

Лекция. Основы программирования на Visual Basic for Applications (VBA)

План:

1. Среда программирования Visual Basic for Applications (VBA).
2. Типы данных и переменные
3. Организация ввода-вывода
4. Форма и элементы управления

1. Среда программирования Visual Basic for Applications (VBA)

Visual Basic for Applications (VBA) - встроенный язык программирования приложений Microsoft Office, в частности, электронных таблиц Excel.

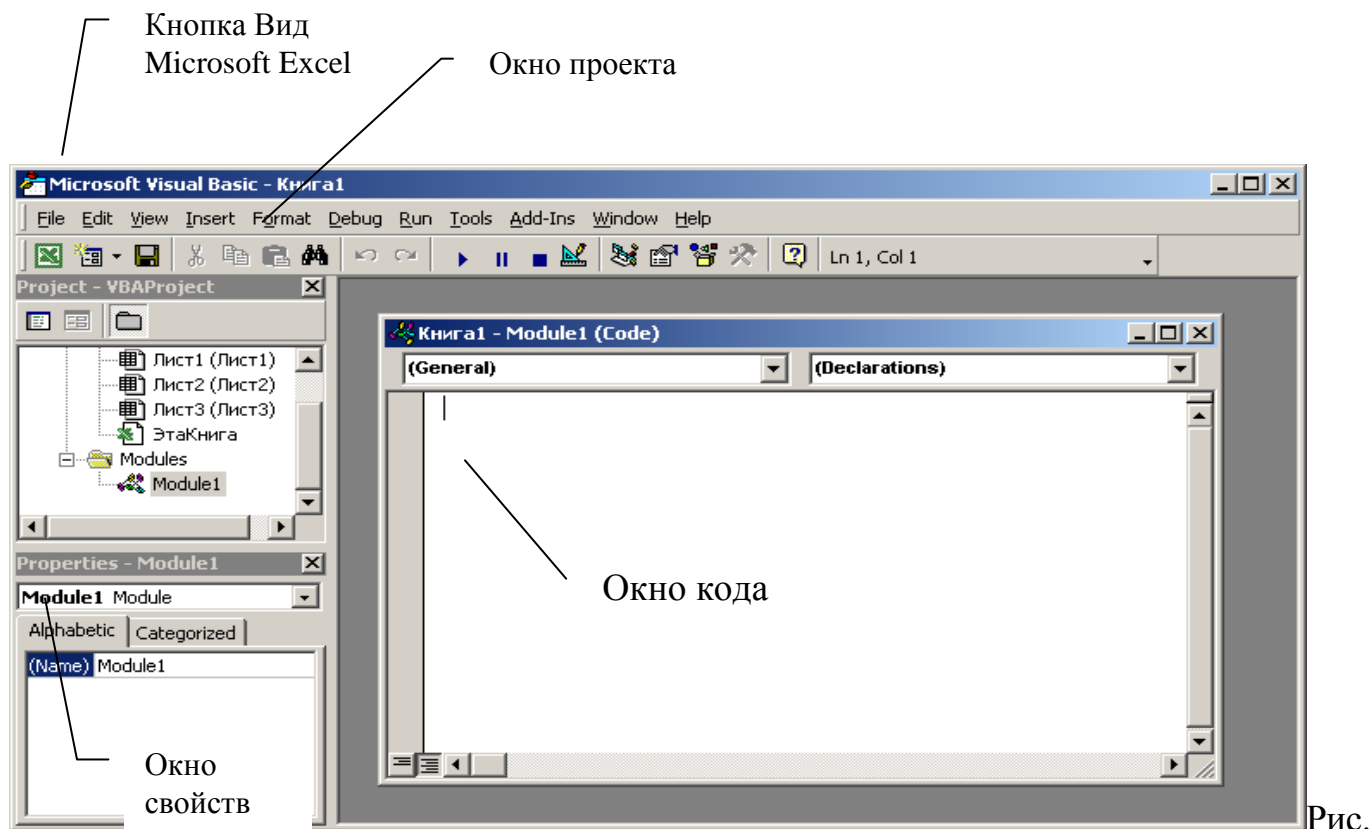
Для создания программы на VBA необходимо находясь в рабочей книге Excel, выполнить команду Сервис/Макрос/Редактор Visual Basic (или Alt/F11).

