

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

А. А. Бабаева

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ

Учебно-методическое пособие по выполнению лабораторных работ
для студентов направления подготовки
10.05.03 Информационная безопасность автоматизированных систем

Калининград, 2022

Рецензент
доцент кафедры информационной безопасности ФГБОУ ВО
«Калининградский государственный технический университет»
А. Г. Жестовский

Бабаева, А. А.

Пакеты прикладных программ: учебно-методическое пособие по выполнению лабораторных работ для студентов направления подготовки 10.05.03 Информационная безопасность автоматизированных систем. - Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 33 с.

Учебно-методическое пособие является руководством по изучению дисциплины

«Пакеты прикладных программ» студентами, обучающимися по специальности 10.05.03 Информационная безопасность автоматизированных систем. Учебно-методическое пособие предназначено для приобретения практических навыков эксплуатации пакетов прикладных программ в различных областях применения и изучения особенностей их использования.

Список лит. – 4 наименования.

Учебно-методическое пособие рассмотрено и одобрено к использованию в качестве электронного методического материала кафедрой информационной безопасности 19 мая 2022 г., протокол № 7

Учебно-методическое пособие рекомендовано к использованию в учебном процессе в качестве электронного методического материала методической комиссией института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 28 июня 2022 г., протокол № 4

© Федеральное государственное бюджетное образовательное учреждение высшего образования «Калининградский государственный технический университет», 2022 г.
© Бабаева А. А., 2022 г.

ОГЛАВЛЕНИЕ

1. Введение.....	4
2. Лабораторная работа №1. Особенности эксплуатации MicrosoftOffice как пакетной среды	5
3. Лабораторная работа №2. Отладка и выполнение программы в среде VBA...9	
4. Лабораторная работа №3. Применение функций математики в ППП. Особенности эксплуатации математических функций в MicrosoftOffice.....	12
5. Лабораторная работа №4. Решение задач в среде MicrosoftOffice с использованием Excel.....	14
6. Лабораторная работа №5. Использование элементов программирования в среде MicrosoftOffice для решения математических задач.....	15
7. Лабораторная работа № 6 Использование дополнительных функций в среде MicrosoftOffice для решения математических задач.....	17
8. Лабораторная работа № 7 Использование сред создания БД (баз данных) в ППП.....	18
9. Лабораторная работа № 8 Использование языка SQLв БД.	20
10. Лабораторная работа № 9 Создание интерфейсов и дополнительного функционала при работе с БД в ППП.	24
11. Заключение.....	30
12. Литература	31

1. ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов направления 10.05.03 Информационная безопасность автоматизированных систем, изучающих дисциплину «Пакеты прикладных программ».

Цель лабораторного практикума по дисциплине: получение обучающимися теоретических знаний и навыков практического использования пакетов прикладных программ общего и специального назначения.

Лабораторный практикум содержит девять лабораторных работ.

В результате выполнения лабораторных работ ожидается, что студенты сформируют систему знаний о теоретических, методических и технологических основах применения пакетов прикладных программ общего и специального назначения.

1. ЛАБОРАТОРНАЯ РАБОТА № 1. ОСОБЕННОСТИ ЭКСПЛУАТАЦИИ MICROSOFTOFFICE КАК ПАКЕТНОЙ СРЕДЫ. ОБЩИЕ СВЕДЕНИЯ

Цель: освоение интерфейса системы Microsoft Office как пакетной среды, работа с основными элементами, создание макросов, *освоение интерфейса системы AutoCAD, настроек графического редактора, команд вычерчивания графических примитивов и выполнения надписей на чертежах*

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office.

Условия допуска к выполнению:

Подготовить конспект по теоретической части задания.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа посвящена изучению офисного пакета, его особенностям и способам создания макросов.

4. Методические указания и порядок выполнения работы

Если какое-то действие часто повторяется, его выполнение можно автоматизировать с помощью макроса. **Макрос** — это серия команд и функций, хранящихся в модуле **Visual Basic**. Их можно выполнять всякий раз, когда необходимо выполнить данную задачу.

Перед тем как записать или написать макрос, необходимо спланировать шаги и команды, которые он будет выполнять. Если при записи макроса была допущена ошибка, ее исправление будет также записано. Каждый раз при записи макроса он сохраняется в новом модуле, присоединенном к книге.

Редактор Visual Basic позволяет изменять макросы, а также копировать их либо из одного модуля в другой, либо между различными книгами. Кроме того, можно переименовывать модули, в которых хранятся макросы, или переименовывать сами макросы.

Загрузите **Excel**. Создайте книгу с именем **Лабораторная работа**, откройте **Лист2**.

Создание макроса в автоматическом режиме с использованием инструмента для запуска макроса.

1. В меню **Сервис** установите указатель на пункт **Макрос** и выберите **Начать запись**.
2. После появления диалогового окна в поле **Имя макроса** введите имя макроса **ЗагрузкаФайла** (имя макроса набирается без пробелов).
3. Нажмите кнопку **ОК**. Выполните процедуру загрузки файла **Лабораторной работы № 4** из Вашей папки (*операцию выполняйте строго без лишних действий!*). После загрузки файла выберите в Меню **Сервис Макрос Остановить запись**.
4. Закройте вызванный файл с Лабораторной работой № 4.
5. В меню **Сервис** установите указатель на пункт **Макрос** и выберите **Макросы**. В появившемся диалоговом окне **Макрос** установите курсор на имя **ЗагрузкаФайла** и нажмите кнопку **Изменить**.
6. Появится окно редактора программ **Microsoft Visual Basic**. В правой его части будет находиться текст программы созданного макроса на языке **Visual Basic** типа (это только пример, название файла может быть другим):

Команда Visual Basic на открытие файла

Путь обращения к Вашему файлу

```
Sub ЗагрузкаФайла()
```

```
ЗагрузкаФайла Макрос
```

```
Макрос записан .....
```

```
Workbooks.Open FileName:="D:\Мешков\Лаб раб N7.xls"
```

```
End Sub
```

Используя структуру команд **Visual Basic**, можно макрос записать вручную и назначить его исполнение через меню. Можно также функции исполнения макроса назначить через панель инструментов **Формы** – инструмент **Кнопка**.

