 200 сетей (глобальных).

С появлением персональных компьютеров стали разрабатывать и усиленно внедрять локальные вычислительные сети (ЛВС). Они существенно повышают эффективность управления производством, улучшают качество обрабатываемой информации, реализуют безбумажную технологию. Создаются новые информационные технологии в рамках задач ЛВС. Объединяются ЛВС и глобальные сети, что обеспечивает доступ к мировым информационным ресурсам.

1.3 Лекция № 4,5 (4 часа)

Тема: «Работа в HTML».

1.3.1 Вопросы лекции:

1. Работа с WWW
2. Способы открытия HTML документов
3. Сохранение документа
4. Учет URL. Сохранение внешних ссылок
5. Обеспечение эффективности работы с Web-сайтом

1.3.2 Краткое содержание вопросов:

1. Работа с www

Для работы со службой WWW (или Web сайтами) используется программа web-браузер, например такая как Internet Explorer. В качестве ополнительной функции обеспечивается работа с FTP сайтами.

Основные функции web-браузера:

- Открытие HTML документов.
- Сохранение HTML документов на локальных дисках.
- Хранение выбранных адресов (URL) и учет URL посещенных пользователем страниц (открытых HTML документов).
- Обеспечение эффективной работы с Web сайтом.
- Загрузка документов, отличных от HTML документов (то есть имеющих иной формат).

2. Способы открытия html документов

- Адресная строка. В ней указывается адрес того документа, который необходимо открыть.
- Использование ссылок. Гиперссылка - это URL того документа, который привязан к одному из элементов Web страницы (текстовому или графическому). Ссылки бывают внутренние и внешние по отношению к открытому Web документу.

Внутренние хранятся внутри открытого Web документа. Внешние - хранятся в папках Windows (избранное и в других).

- Команда открыть в Internet Explorer.
- Команда открыть в Windows.
- Диалог загрузка файла.

3. Сохранение документа

- Сохранение полностью (когда сохраняется и текст и мультимедийные элементы).
- Сохранение только HTML (сохраняется только текстовое содержимое без мультимедийных элементов).
- Сохранение текстового файла (сохраняется текст, но теряется форматирование, то есть внешний вид текста).

4. Учет url. Сохранение внешних ссылок

- Папка избранное является обычной папкой Windows, хранящейся на локальном диске, в которой ссылки хранятся в виде файлов (папки хранятся в виде ярлыков). Internet Explorer предоставляет доступ к этой папке через меню.
- Панель ссылок - это панель инструментов Internet Explorer, на которой отображается содержимое папки ссылки, которая в свою очередь, входит в состав папки избранное.
- Домашняя страница - это Web документ, открывающийся при запуске программы Internet Explorer, а также при щелчке по кнопке Home на основной панели инструментов Internet Explorer. Назначение домашней страницы осуществляется пользователем.
- Журнал - это автоматически заполненный список ссылок на Web документы, когда-либо открывавшиеся пользователем. В журнале имеется возможность сортировки вида ссылок: По узлу (то есть по принадлежности к тому или иному сайту); По дате; По порядку посещения; По посещаемости, то есть по частоте посещения сайта (наиболее часто посещаемые сайты отображаются на первой странице журнала).

5. Обеспечение эффективности работы с Web-сайтом

Некоторые возможности web-браузеров:

- Автозаполнение - это автоматическое завершение набора текста, начатого пользователем в любом текстовом поле.
- Открытие нового окна Internet Explorer. Это позволяет нам работать сразу с несколькими документами.
- Отключение мультимедийных элементов (изображений, видео, аудио, анимации).
- Использование кеш памяти.

Кеш память - это область оперативной памяти компьютера, или папка на локальном диске, или папка на диске удаленного компьютера, предназначенная для промежуточного хранения Web документов и последующего более быстрого доступа к ним.

Кеш оперативной памяти используется посредством команд вперед и назад.

Кеш на локальном диске организуется в папке временных файлов.

Кеш на диске удаленного компьютера подключается посредством настройки прокси-сервера.

- Работа в автономном режиме (без подключения к Internet).
- Исправление ошибок.

Например, изменение вида кодировки. Web документы хранящиеся на сервере с платформой отличной от Windows (например Unix) могут отображаться неправильно по причине использования других принципов хранения файлов. Эта ошибка автоматически

исправляется помошью изменения кодировки. В России используются 2 основные кодировки: "Кириллица Windows-1251 (Windows)" и "Кириллица KOI8-R (Unix)".

1.3 Лекция № 6 (2 часа)

Тема: «Работа с MathCAD»

1.3.1 Вопросы лекции:

1. Формулы и текст.
2. Символьные преобразования.

1.3.2 Краткое содержание вопросов:

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, символьный знак равенства может возвращать выражение.

1.4 Лекция № 7 (2 часа)

Тема: «Решение систем линейных алгебраических уравнений в MathCAD»

1.4.1 Вопросы лекции:

1. Формулы Крамера.
2. Метод Гаусса.

1.4.2 Краткое содержание вопросов:

1. Формулы Крамера.

Метод Крамера (теорема Крамера) — способ решения квадратных СЛАУ с ненулевым определителем основной матрицы. Назван по имени Габриэля Крамера, автора метода.

2. Метод Гаусса.

Метод Гаусса прекрасно подходит для решения систем линейных алгебраических уравнений (СЛАУ). Он обладает рядом преимуществ по сравнению с другими методами:

- во-первых, нет необходимости предварительно исследовать систему уравнений на совместность;
- во-вторых, методом Гаусса можно решать не только СЛАУ, в которых число уравнений совпадает с количеством неизвестных переменных и основная матрица системы невырожденная, но и системы уравнений, в которых число уравнений не совпадает с количеством неизвестных переменных или определитель основной матрицы равен нулю;
- в-третьих, метод Гаусса приводит к результату при сравнительно небольшом количестве вычислительных операций.

1.6 Лекция № 8,9 (4 часа)

Тема: «Графические построения в MathCAD»

1.6.1 Вопросы лекции:

- 1 Двумерные графики в декартовой системе координат
2. Двухмерные графики в полярной системе координат
3. Графики в трехмерном пространстве

1.6.2 Краткое содержание вопросов

MathCAD позволяет легко строить двух- и трехмерные гистограммы, двухмерные графики в декартовых и полярных координатах, трехмерные графики поверхностей, линии уровня поверхностей, изображения векторных полей, пространственные кривые.

Существует три способа построения графиков в системе MathCAD:

- можно воспользоваться позицией Главного меню **Insert**, выбрав команду **Graph** и в раскрывающемся списке - тип графика;
- выбрать тип графика на наборной панели **Graph**, которая включается кнопкой на панели **Math**;

- воспользоваться быстрыми клавишами (они предусмотрены не для всех типов графиков). Рассмотрим более подробно команды меню **Insert->Graph** (слева изображены соответствующие кнопки наборной панели **Math**):

 **X-Y Plot** (X-Y Зависимость) клавиша [@] ¶ Служит для построения графика функции $y=f(x)$ в виде связанных друг с другом пар координат (x_i, y_i) при заданном промежутке изменения для i .

 **Polar Plot** (Полярные координаты) клавиши [Ctrl+7] ¶ Служит для построения графика функции $r(q)$, заданной в полярных координатах, где полярный радиус r зависит от полярного угла q .

 **Surface Plot** (Поверхности) клавиши [Ctrl+2] ¶ Служит для представления функции $z=f(x, y)$ в виде поверхности в трехмерном пространстве. При этом должны быть заданы векторы значений x_i и y_j , а также определена матрица вида $A_{i,j}=f(x_i, y_j)$. Имя матрицы A указывается при заполнении рамки-шаблона. С помощью этой команды можно строить параметрические графики.

 **Contour Plot** (Контурный график) ¶ Строит диаграмму линий уровня функции вида $z=f(x, y)$, т.е. отображает точки, в которых данная функция принимает фиксированное значение $z=const$.

 **3D Scatter Plot** (3D Точечный) ¶ Служит для точечного представления матрицы значений $A_{i,j}$ или отображения значений функции $z=f(x, y)$ в заданных точках. Эта команда может также использоваться для построения пространственных кривых.

 **3D Bar Plot** (3D Диаграммы) ¶ Служит для представления матрицы значений $A_{i,j}$ или отображения значений функции $z=f(x, y)$ в виде трехмерной столбчатой диаграммы.

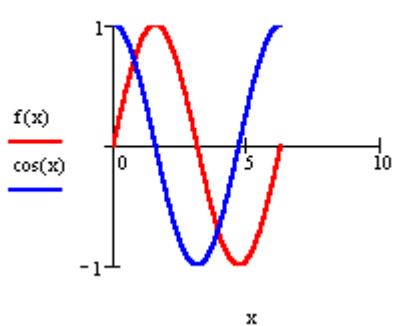
 **Vector Field Plot** (Поле векторов) ¶ Служит для представления двухмерных векторных полей $V=(V_x, V_y)$. При этом компоненты векторного поля V_x и V_y должны быть представлены в виде матриц. При помощи этой команды можно построить поле градиента функции $f(x, y)$.

3D Plot Wizard (вызов мастера для быстрого построения 3-хмерного графика) ¶ При выборе этой команды возникает ряд всплывающих окон, в которых требуется выбрать параметры построения трехмерного графика (задаются тип трехмерного графика, стиль его изображения, цветовая гамма). График по умолчанию строится на промежутке от -5 до +5 (по обеим переменным).

3.1 Двумерные графики в декартовой системе координат

График функции $y=f(x)$.

$$f(x) := \sin(x) \quad x := 0,0.01..2 \cdot \pi$$



При выполнении команды **Inset -> Graph -> Plot** в документ помещается рамка-шаблон с двумя незаполненными ячейками для построения графика. (Клавиша [@]). В ячейке, расположенной под осью абсцисс, указывается независимая переменная x . Ее следует определить заранее как переменную, принимающую значения из промежутка (ранжированная переменная).

В ячейке рядом с осью ординат необходимо задать функцию $f(x)$, график которой мы хотим построить. Если эта функция была определена заранее, то в ячейку достаточно ввести $f(x)$, в противном случае следует ввести изображаемую функцию в явном виде (например, $\cos(x)$). После ввода x и $f(x)$ в графической области появятся еще четыре ячейки, которые не обязательно заполнять. MathCAD автоматически находит подходящие значения для x_{min} , x_{max} , y_{min} , y_{max} . Если же предлагаемые MathCAD значения вас не устраивают, вы можете задать свои. В MathCAD существует возможность строить график функции, не задавая предварительно промежуток изменения независимой переменной. По умолчанию этот промежуток принимается равным [-10, 10].

Для представления на одной диаграмме графиков нескольких функций необходимо выделить ячейку рядом с осью ординат и через запятую ввести вторую функцию. По умолчанию график этой функции будет представлен пунктирной линией другого цвета.

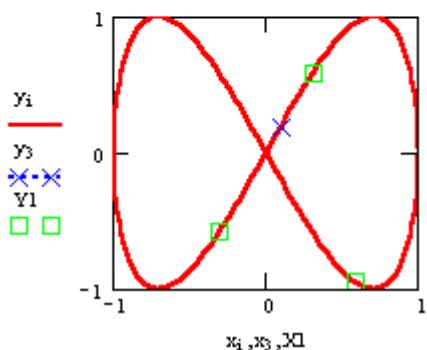
Кривые на плоскости, заданные параметрически.

Уравнения $x=f(t)$, $y=y(t)$, где $f(t)$ и $y(t)$ непрерывны при t из (a, b) , устанавливающие зависимость декартовых координат (x, y) точки плоскости от значения параметра t , определяют на плоскости кривую, заданную в параметрической форме.

$$N := 200 \quad i := 0..N$$

$$x_i := \sin\left(2 \cdot \pi \cdot \frac{i}{N}\right) \quad y_i := \sin\left(4 \cdot \pi \cdot \frac{i}{N}\right)$$

$$X1 := \begin{pmatrix} x_{10} \\ x_{80} \\ x_{190} \end{pmatrix} \quad Y1 := \begin{pmatrix} y_{10} \\ y_{80} \\ y_{190} \end{pmatrix}$$



В случае построения параметрически заданной кривой, вместо независимой переменной x под осью абсцисс необходимо задать индексированную переменную x_i . А рядом с осью ординат необходимо соответственно указать y_i .

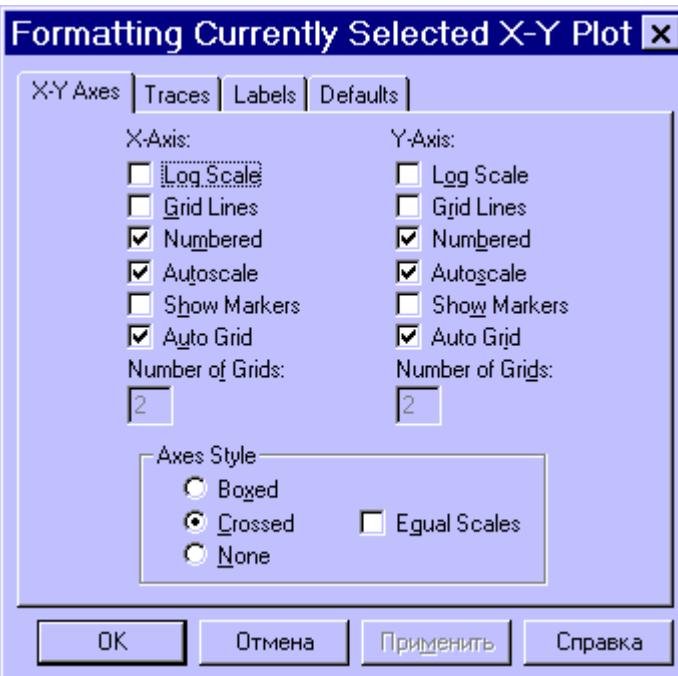
Для нанесения на график функции отдельных точек, их координаты указываются через запятую под осью абсцисс и слева от оси ординат. Если требуется вывести множество точек, то можно сформировать два вектора, один из которых содержит абсциссы точек, а другой - их ординаты. В этом случае на графике в соответствующих ячейках указываются только имена векторов.