Возвратиться из редактора VBA в рабочую книгу можно нажатием кнопки Вид/Microsoft Excel.

Приостановить программу можно нажатием кнопок Ctrl/Pause Break.

Программа размещается в модуле, который добавляется к файлу Excel и будет сохраняться вместе с ним.

Интерфейс редактора VBA состоит из следующих основных компонентов: *окно проекта, окно свойств, окно редактирования кода, окна форм, меню и панели инструментов* (рис. 1).



1 - Интерфейс редактора VBA

В окне проекта представлена иерархическая структура файлов форм и модулей текущего проекта. В проекте автоматически создается модуль для каждого рабочего листа и для всей книги. Кроме того, модули создаются для каждой пользовательской формы макросов и классов.

В окне проекта выводится проект всех открытых рабочих книг. Это позволяет легко копировать формы и коды из одного проекта в другой, что ускоряет процесс создания новых приложений.

В окне свойств перечисляются основные установки свойств выбранной формы или элемента управления. Используя это окно, можно просматривать свойства и изменять их установки. Для просмотра свойств выбранного объекта надо либо щелкнуть кнопку *Окно свойств*, либо выбрать команду Вид/ Окно свойств (View/Property Window).

Программа (код программы) записывается в окне кода. **Окно кода** используется при написании любой программы VBA, будь это код макроса, запуск которого осуществляется при нажатии кнопки в созданной пользователем форме, или подпрограмма. Код программы вводится непосредственно в окно кода, так же как текст в любом текстовом редакторе.

2. Типы данных и переменные

Тип данных определяет, каким образом величина хранится в памяти компьютера. Задавая тип данных, мы налагаем некоторые ограничения на сохраняемую величину, а именно: величина переменной, количество разрядов и др.

Наиболее распространенные типы данных приведены в таблице 1.

Таблица 1

Тип данных	Диапазон
Byte (целый без знака)	От 0 до 255
Integer (целый)	от - 32768 до 32767
Long (длинный целый)	от - 2147483648 до 2147483647
Single (десятичный с одинарной точностью)	От 10^{-35} до 10^{38}
Double (десятичный с двойной точности)	От 10^{-324} до 10^{308}
Boolean (логический)	принимает значения True или False
Date (дата и время)	от 01. 01. 100 до 31. 12.9999
String (строка)	от 0 до 65535 символов
Variant	Любой встроенный тип данных

Область действия переменных и констант

Область действия переменных и констант определяется с помощью ключевых слов: **Private**, **Public** и **Static**.

Private— область действия в пределах конкретного модуля. После завершения выполнения модуля память, отведенная под эти переменные, освобождается.

Public— область действия в пределах приложения.

Static – область действия в пределах конкретного модуля внешних процедур, используемых в данном модуле. После завершения выполнения модуля значения этих переменных сохраняются и могут быть использованы при повторном выполнении модуля.

Переменные и константы, в зависимости от области действия, подразделяются на **глобальные** и **локальные**.

Если переменная или константа описана внутри процедуры, то она является **локальной**, то есть она определена и может использоваться только в пределах данной процедуры.

Если переменная или константа описана вне процедуры, то она будет **глобальной**. Такая переменная или константа может быть использована в нескольких процедурах.

Объявляя переменную, можно указать ее тип, указывая тем самым VBA, какой тип данных должен быть использован при записи переменной в память. Если тип не указан - по умолчанию используется тип Variant, который дает возможность хранить в нем практически любую величину.

Переменная объявляется (описывается) с помощью ключевых слов **Private**, **Public**, **Static**, **Dim**. Чтобы явно указать тип переменной, используется ключевое слово **As**.

Переменная объявляется следующим образом:

Dim <ИмяПеременной> As Тип

Пример.

Dim A As Integer	Переменная A объявляется как целочисленная
Dim Фамилия As String	Переменная Фамилия объявляется как строка символов

Примеры описания простых переменных:**Private X****Public i As Integer, r As Long, c As Date****Static Строка As String****Dim Y****Dim Z As Single** ‘Явный способ объявления переменной. Самый простой и надёжный.Примеры описания индексированных переменных:**Dim Массив1() As Integer****Dim Массив2(12) As String****Dim Массив3(1 to 20) As Single****Dim Массив4(1 to 5, 1 to 7) As Byte**

Константа объявляется с помощью ключевого слова **Const**. При этом можно указать её тип, область действия и присвоить ей значение.