Закройте редактор **Microsoft Visual Basic**.

7. Выполните из меню команды **Вид Панели инструментов Формы**.
8. На появившейся панели нажмите мышкой значок в виде **КНОПКИ**.
9. Переведите курсор мышки в поле рабочего **Листа**. Курсор примет вид перекрестия. Подведите перекрестие к одной из ячеек, нажмите левую клавишу мышки и, передвигая ее,

очертите поле в виде прямоугольника. Отпустите клавишу мышки. Появится диалоговое окно **Назначить макрос объекту** с именами макросов. Установите курсор на имя макроса **ЗагрузитьФайл** и нажмите кнопку **ОК**. В рабочем поле **Листа** появится изображение кнопки с надписью **Кнопка**. Щелкните мышкой по полю кнопки, затрите надпись и введите свою надпись **ЗАГРУЗКА**. Щелкните мышкой в стороне от кнопки.

Расположите кнопку как показано в примере оформления **Листа 2**

10. Переместите курсор экрана в поле кнопки. Он примет вид указательного пальца. Нажмите левую клавишу мышки. Запустится на исполнение макрос, который вызовет с диска Ваш файл с таблицами. Закройте вызванный файл.

Элемент управления «Счетчик»

11. Инициализируйте ячейку с адресом **A1**. В поле панели инструментов **Стандартная** щелкните по кнопке **Вставка функции**. В появившемся окне **Мастер функций** в поле **Категория** выберите опцию **Математические** и в поле **Функция** выберите вид **SIN**. Нажмите кнопку **ОК**. Появится поле ввода **Число**. Наберите аргумент функции в виде: **A4*3,14/180**, нажмите кнопку **ОК**. В ячейке **A1** должно появиться число **0**. Таким образом в ячейку **A1** введена формула в виде функции

$\square x = \sin(x)$, где $x = A4*3.14/180$.

В ячейку **A2** занесите текст *значение sin*. В ячейку **A5** занесите текст *значение угла в градусах*.

12. На панели инструментов **Формы** нажмите мышкой значок с именем **Счетчик**. Переведите курсор мышки в поле рабочего **Листа**. Курсор примет вид перекрестия. Подведите перекрестие к одной из ячеек, нажмите левую клавишу мышки и передвигая ее очертите поле в виде прямоугольника, охватывающего **8** ячеек. Отпустите клавишу мышки. Появятся две кнопки в виде . Переведите курсор в поле этих кнопок и нажмите правую клавишу мышки. Из набора команд выберите **Формат объекта**. Появится диалоговое окно **Формат элемента управления**. В поле **Помещать результат в ячейку** введите адрес **A4** и нажмите **ОК**. Щелкните мышкой в стороне от кнопок счетчика.

Расположите кнопки как показано в примере оформления **Листа 2**.

Понажимайте на кнопки счетчика и посмотрите, как меняются показания в ячейках **A1** и **A4**.

*Нажимая клавишу счетчика Вы устанавливаете значение числа в ячейку **A4**, которое используется для вычисления функции, находящейся в ячейке **A1**.*

Создание макроса для формирования таблицы значений вычисляемой функции

13. Расчертите таблицу как показано в **Приложении 1**. Внесите названия. В столбец **Значение угла** в первую ячейку введите значение **0** (ноль). Щелкните по меню **Правка** и затем задайте последовательно опции **Заполнить Прогрессия...** В появившемся диалоговом окне **Прогрессия** установите режимы в соответствии с рисунком. Нажмите кнопку **ОК**. Весь столбец автоматически заполнится данными.

14. В меню **Сервис** установите указатель на пункт **Макрос** и выберите **Начать запись**.

15. После появления диалогового окна в поле *Имя макроса* введите имя макроса **Таблица**. Нажмите кнопку **ОК**.

15. Выполните копирование значения угла **0** из таблицы в ячейку **A1**.

Установите курсор в ячейку **A1**, дайте команду меню **Правка** и затем **Копировать**.

Установите курсор в первую ячейку столбца **Значение функции** Вашей таблицы, дайте команду меню **Правка** и затем **Специальная вставка ...** В появившемся диалоговом окне **Специальная вставка** щелкнуть мышкой по ячейке **значения**. Нажмите кнопку **ОК**.

Данной командой Вы копируете только значение результата вычисления функции.

16. Повторите такую же процедуру со значением угла равным **10**.

Прервите выполнение макроса, для чего выберите в Меню **Сервис Макрос Остановить запись**.

16. В меню **Сервис** установите указатель на пункт **Макрос** и выберите **Макросы**. В появившемся диалоговом окне **Макрос** установите курсор на имя **Таблица** и нажмите кнопку **Изменить**.