Редактирование графиков в декартовой системе координат.

Если вас не устраивает внешний вид построенных графиков, вы можете его изменить, выделив график (выполнив на нем щелчок, так, чтобы вокруг него появилась рамка) и воспользовавшись командой **Format -> Graph -> X-Y Plot**, или, выполнив на графике щелчок правой кнопкой мыши и выбрав команду **Format** из выпадающего контекстного меню (можно выполнить также двойной щелчок левой кнопкой мыши). В результате на экране появится диалоговое окно **Formatting Currently Selected X-Y Plot**, позволяющее изменить вид графика.

Данное диалоговое окно содержит несколько вкладок: **X-Y Axes** (форматирование осей), **Traces** (тип линий графиков), **Labels** (подписи), **Defaults** (по умолчанию).



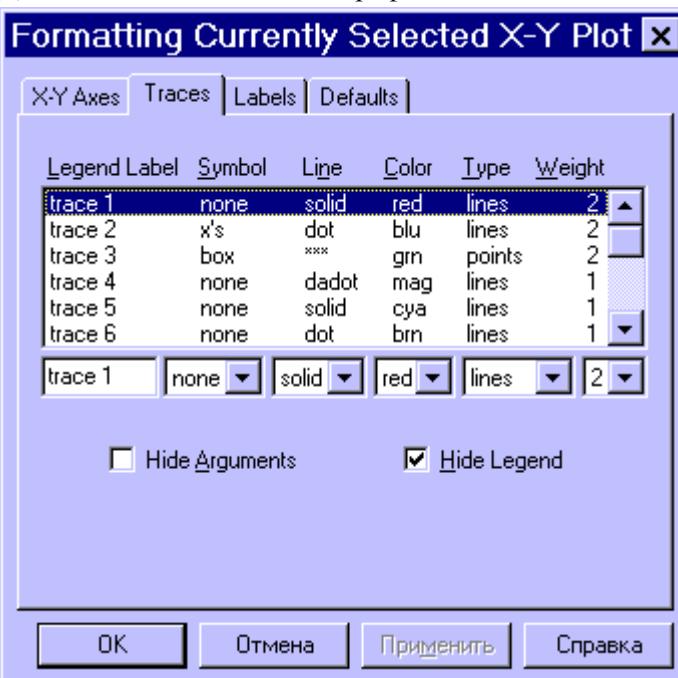
Первая из вкладок позволяет форматировать оси координат:

- **Log Scale** (Логарифмическая шкала) - задает логарифмические оси, в этом случае границы графика должны задаваться положительными числами.
- **Grid Lines** (Вспомогательные линии) - Задает отображение сетки из параллельных осям линий.
- **Numbered** (Нумерация) - Задает отображение подписи к маркировкам на осях.
- **Autoscale** (Автомасштаб) - Задает автоматическое нахождение подходящих границ для осей. Но если вы сами зададите в соответствующих ячейках минимальные и максимальные значения x_{min} , x_{max} , y_{min} , y_{max} , именно эти значения будут использоваться для определения границ графика.

- **Show Markers** (Показать метки) - Если установить эту опцию, то в графической области появятся четыре дополнительные ячейки для создания красных линий маркировки, соответствующих двум специальным значениям x и двум специальным значениям y .
- **Auto Grid** (Автосетка) - При установке этой опции число линий сетки определяет MathCAD.
- **Axes Style** (Вид осей) - Группа кнопок этой области позволяет выбрать следующие варианты представления осей: **Boxed** (ограниченная область), **Crossed** (пересечение) - оси пересекаются в точке с координатами (0,0), **None** (без границ). Флажок **Equal Scales** (равный масштаб) позволяет задать одинаковый масштаб для обеих осей.

Форматирование оси графика можно произвести, выполнив на ней двойной щелчок.

Для изменения типа линий графиков необходимо активизировать вкладку **Traces** (След)



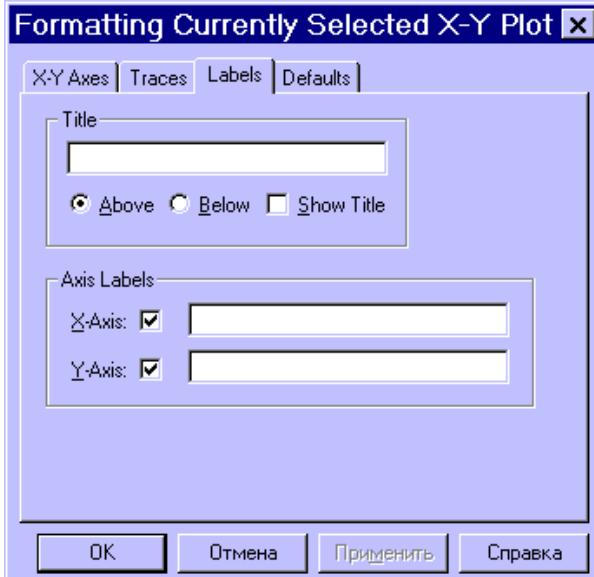
• **Legend Label** (Имя в легенде) - Каждой кривой можно поставить в соответствие некоторый текст, называемый легендой. Легенда отображается в нижней части графической области, а рядом с каждой легендой отображается тип линии соответствующей кривой.

- **Symbol** (Символ) - Позволяет выбрать символ для каждой точки кривой (плюс, крестик, кружок и др.)
- **Line** (Линия) - Можно выбрать один из следующих типов линий: *solid* (сплошная), *dash* (штриховая), *dot* (точечная) или *dadot* (штрихпунктирная). Это поле списка доступно в случае, если в поле **Type** (Тип) выбран элемент *lines*/
- **Color** (Цвет) - Задается цвет представления кривой на экране.

- **Type** (Тип) - Позволяет выбрать один из семи видов графика: в виде кривых (), в виде столбцов () и т. п. Специальным видом графика является тип (погрешность, расхождение), представляющий собой разность двух заданных функций. Величина шага независимой переменной определяет расстояние между отдельными столбцами, ступенями или линиями погрешностей на диаграмме.
- **Weight** (Вес) - Позволяет задавать толщину линий графика.

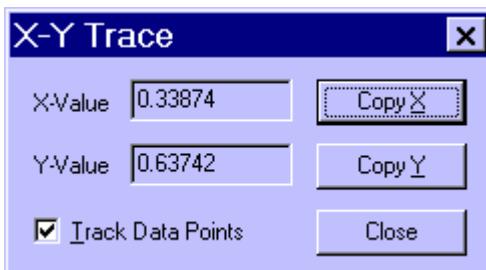
В нижней части вкладки **Traces** расположены опции:

- **Hide Arguments** (Скрыть аргументы) - Эта опция по умолчанию отключена. В этом случае под именем функции рядом с осью ординат указывается текущий тип линий. Если установить данную опцию, указание типа линий исчезнет.
- **Hide Legend** (Скрыть легенду) - По умолчанию легенда не отображается. Если вы хотите отобразить под графиком текст легенды, его необходимо перед этим ввести в поле **Legend Label** (Имя в легенде) и подтвердить ввод, выполнив щелчок на кнопке **Применить**.

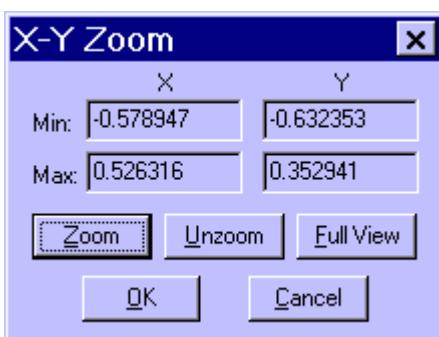


Вкладка **Labels** (Метки) позволяет ввести заголовок графика и подписи для осей.

В подменю **Graph** (Графика) меню **Format** (Формат) содержатся кроме прочих следующие команды:



Trace (След) - При перемещении в области графика указателя мыши при нажатой левой кнопке в полях **X-Value** (значение X) **Y-Value** (значение Y) диалогового окна **X-Y Trace** отображаются координаты точки, на которую указывает курсор. Если установлена опция **Track Data Points** (След точек данных), то курсор-крестик будет перемещаться вдоль графика функции и вы сможете считывать текущее значение аргумента x и соответствующее значение функции $y=f(x)$. Координаты текущей точки можно скопировать в буфер при помощи кнопок **Copy X** (Копировать X) **Copy Y** (Копировать Y).



Zoom (Изменение масштаба) - При помощи этой команды можно увеличить фрагмент графика, предварительно выделив его протаскиванием мыши с нажатой левой клавишей. После отпускания клавиши координаты углов выделенной области будут отображены в полях окна **X-Y Zoom**. При помощи кнопки **Zoom** (Масштаб +) фрагмент можно увеличить, при помощи кнопки **Unzoom** (Масштаб -) отменить выделение фрагмента, а при помощи кнопки **Full View** (Обзор) - восстановить первоначальный вид графика. Если вы увеличили фрагмент графика, то при щелчке на кнопке **OK** в документе будет отображаться только этот фрагмент.

3.2 Двухмерные графики в полярной системе координат

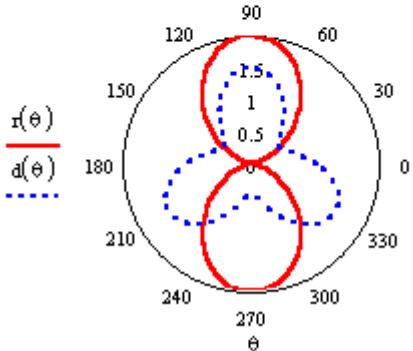
■ Построение полярных графиков.

Для построения полярного графика необходимо выполнить команду **Inset -> Graph -> Polar Plot**. При этом в документ помещается графическая область с двумя незаполненными ячейками для построения графика. (Клавиши [Ctrl +7]).

$$N := 15 \quad \theta := 0, \frac{1}{N} \dots 2 \cdot \pi$$

$$r(\theta) := 1 + \sin\left(2 \cdot \theta + 3 \cdot \frac{\pi}{2}\right)$$

$$d(\theta) := 1 + \frac{\sin(3 \cdot \theta + \pi)}{2}$$



В нижнюю ячейку вводится полярный угол \mathbf{q} . Его следует определить заранее как переменную, принимающую значения из промежутка (ранжированная переменная). В левую ячейку вводится полярный радиус $r(\mathbf{q})$ или $\text{Re}(r(\mathbf{q}))$ и $\text{Im}(r(\mathbf{q}))$. Функция $r(\mathbf{q})$ либо задается заранее, либо ее определение вводится непосредственно в ячейку.

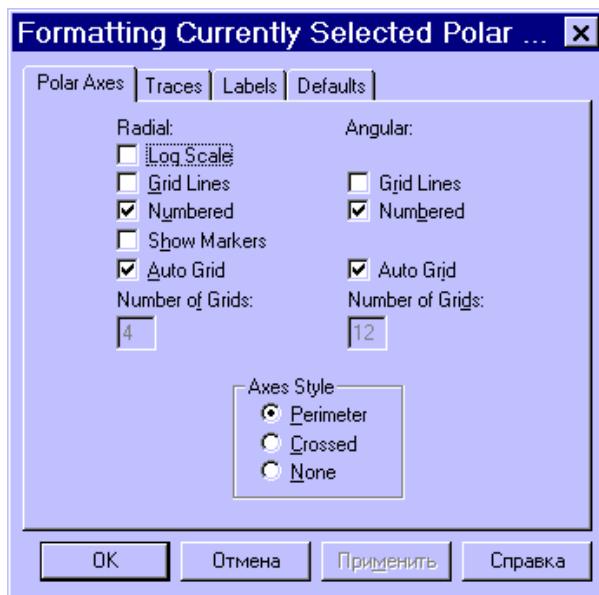
График в полярных координатах можно построить и посредством команды **X-Y Plot**. Только в этом случае необходимо дополнительно задать следующие функции: $x(\mathbf{q})=r(\mathbf{q})\cos(\mathbf{q})$ и $y(\mathbf{q})=r(\mathbf{q})\sin(\mathbf{q})$, а в ячейках для абсцисс и ординат указать соответственно $x(\mathbf{q})$ и $y(\mathbf{q})$.

Если для некоторых углов \mathbf{q} функция не определена (не существует радиуса с действительным значением, он принимает чисто мнимые значения), то MathCAD отображает на графике только действительные значения радиуса. Но при помощи функций **Re** и **Im** можно представить на одной круговой диаграмме графики как действительной, так и мнимой части функции $r(\mathbf{q})$.

Форматирование полярных графиков.

Если вы хотите отредактировать график в полярных координатах, необходимо выделить график (щелчком левой кнопки мыши) и выполнить команду **Format -> Graph -> Polar Plot** или выполнить двойной щелчок на выделенном графике. При этом откроется окно **Formatting Currently Selected Polar Plot** (форматирование полярного графика). Это окно содержит те же вкладки, что и для графиков в декартовой системе координат.

Вкладка **Polar Axes** содержит следующие элементы:



- **Log Scale** (Логарифмическая шкала) - Используется для создания логарифмической r -оси.
- **Grid Lines** (Вспомогательные линии) - Отображает сетку линий, соответствующих уравнениям $r=\text{const}$ и $\mathbf{q}=\text{const}$.
- **Numbered**(Нумерация) - Линии $r=\text{const}$ и $\mathbf{q}=\text{const}$ снабжаются подписями.
- **Show Markers** (Показать метки) - При помощи этой опции можно снабдить график двумя дополнительными пунктирными окружностями $r=\text{const}$. Для этого надо ввести нужные значения радиуса в появившиеся ячейки. Кроме того, справа от графика указывается минимальный и максимальный радиус; можно увеличить или уменьшить график, введя в эти ячейки собственные значения.
- **Auto Grid** (Автосетка) - При установке этой опции число линий сетки определяет MathCAD.