Константа объявляется следующим образом:

Const <имя константы> **As** <тип> = <значение>

или

Const <имя константы> = <значение>

Пример

`Const Процентная Ставка As Single =0.2``Const g=9.8`

Если в константе явно не указан тип данных, то VBA назначает ей тип, который соответствует значению выражения.

Примеры:**Private Const q = 44,55****Public Const pi = 3,1459****Static Const QWER=2,54****Const y = 34****Const Con As Byte = 34****Const z As Single = -3,8374E-22**

Все строковые константы указываются в кавычках.

Примеры:**Const prv As String = “Язык программирования VBA”****Public Const prv= “Язык программирования VBA”**

Логические выражения в результате вычисления принимают логические значения True (Истина) или False (Ложь). Операндами логического выражения могут быть логические константы, переменные логического типа, отношения. В VBA чаще используют 4 логические операции: отрицание — NOT, логическое умножение — AND, логическое сложение — OR, исключающее “или” — XOR.

Функции преобразования типов выполняют преобразование переменной некоторого типа в заданный тип (таблица 2). Чаще всего выполняется преобразование из строки символов в число и числа в его строковое представление.

Таблица 2

Функция	Назначение
Val (строка)	Возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа
Str (число)	Возвращает значение типа variant (String), являющееся строковым представлением числа

3. Организация ввода-вывода

Ввод и вывод данных в VBA может выполняться несколькими способами: с листа Excel, с помощью диалоговых окон, с помощью пользовательской формы.

Для ввода данных с листа или вывода на лист используется объект Worksheets и его методы Range или Cells.

Для **ввода данных** с клавиатуры используется окно ввода InputBox,

Пример.

Имя =InputBox("Введите Ваше имя", "Пример окна ввода")

На экране появится окно.

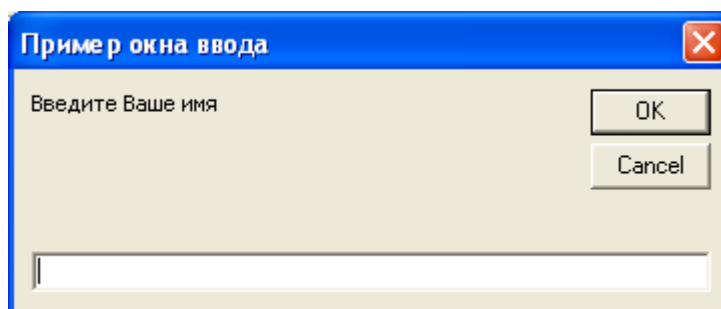


Рисунок 2 – Пример окна ввода

Переменной Имя будет присвоено значение типа String, введенное пользователем.

Следует учесть, что, поскольку введенные пользователем данные считаются текстом, при вводе числовых значений необходимо преобразовать их к одному из числовых типов данных с помощью функции преобразования типа, например `val`.

```
X=Val(InputBox("Введите значение X", "Пример окна ввода", "1,678"))
```

В результате выполнения этой операции на экране появится окно ввода.

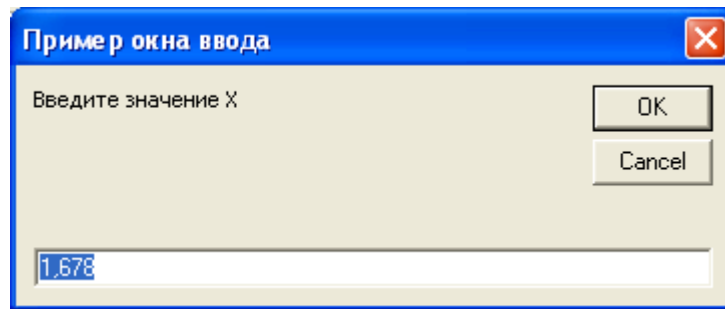


Рисунок 3 – Пример окна ввода

Введенное пользователем значение будет преобразовано к типу `Double` и присвоено переменной `X`. Если пользователь не будет вводить значение, а просто нажмет кнопку `OK`, переменной `X` будет присвоено значение по умолчанию – 1.678.

