

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Пермский государственный аграрно-технологический университет
имени академика Д. Н. Прянишникова»

Факультет прикладной информатики
Кафедра информационных технологий и программирования

КОЗЛОВ А.Н.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО
ПРАКТИКЕ
ПО ПОЛУЧЕНИЮ ПЕРВИЧНЫХ
ПРОФЕССИОНАЛЬНЫХ УМЕНИЙ И НАВЫКОВ,
В ТОМ ЧИСЛЕ УМЕНИЙ И НАВЫКОВ НИР
(учебная практика по информатике)

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

Пермь 2018 г.

УДК 330:004(075.8)
ББК 65.32ф.я73

Козлов, А.Н. Методические рекомендации по практике по получению первичных профессиональных умений и навыков, в том числе умений и навыков НИР 09.03.03 Прикладная информатика./ ФГБОУ ВО Пермский ГАТУ. – Пермь.: Изд: ФГБОУ ВО Пермская ГСХА, - 2018. – 39 с.

Рецензент:

доцент кафедры информатики факультета прикладной информатики ФГБОУ ВО Пермская ГСХА – канд. тех. наук, доцент Зорин А.А.

Методические рекомендации по учебной практике разработаны в соответствии с ФГОС по направлению 09.03.03 «Прикладная информатика» и предназначены для студентов и преподавателей очного и заочного обучения. Основная цель – помочь студентам в подготовке, прохождении и оформлении результатов учебной практики по информатике.

Рекомендовано к изданию и использованию в учебном процессе по решению методической комиссии факультета прикладной информатики ФГБОУ ВО Пермская ГСХА (протокол № 7 от 13.03.2016 г.)

ОГЛАВЛЕНИЕ

Введение	4
1. Общие положения об учебной практике	5
2. Тематика учебной практики	5
3. Требования к выполнению задания по учебной практике	5
4. Подготовка отчета по учебной практике	6
5. Защита отчета о практике.	6
6. Методические указания	6
6.1. Основные понятия и средства машинной графики в Паскале	6
6.2. Макросы и VBA	9
7. Варианты заданий	15
7.1 Темы рефератов	15
7.2. Варианты практического задания	17
Заключение	19
Список используемых источников	20
Приложение 1	21
Приложение 2	22

Введение

Учебная практика является необходимым и важным компонентом образовательной программы. В соответствии с ФГОС ВО по направлению 09.03.03 учебная практика посвящена получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности. Способ проведения учебной практики – стационарная (на кафедре информационных систем Академии).

Учебная практика позволяет сформировать у выпускника следующие компетенции:

способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-4);

способность программировать приложения и создавать программные прототипы решения прикладных задач (ПК-8).

Учебная практика разделена на две части – учебная практика по информатике (2 семестр) и учебная практика по программированию (4 семестр) [2].

Данные методические рекомендации разработаны в соответствии с ФГОС ВО по направлению 09.03.03 « Прикладная информатика», на основе Положения о порядке проведения практик в Академии [1], рабочей программы учебной практики [2], а также опыта проведения практики кафедрами факультета. Основная цель – помочь студентам в подготовке, прохождении и оформлении результатов учебной практики. Пособие предназначено для студентов очного и заочного обучения, а также преподавателей.

1. Общие положения об учебной практике

Учебная практика по информатике (практика по получению первичных профессиональных умений и навыков) имеет целью совершенствование практических навыков студентов по применению современных прикладных программных средств общего назначения в профессиональной деятельности на основе знаний, полученных в ходе изучения курса «Информатика и программирование».

Задачей практики является углубление теоретических знаний и совершенствование навыков:

- программирования на персональном компьютере с применением процедурных языков программирования;
- обработки и визуализации данных в офисных приложениях MS Office.
- программирования в VBA.
- оформления документации в MS Word.

По окончании практики студент предоставляет руководителю практики отчет о проделанной работе и полученных результатах.

Форма итогового контроля по практике – зачет. На зачете студент демонстрирует работу созданных программ и макросов.

При успешном прохождении практики студенту выдается документ о присвоении рабочей специальности.

2. Тематика учебной практики.

Учебная практика выполняется по теме: «Программирование, расчет значений и построение графика тригонометрической функции на заданном интервале» в соответствии с вариантом задания.

3. Требования к выполнению задания по учебной практике.

В соответствии с тематикой учебной практики, требуется:

1. Подготовить реферат на заданную тему.
2. Составить программу на языке Паскаль, которая реализует:
 - Расчет значений тригонометрической функции на заданном интервале;
 - Вывод полученных значений во внешний текстовый файл в виде таблицы;
 - Построение на экране дисплея графика функции по найденным значениям.
3. Выполнить эту же задачу в MS Excel.
4. Отработать создание макросов с помощью VBA (задания №№1-8 в методическом материале «Макросы и VBA»).
5. Создать макрос в MS Excel, позволяющий рассчитать значение заданной согласно варианта тригонометрической функции для заданного (любого)

значения аргумента (см. задание 6 в методическом материале «Макросы и VBA»). Создать кнопку на панели инструментов для вызова данного макроса.