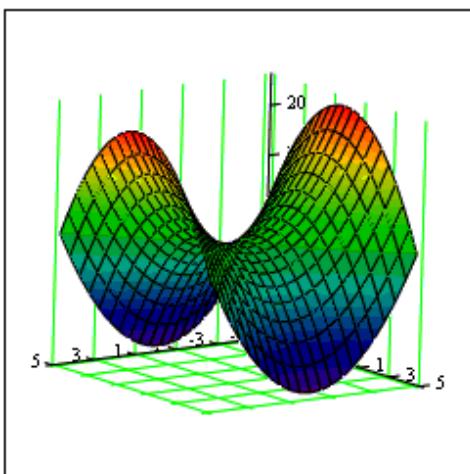
Остальные значения параметров на этой вкладке и на других вкладках те же самые , что и в случае команды **X-Y Plot**.

3.3 Графики в трехмерном пространстве

Построение графика функции $z=f(x,y)$ в виде поверхности в декартовой системе координат.

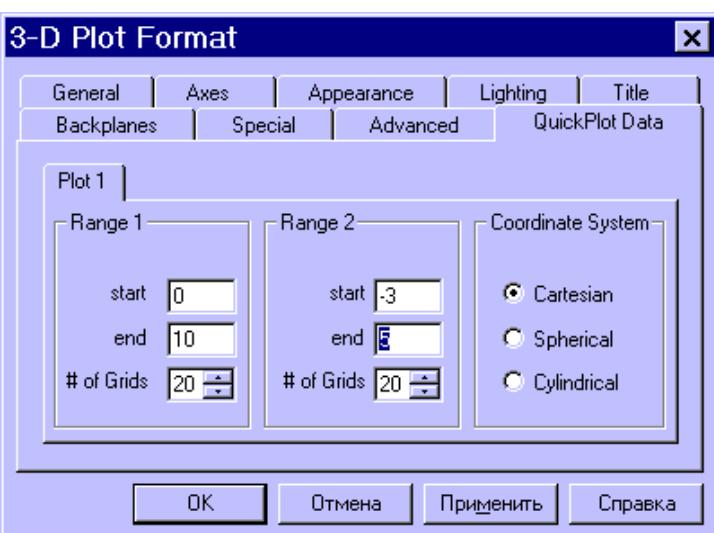
Для построения графика поверхности можно воспользоваться двумя способами:

$$g(x,y) := x^2 - y^2$$



1. Если вам надо только посмотреть общий вид поверхности, то MathCAD предоставляет возможность быстрого построения подобных графиков. Для этого достаточно определить функцию $f(x,y)$ и выполнить команду **Insert -> Graph -> Surface Plot** или нажать соответствующую кнопку наборной панели **Graph** (сочетание клавиш **[Ctrl+7]**). В появившейся графической области под осями на месте шаблона для ввода надо указать имя (без аргументов) функции. MathCAD автоматически построит график поверхности. Независимые переменные x и y принимают значения из промежутка $[-5,5]$.

g



При необходимости этот промежуток может быть уменьшен или увеличен. Для этого необходимо выделить график и воспользоваться командой **Format -> Graph -> 3D Plot** или щелкнуть ПРАВОЙ кнопкой мыши по выделенному графику и в контекстном меню выбрать команду **Format**. В появившемся окне **3-D Plot Format** на вкладке **QuickPlot Data** можно установить другие параметры изменения независимых переменных x и y .

$$N := 15$$

$$i := 0..N \quad j := 0..N$$

$$x_{min} := -1.5 \quad x_{max} := 1.5$$

$$y_{min} := -1.5 \quad y_{max} := 1.5$$

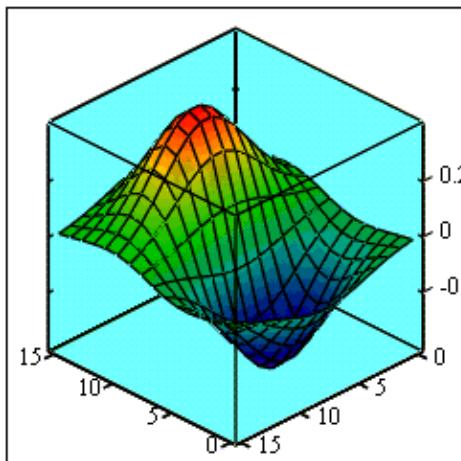
$$x_i := x_{min} + \frac{i}{N} \cdot (x_{max} - x_{min})$$

$$y_j := y_{min} + \frac{j}{N} \cdot (y_{max} - y_{min})$$

$$f(x,y) := \sin(x) \cdot \exp(-x^2 - y^2)$$

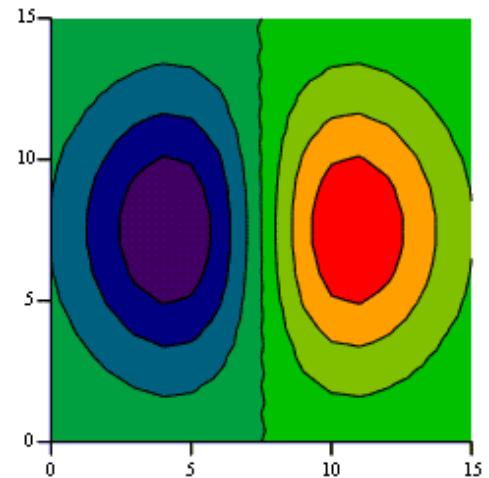
$$A_{i,j} := f(x_i, y_j)$$

Для построения графика поверхности в определенной области изменения независимых переменных или с конкретным шагом их изменения необходимо сначала задать узловые точки x_i и y_j , в которых будут определяться значения функции. После (а можно и до) этого надо определить функцию $f(x,y)$, график которой хотите построить. После этого необходимо сформировать матрицу значений функции в виде: $A_{i,j}=f(x_i, y_j)$.

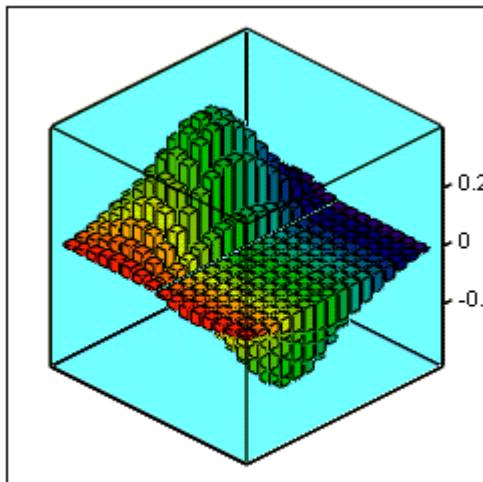


A

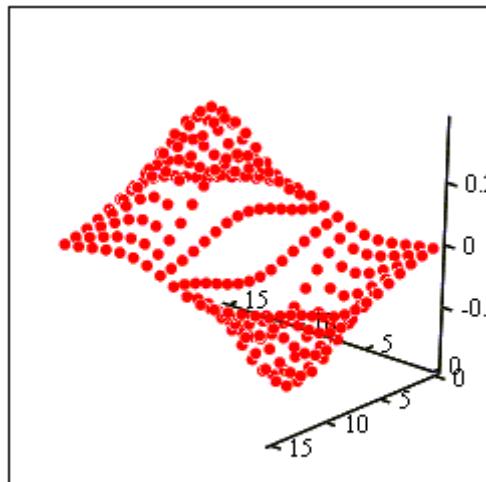
Теперь после выполнения команды **Insert -> Graph -> Surface Plot** в появившейся графической области достаточно ввести имя матрицы (без индексов). Если вы хотите, чтобы узловые точки были расположены через равные промежутки, воспользуйтесь формулами, изображенными на рисунке.



Для построения графика линий уровня данной функции необходимо поступать также как это было описано выше, только вместо команды (Поверхности) следует выбрать команду **Contour Plot** (Контурный). Аналогично, при помощи команды **3D Bar Plot** (3D Диаграммы) можно построить трехмерный столбчатый график данной функции, при помощи команды **3D Scatter Plot** (3D Точечный) - трехмерный точечный график, а при помощи команды **3D Patch Plot** (3D Лоскутный) - трехмерный график поверхности в виде несвязанных квадратных площадок - плоскостей уровня для каждой точки данных, параллельных плоскости X-Y



A



A

Построение графика поверхности, заданной параметрически.

$N := 30 \quad i := 0..N \quad j := 0..N$

$$\phi_i := 3 \cdot \pi \cdot \frac{i}{N} \quad \psi_j := 2 \cdot \pi \cdot \frac{j}{N}$$

$$R(\phi) := 8 \cdot e^{-\frac{\phi}{5}} \quad r(\phi) := 4 \cdot e^{-\frac{\phi}{5}}$$

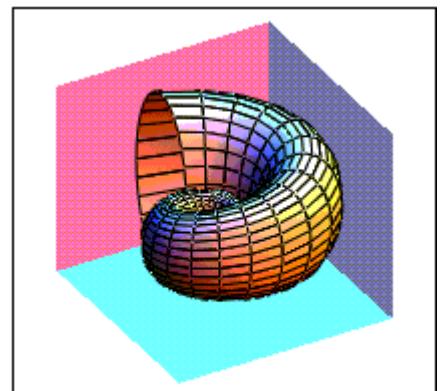
$$x_{i,j} := (R(\phi_i) + r(\phi_i) \cdot \cos(\psi_j)) \cdot \cos(\phi_i)$$

$$y_{i,j} := (R(\phi_i) + r(\phi_i) \cdot \cos(\psi_j)) \cdot \sin(\phi_i)$$

$$z_{i,j} := r(\phi_i) \cdot \sin(\psi_j)$$

Если поверхность задана параметрически, это означает, что все три координаты - x и y и z - заданы как функции от двух параметров u и v . Сначала необходимо задать векторы значений параметров u_i и v_j . Затем необходимо определить матрицы значений функций координат $x(u,v)$, $y(u,v)$ и $z(u,v)$.

После выбора команды **Surface Plot** в MathCAD документе появится графическая область. В свободной ячейке внизу области надо указать В СКОБКАХ имена (без аргументов и индексов) трех матриц - **x,y,z**.



(x, y, z)

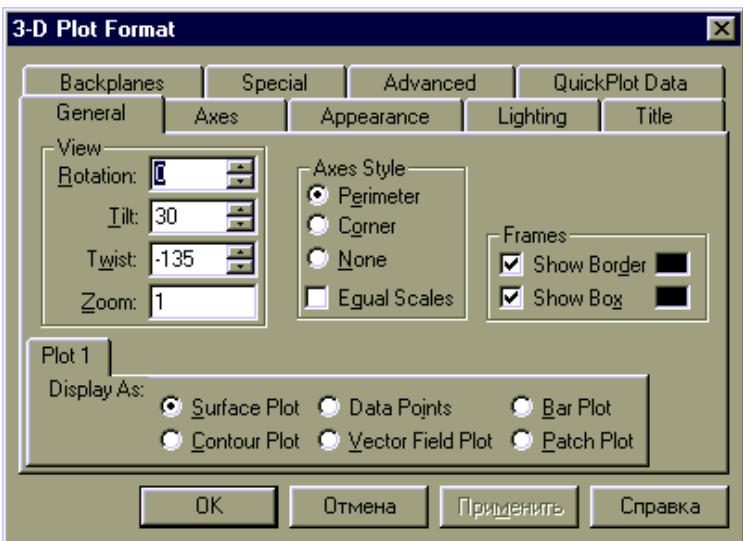
Форматирование трехмерных графиков.

Если вас не устраивает внешний вид созданного трехмерного графика, вы можете изменить его, выполнив команду **Format -> Graph -> 3D Plot** или выполнив двойной щелчок мышкой на графической области. В результате на экране появится диалоговое окно **3-D Plot Format**, позволяющее изменять параметры отображения графика. Мы рассмотрим здесь основные опции. Разобравшись во всех тонкостях управлением видом графика вы можете самостоятельно, построив график и поэкспериментировав, выбирая те или иные опции.

Диалоговое окно **3-D Plot Format** содержит несколько вкладок. Некоторые из них мы рассмотрим более подробно, а для других - опишем лишь их функциональное назначение.

На вкладке **General** (Общие свойства) вы можете

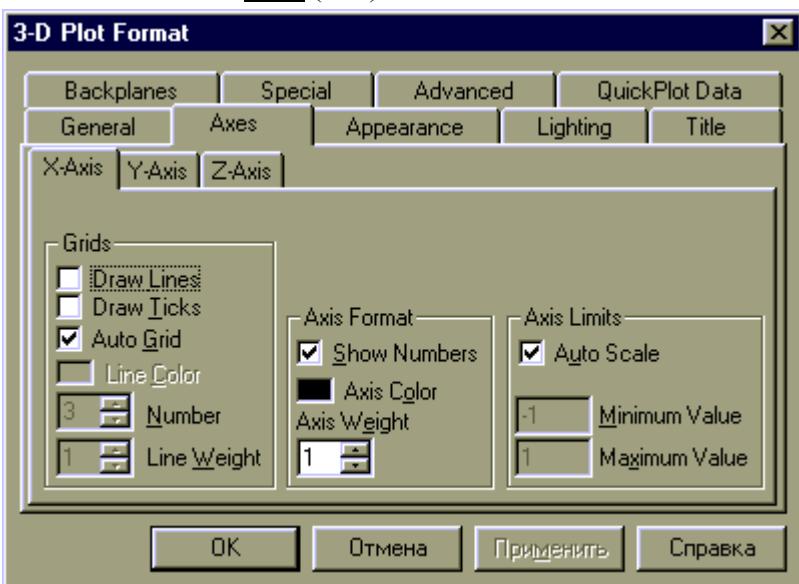
- в области **View** задать направление взгляда наблюдателя на трехмерный график. Значение в поле **Rotation** определяет угол поворота вокруг оси Z в плоскости X-Y. Значение в поле **Tilt** задает угол наклона линии взгляда к плоскости X-Y. Поле **Zoom** позволяет увеличить (уменьшить) графическое изображение в число раз, равное цифре, указанной в поле.