Для **вывода информации** на экран используется окно сообщений `MsgBox`.

Пример.

```
MsgBox "Значение переменной X=" & X
```

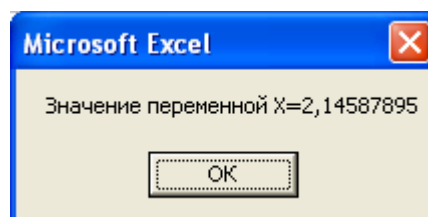




Рисунок 6 – Пример окна сообщений

Знак & (конкатенация) в программном коде позволяет выводить несколько сообщений в одной строке, разделенных этим знаком (т.е. слияние подстрок в строку).

4. Форма и элементы управления

По своей сути форма (или пользовательская форма) представляет собой диалоговое окно, в котором можно размещать различные элементы управления. В приложении может быть как одна, так и несколько форм. Новая форма добавляется в проект выбором команды **Вставка (Insert) → UserForm**.

В VBA имеется обширный набор встроенных элементов управления. Используя этот набор и редактор форм, нетрудно создать любой пользовательский интерфейс, который будет удовлетворять всем требованиям, предъявляемым к интерфейсу в среде Windows. Элементы управления являются объектами. Как любые объекты, они обладают свойствами, методами и событиями. Элементы управления создаются при помощи Панели элементов, которая отображается на экране либо выбором команды **Вид (View) → Панель элементов (Toolbox)**, либо нажатием кнопки  панели инструментов **Standard**. На этой панели представлены кнопки, позволяющие конструировать элементы управления. Для создания элементов управления служат все кнопки панели инструментов, за исключением кнопки **Выбор объекта** . Щелкнув по кнопке **Выбор объекта**, можно выбрать уже созданный в форме элемент управления для последующего его редактирования (изменения размеров или редактирования).

Список основных элементов управления и соответствующих кнопок панели элементов приведен в таблице 3.

Таблица 3

Элемент управления	Имя	Кнопка, его создающая	Элемент управления	Имя	Кнопка, его создающая
Поле	TextBox		Переключатель	OptionButton	
Надпись	Label		Флажок	CheckBox	
Кнопка	CommandButton		Выключатель	ToggleButton	
Список	Listbox		Рамка	Frame	
Поле со списком	ComboBox		Рисунок	Image	
Полоса прокрутки	ScrolBar		Набор страниц	MultiPage	
Счетчик	SpinButton		Набор вкладок	TabStrip	

Для размещения элемента управления на лист или в форму необходимо нажать соответствующую кнопку на панели элементов и с помощью мыши перетащить рамку элемента управления в нужное место. После этого элемент управления можно перемещать, изменять его размеры, копировать в буфер обмена, вставлять из буфера обмена и удалять из формы.

Основные общие свойства элементов управления приводим в таблице 4.

Таблица 4

Свойство	Описание
Caption	Надпись, отображаемая при элементе управления
AutoSize	Допустимые значения: True (устанавливает режим автоматического изменения размеров элемента управления так, чтобы на нем полностью помещался текст, присвоенный свойству Caption) и False (в противном случае)
Visible	Допустимые значения: True (элемент управления отображается во время выполнения программы) и False (в противном случае)
Enabled	Допустимые значения: True (пользователь вручную может управлять элементом управления) и False (в противном случае)
Height и Width	Устанавливают геометрические размеры объекта (высоту и ширину)
Left и Top	Устанавливают координаты верхнего левого угла элемента управления, определяющие его местоположение в форме
ControlTipText	Устанавливает текст в окне всплывающей подсказки, связанной с элементом управления. В следующем примере элементу управления CommandButton1 назначен текст, всплывающей подсказки это кнопка: <code>CommandButton1.ControlTipText = "Это кнопка"</code>
BackColor, ForeColor и BorderColor	Устанавливают цвет заднего и переднего плана элемента управления, также его границы
BackStyle	Устанавливает тип заднего фона
BorderStyle	Устанавливает тип границы. Допустимые значения: <code>fmBorderStyleSingle</code> (граница в виде контура); <code>fmBorderStyleNone</code> (граница невидима)
SpecialEffect	Устанавливает тип границы. Отличается от свойства BorderStyle тем, что позволяет установить несколько типов, но одного цвета. BorderStyle позволяет установить только один тип, но различных цветов
Picture (создание картинки)	Внедряет картинку на элемент управления. Например, на поверхности кнопки картинка отображается с помощью следующей инструкции: <code>CommandButton1. _ Picture =LoadPicture("c:\my doc\Круг.bmp")</code> Функция LoadPicture (ПолноеИмяФайла) считывает графическое изображение. Аргумент ПолноеИмяФайла указывает полное имя графического файла