6. Оформить отчет в MS Word.

4. Подготовка отчета по учебной практике

Отчет по учебной практике выполняется в виде реферата (в обложке) и должен содержать:

1. Титульный лист (см. Приложение 1).
2. Содержание (см. Приложение 2).
3. Теоретическую часть (текст реферата на 8-15 листов).
4. Список использованной литературы.
5. Практическая часть (исходные данные и краткое описание работы).
6. Текст программы на языке Паскаль.
7. Распечатку (скриншот) полученных значений функции и скриншот графика функции.
8. Распечатку значений и графика функции, выполненные в MS Excel;
9. Экранную форму (скриншот) текстов созданных макросов (задания № № 1-8).
10. Приложение: CD с файлом программы на Паскаль, файлом в MS Excel и файлами макросов, в том числе макроса для расчета тригонометрической функции.

Отчет распечатывается на бумаге формата А4 с одной стороны. Поля справа 3 см, слева 1,5 см, сверху и снизу по 2 см. Шрифт Times New Roman размер 14.

5. Защита отчета о практике.

В установленное преподавателем время студент прибывает на кафедру, имея с собой оформленный отчет и программы на **ФЛЭШКЕ**.

После проверки преподавателем реферата студент отчитывается по практической части:

- показывает график функции в Паскале,
- показывает график функции в MS Excel,
- демонстрирует работу макросов,
- выполняет дополнительное задание по модернизации своей программы в Паскале.

6. Методические указания.

6.1. Основные понятия и средства машинной графики в Паскале

Графический модуль Graph

В стандарте Паскаля не предусмотрен графический вывод.

Начиная с 4-ой версии Паскаля появилась библиотека, (модуль) GraphABC.pri, которая содержит большой набор процедур и функция для работы с графикой.

Для того, чтобы в программе можно было обращаться к процедурам и функциям модуля GraphABC.pri, необходимо в начале программы написать строку:

```
Uses GraphABC;
```

Для инициализации (включения) графического режима используется процедура:

```
InitGraph(Driver, Mode : integer; Path : String),
```

где целые переменные Driver и Mode определяют тип графического драйвера и режим его работы. Строковая переменная Path содержит путь к файлу графического драйвера.

Возможно автоматическое определение драйвера и установка режима:

```
Driver := Detect;  
InitGraph(Driver, Mode, "");
```

где Detect – имя константы модуля Graph, соответствующее автоматическому выбору драйвера.

Возвращение в текстовый режим работы дисплея осуществляет процедура:

```
CloseGraph;
```

Графические координаты

Графические оси координат расположены на экране следующим образом:



То есть, горизонтальная ось X направлена слева направо, а вертикальная ось Y – сверху вниз. В соответствии с режимом VGH_i, по горизонтали на экране помещается 640 точек, а по вертикали – 480 точек.

Графические процедуры

Line(X1,Y1,X2,Y2 : integer) – рисует линию от точки (X1,Y1) до точки (X2,Y2);

PutPixel(X,Y : integer; Color : Word) – рисует точку с координатами (X,Y) цветом Color;

OutTextXY(X,Y : integer; Text : string) – отображает Text, начиная с координат (X,Y).

Построение графика функций

Требуется составить программу построения на экране графика функции

$$Y = F(X);$$

Решение задачи будем проводить в следующем порядке:

1. Определим границы значений аргумента, в пределах которых будем строить график функции. Обозначим их: X_{\min} – минимальная граница, X_{\max} – максимальная граница аргумента функции;
2. Для заданной области значений аргумента оценим минимальное и максимальное значения функции: Y_{\min} и Y_{\max} ;
3. Зададим границы графического окна, в пределах которого будем строить график: $[X_{g\min}, X_{g\max}]$, $[Y_{g\min}, Y_{g\max}]$. Поскольку в графических координатах вертикальная ось направлена вниз, то $Y_{g\min} > Y_{g\max}$;
4. Значение функции Y , вычисленное в точке X будем отображать на экране через графические координаты (X_g, Y_g) связанные со значениями (X, Y) по формулам:

$$X_g = X_{g\min} + [(X_{g\max} - X_{g\min}) / (X_{\max} - X_{\min}) * (X - X_{\min})];$$

$$Y_g = Y_{g\min} + [(Y_{g\max} - Y_{g\min}) / (Y_{\max} - Y_{\min}) * (Y - Y_{\min})],$$

Где квадратные скобки означают округление до целого числа (функция Round).

Пример (*в отчете отразить пример в соответствии с вариантом*).

Требуется построить график функции $Y = 2 * \sin(X) + \cos(2 * X)$ на интервале $[0, 2 * \pi]$ с шагом 10 градусов ($2\pi/36$ радиан).