- в области **Axes Style** (Стиль оси) задать вид осей, выбрав селекторную кнопку **Perimeter** (Периметр) или **Corner** (Угол). В первом случае оси всегда находятся на переднем плане. При выборе кнопки **Corner** точка пересечения осей Ox и Oy задается элементом $A_{0,0}$ матрицы A .

- в области **Plot 1 (Plot 2...)** **Display as** (График/ несколько графиков Отобразить как) - имеются селекторные кнопки для представления графика в других видах (контурный, точечный, векторное поле и др.)

Элементы вкладки **Axes** (Ось) позволяют изменять внешний вид осей координат.



- Посредством опций области **Grids** (Сетки) можно отобразить на графике линии, описываемые уравнениями $x, y, z = const$.
 - Если установлены опции **Show Numbers** (Нумерация), отображаются метки на осях и подписи к ним.

При этом рядом с осями Ox и Oy указываются не значения узловых точек x_i, y_j , а значения индексов i и j , в то время как ось Oz размечается в соответствии с промежутком, которому принадлежат элементы матрицы значений $A_{i,j}$.

- Если установлена опция **Auto Grid** (Автосетка), программа самостоятельно задает расстояние между соседними отметками на осях. Вы можете сами указать число линий сетки, если отключите указанную опцию.
- Если установлена опция **Auto Scale** (Авошкала), то MathCAD сам определяет границы построкния графика и масштабы по осям. Можно отключить данную опцию и для каждой оси самостоятельно задать пределы изменения переменных в полях **Minimum Value** (Минимум) и **Maximum Value** (Максимум).

Вкладка **Appearance** (Внешний вид) позволяет изменять для каждого графика вид и цвет заливки поверхности (область **Fill Options**); вид, цвет и толщину дополнительных линий на графике (область **Line Options**); наносить на график точки данных (опция **Draw Points** области **Point Options**), менять их вид, размер и цвет.

Вкладка **Lighting** (Освещение) при включении опции **Enable Lighting** (Наличие подсветки) позволяет выбрать цветовую схему для освещения, "установить" несколько источников света, выбрав для них цвет освещения и определив его направление.

Вкладка **Backplanes** (Задние плоскости) позволяет изменить внешний вид плоскостей, ограничивающих область построения: цвет, нанесение сетки, определение ее цвета и толщины, прорисовка границ плоскостей.

На вкладке **Special** (Специальная) можно изменять параметры построения, специфичные для различных типов графиков.

Вкладка **Advanced** позволяет установить параметры печати и изменить цветовую схему для окрашивания поверхности нрафика, а также указать направление смены окраски (вдоль оси Ox , Oy или Oz). Включение опции **Enable Fog** (Наличие Тумана) делает график нечетким, слегка размытым (полупрозрачным). При включении опции **Perspective** (Перспектива) появляется возможность указать в соответствующем поле расстояние до наблюдателя.

Вкладка **Quick Plot Data** обсуждалась ранее в начале раздела.



Кривая в пространстве.

Трехмерные точечные графики можно использовать для построения изображения пространственных кривых. Пространственные кривые задаются, как правило, в виде $(x(t), y(t), z(t))$, где t представляет собой непрерывный действительный параметр.

$N := 36$ Число значений параметров

$i := 0..N - 1$ Индексная переменная i

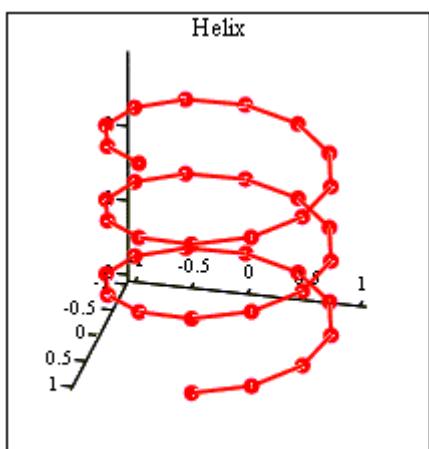
$$x_i := \cos\left(\frac{i}{N} \cdot 6 \cdot \pi\right)$$

Задание векторов координат x , y и z , которые являются функциями от индекса i . Данные векторы координат описывают винтовую линию.

$$y_i := \sin\left(\frac{i}{N} \cdot 6 \cdot \pi\right)$$

$$z_i := \frac{i}{N} \cdot 3$$

Поскольку при построении трехмерной точечной диаграммы MathCAD позволяет отображать на графике только отдельные точки и соединяющие их линии, необходимо сначала определить три вектора координат - x_i , y_i , z_i .



Пространственная кривая создается командой **Insert3D -> Graph ->Scatter Plot**. Можно использовать наборную панель Graph, выбрав соответствующую пиктограмму. Для соединения точек необходимо на вкладке **Appearance** окна форматирования графиков указать опцию **Line**.

(x, y, z)

Векторные и градиентные поля.

Команда **Insert -> Graph -> Vector Field Plot** (Поле векторов) служит для представления двумерных векторных полей $v = (vx, vy)$.

$$v(x, y) := \begin{pmatrix} -y \\ \sqrt{x^2 + y^2} \\ x \\ \sqrt{x^2 + y^2} \end{pmatrix} \quad N := 20$$

$i := 0..N-1$

$j := 0..N-1$

$$a := -2 \quad b := 2 \quad x_i := a + \frac{i}{N-1} \cdot (b-a)$$

$$c := -2 \quad d := 2 \quad y_j := c + \frac{j}{N-1} \cdot (d-c)$$

$$v_{i,j} := v(x_i, y_j)$$

$$v_{x_{i,j}} := \left(v_{i,j} \right)_0 \quad v_{y_{i,j}} := \left(v_{i,j} \right)_1$$

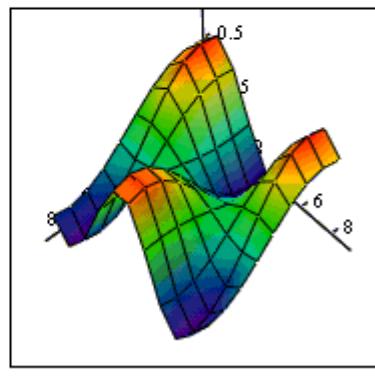
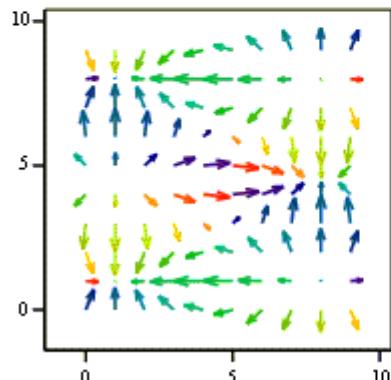
$$f(x, y) := \sin(x) \cdot \cos(y)$$

$$\text{grad}(x, y) := \begin{pmatrix} \frac{d}{dx} f(x, y) \\ \frac{d}{dy} f(x, y) \end{pmatrix} \quad Nx := 10 \quad a := -2 \quad b := 2 \quad i := 0..Nx-1$$

$Ny := 10 \quad c := -4 \quad d := 4 \quad j := 0..Ny-1$

$x_i := a + i \cdot \frac{b-a}{Nx-1} \quad y_j := c + j \cdot \frac{d-c}{Ny-1}$

$$F_{i,j} := f(x_i, y_j) \quad v_{i,j} := \text{grad}(x_i, y_j) \quad v_{x_{i,j}} := \left(v_{i,j} \right)_0 \quad v_{y_{i,j}} := \left(v_{i,j} \right)_1$$

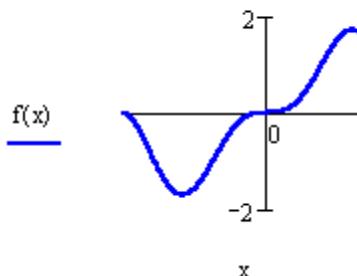


(v_x, v_y)

Поверхности, полученные вращением кривых вокруг осей.

Интересные объемные фигуры можно получить, вращая некоторую кривую вокруг той или иной оси. Построение этих фигур вращения сродни параметрически заданным поверхностям.

$$f(x) := x \cdot \sin(x)^2 \quad a := -\pi \quad b := 2$$



При этом необходимо обеспечить пересчет координат точек фигуры по известным из геометрии формулам. В MathCAD встроена функция **CreateMesh**, с помощью которой можно построить параметрически заданные поверхности.

Вращение кривой вокруг оси Ох

$$F(u, v) := u$$

mesh := 30

$$G(u, v) := f(u) \cdot \cos(v)$$

$$H(u, v) := f(u) \cdot \sin(v)$$

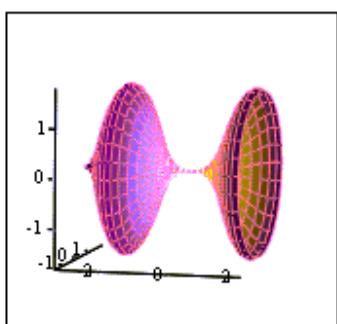
Вращение кривой вокруг оси Оу

$$X(u, v) := u \cdot \sin(v)$$

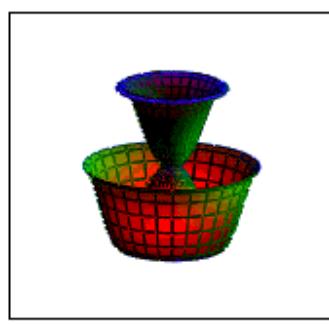
$$Y(u, v) := u \cdot \cos(v)$$

$$Z(u, v) := f(u)$$

SX := CreateMesh(F, G, H, a, b, 0, 2π, mesh) SY := CreateMesh(X, Y, Z, a, b, -π, π, mesh)



SX



SY

1.7 Лекция № 10 (2 часа)

Тема: «Понятие базы данных».

1.7.1 Вопросы лекции:

1. Базы данных и информационные системы.
2. Администрирование баз данных.

1.7.2 Краткое содержание вопросов:

1. Базы данных и информационные системы.

Информационная система представляет собой систему, реализующую автоматизированный сбор, обработку и манипулирование данными и включающую технические средства обработки данных, программное обеспечение и обслуживающий персонал.

База данных обеспечивает хранение информации и представляет собой поименованную совокупность данных, организованных по определенным правилам, включающим общие принципы описания, хранения и манипулирования данными.

2. Администрирование баз данных.

Поскольку система баз данных может быть весьма большой и может иметь много пользователей, должно существовать лицо или группа лиц, управляющих этой системой. Такое лицо называется администратором базы данных (АБД).

В любой базе данных должен быть хотя бы один человек, выполняющий административные обязанности; если база данных большая, эти обязанности могут быть распределены между несколькими администраторами.

В обязанности администратора могут входить:

- инсталляция и обновление версий сервера и прикладных инструментов
- распределение дисковой памяти и планирование будущих требований системы к памяти
- создание первичных структур памяти в базе данных (табличных пространств) по мере проектирования приложений разработчиками приложений
- создание первичных объектов (таблиц, представлений, индексов) по мере проектирования приложений разработчиками
- модификация структуры базы данных в соответствии с потребностями приложений
- зачисление пользователей и поддержание защиты системы
- соблюдение лицензионного соглашения управление и отслеживание доступа пользователей к базе данных
- отслеживание и оптимизация производительности базы данных
- планирование резервного копирования и восстановления
- поддержание архивных данных на устройствах хранения информации
- осуществление резервного копирования и восстановления
- обращение в корпорацию за техническим сопровождением

1.8 Лекция № 11 (2 часа)

Тема: «Архитектура СУБД»

1.11.1 Вопросы лекции:

1. Реляционная модель данных.
2. Иерархическая модель данных.
3. Сетевая модель данных.

1.11.2 Краткое содержание вопросов:

1. Реляционная модель данных.

В основе реляционных систем лежит реляционная модель данных. Принципы реляционной модели были заложены в 1969–1970 гг. американским ученым Е. Ф. Коддом (E. F. Codd), в то время работавшим в корпорации IBM. Будучи математиком по образованию, он привнес в область управления базами данных строгие математические принципы и точность, которых не хватало ранним системам. Хотя реляционный подход утвердился не сразу, можно отметить, что почти все созданные с конца 70-х гг. продукты баз данных основаны именно на реляционном подходе. Подавляющее большинство научных исследований в области баз данных в течение последних 35 лет также проводилось именно в этом направлении.

2. Иерархическая модель данных.

Иерархическая модель данных — это модель данных, где используется представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней. Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня.

3. Сетевая модель данных.

Сетевая модель данных определяется в тех же терминах, что и иерархическая. Она состоит из множества записей, которые могут быть владельцами или членами групповых отношений. Связь между записью-владельцем и записью-членом также имеет вид 1:N.

Основное различие этих моделей состоит в том, что в сетевой модели запись может быть членом более чем одного группового отношения. Согласно этой модели каждое групповое отношение именуется и проводится различие между его типом и экземпляром. Тип группового отношения задается его именем и определяет свойства общие для всех экземпляров данного типа. Экземпляр группового отношения представляется записью-владельцем и множеством (возможно пустым) подчиненных записей. При этом имеется следующее ограничение: экземпляр записи не может быть членом двух экземпляров групповых отношений одного типа (т.е. сотрудник из примера в п.3.1, например, не может работать в двух отделах).