Свойство	Описание
Picture (удаление картинки)	После того как картинка создана на элементе управления, иногда возникает необходимость ее удалить. Это легко достигается присвоением свойству Picture значения LoadPicture("") <code>CommandButton1.Picture = LoadPicture("")</code>

Наиболее часто используемые свойства элементов управления приводим в таблице 5.

Таблица 5

Свойство	Описание
TextBox (поле) используется для ввода текста пользователем или для вывода в него результатов расчетов программ	
Text	Возвращает текст, содержащийся в поле
Multiline	Допустимые значения: True (устанавливает многострочный режим ввода текста в поле) и False (однотрочный режим)
WordWrap	Допустимые значения: True (устанавливает режим автоматического переноса) и False (в противном случае)
Label (надпись) используется для отображения надписей, например, заголовков элементов управления, не имеющих свойства Caption	
Caption	Возвращает текст, отображаемый в надписи
Multiline	Допустимые значения: True (устанавливает многострочный режим ввода) и False (однотрочный режим)
WordWrap	Допустимые значения: True (устанавливает режим автоматического переноса) и False (в противном случае)
CommandButton (кнопка) используется для инициирования выполнения некоторых действий, вызываемых нажатием кнопки, например запуск программы или остановка ее выполнения, печать и т.д.	
Caption	Возвращает текст, отображаемый на кнопке
Cancel	Допустимые значения: True (устанавливаются отменяющие функции для кнопки, т.е. нажатие клавиши <Esc> приводит к тем же результатам, что и нажатие кнопки) и False (в противном случае)
Accelerator	Назначает клавишу, при нажатии на которую одновременно с клавишей <Alt> происходит запуск действий, связанных с кнопкой. Например <code>CommandButton1.Accelerator="C"</code>
Default	Задаёт кнопку по умолчанию, т.е. устанавливает ту кнопку, для которой действия, связанные с ней, будут выполняться при нажатии клавиши <Enter>
Frame (рамка) используется для визуальной группировки элементов управления	
Capture	Отображает надпись рамки
OptionButton (переключатель) позволяет выбрать один из нескольких взаимоисключающих параметров. Переключатели обычно отображаются группами, обеспечивая возможность выбора альтернативного варианта.	
Capture	Надпись, отображаемая рядом с переключателем
Value	Возвращает True, если переключатель выбран и False в противном случае

Пример.

В качестве примера работы с формой составим программу определения значений величин X и Y

Загрузим систему VBA: Alt/F11, выполнив команду **Insert (Вставка) → UserForm**, перенесем на форму командную кнопку CommandButton и выполним на ней ДвЩел ЛевКл мыши.

Появляется окно программного кода с автоматически созданными двумя строками:

```
Private Sub CommandButton1_Click()
```

```
End Sub
```

Запишем внутри них строки нашей программы

```
x=10
```

```
y=5
```

```
While y<50
```

```
x=x+3
```

```
y=y*3
```

```
MsgBox X & " " & Y
```

```
Wend
```

Здесь: языковая конструкция While ... Wend переводится как «до тех пор, пока условие $y < 50$ будет истинным, необходимо выполнять строки программы до Wend, возвращаясь каждый раз на строку $x = x + 3$ и далее повторить по тексту программы».

Для получения ответа выберем команду **Run → Run Sub/UserForm**, либо нажмем клавишу <F5>, либо кнопку панели инструментов Standard, выполняется клик по командной кнопке CommandButton1, расположенной на форме. Ответ – в окне сообщения MsgBox.

Для получения всей таблицы значений X и Y вместо строки

```
MsgBox X & " " & Y
```

необходимо записать

```
Debug.Print X & " " & Y
```