Решение. Для построения графика функции на экране, необходимо связать найденные в программе значения (X, Y) с графическими координатами (X_g, Y_g) :

1. Определим границы значений аргумента, в пределах которых будем строить график: $X_{\min} = 0$, $X_{\max} = 2 * \pi$;
2. Для заданной области значений аргумента оценим минимальное и максимальное значения функции $Y = 2 * \sin(X) + \cos(2 * X)$: на интервале $[0, 2 * \pi]$, значения функции $2 * \sin(X)$ изменяются в пределах от -2 до 2, а значения функции $\cos(2 * X)$ – в пределах от -1 до 1. Тогда значения функции $Y = 2 * \sin(X) + \cos(2 * X)$ на интервале $[0, 2 * \pi]$ будут в пределах от -3 до 3, т.е. $Y_{\min} = -3$ и $Y_{\max} = 3$;
3. Зададим границы графического окна, в пределах которого будем строить график: график будем строить в окне экрана от 10 до 210 точек по оси X и от 40 до 240 точек по оси Y , т.е. $X_{g\min} = 10$, $X_{g\max} = 210$, $Y_{g\min} = 240$, $Y_{g\max} = 40$;
4. Значение функции Y , вычисленное в точке X будем отображать на экране через графические координаты (X_g, Y_g) , связанные со значениями (X, Y) по формулам:

$$Xg = 10 + [(210-10)/(2*\pi-0)*(X-0)] = 10 + [100*X/\pi];$$

$$Yg = 240 + [(40-240)/(3-(-3))*(Y-(-3))] = 140+[-100*Y/3].$$

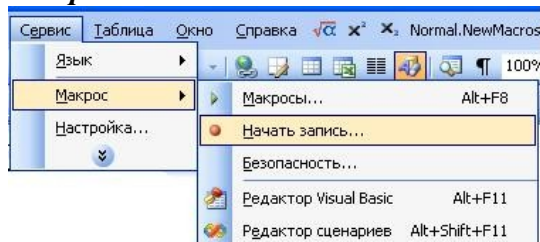
Вместе с графиком функций выводим на экран оси координат. Ось X имеет координату $Yg = (Ygmin+Ygmax)/2 = 140$, ось Y имеет координату $Xg = Xgmin = 10$;

6.2. Макросы и VBA.

Задание №1. Запись макроса в Word

Запустите редактор *Microsoft Office Word*.

Выберите на панели *Сервис / Макрос / Начать запись* или нажмите индикатор  *Начать запись* на статусной панели.




Замените имя макроса (*Макрос1*), предложенное по умолчанию, на имя *Создание_Проекта_Распоряжения*.

Добавьте следующий текст в поле *Описание* к автоматически созданному тексту: *Проект распоряжения согласно указания № XXX*.

Примечание. Не используйте ни одну из настроек *Назначить макрос* (ни панели, ни клавиши).

5. Нажмите *ОК*, чтобы закрыть диалоговое окно *Запись макроса* и запустить процесс записи.

При этом на экране появится диалоговое окно *Остановить запись*  и индикатор *REC* на статусной панели изменит цвет (показатель активности записи).

Нажмите кнопку *Создать* на стандартной панели инструментов, чтобы создать новый чистый документ.

Наберите заголовок текста *Проект служебного распоряжения*.

8. Выберите *Вставка / Поле*, чтобы открыть окно *Поле*, выберите в *Категориях* пункт *Все*, если он не подсвечен автоматически, затем выберите поле *Create Date* и задайте формат M/D/YYYY H:MM:SS A/P (например, 20/11/2006 4:15:07 PM) и нажмите *ОК*.

9. Набрать следующий текст:

№ 056

г.

Пермь

[О создании ...]

ПРЕДЛАГАЮ:

1. Создать ...
2. Определить ...
3. Считать ...
4. Распоряжение довести по принадлежности.
5. Контроль за исполнение распоряжения возложить на заместителей начальника по курируемым направлениям.

10. Выполните команду *Сохранить как* (или нажмите *Ctrl+S*), чтобы открыть диалоговое окно.

11. Сохраните документ под именем *Проект1.doc* в папке *Мои документы*.

12. Нажмите кнопку (*Остановить запись*) на панели инструментов или на индикаторе.

13. Протестируйте макрос. Для этого выберите **Сервис / Макрос / Макросы**, а затем выберите имя вашего макроса в списке и нажмите кнопку **Выполнить**. VBA создаст новый документ, введет текст в него и сохранит документ.

Задание № 2. Запись макроса в Excel

Запустите **Microsoft Office Excel**.

Выберите **Сервис / Макрос / Начать запись**.

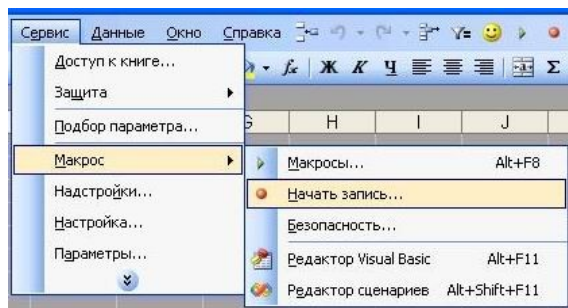
Замените предложенное по умолчанию имя макроса (**Макрос1**), на имя

Создание Рабочего Бюджета.