1.9 Лекция № 12,13 (2 часа)

Тема: «СУБД ACCESS»

1.9.1 Вопросы лекции:

1. Объекты Access.
2. Режимы работы с Access.
3. Таблицы. Создание таблиц
4. Особенности таблиц баз данных

Системы управления базами данных (СУБД) — это программные средства, с помощью которых можно создавать базы данных, наполнять их и работать с ними. В мире существует немало различных систем управления базами данных. Многие из них на самом деле являются не законченными продуктами, а специализированными языками программирования, с помощью которых каждый, освоивший данный язык, может сам создавать такие структуры, какие ему удобны, и вводить в них необходимые элементы управления. К подобным языкам относятся Clipper, Paradox, FoxPro и другие.

Необходимость программировать всегда сдерживала широкое внедрение баз данных в малом бизнесе. Крупные предприятия могли позволить себе сделать заказ на программирование специализированной системы «под себя». Малым предприятиям зачастую не по силам было не только решить, но даже и правильно сформулировать эту задачу.

Положение изменилось с появлением в составе пакета Microsoft Office системы

управления базами данных Access. Ранние версии этой программы имели номера Access 2.0 и Access 95.

С помощью Access обычные пользователи получили удобное средство для создания и эксплуатации достаточно мощных баз данных без необходимости что-либо программировать. В то же время работа с Access не исключает возможности программирования. При желании систему можно развивать и настраивать собственными силами. Для этого надо владеть основами программирования на языке Visual Basic.

Еще одним дополнительным достоинством Access является интегрированность этой программы с Excel, [Word](#) и другими программами пакета Office. Данные, созданные в разных приложениях, входящих в этот пакет, легко импортируются и экспортируются из одного приложения в другое.

^ Объекты Access Исходное окно Access отличается простотой и лаконичностью. Шесть вкладок этого окна представляют шесть видов объектов, с которыми работает программа.

1. *Таблицы* — основные объекты базы данных. С ними мы уже знакомы. В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц.
2. *Запросы* — это специальные структуры, предназначенные для обработки данных базы. С помощью запросов данные упорядочивают, фильтруют, отбирают, изменяют, объединяют, то есть обрабатывают.
3. *Формы* — это объекты, с помощью которых в базу вводят новые данные или просматривают имеющиеся.
4. *Отчеты* — это формы «наоборот». С их помощью данные выдают на принтер в удобном и наглядном виде.
5. *Макросы* — это макрокоманды. Если какие-то операции с базой производятся особенно часто, имеет смысл сгруппировать несколько команд в один макрос и назначить его выделенной комбинации клавиш.
6. *Модули* — это программные процедуры, написаны на языке Visual Basic. Если стандартных средств Access не хватает, программист может расширить возможности системы, написав для этого необходимые модули.

^ Режимы работы с Access



<£? Макросы <^£ Модули ѡ Открыть

Конструктор

Кнопка раскрывающего списка выбора типа поля

|||||

III

и

: Описание-С организационной точки зрения в работе с любой базой данных есть два разных режима:

1. *проектировочный* и
2. *эксплуатационный* (пользовательский).

Создатель базы имеет право создавать в ней новые объекты (например, таблицы), задавать их структуру, свойства полей, устанавливать необходимые связи. Он работает со *структурой базы* и имеет полный доступ к базе. У одной базы может быть один, два или несколько разработчиков.

Пользователь базы — это лицо, которое наполняет ее информацией с помощью форм, обрабатывает данные с помощью запросов и получает результат в виде *результатирующих таблиц* или *отчетов*. У одной базы могут быть миллионы пользователей, и, конечно, доступ к структуре базы для них закрыт.

Взгляните на стартовое окно базы данных. Кроме шести вкладок для основных объектов оно содержит три командные кнопки: Открыть, Конструктор, Создать. С их помощью и выбирается режим работы с базой.

1. Кнопка Открыть открывает избранный объект. Если это таблица, ее можно просмотреть, внести новые записи или изменить те, что были внесены ранее.
2. Кнопка Конструктор тоже открывает избранный объект, но по-другому. Она открывает его структуру и позволяет править не ее содержимое, а устройство. Если это таблица, в нее можно вводить новые поля или изменять свойства существующих полей. Если это форма, в ней можно изменять или создавать *элементы управления*. Очевидно, что этот режим служит не для пользователей базы, а для ее разработчиков.
3. Действие командной кнопки Создать соответствует ее названию. Она служит для создания новых объектов. Этот элемент управления тоже предназначен для проектировщиков базы. Таблицы, запросы, формы и отчеты можно создавать несколькими разными способами: автоматически, вручную или с помощью Мастера.

Таблицы. Создание таблиц

Таблицы — основные объекты базы данных. Без запросов, форм, отчетов и прочего можно обойтись, но если нет таблиц, то данные некуда записывать, а значит, нет и базы. Создание базы начинается с создания первой таблицы.

Создание таблицы состоит в задании ее полей и назначении их свойств. Оно начинается с щелчка на кнопке Создать в окне База данных.

Есть несколько способов создания новой таблицы, отличающихся уровнем [автоматизации](#).

1. Самый «автоматичный» способ состоит в импорте таблиц из другой базы, может быть, даже созданной в другой системе. В зависимости от обстоятельств из импортируемой таблицы может поступить структура полей, их названия и свойства, а также и содержимое базы. Если что-то импортируется не совсем так, как надо, необходимые правки (например, в свойства полей) вносят вручную.
2. В тех случаях, когда речь идет о чужой таблице, которая находится на удаленном сервере и которую нельзя импортировать целиком, пользуются режимом Связь с таблицами. Это напоминает подключение к таблице для совместного использования ее данных.

3. Опытные разработчики пользуются Мастером таблиц. Это программа, ускоряющая создание структуры таблицы. Мастер задает ряд вопросов и, руководствуясь полученными ответами, создает структуру таблицы автоматически. Несмотря на то, что этот режим служит для упрощения работы, начинающим пользоваться им не рекомендуется, поскольку, не владея всей терминологией, легко запутаться в вопросах и ответах. Первые таблицы стоит попробовать создать вручную.

4. Пункт Режим таблицы открывает заготовку, в которой все поля имеют формальные имена: Поле1, Поле2... и т. д. и один стандартный текстовый тип. Такую таблицу можно сразу наполнять информацией.

5. Наиболее универсальный ручной метод предоставляет пункт Конструктор. В этом режиме можно самостоятельно задать имена полей, выбрать их тип и настроить свойства.

Для изменения свойств полей надо перейти в режим Конструктор щелчком на кнопке Вид. Чтобы вставить новое поле, надо установить указатель мыши на маркер поля и нажать клавишу INSERT. Чтобы удалить поле, его надо выделить и нажать клавишу DELETE. Закончив создание структуры, можно щелкнуть на кнопке Вид и перейти в Режим таблицы для заполнения ее данными.

^ Особенности таблиц баз данных

Прежде чем мы приступим к изучению приемов работы с таблицами баз данных, надо обратить внимание на одну особенность всех баз данных, связанную с сохранением информации. Тех, кто привык работать с другими классами программ, она поначалу обескураживает.

Обычно с документом в программах можно делать все что угодно, пока не настала пора его сохранять. Испортив неаккуратными действиями исходный документ, можно отказаться от сохранения и вернуться к работе с прежней копией. В базах данных это не так.

Таблицы баз данных не являются самостоятельными документами. Сама база — это документ. Ей соответствует файл на диске, и мы можем сделать его копию. Структура таблиц — тоже документ. В некоторых системах она имеет отдельный файл, а в некоторых (например, в Access) такого файла нет, но структура таблиц входит в состав общего файла базы данных наряду с запросами, формами, отчетами и другими объектами. При изменении структуры таблицы система управления базой данных всегда выдает запрос на сохранение изменений.

Но содержание таблиц — это совсем другое дело. Его нельзя сохранить принудительной командой или, наоборот, отказаться от его сохранения. Все изменения в таблицах сохраняются автоматически *в режиме реального времени*. Режим реального времени означает, что, пока мы работаем с таблицей, происходит ее непрерывное сохранение. Как

только заканчивается ввод данных в одно поле и происходит переход к следующему полю, данные немедленно записываются на жесткий диск.

Профессионалы высоко ценят эту особенность систем управления базами данных, а начинающих она иногда вводит в заблуждение. Экспериментируя с таблицами, надо знать, что все изменения, которые вносятся в их содержание, имеют необратимый характер. Нельзя что-то изменить, удалить, а потом отказаться от сохранения и вернуться к исходному варианту.

Эта особенность систем управления базами данных требует аккуратного отношения к работе с таблицами. Для экспериментов надо создавать отдельные копии базы или таблиц и работать с ними.

1.9.2 Краткое содержание вопросов:

1.10 Лекция № 14 (2 часа)

Тема: «Аппаратные средства связи»

1.10.1 Вопросы лекции:

1. Процедура передачи данных.
2. Архитектура сетей.

1.10.2 Краткое содержание вопросов:

1. Процедура передачи данных.

Физическая среда является основой, на которой строятся физические средства соединения. Сопряжение с физическими средствами соединения посредством физической среды обеспечивает Физический уровень. В качестве физической среды широко используются эфир, металлы, оптическое стекло и кварц. На физическом уровне находится носитель, по которому передаются данные. Среда передачи данных может включать как кабельные, так и беспроводные технологии. Хотя физические кабели являются наиболее распространенными носителями для сетевых коммуникаций, беспроводные технологии все более внедряются благодаря их способности связывать глобальные сети.

2. Архитектура сетей.

Архитектура сети — это реализованная структура сети передачи данных, определяющая её топологию, состав устройств и правила их взаимодействия в сети. В рамках архитектуры сети рассматриваются вопросы кодирования информации, её адресации и передачи, управления потоком сообщений, контроля ошибок и анализа работы сети в аварийных ситуациях и при ухудшении характеристик.

1.11 Лекция № 15 (2 часа)

Тема: «Локальные и глобальные вычислительные сети»

1.11.1 Вопросы лекции:

1. Структурная организация локальных вычислительных сетей.
2. Глобальная сеть Internet.

1.11.2 Краткое содержание вопросов:

1. Структурная организация локальных вычислительных сетей.

В состав локальной сети (ЛВС) входит следующее оборудование:
Активное оборудование – коммутаторы, маршрутизаторы, медиаконверторы;
Пассивное оборудование – кабели, монтажные шкафы, кабельные каналы, коммутационные панели, информационные розетки;
Компьютерное и периферийное оборудование – серверы, рабочие станции, принтеры, сканеры.

2. Глобальная сеть Internet.

Internet – всемирная информационная компьютерная сеть, представляющая собой объединение множества региональных компьютерных сетей и компьютеров, обменивающих друг с другом информацией по каналам общественных телекоммуникаций (выделенным телефонным аналоговым и цифровым линиям, оптическим каналам связи и радиоканалам, в том числе спутниковым линиям связи).

Информация в Internet хранится на серверах. Серверы имеют свои адреса и управляются специализированными программами. Они позволяют пересыпалть почту и файлы, производить поиск в базах данных и выполнять другие задачи.

1.12 Лекция № 16 (2 часа)

Тема: «Организация беспроводных сетей».

1.12.1 Вопросы лекции:

1. Основы передачи данных в беспроводных сетях.
2. Wi-Fi.

1.12.2 Краткое содержание вопросов:

1. Основы передачи данных в беспроводных сетях.

Беспроводные технологии служат для передачи информации на расстояние между двумя и более точками, не требуя связи их проводами. Для передачи информации может использоваться инфракрасное излучение, радиоволны, оптическое или лазерное излучение.

В настоящее время существует множество беспроводных технологий, наиболее часто известных пользователям по их маркетинговым названиям, таким как Wi-Fi,

WiMAX, Bluetooth. Каждая технология обладает определёнными характеристиками, которые определяют её область применения.

Существуют различные подходы к классификации беспроводных технологий.

2. Wi-Fi.

Разработан консорциумом Wi-Fi Alliance на базе стандартов IEEE 802.11, «Wi-Fi» — торговая марка «Wi-Fi Alliance». Название технологии - Wireless-Fidelity («беспроводная точность») по аналогии с Hi-Fi.

В начале использования установка Wireless LAN рекомендовалась там, где развертывание кабельной системы было невозможно или экономически нецелесообразно. В настоящий момент во многих организациях используется Wi-Fi, так как при определенных условиях скорость работы сети уже превышает 100 Мбит/сек. Пользователи могут перемещаться между точками доступа по территории покрытия сети Wi-Fi.

1.13 Лекция № 17,18 (4 часа)

Тема: «Использование Visual Basic for Applications».

1.13.1 Вопросы лекции:

- 1 Обоснование языка программирования
- 2 Введение в Visual Basic for Application
- 2.1 Об объектах и коллекциях
- 2.2 Примеры использования VBA в среде Access.
3. Разработка и эксплуатация АИС

1.13.2 Краткое содержание вопросов:

- 1 Обоснование выбора языка программирования

Лучшей среды программирования для создания информационного обеспечения в контуре любой АИС нашего бизнес - проекта можно считать MS Office. Одним из наиболее важных и полезных аспектов использования MS Office является автоматизация некоторых процессов взаимодействия пользователя и приложений Office(Excel, Access, Word), которые позволяют решать, практически, все современные бизнес - задачи : от создания простых документов и отчетов до полной автоматизации документооборота с использованием систем управления БД. Это связано с тем, что все приложения Office поддерживают язык программирования Basic for application (VBA). Важнейшим достоинством является возможность объединить любые приложения MS Office для выполнения одной или нескольких задач.

В последнее время все более и более возрастает значимость разработки приложений на базе электронных таблиц Excel - продукта с высокой степенью программируемости из-за поддержки VBA. Таким образом, для разработчиков АИС важными являются следующие возможности использования VB в среде MS Excel:

- 1) Файловая структура. Ориентация на многолистовую структуру позволяет легко организовывать элементы приложения и хранить его в единственном файле;
- 2) VBA - это макроязык, позволяет создавать структурированные программы непосредственно в Excel.
- 3) Excel позволяет довольно легко вставить в рабочий лист различные элементы управления, например, кнопки, поля со списком, переключатели. Можно также легко создавать диалоговые окна, имеющие профессиональный внешний вид.
- 4) Для упрощения формул и вычислений можно с помощью VBA создавать пользовательские функции рабочих листов.