17. Появится окно редактора программ **Microsoft Visual Basic**. В правой его части будет находиться текст программы созданного макроса на языке **Visual Basic** в виде:

```
Range("H5").Select
```

```
Selection.Copy
```

```
Range("A4").Select
```

```
ActiveSheet.Paste
```

```
Range("A1").Select
```

```
Application.CutCopyMode = False
```

```
Selection.Copy
```

```
Range("I5").Select
```

```
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
```

```
False, Transpose:=False
```

```
Range("H6").Select
```

```
Application.CutCopyMode = False
```

```
Selection.Copy
Range("A4").Select
ActiveSheet.Paste
Range("A1").Select
Application.CutCopyMode = False
Selection.Copy
Range("I6").Select
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
False, Transpose:=False
End Sub
```

Установите курсор перед **End Sub**. Нажмите **Enter**.

Скопируйте текст программы, начиная со строки **Range("H5").Select** и кончая последней строкой **False, Transpose:=False**. Установите курсор в пустую строку перед **End Sub** и выполните команду **Вставить**. Повторите команду **Вставить** 6 раз. В добавленном листинге откорректируйте адреса ячеек таблицы, откуда копируются данные и куда вставляются (в данном примере это ячейки **H** и **I**).

Закройте редактор **Microsoft Visual Basic**.

18. Создайте кнопку с именем **Таблица** и свяжите ее с макросом **Таблица**.

19. Щелкните по кнопке **Таблица**. Должен заполниться столбец **Значение функции** данными вычисления.

Сохраните **Рабочую книгу** под именем **Панель управления** на винчестере компьютера.

Подготовить отчет, разместить в ЭИОС и защитить у преподавателя.

Содержание отчета: титульный лист с указанием названия дисциплины, названия работы и фамилии студента, скриншоты выполнения этапов работы.

ЛАБОРАТОРНАЯ РАБОТА № 2. ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММЫ В СРЕДЕ VBA

1. Общие сведения

Цель: научиться работать с таблицами на нескольких листах, делать по ним

сводные отчеты, познакомиться с макросами, как средством автоматизации электронных вычислений, а также получить навыки электронного моделирования при помощи построения трендовой модели средствами Excel.

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office.

Условия допуска к выполнению:

Успешное выполнение и защита лабораторной работы № 1.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа: посвящена созданию макросов и работе с ними.

4. Методические указания и порядок выполнения работы

Построить сводную таблицу на основе результатов двух таблиц, расположенных на разных листах документа **Excel**. Оформить сводную таблицу для представления в отчете, и построить на ее основе сводную диаграмму по заданному условию. Произвести запись макроса, выполняющего автоматическое форматирование содержимого ячеек таблицы **Excel**, назначить его как функцию элемента управления. По данным из двух таблиц построить диаграмму и произвести на ней прогнозирование будущего значения, используя трендовую модель.

Порядок выполнения работы

1). Работа с несколькими листами:

- Создать лист с именем **«Исходная»** с таблицей. Таблицу выбрать согласно варианту в пункте **«Варианты заданий к работе»**. Номер своего варианта узнать у преподавателя.
- В произвольной пустой ячейке этого листа введите число равное номеру вашего варианта.
- Создать лист **«Сводная»** с таблицей, изображенной в пункте **«Сводная таблица» «Вариантов заданий к работе»**.
- Добавить к сводной таблице столбец с именем **«Кол-во1»**. Значения в нем должны быть заполнены числами, равными исходным значениям столбца **«Кол-во»**, сложением с ячейкой, содержащей номер варианта на листе **«Исходная»**;

2). Работа со сводными таблицами:

- На основе этой таблицы создайте отчёт сводной таблицы, используя меню **«Данные – Сводная таблица...»**.
- Следовать указаниям **«Мастера сводных таблиц»**, указав источник данных и место размещение результата;
- разместить поля по областям следующим образом:

Страница – **«Квартал»**

Строка – **«Инв. №»**

Столбец – **«Канал»**

Данные – сумма по полю **«Кол-во1»**

- Используя переключатели в отчёте сводной таблицы, найти:
 - а) Общее количество заказов по почте в 1 квартале;
 - б) Общее количество проданных изданий № 30782;
 - в) Количество изданий № 26059, проданных по международным каналам;
 - г) Общее количество изданий, проданных во втором квартале.
- Отобразить все значения по всем полям сводной таблицы;
- Построить отчёт сводной диаграммы для всех изданий, проданных по международным каналам;
- Отобразить все значения по всем полям сводной таблицы;
- Построить отчёт сводной диаграммы для издания **№ 41210**;
- Отобразить все значения по всем полям сводной таблицы;
- Приведите отчёт сводной таблицы к структурированному формату, используя

автоформат «Отчёт4».

3). Создание и выполнение макросов:

- Перейти на лист «Исходная» и выделить на нем произвольный диапазон числовых ячеек.
- Произвести запись макроса, выполняющего установку полужирного курсивного шрифта в любом выделенном диапазоне ячеек;
- Добавить на лист «Исходная» элемент управления «Кнопка» через меню панель инструментов «Формы» в меню «Вид»;
- Назначить макрос элементу управления «Кнопка» с надписью «Полужирный+Курсив»

4). Построение трендовых моделей:

- Создать новый лист с именем «Трендовая модель»;
- Создать на листе таблицу, содержащую две колонки: «Код» и «Кол-во1». Колонка **Код** заполняется, начиная с числа равного номеру вашего студенческого, шаг заполнения +1. Колонка «Кол-во1» – содержит ссылки на ячейки в одноименной колонке на листе «Сводная».
- Построить трендовую модель по данным получившейся таблицы;
- По уравнению получившейся трендовой модели в отдельной ячейке вычислить значение столбца «Кол-во1» для кода, превышающего последнее значение столбца *Код* на 10.

