

# Лабораторная работа №1

## Тема: Ознакомление с базовыми понятиями в VBA.

**Цель работы:** закрепить знания о понятиях VBA: типы данных, выражения, массивы, процедуры, функции, условные операторы, циклы

## Теоретическая часть:

Лекции 1-3

### Запуск редактора VBA

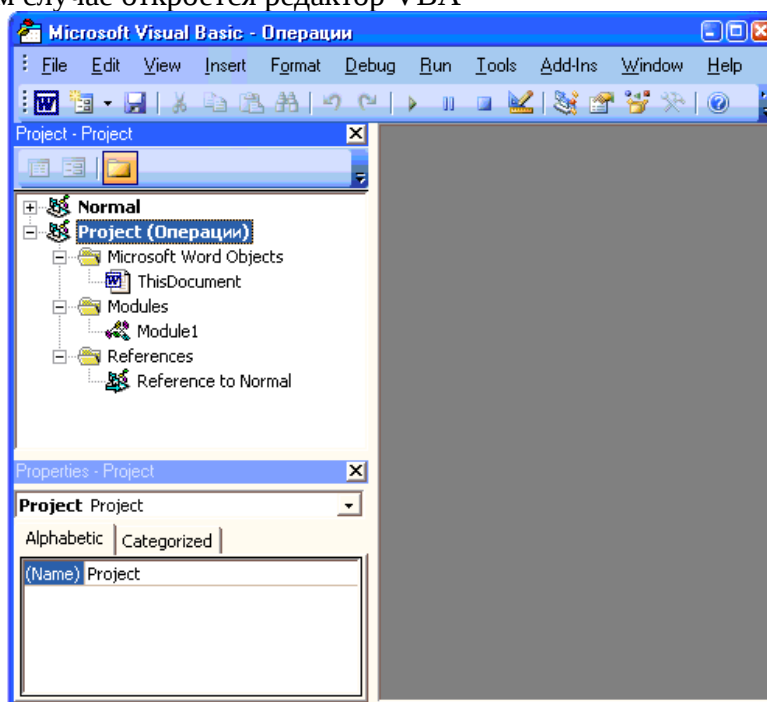
Как и любые среды программирования, редактор VBA необходимо сначала запустить. Для запуска можно использовать два способа:

- 1) активизировать любое приложение пакета MS Office (Word, Excel);
- 2) выполнить команду меню: Сервис + Макрос + Редактор Visual Basic.

Или:

- 1) активизировать любое приложение пакета MS Office (Word, Excel);
- 2) нажать комбинацию клавиш Alt+F11.

И в том, и в другом случае откроется редактор VBA



В левой части окна редактора появляется строение разрабатываемого проекта (аналог с Проводником). Необходимо обратить внимание на объект Project (Операции) — это объект содержит информацию о документе, с которым в данный момент идет работа, т. е. дается подсказка, в каком документе необходимо работать и где создаются модули, процедуры, приложения.

### Ход работы

**1. Создать программу, высчитывающую выражение:  $\sin(a^2) + \cos(b^2) + \text{Pi}$  и выводящую полученное значение на экран , где Pi — константа Пи, a,b — вводятся пользователем с экрана.**

**Сведения для выполнения задания:**

Функция **InputBox** выводит на дисплей диалоговое окно, которое содержит командные кнопки OK и Cancel, текстовое поле для ввода входных данных и некоторый текст, который запрашивает у пользователя ввод этих данных. В общем случае функция InputBox имеет следующий синтаксис:

***stringvar=InputBox (Prompt [, Title] [, Default] [, XPos] [, YPos] [, HelpFile] [, Context])***

где *stringvar* - любая переменная, сохраняющая строку;

*Prompt* - любое строковое выражение, содержащее информацию, выводимую в диалоговом окне (единственный обязательный аргумент этой функции);

*Title* - строка для заголовка диалогового окна;

*Default* - строковое выражение, используемое как значение по умолчанию для пользовательского ввода;

*XPos* - горизонтальное расстояние от левого края окна;

*YPos* - вертикальное расстояние от верхнего края окна;

*HelpFile* - строковое выражение, содержащее имя справочного файла Windows;

*Context* - численное выражение, указывающее раздел в справочном файле, относящийся к отображаемому диалоговому окну.