В списке **Сохранить в** выберите **Личная книга макросов**.

Добавьте следующий текст к началу автоматически созданного описания: **Создание проекта рабочего Бюджета**.

Нажмите **ОК**, чтобы закрыть диалоговое окно **Запись макроса**. При этом запустится процесс



записи и появится диалоговое окно **Остановить запись**.

ПРИМЕЧАНИЕ: И начинаем записывать макрос, как музыку на диске: начинаем записывать все выполняемые вами операции с целью дальнейшего запуска. *Будьте очень внимательны при записи макроса – не делайте лишних действий и ошибок.*

Нажмите кнопку **Создать** на стандартной панели инструментов, чтобы создать новую книгу.

8. Наберите таблицу, предложенную ниже:

ПЕРИОД ВРЕМЕНИ						
Дата / Время		ПРОВодКА				
ОПЕРАЦИЯ		ПРОВодКА				
№ п/п	Дата	ОПИСАНИЕ	Сумма	Дебит	Кредит	ДОКУМЕНТ

9. Выполните команду **Сохранить**, чтобы открыть диалоговое окно **Сохранить как**.

Сохраните книгу под именем **Бюджет.xls** в папке по вашему усмотрению. Например, используем папку **d:\ Общие**.

Нажмите кнопку **Остановить запись** кнопку на индикаторе или на панели инструментов, выбрав **Сервис / Макрос / Остановить запись**.

10. Протестируйте получившийся макрос. Для этого выберите **Сервис / Макрос / Макросы** или нажмите **Alt+F8**. Затем выберите имя вашего макроса в списке, и нажмите кнопку **Выполнить**. VBA создаст новую книгу, создаст таблицу, введет в нее соответствующую информацию и сохранит книгу. Заметьте, что **Excel** покажет вам предупреждение о перезаписи, сохраняемой книга, в то время как **Word** делает это молча.

Задание № 3

Рассмотрим создание макроса переход по страницам любого уровня в тексте документа **Word**. Такой макрос даст нам возможность быстро перемещаться по тексту.

1. В меню **Сервис / Макрос** и активизируем пункт **Начать запись**.

2. В окне **Запись макроса** в качестве имени макроса введём **Переход_страниц**.

3. Для задания горячей комбинации клавиш нажмём вторую кнопку со словом **Клавишам** из группы кнопок под именем **Назначить макрос**.

4. В открывшемся окне *Настройка клавиатуры* нажимаем комбинацию **Ctrl+N**, находясь в строке *Новое сочетание*. Нажмите сначала кнопку *Назначить*, а затем - кнопку *Закрывать*. Горячая комбинация для вызова макроса назначена, и запись макроса началась.
5. В меню *Правка* нажмите *Перейти*.
6. В раскрывшемся окне *Найти и заменить* в списке *Объект перехода* выставляем значение *Страница*. Проверяем, что строка под названием *Введите номер страницы* пуста. Нажимаем кнопку *Далее*. Затем нажмите кнопку *Закрывать*. Диалог *Найти и заменить* закрывается, курсор оказывается на следующей странице в тексте документа.
7. В меню *Сервис / Макрос* активизируем пункт *Остановить запись*.
8. Протестируйте получившейся макрос. Теперь в любой момент мы можем нажать комбинацию **Ctrl+N**, и курсор тут же переместится к следующей странице. Таким образом, быстро можно просмотреть любой объемный документ.
9. Создайте аналогичный макрос для перехода к предыдущей странице. С его помощью можно быстро перемещаться по тексту и в обратном направлении. Для этого горячей комбинацией этого макроса необходимо назначить комбинацию **Ctrl+Shift+N**. А в строке *Введите номер страницы* окна *Найти и заменить* поставьте знак минус.

Задание № 4

1. Запустите *Microsoft Office Excel*.
2. Откройте *Лист Excel* и в ячейке **A1** вставьте цифру 7.
3. Выберите *Сервис / Макрос / Начать запись*.
4. В появившемся окне *Запись макроса* в поле *Имя макроса* указать **ПРОБА**.
5. Задайте комбинацию клавиш для его запуска (например, **Ctrl + q**) и нажмите кнопку **ОК**.
6. Выполнить следующие операции:

ПРИМЕЧАНИЕ: Будьте очень внимательны при записи макроса – не делайте лишних действий и ошибок.