5) Имеется возможность вносить изменения в элементы меню, добавлять в имеющиеся меню новые элементы или создавать полностью новые меню.

В связи с этим в теоретическом разделе наряду с вопросами разработки АИС, рассматриваются также вопросы связанные с использованием VBA и MS Excel.

1.2 Введение в Visual Basic For Application

Программирование в Excel , в основном, сводится к управлению объектами. Эта задача выполняется с помощью инструкций, введенных на языке, понятном Excel. Рассмотрим далее этот язык, а также объекты, включенные в Excel .

История языка BASIC

Многие опытные программисты не воспринимают идею программирования на BASIC. Само название (аббревиатура от Beginner's LP-purpose Symbolic Instruction Code --универсальный символический язык инструкций для начинающих) предполагает, что это не профессиональный язык.

Действительно, BASIC был разработан в начале 1960-х годов и задумывался как наглядное средство преподавания методов программирования студентам колледжей. BASIC довольно быстро приобрел большую популярность, и сейчас поддерживается во многих типах компьютеров.

С годами BASIC развивался и улучшался. Например, во многих ранних версиях он был интерпретируемым языком. Каждая строка перед выполнением интерпретировалась, чем и была обусловлена медленная скорость обработки кода. В большинстве современных вариантов языка BASIC программа компилируется, и в результате выполнение программы значительно ускорилось, а перемещаемость программы улучшилась.

BASIC стал намного популярнее в 1991 году, когда компания Microsoft выпустила Visual Basic для Windows (на данный момент существующий в версии 6.0). Этот продукт облегчил массовую разработку самостоятельных приложений для Windows. У Visual Basic мало общего с ранними версиями BASIC, но последний представляет собой основу, на которой построен VBA. Обзор VBA Excel 5 -- это первое приложение на рынке, в котором появился Visual Basic for Applications (VBA). VBA считается стандартным языком написания сценариев для приложений Microsoft, и в настоящее время входит в состав всех приложений Office 2002 и даже приложений других компаний. Следовательно, овладев VBA для Excel, вы сможете сразу перейти к созданию макросов для других программных продуктов Microsoft (равно, как и приложений других компаний). Более того, вы сможете создавать полноценные программные продукты, одновременно использующие функции самых разных приложений.

Объектные модели

Секрет использования VBA заключается в правильном понимании объектной модели В каждом отдельном приложении. Следует отметить. VBA всего лишь управляет объектами, а у каждого программного продукта (Excel. Word. Access. PowerPoint и т.п.) своя объектная модель. Приложением можно управлять программным образом только с помощью объектов, которые представлены в этом приложении.

Например, в объектной модели Excel представлено несколько мощных объектов анализа данных, например, рабочие листы, диаграммы, сводные таблицы, сценарии, а также многочисленные математические, финансовые, инженерные и общие функции. С помощью VBA вы можете работать с этими объектами и разрабатывать автоматизированные процедуры.

Сравнение VBA и XLM

До появления Excel 5 разработчиками использовался мощный (но сложный для понимания) язык макросов под названием XLM. Более поздние версии Excel все еще выполняют макросы XLM, но, начиная с Excel 97, пользователи не имеют возможности записывать макросы на языке XI.M. Для современных разработок используется исключительно VBA.

Основы VBA

Действия в VBA осуществляются в результате выполнения кода VBA. * Вы создаете (или записываете) программу VBA, которая сохраняется в модуле VBA.

* Модуль VBA состоит из процедур.

Процедура, по существу, представляет собой элемент компьютерной программы, выполняющей определенное действие. Ниже приведен пример простой процедуры под названием Test: она вычисляет сумму, а затем отображает результат в окне сообщений:

```
Sub Test()
```

```
Sum = 1 + 1
```

```
MsgBox "Ответ: " & Sum
```

End Sub

* Кроме процедур Sub, в модуле VBA может использоваться второй тип процедур -- функции.

Процедура функции возвращает одно значение (или массив). Функция может быть вызвана из другой процедуры VBA или использоваться в формуле рабочего листа. Далее приведен пример функции с названием AddTwo:

```
Function AddTwo(arg1, arg2)
```

```
    AddTwo = arg1 + arg2
```

```
End Function
```

* VBA управляет объектами, которые представлены запускающим приложением (в данном случае Excel).

Excel позволяет управлять более, чем ста классами объектов, включая рабочую книгу, рабочий лист, диапазон ячеек рабочего листа, диаграмму и нарисованный прямоугольник. В вашем распоряжении находятся и другие объекты, с которыми можно работать в VBA.

*Классы объектов организованы в иерархическую структуру.

Объекты могут быть контейнерами для других объектов. Например, Excel -- это объект под названием Application, он содержит другие объекты, например Workbook (Рабочая книга). Объект Workbook может состоять из других объектов, например Worksheet (Рабочий лист) и Chart (Диаграмма). Объект Worksheet также содержит объекты, например Range (Диапазон), PivotTable (Сводная таблица) и т.д. Организацию таких объектов называют объектной моделью Excel.

Однократные объекты формируют коллекцию. Например, коллекция Worksheets состоит из всех рабочих листов конкретной рабочей книги, а коллекция CommandBars -- из всех объектов CommandBar. Коллекции -- это объекты в себе.

При ссылке на объект, вложенный в другой объект, положение в иерархической структуре объектной модели задается с помощью точки-разделителя. Например, на рабочую книгу с названием Книга1.xls можно сослаться следующим образом:

```
Application.Workbooks("Книга1.xls")
```

Это ссылка на рабочую книгу Книга1.xls в коллекции Workbooks. Коллекция Workbooks находится в объекте Application. Переходя на следующий уровень, вы можете сослаться на лист Лист1 в книге Книга1.xls:

```
Application.Workbooks("Книга1.xls").Worksheets("Лист1")
```

После чего необходимо сослаться на отдельную ячейку:

```
Application.Workbooks("Книга1.xls").Worksheets("Лист1").Range("A1")
```

При опущенной ссылке на объект Excel по умолчанию используются активные объекты. Если книга Лист1 -- активная рабочая книга, то предыдущую ссылку можно упростить:

```
Worksheets("Лист1").Range("A1")
```

Если вы знаете, что лист Лист1 -- активный, то ссылку можно упростить еще больше:

```
Range("A1")
```

Объекты имеют свойства.

Свойство можно считать параметром или настройкой объекта. Например, объект диапазона имеет такие свойства, как Value (Значение) и Name (Имя). Объект диаграммы обладает такими свойствами, как HasTitle (Заголовок) и Type (Тип). Вы вправе использовать VBA, чтобы задать свойства объектов и их изменить.

Свойства в программном коде отделяются от названия объекта точкой. Например, вы можете сослаться на значение в ячейке A1 листа Лист1 следующим образом: Worksheets("Лист1").Range("A1").Value

Рабочая книга Excel является объектом, но она также содержит другие объекты, например, рабочие листы, диаграммы, модули VBA и т.д. Более того, каждый объект в рабочей книге может содержать собственные объекты. Например, объект worksheet (Рабочий лист) включает объекты Range (Диапазон), pivottable (Сводная таблица), Shape (Форма) и т.д.

Помимо свойств, объекты Excel также располагают методами, выполняющими операции над объектом. Например, метод clearContents, удаляет содержимое объекта Range.

В Excel методы иногда используются для изменения свойств объекта. Метод clearContents объекта Range изменяет свойство Value объекта Range.

В VBA существует возможность писать процедуры для управления объектами Excel.

Вы вправе присваивать значения переменным VBA. Переменную можно считать константой, которая используется для хранения конкретного значения. Чтобы присвоить значение ячейки A1 листа Лист1 переменной с названием Interest, используйте следующий оператор VBA:

```
Interest = Worksheets("Лист1").Range("A1").Value
```

У объектов есть методы. Метод -- это действие, которое выполняется над объектом. Например, один из методов объекта Range -- ClearContents. Этот метод удаляет содержимое диапазона ячеек.

Методы вводятся после названия объекта с методом, в роли разделителя выступает точка.

Например, для удаления содержимого ячейки A1 активного рабочего листа используется следующая команда:

R a n g e (" A1 "). ClearContents

VBA также включает конструкции современных языков программирования (в том числе массивы, циклы и т.д.).

Знакомство с редактором Visual Basic

Для работы и просмотра модулей VBA используется редактор Visual Basic (VBE-- Visual Basic Editor).

Модули VBA все еще сохраняются вместе с файлами рабочей книги; просто они не видны до тех пор, пока не запущен редактор VBE.

VBE-- это отдельное приложение, запускающееся в Excel. Программа сама выполняет всю операцию по запуску VBE, когда это необходимо. VBE не может запускаться отдельно от Excel; для запуска VBE используется только Excel.

Запуск VBE

Во время работы в Excel вы можете перейти к окну VBE с помощью одного из следующих способов.

1) нажать <Alt+F11>.

2) Выбрать команду Сервис - Макрос - Редактор Visual Basic.

3) Щелкнуть на кнопке Редактор Visual Basic, расположенной на панели инструментов Visual Basic.

Окна VBE

VBE состоит из ряда элементов. В следующий разделах кратко описаны ключевые компоненты редактора Visual Basic.

Строка меню

Строка меню VBE, естественно, работает, как и строка меню любого другого приложения.

Она содержит команды, используемые для управления различными компонентами VBA.

Кроме того, для выполнения многих команд меню используются комбинации клавиш. Например, для команды ViewAImmediate Window (Вид - Окно отладки) применяется комбинация клавиш <Ctrl+G>.

В VBE также представлены контекстные меню. Щелкнув правой кнопкой мыши практически на любых элементах окна VBE, вы увидите меню, предлагающие ряд команд.

Панели инструментов

Стандартная панель инструментов Standard, которая по умолчанию находится под строкой меню, -- это одна из шести панелей инструментов, используемых в VBE (строка меню тоже считается панелью инструментов). Панели инструментов VBE работают, как и в Excel: вы можете задавать специальные настройки для панелей инструментов, перемещать их, отображать другие панели инструментов и т.д. Для управления панелями инструментов VBE используется команда ViewAToolbarsACustomic (Вид-Панели инструментов-Настройка).

Окно Project Explorer

В окне Project Explorer отображается диаграмма-дерево, состоящая из всех открытых в данный момент в Excel рабочих книг (включая дополнительные элементы и скрытые рабочие книги).

Каждая рабочая книга известна как проект.

Если в редакторе Visual Basic окно Project Explorer не отображено, нажмите <Ctrl+R>. Чтобы скрыть его, щелкните на кнопке закрытия строки заголовка (или щелкните правой кнопкой мыши в любом месте окна и выберите Hide из контекстного меню).

Окно кода

Окно кода (которое иногда называют Module) содержит код VBA. Для каждого элемента проекта представлено собственное окно кода. Чтобы просмотреть код объекта, дважды щелкните мышью на этом объекте в окне Project Explorer. Например, чтобы просмотреть код объекта Лист1, дважды щелкните на элементе Лист1 в окне Project Explorer. Если вы не создавали для него VBA-кода, это открывшееся окно будет пустым. Существует еще один способ просмотреть код объекта -- выделите этот объект в окне Project Explorer, а затем щелкните на кнопке View Code (Просмотр кода) на панели инструментов вверху окна Project Explorer.

Окно Immediate

Окно Immediate предназначено для непосредственного выполнения операторов VBA. Тестирования операторов и отладки кода. Это окно может отображаться и скрываться. Если окно Immediate в данный момент не отображается на экране, нажмите <Ctrl+G>. Чтобы закрыть окно Immediate, щелкните на кнопке закрытия в его строке заголовка (или щелкните правой кнопкой мыши в любом месте окна и выберите Hide из контекстного меню).

Работа с Project Explorer

При работе в редакторе Visual Basic каждая рабочая книга Excel и открытые в данный момент надстройки рассматриваются как проекты. Проект можно считать коллекцией объектов, организованных в виде иерархической структуры. Вы раскроете проект, если щелкнете на знаке "плюс" слева от названия проекта в окне Project Explorer. Проект сворачивается при щелчке на знаке "минус" слева от названия проекта. Кроме того, для разворачивания и сворачивания проекта можно использовать кнопку ToggleFolders (Показать папки) на панели инструментов окна Project Explorer. При попытке развернуть проект, защищенный паролем, отображается окно ввода пароля.

Дерево каждого проекта в развернутом виде имеет как минимум один узел под названием Microsoft Excel Objects. В этом узле содержатся элементы каждого рабочего листа и лист диаграмм рабочей книги (рабочий лист считается объектом), а также объект под названием ЭтаКнига, представляющий объект Active Workbook. Если в проекте используются модули VBA, то в дереве отображается также узел Modules, в котором перечислены модули. Проект может также содержать узел Forms, содержащий объекты UserForm (пользовательские формы, известные как пользовательские диалоговые окна). Если в проекте находятся модули классов, то в дереве отображается узел под названием Class Modules.

В Excel 2002 при добавлении ссылки на проект (с помощью команды Tools\References) в дереве проекта появляется еще один узел: References. Каждая ссылка является отдельным объектом. Объекты, перечисленные в этом узле, не имеют окна кода.

Добавление нового модуля VBA

Чтобы добавить в проект новый модуль VBA, выделите название проекта в окне Project Explorer и выберите команду Insert - Module (Вставка - Модуль). Также можно щелкнуть правой кнопкой мыши на названии проекта и выбрать команду Insert - Module в контекстном меню. При записи макроса Excel автоматически вставляет модуль VBA для хранения записанного кода.

