

1.Задание.

1. 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.
2. Выполнить эту же задачу в MS Excel.
3. Отработать создание макросов с помощью VBA
4. Создать макрос в MS Excel, позволяющий рассчитать значение заданной согласно варианту тригонометрической функции для заданного (любого) значения аргумента. Создать кнопку на панели инструментов для вызова данного макроса.

7	$\sin(2x) - \cos(x)$	$[0, 2\pi]$	$2\pi / 36$	$y = \begin{cases} F(x) & \text{на инт-ле } [0, 2\pi] \\ 0.5, & \text{если } F(x) < 0 \text{ на инт-ле } [\pi, 1.5\pi] \end{cases}$
---	----------------------	-------------	-------------	---

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.tpu, 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.tpu, 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 VGHi, 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: Xmin – минимальная граница, Xmax – максимальная граница аргумента функции;

2. Для заданной области значений аргумента оценим минимальное и максимальное значения функции: Ymin и Ymax;

3. Зададим границы графического окна, в пределах которого будем строить график: $[X_{gmin}, X_{gmax}]$, $[Y_{gmin}, Y_{gmax}]$. Поскольку в графических координатах вертикальная ось направлена вниз, то $Y_{gmin} > Y_{gmax}$;

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:

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

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

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. $X_{gmin} = 10$, $X_{gmax} = 210$, $Y_{gmin} = 240$, $Y_{gmax} = 40$;

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:

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

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

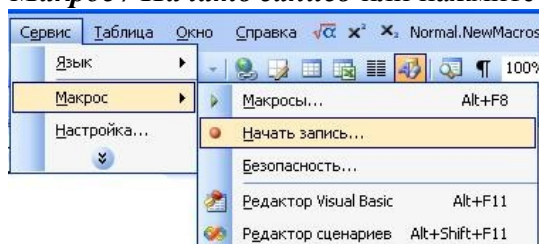
Вместе с графиком функций выводим на экран оси координат. Ось X имеет координату $Y_g = (Y_{gmin} + Y_{gmax}) / 2 = 140$, ось Y имеет координату $X_g = X_{gmin} = 10$;

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 М/Д/YYYY Н:ММ:СС А/Р (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*.

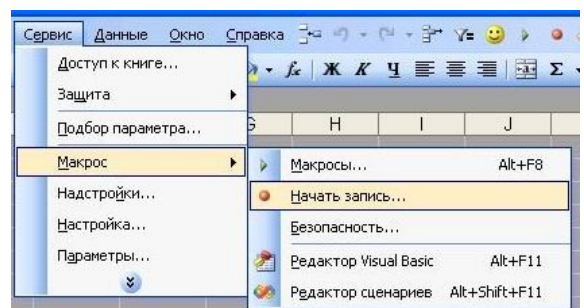
Замените предложенное по умолчанию имя макроса (*Макрос1*), на имя *Создание_Рабочего_Бюджета*.

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						
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 **Заккрыть**. 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 **Заккрыть**. 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 переместиться 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** поставить 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
```

ПРИМЕЧАНИЕ: Процедура - это общее название всех самостоятельных частей кода. Процедура - участок кода, выделенный ключевыми словами **Sub** (в начале процедуры: за ним следует ее имя) и **End Sub** (окончание процедуры: помещается на отдельной строке). Обычно процедура выполняет одно или несколько действий.

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 перейти 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

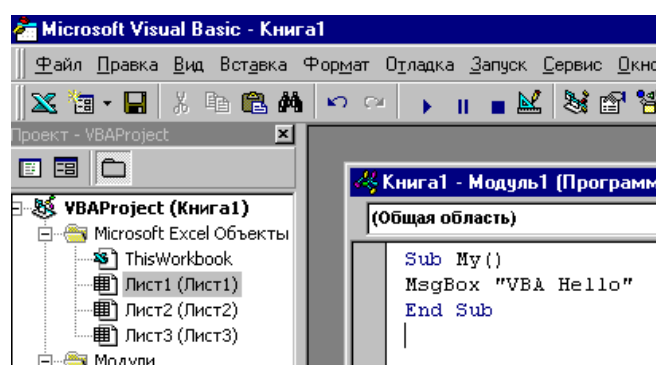
1. Запустите *Microsoft Office Excel*.

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

3. Создайте макрос. Для этого в поле *Имя* наберите *МУ*, 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*).

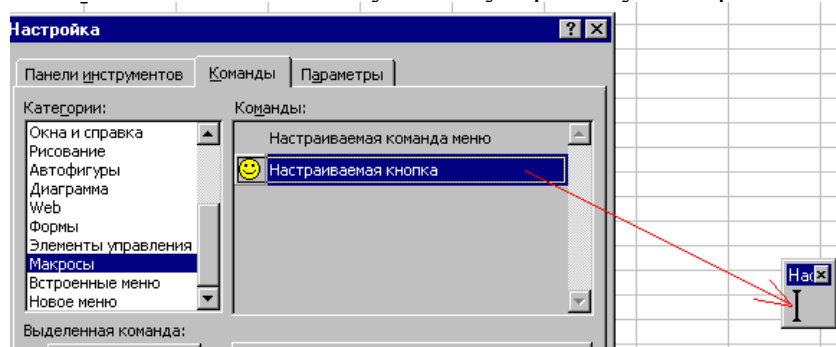


ПРИМЕЧАНИЕ: Здесь используется функция *MsgBox*, которая выводит на экран окно сообщения.

6. Закройте редактор *VBA* и закройте файл.

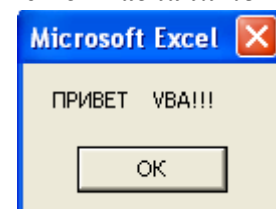
7. Привяжите макрос к кнопке. Для этого через меню *Сервис / Настройка / Команды* в разделе *Категории* выберите *Макросы*.

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:

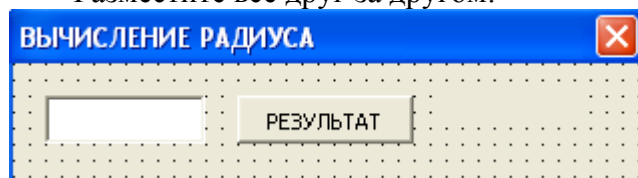
```
RADfrm.Show
```

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

End
End If
End Sub

8. Протестируйте созданный макрос *RAM*.