2. ЛАБОРАТОРНАЯ РАБОТА № 3. ПРИМЕНЕНИЕ ФУНКЦИЙ МАТЕМАТИКИ В ППП. ОСОБЕННОСТИ ЭКСПЛУАТАЦИИ МАТЕМАТИЧЕСКИХ ФУНКЦИЙ В MICROSOFTOFFICE

Цель: научиться применять функции математики в ППП.

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office.

Условия допуска к выполнению:

Успешное выполнение и защита лабораторной работы № 2.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа: посвящена функциям математики в ППП.

4. Методические указания и порядок выполнения работы

Согласуйте с преподавателем выбранный вами вариант задания (будет предоставлен преподавателем). Предполагается, что в рамках одной учебной группы варианты заданий не повторяются. Разработайте и заполните таблицу и запрограммируйте в ней необходимые вычисления. При необходимости воспользуйтесь функциями. Убедитесь в правильности вычислений. Оформите таблицу, задайте шрифты, границы и т. п. В качестве примера будет рассматриваться таблица, предназначенная для расчета налогов и определения суммы заработной платы.

Используя копию созданной таблицы, создайте и изучите макросы, позволяющие программировать вычисления в таблице.

Порядок выполнения работы

1. Создайте новую рабочую книгу Excel. Сделайте ее настройку:
2. Проверьте работоспособность ранее созданного макроса, для чего воспользуйтесь командой **Сервис, Макрос, Макросы**. Убедитесь в правильности вычислений.
3. Командой **Сервис, Макрос, Редактор Visual Basic** запустите редактор Visual Basic. Сделайте настройку интегрированной среды разработки VBA:

- выполните команду **Tools, Options** и в диалоговом окне выберите вкладку *Editor*, установив следующие параметры: Code Settings: Auto Syntax Check, Require Variable Declaration, Auto List members, Auto Quick Info, Auto Data Tips - включено.
Window Settings: Drag-and-Drop Text Editing, Default to Full Module View, Procedure Separator - включено.
Auto Indent - включено.
Tab - 4.

Задайте режим обязательного объявления переменных, для чего выше заголовка вставьте строку `Option explicit`.

5. Напишите коды объявления внутренних переменных своей программы и задайтесь их типом данных. Прокомментируйте их в тексте программы.
6. Напишите коды программы считывания исходных данных из таблицы Excel во внутренние переменные программы VBA. Прокомментируйте их в тексте программы.
7. Напишите коды вычислений результирующих значений так, чтобы их результаты оказались во внутренних переменных программы VBA. Прокомментируйте их в тексте программы.
8. Напишите коды программы записи рассчитанных значений в соответствующие ячейки таблицы Excel. Прокомментируйте их в тексте программы.
9. Удалите в таблице Excel результаты вычислений и проверьте работоспособность программы в режиме **Debug, Run To Cursor**. Проконтролируйте значения в окне локальных переменных Locals. Введите точки останова командой **Debug, Toggle Breakpoint** и убедитесь в правильности работы программы.
10. Проверьте правильность комментариев с учетом изменений в тексте программы, дополните и, при необходимости, скорректируйте их.

5. Требования к отчету и защите

Подготовить отчет, разместить в ЭИОС и защитить у преподавателя.

Содержание отчета: титульный лист с указанием названия дисциплины, названия работы и фамилии студента, скриншоты выполнения этапов работы.

3. ЛАБОРАТОРНАЯ РАБОТА № 4. РЕШЕНИЕ ЗАДАЧ В СРЕДЕ MICROSOFT OFFICE С ИСПОЛЬЗОВАНИЕМ EXCEL

Цель: решение задач в среде ППП с использованием EXCEL.

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office.

Условия допуска к выполнению:

Успешное выполнение и защита лабораторной работы № 3.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа: решению задач в среде Microsoft с использованием Excel.

4. Методические указания и порядок выполнения работы

Порядок выполнения работы

1. Внедрите на рабочий лист кнопку, позволяющую при нажатии скопировать таблицу и ее исходные данные на новый лист. На этот лист внедрите кнопку, позволяющую вызвать процедуру вычисления данных в таблице. Далее в качестве рабочего будет использоваться этот лист.

2. Создайте на рабочем листе флажок. Напишите процедуру обработки факта нажатия флажка.

3. Создайте на рабочем листе выключатель. Напишите процедуру обработки факта нажатия выключателя.

4. Внедрите три переключателя на рабочий лист. Убедитесь, что созданная вами группа переключателей обеспечивает выбор только одного из возможных режимов. Напишите процедуру обработки состояния переключателей.

5. Внедрите на рабочий лист список и поле со списком. Напишите программу инициализации списков с учетом данных, имеющихся в таблице.

6. Внедрите на рабочий лист кнопку, обеспечивающую обработку информации, выбранной в списках, в результате ее нажатия.

7. Внедрите на рабочий лист надпись. Напишите процедуру изменения текста надписи, которая должна учитывать результат выбора в группе переключателей.

8. Внедрите на рабочий лист поле. Напишите процедуру считывания текущего содержания поля и отображения введенного текста в виде надписи.

9. Убедитесь в работоспособности созданных программ.