Удаление модуля VBA

Чтобы удалить из проекта модуль VBA или модуль класса, выделите название модуля в окне Project Explorer и используйте команду File - Remove xxx (где xxx-- название модуля). Кроме того, вы можете щелкнуть правой кнопкой мыши на названии модуля и выбрать команду Remove xxx из контекстного меню.

Экспорт и импорт объектов

За исключением объектов, перечисленных в узле References, каждый объект в проекте можно сохранить в отдельном файле. Сохранение отдельного объекта в проекте называется экспортом. Соответственно, вы можете также импортировать объекты в проект. Экспорт и импорт объектов полезен, если уже существующий объект (например, модуль VBA или форму UserForm) можно использовать в другом проекте.

Чтобы экспорттировать объект, выберите его в окне Project Explorer и выполните команду File - Export File (или нажмите <Ctrl+E>). При этом отображается диалоговое окно, запрашивающее имя файла. Обратите внимание, что сам объект остается в проекте (а экспортируется только его копия). Если вы экспортите объект UserForm, экспортируется также весь код, связанный с формой UserForm.

Чтобы импортировать файл в проект, выберите имя проекта в окне Project Explorer и выполните команду File - Import File. Появится диалоговое окно, в котором необходимо указать имя файла. Вы можете импортировать только те файлы, которые экспортированы с помощью команды File--Export File.

Если вы решили скопировать в другой проект модуль или объект UserForm, не обязательно использовать функции экспортации и импорта. Убедитесь, что оба проекта открыты, затем активизируйте окно Project Explorer и перетащите необходимый объект из одного проекта в другой.

Работа с окнами кода

Каждому объекту в проекте соответствует свое окно кода. Такими объектами могут быть: сама рабочая книга (ЭтаКнига в окне Project Explorer);

рабочий лист или лист диаграмм рабочей книги (например. Лист1 или Диаграмме 1 в окне Project);

модуль VBA;

модуль класса (специальный тип модуля, позволяющий создавать новые классы объектов);
форма UserForm.

Сохранение программы VBA

Как правило, окно кода содержит четыре типа кода.

Процедуры (процедуры Sub). Процедура -- это набор инструкций, выполняющих определенное действие.

Процедуры функции. Функция -- это набор инструкций, возвращающий значение или массив значений (концепция функции VB A подобна такой же функции Excel).

Процедуры свойств. Специальные процедуры, используемые в модулях классов.

Объяснение -- это информация о переменной, предоставляемая VBA. Например, можно объявить тип данных для переменных, которые вы планируете использовать в коде.

В отдельном модуле VBA может храниться любое количество процедур, функций и объявлений. Способ организации модуля VBA зависит только от вашего желания. Некоторые предпочитают записывать весь код VBA приложения в одном модуле VBA; другие разделяют код на несколько разных модулей.

Несмотря на то, что предоставляются широкие возможности по определению места хранения кода VBA, существуют некоторые ограничения на его расположение. Процедуры обработки событий должны содержаться в окне кода объекта, которому соответствует это событие.

Например, если вы пишете процедуру, которая выполняется при открытии рабочей книги, то эта процедура должна располагаться в окне кода для объекта ЭтаКнига и иметь специальное название.

Введение кода VBA

Для того чтобы выполнить одно из действий программным образом, необходимо написать программу VBA в окне кода. Код VBA располагается в процедуре. Процедура состоит из операторов VBA. На данном этапе (для примера) остановимся только на одном типе окна кода: модуль VBA.

Вы можете добавить код в модуль VBA тремя способами.

Ввести код традиционным способом: с клавиатуры.

Использовать функцию записи макросов в Excel, чтобы записать действия и преобразовать их в код VBA.

Скопировать текст программы из другого модуля и вставить его в модуль, над которым работаете.

Ввод кода вручную

Иногда самый простой путь является наилучшим. Непосредственное введение кода связано с использованием клавиатуры, т.е. вы вводите код программы с помощью клавиатуры. Клавиша <Tab> при этом используется с целью задать отступ в строках, которые логически принадлежат одной группе (например, условные операторы If и End If)- Это совершенно не обязательно, но помогает быстрее освоить программу, анализируя ее блочную структуру. Именно поэтому подобный подход в программировании называется "хорошим стилем".

Ввод и редактирование кода в модуле VBA выполняется обычным образом. Вы можете выделять текст, копировать, вырезать его, а затем вставлять в другое место программы.

Отдельная инструкция в VBA может иметь произвольную длину. Однако, чтобы обеспечивать удобочитаемость кода, длинные инструкции лучше разбить на две или более строк. Для этого следует в конце строки ввести пробел и символ подчеркивания, а затем нажать <Enter> и продолжить инструкцию в следующей строке. Например, ниже приведен один оператор VBA, разбитый на четыре строки.

```
MsgBox "Невозможно найти" & UCASE(SHORCUTMENUFILE) _  
& vbCrLf & vbCrLf & "Файл должен находиться в _  
" & ThisWorkbook.Path & vbCrLf & vbCrLf & _  
"Возможно, требуется переустановить BudgetMan", vbCritical, APPNAME
```

Обратите внимание, что три последние строки этого оператора введены с отступом. Это необязательное условие, однако таким образом вы указываете что на самом деле эти четыре строки являются одним оператором.

Как и в Excel, в VBE есть несколько уровней отмены выполненных операций. Поэтому, если вы по ошибке удалили инструкцию, можете несколько раз щелкнуть на кнопке Undo (Отменить)

или нажать <Ctrl+Z>, и инструкция вновь появится в коде. После отмены операции можно щелкнуть на кнопке Redo (Вернуть), чтобы вернуть изменения, которые ранее отменены. Эта функция поможет исправить критически важные ошибки, поэтому не пренебрегайте ее использованием.

Использование функции записи макросов

Одним из способов создания кода модуля VBA является запись последовательности действий с помощью специальной функции записи макросов Excel.

В следующем примере показано, как записать макрос, изменяющий ориентацию страницы на альбомную. Если вы хотите получить его самостоятельно, то начните работу с пустой рабочей книги и выполните следующие действия.

1. Активизируйте рабочий лист в книге (подойдет любой лист).

2. Выберите команду Сервис - Макрос - Начать запись.

При этом Excel отображает диалоговое окно Запись макроса.

3. Щелкните на кнопке OK, чтобы принять параметры по умолчанию.

Excel автоматически вставляет новый модуль VBA в проект. Начиная с этого момента Excel, преобразовывает ваши действия в код VBA. При записи в строке состояния отображается слово Запись, кроме того, в окно добавляется небольшая плавающая панель инструментов, содержащая две кнопки (Остановить запись и Относительная ссылка).

4. Выполните команду Файл - Параметры страницы.

Excel отображает диалоговое окно Параметры страницы.

Выберите переключатель Альбомная и щелкните на кнопке OK, чтобы закрыть диалоговое окно.

Щелкните на кнопке Остановить запись на панели инструментов (или выберите Сервис--Макрос - Остановить запись).

Excel прекращает записывать ваши действия.

Чтобы просмотреть макрос, запустите VBE (проще всего нажать <Alt+F11>) и найдите проект в окне Project Explorer. Щелкните на узле Modules, чтобы развернуть его. Затем щелкните на элементе Module 1, чтобы отобразить окно кода (если в проекте уже присутствовал модуль Module1, новый макрос будет находиться в модуле Module2). Код, созданный одной командой, представлен в листинге 1. Если вы используете не Excel 2002, а иную версию, текст программы может немного отличаться.

2.Методические материалы по выполнению лабораторных работ

2.1.Лабораторная работа № ЛР-1-5 (10 часов)

Тема: «Работа в HTML».

2.1.1 Цель работы: научится работать в HTML

2.1.2 Задачи работы:

1. ознакомиться с основными понятиями: тег, атрибут тега.
2. ознакомиться со структурой HTML документа
3. создать простейшую страницу HTML

2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:

ПК, методические рекомендации

2.1.4 Описание (ход) работы:

1. создайте на локальном диске папку Lab_1.
2. Откройте текстовый редактор Блокнот.

```
<html>
<head></head>
<body></body>
</html>
```

3. В текстовом редакторе напишите следующий код:
4. сохраните данный документ в папке Lab_1 и назовите его Index1.html
5. При сохранении в поле кодировка выберите кодировку UTF-8 (она необходима для корректного отображения браузером символов).
6. Откройте сохраненный файл с помощью браузера (у вас должна открыться пустая страница)
7. Откройте файл Index1.html в блокноте
8. Внутри тега <HEAD> добавьте тег <TITLE> Моя первая страница </TITLE>
9. Сохраните файл и снова откройте его с помощью браузера. Теперь ваша вкладка должна называться МОЯ ПЕРВАЯ СТРАНИЦА.
10. Добавьте внутрь тега <BODY></BODY> Текст ПРИВЕТ МИР!!!
11. Сохраните файл и посмотрите результат в браузере.
12. Для тега <BODY> добавьте атрибут **bgcolor="#ff00ff** (<body bgcolor="#ff00ff>)
13. Сохраните и посмотрите результат в браузере (цвет фона должен поменяться)
14. Для тега <BODY> добавьте атрибут **text="#ff0000**
15. Сохраните и посмотрите результат в браузере (должен поменяться цвет текста)

По завершению работы у вас должен быть файл Index1.html который сохранен в папке Lab_1.

```
<html>
<head></head>
<body></body>
</html>
```

Который содержит следующий код:

Практическое задание.

Вам необходимо создать страницу которая:

1. в качестве названия вкладки содержит ваше имя и фамилию

2. Фон страницы залит зеленым цветом

3. Цвет текста на странице красный

4. Текст на странице:

Студент(ка) ФИО, Группа: ХХ ИТ ХХ, Специальность: НАЗВАНИЕ ВАШЕЙ СПЕЦИАЛЬНОСТИ

Файл сохранить в папке Lab_1 с названием Index2.html

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скрином экрана.

2.2.Лабораторная работа № ЛР-6-9 (8 часов)

Тема: «Работа с MathCAD».

2.2.1 Цель работы: Научится пользоваться основными функциями

2.2.2 Задачи работы:

1. Ознакомится с интерфейсом

2. работа с функциями

2.2.3 Перечень приборов, материалов, используемых в лабораторной работе:

ПК, методические рекомендации

2.2.4 Описание (ход) работы:

I).Найдите значения функций при $x=1$.

II). Найдите производные указанных функций.

Варианты заданий 1–20.

I).Найдите значения функций при $x=1$.

II). Найдите производные указанных функций.

1. а) $f(x) = \frac{3}{5}x^5 - \frac{1}{2x^4} + \frac{2}{\sqrt[4]{x^3}} + 7$; б) $f(x) = \frac{e^x - \sin x}{\cos x + \sqrt{x}}$;
в) $f(x) = \sqrt[4]{x^2 + \ln x}$.

2. а) $f(x) = \frac{2}{3}x^3 - \frac{1}{3x^8} + \frac{5}{\sqrt[5]{x^3}} - 6$; б) $f(x) = (1-x^2)(\lg x + 3^x)$;
в) $f(x) = e^{\sin 5x - 3}$.

3. а) $f(x) = \frac{4}{5}x^5 - \frac{1}{6x^6} + \frac{7}{\sqrt[4]{x^3}} + 2$; б) $f(x) = \frac{\ln x - \operatorname{tg} x}{7^x - 5}$;
в) $f(x) = \sqrt{x^5 + \sin 5x}$.

4. а) $f(x) = 3x^2 - \frac{1}{7x^7} + \frac{3}{\sqrt[3]{x^2}} + 1$; б) $f(x) = \frac{5^x - \ln x}{\cos x - 3}$;
в) $f(x) = \arcsin(5x^3 + 1)$.

5. а) $f(x) = 4x^5 - \frac{7}{4x^4} + \frac{3}{\sqrt[3]{x^2}} + 2$; б) $f(x) = \frac{\sin x - \cos x}{3^x - \ln x}$;
в) $f(x) = \cos(2x^2 + 3)$.

6. а) $f(x) = 2x^5 + \frac{4}{5x^5} - \frac{2}{\sqrt{x}} + 3$; б) $f(x) = (x^2 - 3)(\sin x + 5^x)$;
в) $f(x) = e^{\sin 7x + 3}$.

2.3.Лабораторная работа № ЛР-10-13 (8 часов)

Тема: «Решение систем линейных алгебраических уравнений в MathCAD»

2.3.1 Цель работы: научиться решать уравнения и системы уравнений

2.3.2 Задачи работы:

1. Ознакомится с интерфейсом

2. работа с функциями

2.3.3 Перечень приборов, материалов, используемых в лабораторной работе:

ПК, методические рекомендации

2.1.4 Описание (ход) работы:

уравнений

Варианты заданий 1–20.