 - сначала выделите ячейку **A1**, нажав на неё, затем копируйте её содержание и вставляйте в ячейку **C1** (простая операция копирования).
7. Нажмите кнопку *Остановить запись* кнопку на индикаторе или на панели инструментов, выбрав *Сервис / Макрос / Остановить запись*.
8. Удалите данные с листа, а затем в ячейку **A1**, поставьте любое значение и нажмите **Ctrl + q**. Если вы всё сделали правильно, в ячейку **C1** должны автоматически копироваться значения **A1**.
9. Перейдем к редактору *Visual Basic*. Для этого через меню *Сервис / Макрос / Макросы* нажмите кнопку *Изменить* – откроется окно редактора *Visual Basic*.
10. Рассмотрите более подробно автоматически созданный программный код. Вот как выглядит макрос или процедура (программный код) задачи копирования ячейки **A1** в ячейку **C1**.


```
Sub Макрос2 ()
    Макрос2 Макрос
    Макрос записан
    Сочетание клавиш: Ctrl+q
    Range("A1").Select
    Selection.Copy
    Range("C1").Select
    ActiveSheet.Paste
End Sub
```
11. Выделите строки программного кода от *«Range...»* до *«...Paste»* и копируйте их либо через меню *Edit / Copy*, либо с помощью правой кнопки мыши.

12. Сместите строку «*End Sub*» на одну строку ниже и в пустую строку вставьте скопированный программный код.
13. Во вставленном программном коде измените «*A1*» на «*C1*» и «*C1*» на «*E1*».
14. Через панель задач перейдите в *Microsoft Office Excel* и очистите все ячейки, а затем наберите любое значение в ячейке *A1*.
15. Запустите на выполнение измененный макрос либо нажатием комбинации клавиш *Ctrl + q*, либо через меню *Сервис / Макрос / Макросы / Выполнить*.

Задание № 5

Запустите *Microsoft Office Excel*.

Откройте *Лист Excel*.

Выберите *Вид / Панели инструментов / Visual Basic*.

На открывшейся панели редактора *Visual Basic* нажмите кнопку *Панель элементов управления* и перенести элемент «*Кнопка*» на рабочее поле.

На *Панели элементов управления* нажмите кнопку *Окно свойства* и в появившемся окне *Properties* в графе *Caption* переименуйте название кнопки *CommandButton1* на *Создание Овала*.

6. Выберите *Сервис / Макрос / Начать запись*.

7. В появившемся окне *Записи Макроса* укажите *Имя макроса* и комбинацию клавиш для его запуска (*Ctrl + a*) и нажмите *ОК*.

8. С *панели Рисование* перенесите на рабочее поле *Овал*, а затем выделите этот объект и зайдите в *Формат Автофигуры*, а затем зайдите *Цвет Заливки* – Красный, *Цвет линии* – черный, *Толщина линии* – 3 пт.

9. В меню *Сервис / Макрос* активизируйте пункт «*Остановить запись*».

10. Зайдите в *Microsoft Visual Basic* в меню *Редактор Visual Basic* в модулях найти наш *Макрос*, скопируйте код макроса и вставьте в процедуру:
Private Sub CommandButton1_Click()

End Sub

11. Протестируйте макрос. Теперь при нажатии кнопки *Создать Овал* будет появляться на рабочем поле соответствующая геометрическая фигура.

Задание № 6

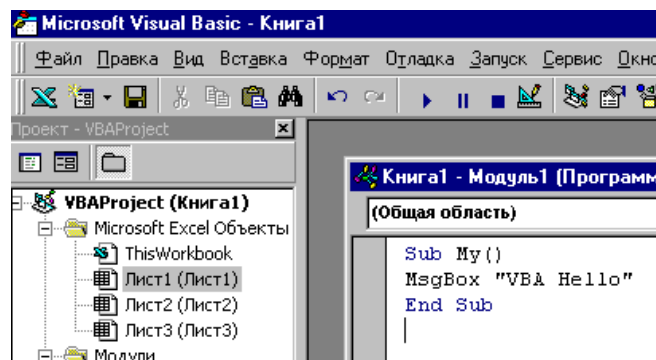
1. Запустите *Microsoft Office Excel*.

2. Откройте окно *Макрос* либо через меню *Сервис / Макрос / Макросы*, либо нажатием кнопок (*Alt + F8*).

3. Создайте макрос. Для этого в поле *Имя* наберите *MY*, а в поле *Находится в* выберите - *Эта книга* (место нахождения этого макроса).

4. Нажмите кнопку *Создать*, если она станет активной. Появится редактор *VBA*.

5. Введите в пустую строку программный код `MsgBox "ПРИВЕТ VBA!!!"`, как на рисунке (Пустая строка между *Sub* и *End Sub*).

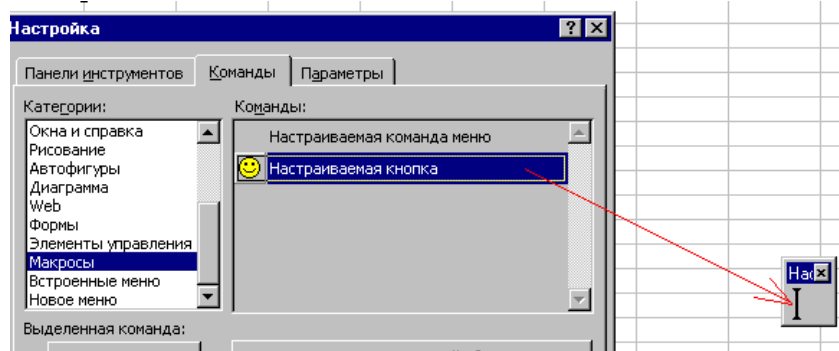


ПРИМЕЧАНИЕ: Здесь используется функция *MsgBox*, которая выводит на экран окно сообщения.