10. Проверьте правильность комментариев с учетом изменений в тексте программы, дополните и, при необходимости, скорректируйте их.

4. ЛАБОРАТОРНАЯ РАБОТА № 5. ИСПОЛЬЗОВАНИЕ ЭЛЕМЕНТОВ ПРОГРАММИРОВАНИЯ ДЛЯ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ

1. Общие сведения

Цель: использование элементов программирования для решения математически задач.

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office, пакет SciLab.

Условия допуска к выполнению:

Успешное выполнение и защита лабораторной работы № 4.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа: посвящена математическим пакетам прикладных программ

4. Методические указания и порядок выполнения работы

Составить модель задачи и решить в Scilab.

1. Найдите длину окружности, имеющей радиус 5 см.

2. Найдите сумму двух произвольных чисел, затем вычтите 1 из суммы, используя переменную ans.

3. Вычислите значения выражения $5^3 + \frac{4}{3} - 6 * 8$.

4. Определите, какое из чисел больше, $\frac{5}{139}$ или $\frac{9}{901}$.

5. Вычислите значение выражения $\frac{153}{5 + 64 * (\frac{1}{33} - 7)^3 + \frac{7}{8}}$.

6. Найдите угол между касательной в точке $x=4$ к кривой

$$y = x^2 + x^5 \quad \text{и осью } O_x.$$

7. Вычислите производную функции $f(x) = \frac{x-3}{x}$ в точках $x=3$ и $x=8$.

8. Найдите работу по перемещению тела на расстояние в 30 см под действием силы;

$$F(x) = \frac{5+x^2}{x-1}.$$

9. Вычислите интеграл $\int_2^5 (-\sin \pi x) dx$.

10. Вычислите площадь фигуры, ограниченной линиями

$$y = \left(\sin \frac{x}{4} + \cos \frac{x}{4} \right)^2 \quad \text{и} \quad y = \left(\sin \frac{x}{4} - \cos \frac{x}{4} \right)^2.$$

5. Требования к отчету и защите

Подготовить отчет, разместить в ЭИОС и защитить у преподавателя.

Содержание отчета: титульный лист с указанием названия дисциплины, названия работы и фамилии студента, скриншоты выполнения этапов работы.

7. ЛАБОРАТОРНАЯ РАБОТА № 6. ИСПОЛЬЗОВАНИЕ ДОПОЛНИТЕЛЬНЫХ ФУНКЦИЙ ПРОГРАММИРОВАНИЯ ДЛЯ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ

1. Общие сведения

Цель: Знакомство с СКМ Scilab Освоение интерфейса, просмотр демосов, работа в режиме калькулятора.

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office, пакет SciLAB.

Условия допуска к выполнению:

Успешное выполнение и защита лабораторной работе №5.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа: Знакомство с СКМ Scilab Освоение интерфейса, просмотр демосов, работа в режиме калькулятора.

4. Методические указания и порядок выполнения работы

Задание 1. Изучить интерфейс Scilab.

Задание 2. Ознакомиться с демонстрационными примерами Scilab.

Задание 3. Выполнить в режиме калькулятора следующие действия:

- Ввод исходных операндов.
- Выполнить над операндами 1 и 2 операцию 1.
- Выполнить над результатом и операндом 1 операцию 2.
- Выполнить над результатом и операндом 2 операцию 3.
- Возвести операнд 1 почленно в степень 3.

ГИП с независимыми переключателями

Создать ГИП с двумя независимыми переключателями, которым соответствуют обработчики, рисующие в графическом окне функции $f_1(x)$ и $f_2(x)$. Команда рисования – нажатие кнопки. В ГИП рисуются все выбранные графики.

5. Требования к отчету и защите

Подготовить отчет, разместить в ЭИОС и защитить у преподавателя.

Содержание отчета: титульный лист с указанием названия дисциплины, названия работы и фамилии студента, скриншоты выполнения этапов работы.

8. ЛАБОРАТОРНАЯ РАБОТА № 7. ИСПОЛЬЗОВАНИЕ СРЕЗ БАЗ ДАННЫХ В ППП

1. Общие сведения

Цель: направлена на изучение работы с базами данных через пакеты прикладных программ и СУБД.

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office, MySql

Условия допуска к выполнению:

Успешное выполнение и защита лабораторной работы №6.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа: направлена на изучение работы с базами данных через пакеты прикладных программ и СУБД.

4. Методические указания и порядок выполнения работы

ПОДКЛЮЧЕНИЕ К СЕРВЕРУ БАЗ ДАННЫХ

Подключение к серверу выполняется при помощи команды:

```
mysql -u user_name -p
```

Где **user_name** – Ваше имя пользователя, заданное при установке сервера, обычно используется имя root.

В следующей строке введите пароль, заданный при установке (отображается звёздочками) и нажмите Enter.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ MYSQL

Клиентская программа MySQL представляет собой утилиту командной строки. Эта программа подключается к серверу по сети. Команды, выполняемые сервером, обычно связаны с чтением и записью данных на жестком диске.

Клиентские программы могут работать не только в режиме командной строки. Есть и графические клиенты, например MySQL GUI, PhpMyAdmin и др. Но они – тема отдельного курса.

MySQL взаимодействует с базой данных на языке, называемом SQL (Structured Query Language — язык структурированных запросов).

SQL предназначен для манипуляции данными, которые хранятся в Системах управления реляционными базами данных (RDBMS). SQL имеет команды, с помощью которых данные можно извлекать, сортировать, обновлять, удалять и добавлять. Стандарты языка SQL определяет ANSI (American National Standards Institute). В настоящее время действует стандарт, принятый в 2003 году (SQL-3).