Решите систему линейных уравнений и сделайте проверку.

$$1. \begin{cases} x + 2y - z = 5, \\ 2x - y + 5z = -7, \\ 5x - y + 2z = -4. \end{cases} \quad 2. \begin{cases} 2x + 3y - 5z = 1, \\ 3x + 4y - 3z = 2, \\ x - 3y + 7z = 5. \end{cases} \quad 3. \begin{cases} 7x - 3y + z = 5, \\ x + 2y - z = -4, \\ 3x + y - z = -3. \end{cases}$$

$$4. \begin{cases} 5x + y + 6z = -3, \\ 4x + 3y - z = 2, \\ x + 2y - 5z = 3. \end{cases} \quad 5. \begin{cases} 5x - 3y + z = -3, \\ 3x - y + 2z = 1, \\ x + 5y + z = 1. \end{cases} \quad 6. \begin{cases} 8x + 2y - 7z = 3, \\ x - 3y + 5z = 3, \\ 5x - 2y + 4z = 7. \end{cases}$$

$$7. \begin{cases} 3x - 4y + z = 5, \\ 2x - y + 3z = 1, \\ x + 5y - z = 3. \end{cases} \quad 8. \begin{cases} 7x - y + 2z = 5, \\ 2x + y - 3z = -7, \\ x - 5y + z = 7. \end{cases} \quad 9. \begin{cases} x - 4y - z = -3, \\ 3x + 7y + z = -1, \\ 2x + 3y - z = -4. \end{cases}$$

$$10. \begin{cases} x + y + z = 3, \\ 3x - 2y + z = 2, \\ 5x + 2y - 7z = 0. \end{cases} \quad 11. \begin{cases} x - 5y + z = 1, \\ 3x + y - 2z = -7, \\ 2x + 7y + z = 0. \end{cases} \quad 12. \begin{cases} 3x - 4y + 7z = -1, \\ x + 7y + 2z = 0, \\ 2x - 3y + z = 3. \end{cases}$$

$$13. \begin{cases} 5x - 3y + z = 9, \\ 3x - 7y + 6z = 0, \\ x + 2y + z = 1. \end{cases} \quad 14. \begin{cases} x + 2y + 5z = -1, \\ 5x + y - 3z = 5, \\ 7x - 4y - 3z = -5. \end{cases} \quad 15. \begin{cases} x - y + 7z = -3, \\ 2x + y - 5z = 0, \\ 3x + 2y - 5z = 1. \end{cases}$$

2.4.Лабораторная работа № ЛР-14-17

Тема: «Графические построения в MathCAD»

2.4.1 Цель работы: научиться строить графики

2.4.2 Задачи работы:

1. Двумерные графики

2. трехмерные графики

2.4.3 Перечень приборов, материалов, используемых в лабораторной работе:

ПК, методические рекомендации

2.1.4 Описание (ход) работы:

Варианты заданий 1–20.

Постройте график функции на интервале x от –10 до 10.

1. $f(x) = x^3 - 9x^2 + 24x - 15.$
2. $f(x) = -x^3 - 12x^2 - 45x + 51.$
3. $f(x) = x^3 - 3x + 2.$
4. $f(x) = -x^3 + 9x^2 - 24x + 21.$
5. $f(x) = x^3 + 3x^2 - 2.$
6. $f(x) = -x^3 - 3x^2 - 1.$
7. $f(x) = x^3 - 9x^2 + 24x - 12.$
8. $f(x) = -x^3 + 9x^2 - 24x + 15.$
9. $f(x) = x^3 - 12x^2 + 45x - 45.$
10. $f(x) = -x^3 + 3x - 7.$
11. $f(x) = x^3 + 6x^2 + 9x + 3.$
12. $f(x) = -x^3 - 9x^2 - 24x - 18.$
13. $f(x) = x^3 - 3x^2 + 9.$
14. $f(x) = -x^3 - 6x^2 - 9x - 6.$
15. $f(x) = x^3 - 6x^2 + 9x + 2.$
16. $f(x) = -x^3 + 18x^2 - 105x + 193.$
17. $f(x) = x^3 + 6x^2 + 9x + 6.$
18. $f(x) = -x^3 + 15x^2 - 72x + 107.$
19. $f(x) = x^3 - 12x^2 + 45x - 51.$
20. $f(x) = -x^3 + 3x^2 - 6.$

3. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

3.1 Практическое занятие №1 (2 часа).

Тема: «Общие сведения об информационных технологиях».

3.1.1 Задание для работы:

1. Сущность информационных технологий.
2. Характеристика информационных технологий.

3.1.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.
2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Информационные технологии - это

- 1) совокупность методов и приемов решения типовых задач обработки данных;
- 2) программное обеспечение, используемое для решения типовых задач обработки данных;
- 3) технические устройства, используемые при решении типовых задач обработки данных;

4) способ организации труда разработчиков при решении типовых задач обработки данных.

Ответ: совокупность методов и приемов решения типовых задач обработки данных

2. Информационные технологии включают в себя:

- 1) только сбор информации;
- 2) только хранение информации;
- 3) сбор, хранение, передача, уничтожение информации;
- 4) сбор, хранение, передача, обработка информации.

Ответ: сбор, хранение, передача, обработка информации

3. Совокупность фактов, явлений, событий, представляющих интерес и подлежащих регистрации и обработке:

- 1) данные;
- 2) информационные технологии;
- 3) информация;
- 4) информатика;
- 5) индустрия информатики.

Ответ: информация

4. Информация:

- 1) организованное множество, образующее целостное единство, направленное на достижение определённой цели;
- 2) мера устранения неопределенности в отношении исхода интересующего нас события;
- 3) комплекс средств и методов, обеспечивающих процессы сбора, обработки, хранения и передачи информации;
- 4) верно утверждение в предлагаемых вариантах один и три.

Ответ: мера устранения неопределенности в отношении исхода интересующего нас события

5. Информационная технология АСУ – это:

- 1) система, управляющая работой станка с числовым программным управлением;
- 2) комплекс технических и программных средств, организующих управление объектами в производстве;
- 3) система, помогающая осваивать новый материал, контролирующая знания;
- 4) программно-аппаратный комплекс, который позволяет эффективно проектировать механизмы, здания, узлы агрегатов.

Ответ: комплекс технических и программных средств, организующих управление объектами в производстве

3.2 Практическое занятие № 2 (2 часа).

Тема: «Классификация информационных технологий»

3.2.1 Задание для работы:

1. Современные информационные технологии.
2. Сетевые информационные технологии.

3.2.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.
2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Система защиты информации:
 - 1) осуществление мероприятий с целью системного обеспечения передаваемой, хранимой и обрабатываемой информации;
 - 2) совокупность мер, направленные на обеспечение физической целостности информации;
 - 3) совокупность организационно-технологических мер, технических средств, правовых норм, направленных на противодействие угрозам;
 - 4) верны утверждения предложенных вариантов ответов один и два.
2. Организационно-административные средства защиты:
 - 1) регистрация пользователей компьютерных средств в журналах;
 - 2) разграничение доступа к информации в соответствии с функциональными обязанностями должностных лиц;
 - 3) использование автономных средств защиты аппаратуры;
 - 4) создание контрольно-пропускного режима на территории расположения средств обработки информации;
3. Несуществующий вид компьютерных сетей
 - 1) простейшая компьютерная;
 - 2) локальная вычислительная;
 - 3) распределенная;
 - 4) терминальная;
 - 5) глобальная.

3.3 Практическое занятие № 3 (2 часа).

Тема: «Работа в HTML».

3.3.1 Задание для работы:

1. Основные элементы HTML.
2. Теги HTML.

3. Фреймы в HTML.
4. Изображения в HTML.

3.3.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.
2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Тестовые задания:

1) Три вида топологии сетей это ...

+1. звезда, шина, кольцо

2. дерево, шина, кольцо

3. звезда, дерево, кольцо

4. звезда, шина, дерево

2) В записи адреса сайта “.ru” означает ...

1. сайт на русском языке

+2. домен на территории России

3. сайт создали российские программисты

4. сайт принадлежит организации в России

3) Первые две буквы в аббревиатуре HTML означают ...

1. гипербола

2. гипертаблица

+3. гипертекст

4. гиперссылки

2. Выборочная проверка самостоятельного изучения вопросов: «Создание WEB-страниц»

3.4 Практическое занятие № 4 (2 часа).

Тема: «Работа с MathCAD».

3.4.1 Задание для работы:

1. Формулы и текст.
2. Символьные преобразования.

3.4.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.
2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1) Безбумажные информационные технологии реализуются пакетом программ

+1. Microsoft Office

2. Visual Basic for Applications
 3. MathCAD
 4. Access
- 2) Табличный процессор в Microsoft Office это ...
1. Word
 2. Access
 3. MathCAD
- +4. Excel
- 3) VBA расшифровывается как ...
1. Virtual Basic for Applications
 2. Visual Basic for Access
- +3. Visual Basic for Applications
4. Virtual Basic for Access
2. Выборочная проверка самостоятельного изучения вопросов: «MatLAB»

3.5-6 Практическое занятие № 5 (2 часа).

**Тема: «Решение систем линейных алгебраических уравнений в MathCAD,
Графические построения в MathCAD »**

3.5-6.1 Задание для работы:

1. Формулы Крамера.
2. Метод Гаусса.

3.5-6.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.
2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Тестовые задания:

- 1) Для чего предназначены формы:
 1. для хранения данных базы
 2. для отбора и обработки данных базы

+3. для ввода данных базы и их просмотра

 4. для автоматического выполнения группы команд

- 2) Просмотреть в форме только определенные записи позволяет ...
 - +1. Фильтрация
 2. Сортировка
 3. Отбор
 4. Отчет

3) Для чего предназначены запросы:

1. для хранения данных базы
- +2. для отбора и обработки данных базы
3. для ввода данных базы и их просмотра
4. для вывода обработанных данных базы на принтер

2. Выборочная проверка самостоятельного изучения вопросов: «MatLAB».

3.7 Практическое занятие № 7 (2 часа).

Тема: «Понятие базы данных».

3.7.1 Задание для работы:

1. Базы данных и информационные системы.
2. Администрирование баз данных.

3.7.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.
2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Для чего предназначены формы:

1. для хранения данных базы
2. для отбора и обработки данных базы
- +3. для ввода данных базы и их просмотра
4. для автоматического выполнения группы команд

2. Просмотреть в форме только определенные записи позволяет ...

- +1. Фильтрация
2. Сортировка
3. Отбор
4. Отчет

3. Для чего предназначены запросы:

1. для хранения данных базы
- +2. для отбора и обработки данных базы
3. для ввода данных базы и их просмотра
4. для вывода обработанных данных базы на принтер

3.8 Практическое занятие № 8 (2 часа).

Тема: «Архитектура СУБД»

3.8.1 Задание для работы:

1. Реляционная модель данных.
2. Иерархическая модель данных.

3. Сетевая модель данных.

3.8.2 Краткое описание проводимого занятия:

1. Ответы на вопросы семинарского (практического) занятия.

2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Объект базы данных, представляющий собой бланк, подлежащий заполнению, или маску, накладываемую на набор данных

1) форма;

2) отчет;

3) запрос;

4) таблица.

2. Основные достоинства реляционной модели данных:

1) простота и доступность;

2) все данные зависят друг от друга;

3) возможность непрограммных запросов;

4) применение промышленных СУБД.

3. Основные требования к организации БД

1) производительность - запросы на данные удовлетворяются с такой скоростью, которая требуется для использования данных;

2) возможность обновления данных любым пользователем;

3) сложность внесения изменений - для предотвращения порчи и искажения в БД;

4) гибкость использования - обращение к данным или их поиск осуществляется с помощью различных методов доступа.

3.9 Практическое занятие № 9 (2 часа)

Тема: «СУБД ACCESS»

3.8.1 Задание для работы:

1. Создание таблиц и форм.

2. Создание запросов и отчетов.

3.8.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.

2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. В каких режимах можно отображать отчеты базы данных Access:

1) в режиме конструктора;

2) в режиме просмотра образца;

- 3) в режиме предварительного просмотра;
- 4) в режиме схемы данных.

2. Какие существуют типы запросов в базе данных Access:

- 1) запросы на выборку;
- 2) параметрические;
- 3) перекрестные;
- 4) запросы на бездействие.

3. Укажите базовые объекты MS Access:

- 1) таблицы;
- 2) запросы;
- 3) фильтры;
- 4) отчеты;
- 5) формы;
- 6) столбцы;
- 7) записи.

3.10 Практическое занятие № 10 (2 часа).

Тема: «Аппаратные средства связи»

3.10.1 Задание для работы:

1. Процедура передачи данных.
2. Архитектура сетей.

3.10.2 Краткое описание проводимого занятия:

1. Ответы на вопросы семинарского (практического) занятия.
2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Базы данных и информационные системы
2. Администрирование баз данных
3. Выборочная проверка самостоятельного изучения вопросов: «Создание WEB-страниц»

3.11 Практическое занятие № 11 (2 часа).

Тема: «Локальные и глобальные вычислительные сети»

3.11.1 Задание для работы:

1. Структурная организация локальных вычислительных сетей.
2. Глобальная сеть Internet.

3.11.2 Краткое описание проводимого занятия:

1. Ответы на вопросы семинарского (практического) занятия.

2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Тестовые задания:

1) Числовые переменные бывают двух видов это ...

+1. целые и вещественные

2. дробные и вещественные

3. целые и рациональные

4. рациональные и иррациональные

2) Двумерный массив данных называется ...

1. таблица

2. решетка

+3. матрица

4. список

3) Степени переменных в линейных уравнениях могут быть не более ...

1. нулевой

2. второй

3. третьей

+4. Первой

2. Выборочная проверка самостоятельного изучения вопросов: «Создание WEB-страниц»

3.12 Практическое занятие № 12 (2 часа).

Тема: «Организация беспроводных сетей»

3.12.1 Задание для работы:

1. Основы передачи данных в беспроводных сетях.

2. Wi-Fi.

3.12.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.

2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Процедура передачи данных

2. Архитектура сетей

3. Структурная организация локальных вычислительных сетей

4. Глобальная сеть Internet

5. Выборочная проверка самостоятельного изучения вопросов: «Создание WEB-страниц»

3.13 Практическое занятие № 13 (2 часа).

Тема: «Использование Visual Basic for Applications»

3.13.1 Задание для работы:

1. Среда VBA.
2. Создание макросов.

3.13.2 Краткое описание проводимого занятия:

1. Ответы на вопросы практического занятия.
2. Проведение текущего контроля успеваемости

Задания для проведения текущего контроля успеваемости

1. Безбумажные информационные технологии реализуются пакетом программ ...

- +1. Microsoft Office
- 2. Visual Basic for Applications
- 3. MathCAD
- 4. Access

2. Табличный процессор в Microsoft Office это ...

- 1. Word
- 2. Access
- 3. MathCAD
- +4. Excel

3. VBA расшифровывается как ...

- 1. Virtual Basic for Applications
- 2. Visual Basic for Access
- +3. Visual Basic for Applications
- 4. Virtual Basic for Access