6. Закройте редактор *VBA* и закройте файл.

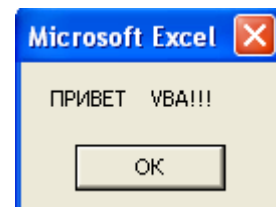
7. Привяжите макрос к кнопке. Для этого через меню *Сервис / Настройка / Команды* в разделе *Категории* выберите *Макросы*.

8. Захватите левой кнопкой мыши «веселую желтую рожицу» и перетащите ее на панель.



9. Нажмите правой кнопкой мыши на «рожице» и выберите пункт меню **Назначить макрос**. 10. Выберите макрос **МУ** и нажмите **Ок**, а затем закройте окно настройки.

11. Протестируйте макрос. Для этого нажмите кнопку - макрос выполняется и появляется надпись.



Задание № 7

Запустите редактор **Microsoft Office Word**.

Создайте новый макрос с именем **Радиус** через **Сервис / Макрос / Начать запись** и нажмите **Остановить запись**.

Откройте окно редактора **Visual Basic** и найдите в окне **Code** программный код:

```
Sub Радиус ()
'
' Радиус Макрос
' Макрос создан < Дата > < Ваше имя >
'
```

End Sub

4. Запишите в пустой строке следующий программный код:

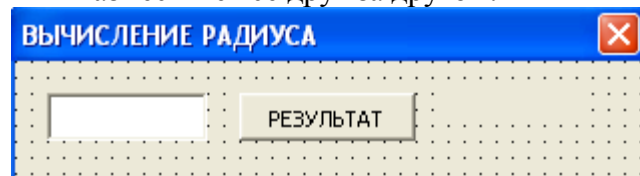
RADfrm.Show

ПРИМЕЧАНИЕ: Этой строкой вызывается форма **RADfrm**.

5. Создайте саму форму. Для этого в окне редактора **Visual Basic** найдите кнопку (**Insert UserForm**) перед вами появится окно с пустой формой, перейдите на панель **Properties** (слева, если у вас его нет нажмите кнопку **F4**). В ней найдите свойство (**Name**) и измените его значение (по умолчанию **UserForm1**) на **RADfrm**, в свойстве **Caption** (**Надпись**) наберите **ВЫЧИСЛЕНИЕ РАДИУСА**.

6. На саму форму перенесите **Textbox** (окно ввода), **Button** (кнопку) и **Label** (метку).

Разместите всё друг за другом:



7. Задайте следующие параметры через на панель **Properties**:

Textbox:

Name: txtRadius

Text: (пусто)

Button:

Name: cmdCalc

Caption: РЕЗУЛЬТАТ

Label:

Name: lblLenght

Caption: (пусто)

8. Разберитесь в том, что должно получиться: в поле ввода вводим цифру, нажимаем на кнопку **Результат**, и на Label появляется сосчитанная по формуле длина окружности.
9. Откройте окна программного кода созданной формы. Для этого щелкните два раза по кнопке **Результат**.

10. Напишите в окне кода следующее:

'**Объявляем переменные радиуса и длины окружности**

Dim Radius As Integer

Dim Lenght As Integer

' **Объявление постоянного числа Pi**

Const Pi = 3.14159

Private Sub cmdCalc_Click()

'**Считывание чисел в окне ввода**

Radius = txtRadius.Text

' **Считаем...**

Lenght = 2 * Radius * Pi

'**Полученный результат выводим на Label**

lblLenght.Caption = Lenght

End Sub

11. Протестируйте макрос. Для этого перейдите через **Панель задач** в **Word**, запустите на выполнение макрос, в появившемся окне **ВЫЧИСЛЕНИЕ РАДИУСА** в поле **Textbox** наберите любое число и нажмите кнопку **Результат**.

Задание № 8

Это задание позволяет сделать текущий запрос о свободной оперативной памяти. К примеру, на слабых машинах, из-за сильной их загруженности, приложения **Microsoft Office** вылетают с сообщением об ошибке. Причина, которой можно объяснить это - нехватка оперативной памяти, и ошибка при переходе на использование «виртуальной» памяти.

ПРИМЕЧАНИЕ: Код этого макроса уже намного серьезнее предыдущих. Стандартными средствами **Visual Basic**, на котором пишутся макросы, эту задачу не решить. На помощь приходят **API - функции Windows**. **API** - это сокращение от **Application Programming Interface**. **API** состоит из целого ряда функций, которые позволяют использовать системные **Windows-конструкции**. Чтобы использовать **API - функции**, нужно сначала их декларировать (объявить).