Функция **MsgBox** служит для вывода сообщений пользователю процедуры:

**MsgBox "Hello, World!"**

**Опорный пример**

```
Dim a As Double
```

```
Dim b, c, d As Double
```

```
Dim k As String
```

```
Const Pi As Double = 3.14
```

```
Sub list1()
```

```
a = 5
```

```
b = 25.6
```

```
c = Sin(a ^ 2) + Cos(b) + Pi
```

```
d = Sqr(c)
```

```
MsgBox (d)
```

```
End Sub
```

Запустите программу на выполнение, предварительно ее откомпилировав (проверив синтаксические ошибки). Для компиляции проекта выполните команду меню: Debug + Compile Project. Если ошибок нет, запустите приложение при помощи кнопки **Запуск** панели инструментов или клавиши F5. Если все действия выполнены верно, то на экране появится диалоговое окно MsgBox с итоговым сообщением.

**2. Создать программу, которая производит сложение строковых переменных и результат выводит в диалоговое окно msgbox.**

## Технология выполнения

1. В том же документе выделите правой кнопкой мыши папку Modules и выполните команду: Insert + Module
2. В результате появится новый модуль, где можно создать новую программу (модуль). Пропишите следующий код.

```
Dim a, b, c As String
Sub list2()
a = «Привет!»
b = « Пока не сложно?»
c = a + b
MsgBox (c)
End Sub
```

В данном примере переменным а и b присваиваются строковые значения (в кавычках!), после чего происходит сложение строк.

3. Прокомпилируйте программу и запустите ее на выполнение.  
Примечание.

Если в одном документе находятся несколько модулей, то при запуске программ может появляться диалоговое окно выбора макроса (модуля), в котором необходимо выбрать макрос, выделить его и нажать кнопку «Run»

**3. Создать программу, которая, используя инструкцию if ... then, выполняет следующие действия: если переменной а присваивается с клавиатуры значение больше нуля, то находится сумма чисел а и b, если меньше нуля, то находится произведение. Результат выводится в стандартное диалоговое окно msgbox.**

## Опорный пример

```
Dim a, b, c As Integer
Sub poradok()
a = -5
b = 25
If a > 0 Then
c = a + b
MsgBox (c)
End If
If a < 0 Then c = a * b
MsgBox (c)
End If
End Sub
```

**4. Используя инструкцию case, создать программу, которая в зависимости от введенного значения переменной а производит различные вычисления с переменными b и c. Если значение переменной а не совпадает с программными, то выдается сообщение «Введено не то значение». Все переменные вводятся в текстовые поля формы. При нажатии на кнопку «Результат» происходят выбор действия и вывод полученного значения в специальную метку формы.**

## Технология выполнения

1. Активизируйте приложение Word, сохраните документ под именем Case.
2. Перейдите в редактор VBA и создайте форму (рис. 18).
3. Пропишите обработчик кнопки «Результат».

```

Dim a, b, c, d As Integer
Private Sub CommandButton1_Click()
a = Val(TextBox1.Text)
b = Val(TextBox2.Text)
c = Val(TextBox3.Text)
Select Case a
Case 5
d = b + c
Label4.Caption = «Результат: d=» & d
Case 0
d = - b - c
Label4.Caption = «Результат: d=» & d
Case 10
d = b * c
Label4.Caption = «Результат: d=» & d
Case Else
Label4.Caption = «Введено не то значение»
End Select
End Sub

```

**5. Создать программу, которая, используя пользовательское диалоговое окно (форму), выполняет следующие действия: при вводе трех переменных в текстовые поля она считывает данные и сравнивает с первой переменной а. Результат выдается в метку на форме**

```

Private Sub CommandButton1_Click()
Dim a, b, c As Integer
a = Val(TextBox1.Text)
b = Val(TextBox2.Text)
c = Val(TextBox3.Text)
If a > b And a > c Then
Label1.Caption = «Значение а > b и а >с»
Else
Label1.Caption = «Значение а не всегда больше b и c» End If End Sub

```