SQL можно использовать с такими RDBMS как MySQL, mSQL, PostgreSQL, Oracle, Microsoft SQL Server, Access, Sybase, Ingres. Эти системы RDBMS поддерживают все важные и общепринятые операторы SQL, однако каждая из них имеет множество своих собственных патентованных операторов и расширений.

Задание:

1. Изучить основные характеристики СУБД MySQL.
2. Научиться создавать базы данных, таблицы и записи в них.
3. Ознакомиться с типами данных, используемыми в MySQL.
4. Научиться работе с таблицами; внесению, извлечению, поиску и удалению данных.

5. Научиться применять логические операторы в SQL – запросах.
6. Изучить команды обработки данных.
7. Научиться применять в SQL - запросах математические функции.
8. Изучить способы работы с датой и временем.
9. Изучить команды обработки строковых данных.
10. Реализовать при помощи СУБД MySQL, задание лабораторной работы № 2.

5. Требования к отчету и защите

Подготовить отчет, разместить в ЭИОС и защитить у преподавателя.

Содержание отчета: титульный лист с указанием названия дисциплины, названия работы и фамилии студента, скриншоты выполнения этапов работы.

9. ЛАБОРАТОРНАЯ РАБОТА № 8. ИСПОЛЬЗОВАНИЕ ЯЗЫКА SQL В БД

1. Общие сведения

Цель: научиться использовать язык sql для решения задач в базах данных ППП.

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office, MySql

Условия допуска к выполнению:

Успешное выполнение и защита лабораторной работе № 7.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа: направлена на изучение использования языка sql для решения задач в базах данных ППП.

4. Методические указания и порядок выполнения работы

Команда на языке mysql обычно состоит из выражения, за которым следует точка с запятой. Когда пользователь вводит команду, mysql отправляет ее серверу для выполнения и выводит на экран Сначала результаты, а затем – новую строку **mysql>**, что означает готовность к выполнению новых команд.

Строка mysql> называется приглашением командной строки.

Утилита mysql выводит результаты работы запроса, если таковые есть, в виде таблицы (строк и столбцов). В первой строке этой таблицы содержатся заголовки столбцов, а в следующих строках – собственно результаты. Обычно заголовками столбцов становятся имена, полученные из таблиц базы данных.

Утилита mysql также сообщает количество возвращаемых строк и время выполнения запроса, что позволяет в некоторой степени составить представление о производительности сервера.

Для mysql признаком завершения команды является точка с запятой, а не конец строки. Команды собираются, но не исполняются до тех пор, пока программа не обнаружит точку с запятой. Например, команда, требующая распечатать список баз данных сервера, может быть дана двумя строками так:

```
mysql> SHOW  
-> DATABASES;
```

Обратите внимание на то, как изменилась метка командной строки (с mysql> на ->) после ввода первой строки этого запроса. Таким образом, mysql показывает, что завершено выражения она пока что не получила и ожидает его полного ввода. Эта метка полезна, так как предоставляет весьма ценную информацию о состоянии программы. С ее помощью всегда можно узнать, чего ждет mysql.

Работа с таблицами базы данных

Создадим процедуру, которая подсчитывает число записей в таблице Post.

```
mysql> USE Postavki //  
Database changed  
mysql> CREATE PROCEDURE num_Post (OUT total INT)  
-> BEGIN  
-> SELECT COUNT (*) INTO total FROM Post ;
```

```
-> END
```

```
-> //
```

Вызов этой процедуры:

```
mysql> USE Postavki //
```

```
mysql> CALL num_Post (@a) //
```

```
mysql> SELECT @a //
```

Будет выведено:

```
@a
```

```
7
```

В качестве параметра процедуры передается пользовательская переменная @a.

Создадим хранимую процедуру proc_PIM, которая будет возвращать по первичному ключу PN имя поставщика PIM из таблицы Post. Для этого потребуется определить параметр pn_post как IN, а pim_post как OUT.

```
mysql> CREATE PROCEDURE proc_pim (IN pn_post INT, OUT pim_post CHAR)
```

```
-> BEGIN
```

```
-> SELECT PIM INTO pim_post FROM Post
```

```
-> WHERE PN = pn_post;
```

```
-> END
```

```
-> //
```

```
mysql> SET @pn_post := 2 //
```

```
mysql> CALL proc_pim (@pn_post, @pim_post) //
```

```
mysql> SELECT @pn_post, @pim_post //
```

Будет выведено:

```
@pn_post, @pim_post
```

2 Петров

Создадим хранимую процедуру pim_gor, которая по имени поставщика (PIM) возвращает город (GOR), в котором размещен этот поставщик.

```
mysql> CREATE PROCEDURE pim_gor (IN pim_post CHAR, OUT gor_post
```

```
-> CHAR)
```

```
-> BEGIN
```

```
-> DECLARE pn_post INT;
```

```
-> SELECT PN INTO pn_post FROM Post
```

```
-> WHERE PIM = pim_post LIMIT 1;
```

```
-> CALL proc_pim (pn_post, );
```

```
-> END
-> //
mysql> CALL pim_gor ('Иванов', @gorod) //
mysql> SELECT @gorod //
@gorod
СП6
```

В приведенном примере по имени поставщика с помощью запроса находится значение первичного ключа. Полученное значение помещается во временную переменную `pn_post`, которая передается в качестве первого аргумента при вызове процедуры `proc_pim`, возвращающей имя поставщика и помещающей результат работы в переменную `gor_post`, являющуюся выходным параметром процедуры `pim_gor`.