Запустите **Microsoft Office Excel**.

Создайте новый макрос с именем **RAM** через **Сервис / Макрос / Начать запись** и нажмите **Остановить запись**.

Откройте окно редактора **Visual Basic** и найдите в окне **Code** программный код:

```
Sub RAM ()
```

```
,
```

```
' RAM Макрос
```

```
' Макрос создан < Дата > < Ваше имя >
```


```
,
```

```
End Sub
```

4. Запишите в пустой строке

```
Ramfrm.Show
```

ПРИМЕЧАНИЕ: Этой строкой вызывается форма **Ramfrm**.

Создайте саму форму. Для этого в окне редактора **Visual Basic** найдите кнопку  (Insert UserForm) и перед вами появится окно с пустой формой. Перейдите на панель **Properties** (слева, если у вас его нет нажмите **F4**). В ней найдите свойство (**Name**) и измените его значение (по умолчанию **UserForm1**) на **Ramfrm**.

Нажмите **F7**, появится окно редактора кода. Удалите все, что там есть и внесите туда следующий код:

```
Private Declare Sub GlobalMemoryStatus Lib "kernel32" (lpBuffer As MEMORYSTATUS)
```

```
Private Type MEMORYSTATUS
```

```
dwLength As Long
```

dwMemoryLoad As Long
 dwTotalPhys As Long
 dwAvailPhys As Long
 dwTotalPageFile As Long
 dwAvailPageFile As Long
 dwTotalVirtual As Long
 dwAvailVirtual As Long
 End Type

7. Наберите программный код процедуры, которая выполняется при старте формы:

```

Private Sub UserForm_Initialize()
Dim w1 As MEMORYSTATUS
Dim q1 As Single
Dim q2 As Single
Dim d2 As String
Dim e As Integer
w1.dwLength = Len(w1)
GlobalMemoryStatus w1
q1 = w1.dwMemoryLoad
q2 = w1.dwTotalPhys
d2 = (q2 - q1 * 0.01 * q2) / 1024 ^ 2
d2 = Int(d2 * 10) / 10
d2 = d2 & " Mb"
e = MsgBox("Свободно: " & d2, vbOKOnly + vbInformation, "Информация о RAM")
If e = 1 Then
End
End If
End Sub
  
```

8. Протестируйте созданный макрос *RAM*.

7. Варианты заданий.

Номер варианта (реферата и практического задания) определяются в табл.1 по последней и предпоследней цифрам номера зачетной книжки студента (шифра). Например, студент, у которого номер зачетной книжки оканчивается на 58, должен выбрать вариант №29 по списку тем рефератов и по табл. 2.

Таблица 1

Таблица 1. Номер варианта.

Предпоследняя цифра шифра	Последняя цифра шифра									
	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	11	12	13	14	15	16	17	18	19	20
2	21	22	23	24	25	26	27	28	29	30
3	1	2	3	4	5	6	7	8	9	10
4	11	12	13	14	15	16	17	18	19	20
5	21	22	23	24	25	26	27	28	29	30
6	1	2	3	4	5	6	7	8	9	10
7	11	12	13	14	15	16	17	18	19	20
8	21	22	23	24	25	26	27	28	29	30
9	1	2	3	4	5	6	7	8	9	10

7.1 Темы рефератов.

1. Информация. Информационные процессы. Информатика.
2. История развития средств вычислительной техники.

3. Программное обеспечение ПК: базовое, системное, служебное, прикладное.
4. Внутреннее устройство системного блока. Базовая аппаратная конфигурация.
5. Устройства, расположенные на материнской плате. Оперативная память.
6. Устройства, расположенные на материнской плате. Процессор.
7. Устройства, расположенные на материнской плате. Микросхема ПЗУ и система BIOS.
8. Устройства, расположенные на материнской плате. Энергонезависимая память CMOS. Микропроцессорный комплект (чипсет).
9. Устройства, расположенные на материнской плате. Шинные интерфейсы материнской платы (ISA, EISA, VLB, PCI, FSB, AGP, USB и т.д.)
10. Периферийные устройства персонального компьютера. Сканеры.
11. Периферийные устройства персонального компьютера. Принтеры.
12. Устройства вывода данных. Мониторы. Виды, разрешение экрана, цветное разрешение (глубина цвета).
13. Внутреннее устройство системного блока. Видеокарта (видеоадаптер). Стандарты MDA, CGA, EGA, VGA, SVGA. Звуковая карта.
14. Внешние запоминающие устройства. Жесткий диск.
15. Внешние запоминающие устройства. Гибкие диски, CD, DVD, флэш-карты.
16. Операционные системы. Виды, функции. ОС MS-DOS.
17. Операционные системы. Виды, функции. ОС Windows.
18. Операционные системы. Виды, функции. ОС Unix.
19. Операционные системы. Виды, функции. ОС Linux.
20. Файл. Файловая система. Структура, функции, виды. FAT 16, FAT 32, NTFS.
21. Сжатие данных. Обратимые и необратимые методы. Архивирование.
22. Кодирование данных. Двоичное, восьмеричное, шестнадцатеричное кодирование. Кодирование текстовых данных. Системы кодирования ASCII, UNICODE
23. Кодирование графических и звуковых данных. Принцип декомпозиции. Система кодирования RGB и CMYK. Режимы True Color, High Color. Метод FM, метод таблично-волнового синтеза.
24. Виды угроз безопасности информации.
25. Средства и методы защиты информации.
26. Правовая охрана программ и данных. Защита информации. Шифрование информации, электронная подпись, электронные сертификаты.
27. Глобальная сеть INTERNET. История возникновения. Топология и состав.
28. Протоколы сети INTERNET . адресация в сети INTERNET .
29. Сервисы INTERNET .