В процедуре `pim_gor` потребовалась временная переменная `pim_gor`. Для использования любой переменной в процедуре требуется ее объявление при помощи оператора `DECLARE`, который имеет следующий синтаксис:

```
DECLARE имя_переменной [, ...] тип [DEFAULT значение ]
```

Одним оператором `DECLARE` можно объявить несколько переменных, например:

```
DECLARE id, num INT DEFAULT 0;
```

Оператор `DECLARE` может быть использован только внутри блока `BEGIN ... END`, область видимости объявленной переменной также ограничена этим блоком. Это означает, что в разных блоках `BEGIN ... END` могут быть объявлены переменные с одинаковым именем, и действовать они будут только в рамках данного блока, не пересекаясь с переменными других блоков.

5. Требования к отчету и защите

Подготовить отчет, разместить в ЭИОС и защитить у преподавателя.

Содержание отчета: титульный лист с указанием названия дисциплины, названия работы и фамилии студента, скриншоты выполнения этапов работы.

10. ЛАБОРАТОРНАЯ РАБОТА № 9. СОЗДАНИЕ ИНТЕРФЕЙСОВ И ДОПОЛНИТЕЛЬНОГО ФУНКЦИОНАЛА ПРИ РАБОТЕ С БД В ППП

1. ОБЩИЕ СВЕДЕНИЯ

Материалы, оборудование, программное обеспечение:

Для выполнения данной лабораторной работы потребуется персональный компьютер с установленным на нем офисным пакетом Microsoft Office, MySQL

Условия допуска к выполнению:

Успешное выполнение и защита лабораторной №8.

Критерии положительной оценки: предоставить преподавателю отчет (скриншоты этапов выполнения работы) в электронном виде.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 2 ч.

Время самостоятельной подготовки: 1 ч.

2. Теоретическое введение

Контрольные вопросы для самопроверки: не предусмотрены.

3. Задание к лабораторной работе

Практическая работа: научиться создавать интерфейсы и дополнительный функционал при работе с БД.

4. Методические указания и порядок выполнения работы

Хранимые процедуры-функции

Процедура-функция в отличие от процедуры-подпрограммы может вызываться непосредственно без использования оператора CALL. Она возвращает только одно значение.

Создадим простейшую хранимую функцию say_hello(), которая будет принимать единственный входной параметр с именем **name** и возвращать фразу “Hello, name!”, где вместо подстроки name будет подставлено значение параметра **name**. Процедура имеет вид:

```
mysql> CREATE FUNCTION say_hello (name CHAR(20)) RETURNS CHAR(50)
-> BEGIN
-> RETURN CONCAT ('Hello, ',name,'!');
-> END
```

```
-> //  
mysql> SELECT say_hello ('world'), say_hello ('softtime') //
```

Выведется:

```
say_hello('world') say_hello ('softtime')
```

```
Hello, world! Hello, softtime!
```

В описании функции после объявления параметров функции следует оператор RETURNS, который задает тип возвращаемого значения. Вернуть значение из функции можно при помощи оператора RETURN, который может быть написан в любой точке функции. Исполнение оператора RETURN означает, что функция должна немедленно завершить выполнение и вернуть значение, переданное в качестве аргумента оператора RETURN.

Следует отметить, что при объявлении параметров функции использование ключевых слов IN, INOUT и OUT недопустимо. Все параметры, передаваемые функции, являются входными.

Функция обязательно должна содержать оператор RETURNS, устанавливающий тип возвращаемого значения, и хотя бы один оператор RETURN в теле функции, который возвращает это значение.

Рассмотрим еще один пример. Пусть требуется создать две процедуры-функции, одна из которых (count_cena ()) вычислит суммарную цену по всем поставленным деталям, а вторая (count_kol ()) вычислит суммарное количество всех поставленных деталей.

Для получения желаемого результата функция count_cena () использует два запроса. Первый запрос выдает номера поставляемых деталей, которые сохраняет в таблице new (запрос на создание таблицы). Можно было получить желаемый результат иначе. Предложите свой вариант.

```
mysql> CREATE FUNCTION count_cena ( ) RETURNS DECIMAL  
-> BEGIN  
-> DECLARE sum_cena DECIMAL;  
-> SELECT PD.DN INTO new  
-> FROM det INNER JOIN PD ON det.DN = PD.DN  
-> WHERE (((det.DN) In (select distinct DN from PD)))  
-> GROUP BY PD.DN;  
-> SELECT Sum([CENA]) INTO sum_cena  
-> FROM new INNER JOIN det ON new.DN = det.DN;  
-> RETURN sum_cena;  
-> END
```

```

-> //
mysql> CREATE FUNCTION count_kol ( ) RETURNS INT
-> BEGIN
-> DECLARE sum_kol INT;
-> SELECT Sum([KOL])
-> FROM PD;
-> RETURN sum_kol ;
-> END
-> //

```

Теперь вычислим суммарную стоимость всех поставленных деталей:

```

mysql> SELECT count_cena ( ) AS sum_cena, count_kol ( ) AS sum_kol,
-> count_cena ( ) * count_kol ( ) AS total_cena //

```

Ответ будет примерно такой (все зависит от данных соответствующих таблиц):

```

sum_cena sum_kol total_cena
530 657 348210

```

Операторы управления потоком данных

Хранимые процедуры – это не просто удобные контейнеры для группы запросов, они позволяют реализовать достаточно сложную логику, используя операторы ветвления и циклы.