30. Компьютерные вирусы. Понятие, классификация. Средства антивирусной защиты.

7.2. Варианты практического задания.

Таблица 2. Варианты заданий.

№	Функция F(x)	интервал	шаг	Вывести график функции y(F(x))
1	$\sin(x) + \cos(3x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
2	$2\sin(2x) + 0.5\sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
3	$\sin(x) - \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
4	$2\sin(x) + \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
5	$\sin(3x) + \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
6	$2\cos(2x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
7	$\sin(2x) - \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
8	$\cos(x) - \sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
9	$\cos(3x) + \sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
10	$3\sin(x) + \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
11	$3\cos(x) + \sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
12	$\sin(x) + 2\cos(3x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
13	$2\sin(3x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
14	$3\sin(x) - \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
15	$\sin(2x) + \cos(2x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \end{cases}$

				-1, если $F(x) < 0$ на инт-ле $[\pi, 1.5\pi]$
16	$\sin(3x) + \cos(3x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
17	$3\cos(2x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
18	$\sin(2x) - \cos(2x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
19	$\cos(x) - 3\sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
20	$\cos(3x) + 2\sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
21	$\sin(0.5x) + \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
22	$\cos(0.5x) + \sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
23	$\cos(2x) + \cos(0.5x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
24	$\sin(3x) + 0.5\sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
25	$0.5\sin(x) + \cos(3x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
26	$0.5\sin(2x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
27	$\sin(0.5x) - \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
28	$2\sin(x) + 0.5\cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
29	$\sin(3x) + 0.5\sin(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
30	$2\cos(2x) + \cos(0.5)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ -1, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$

Заключение

Методические рекомендации являются основой для прохождения практики и оформления отчета.

Во время учебной практики студенты приобретают практические навыки решения стандартных задач в профессиональной деятельности, а также навыков программирования, что позволяет им подготовиться к изучению последующих дисциплин, связанных с программированием и к выполнению предстоящей выпускной квалификационной работы.

Список использованных источников

1. Положения о порядке проведения практики студентов ФГБОУ ВО Пермская ГСХА (<http://pgsha.ru/web/studyprocess/declaration/>).
2. Рабочий учебный план по направлению 09.03.03 «Прикладная информатика». Утвержден ученым советом ФГБОУ ВО Пермская ГСХА 28.04.2016. (<http://pgsha.ru/web/studyprocess/rup/>)

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Пермский государственный аграрно-технологический университет
имени академика Д. Н. Прянишникова»

ОБРАЗЕЦ

Кафедра Информационных технологий и
программирования

ОТЧЕТ О ПРАКТИКЕ ПО ПОЛУЧЕНИЮ ПЕРВИЧНЫХ ПРОФЕССИОНАЛЬНЫХ УМЕНИЙ И НАВЫКОВ, В ТОМ ЧИСЛЕ УМЕНИЙ И НАВЫКОВ НИР (по информатике)

«Устройства вывода данных. Мониторы. Виды, разрешение экрана, цветовое разрешение (глубина цвета)»

«Программирование, расчет значений и построение графика тригонометрической функции на заданном интервале»

Название в соответствии со своим вариантом реферата

Выполнила:
студентка 1 курса
направления 09.09.03 «Прикладная информатика»
шифр **Пиб-15-404**
Суханова Любовь Сергеевна

Проверил:
профессор
Козлов Алексей Николаевич

Пермь-20....

СОДЕРЖАНИЕ

I. Теоретическая часть: реферат на тему «...*(согласно варианта)* ...».

Введение.

- 1. ...
- 1.1.
- 1.2. ...
- 2. ...
- 2.1. ...
- 2.2. ...

Основная часть реферата (2-3 раздела с подразделами).

Заключение.

Список литературы. *(5-10 наименований)*

II. Практическая часть: расчет значений тригонометрической функции.

- 1. Исходные данные и краткое описание работы.
- 2. Текст программы на языке Паскаль.
- 3. Полученные значения функции и график функции.
- 4. Значения и график функции, выполненные в MS Excel.
- 5. Макросы.

Приложение: CD с файлами.

**Не забудь указать
страницы**