Оператор IF ... THEN ... ELSE

Этот оператор имеет следующий синтаксис:

```

IF условие_1 THEN операторы_1
[ELSEIF условие_2 THEN операторы_2] ...
[ELSE операторы_3]
END IF

```

Отметим, что кроме оператора IF существует в MySQL функция IF ().

Оператор CASE

Этот оператор позволяет осуществить множественный выбор и имеет две формы. Первая форма имеет следующий синтаксис:

```

CASE значение_case
WHEN значение_1 THEN операторы_1
[WHEN значение_2 THEN операторы_2] ...
[ELSE операторы_3]
END CASE

```

Оператор CASE сравнивает значение выражения **значение_case** со значением **значение_1**, **значение_2** и так далее. Как только соответствие будет найдено, то выполнится соответствующий блок операторов **операторы_1**, **операторы_2** и так далее. Если ни одного соответствия не найдено, выполняются **операторы_3**.

Вторая форма оператора CASE позволяет осуществлять сравнение непосредственно в конструкции WHEN. При этом как только будет найдено первое истинное значение, выполняется соответствующий блок операторов и осуществляется выход из оператора CASE:

```
.....  
CASE  
WHEN cc = 0 THEN  
SESELECT ....  
WHEN cc = 1 THEN  
SESELECT ....  
WHEN cc = 3 THEN  
SESELECT ....  
.....  
ELSE  
SESELECT ....  
END CASE;
```

.....

Оператор **WHILE**

Это – оператор цикла с предусловием. Синтаксис:

```
[метка:] WHILE условие DO  
блок операторов  
END WHILE [метка]
```

Блок операторов выполняется, пока условие истинно.

Пример. Создадим хранимую процедуру, которая выводит текущую дату **num** раз, где **num** - параметр, задаваемый пользователем.

```
mysql> CREATE PROCEDURE nown (IN num INT)  
-> BEGIN  
-> DECLARE i DEFAULT 0;  
-> IF (num > 0) THEN  
-> wet : WHILE i < num DO  
-> SELECT NOW ( );
```

```
-> SET i = i + 1;
-> END WHILE wet;
-> ELSE
-> SELECT 'Ошибочное значение параметра';
-> END IF;
-> END
-> //
```

Вызов процедуры:

```
mysql> CALL NOWN (2) //
```

Метка в цикле предназначена не только для того, чтобы облегчить чтение кода при очень длинных циклах, она позволяет осуществлять досрочный выход из цикла с помощью оператора LEAVE, который имеет следующий синтаксис:

LEAVE метка

Оператор LEAVE прекращает выполнение блока, помеченного меткой, например:

```
.....
-> IF (num > 0) THEN
-> wet: WHILE i < num DO
-> IF i > 5 THEN LEAVE wet;
-> END IF;
.....
```

Еще одним оператором, выполняющим досрочное прекращение цикла, является оператор ITERATE. В отличие от оператора LEAVE оператор ITERATE не прекращает выполнение цикла, он лишь выполняет досрочное прекращение текущей итерации.

Оператор **REPEAT**

Этот оператор также реализует цикл, но с постусловием. Синтаксис:

```
[метка:] REPEAT
блок операторов
UNTIL условие
END REPEAT [метка]
```

Блок операторов выполняется, пока условие ложно. Проверка условия осуществляется после выполнения блока операторов.

Оператор **LOOP**

Этот оператор предназначен для реализации циклов и имеет следующий синтаксис:

```
[Метка:] LOOP
```

блок операторов

END LOOP [метка]

Этот цикл, в отличие от операторов WHILE и REPEAT, не имеет условий выхода. Поэтому выход из цикла нужно предусмотреть, например, с помощью оператора LEAVE.

Оператор **GOTO**

Этот оператор осуществляет безусловный переход по метке.

11. ЗАКЛЮЧЕНИЕ

Учебно-методическое пособие позволяет студентам освоить базовые знания и приобрести навыки использования пакетов прикладных программ общего и специализированного проектирования. В работе с офисными пакетами и использовании макросов и языка программирования VBA, работе с СУБД MySQL, работе с математическими пакетами и пакетами для промышленного проектирования.

Для углубления своих знаний в области использования и особенностях применения пакетов прикладных программ студенты могут использовать учебные ресурсы, часть из которых представлена в списке литературы.

12. ЛИТЕРАТУРА

1. Федотова, Е. Л., Информационные технологии и системы: учебное пособие / Е. Л. Федотова. - Москва : ИД "Форум". - Москва : ИНФРА-М, 2013.
2. Фуфаев, Э. В., Пакеты прикладных программ : учебное пособие / Э. В. Фуфаев, Л. И. Фуфаева. - 7-е изд. – Москва : Academia, 2013. - 352 с.
- 3., Статистические и динамические экспертные системы: учебное пособие / Под ред. Э. В. Попова, И. Б. Фоминых, Е. Б. Кисель. - Москва: Финансы и статистика, 1996
4. Головицына, М. Интеллектуальные САПР для разработки современных конструкций и технологических процессов / М. Головицына - Москва: Национальный Открытый Университет «ИНТУИТ», 2016.

Локальный электронный методический материал

А. А. Бабаева

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ»

Редактор Г. А. Смирнова

Уч.-изд. л. 2,1. Печ. л. 2,1