

ЛЕКЦИЯ 5
ОСНОВЫ
ПРОГРАММИРОВАНИЯ НА VBA
(Visual Basic for Application)

Языки программирования

Язык программирования – это совокупность набора символов (алфавит) системы, правил образования (синтаксис) и истолкования конструкций из символов (семантика) для задания алгоритмов с использованием символов естественного языка.

- Языки программирования – это искусственно созданные языки для описания алгоритмов решения задач с помощью ЭВМ. Алгоритм, записанный на языке программирования, называется *программой*.
- Различают языки низкого и высокого уровня. К языкам низкого уровня относятся машинные языки (языки машинных команд данной модели компьютера) и полумашинные языки (ассемблеры).
- Программа, записанная на языке высокого уровня, представляет собой набор операторов.
- К языкам высокого уровня относятся Basic, Pascal, C++, Lisp, Prolog, Visual Basic, Delphi и др. Языки высокого уровня называют также алгоритмическими.

Поскольку машина “понимает” только свой машинный язык, программа на алгоритмическом языке перед выполнением переводится на этот язык с помощью специальной программы – **транслятора**.

Существует **два способа трансляции**:

- **Интерпретация** – метод выполнения программы, при котором инструкция исходной программы переводится и сразу выполняется;
- **Компиляция** – метод выполнения программы, при котором вся программа переводится на машинный язык и затем выполняется.

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА VBA

- ***Visual Basic For Application (VBA)*** – это сочетание одного из самых простых языков программирования и всех вычислительных возможностей табличного процессора Excel в MS Office, а в Libre Office- Calc.
- С помощью VBA можно легко и быстро создавать разнообразные приложения, даже не являясь специалистом в области программирования. VBA содержит графическую среду, позволяющую наглядно конструировать экранные формы и управляющие элементы.
- В результате своей эволюции Visual Basic превратился в объектно-ориентированный язык программирования.
- ***Объекты*** – это естественные детали VBA-ландшафта. Посредством объектов вы получаете доступ к функциональным возможностям VBA-приложения.

С практической точки зрения объект – это просто именованный элемент, имеющий:

- свойства, то есть установки, которые вы можете проверить или изменить;
- методы, то есть действия, которые объект может выполнить, если программа попросит об этом;
- события, то есть возможные для объекта ситуации, на которые он может ответить заранее определенными действиями.
- Прежде чем программировать на VBA, следует запустить табличный процессор Excel MS Office или Calc в Libre Office. Редактор Visual Basic является командным центром для работы в VBA. Вызов редактора Visual Basic осуществляется при помощи команды **Вид – Макросы – Макросы** главного меню Excel.

При составлении программ приходится сталкиваться с такими понятиями, как оператор, процедура, модуль.

- **Оператор** – это наименьшая, способная выполняться единица VBA-кода. Оператор может объявлять или определять переменную, устанавливать параметр компилятора VBA или выполнять какое-либо действие в программе. У допустимого оператора много сходства с законченным предложением – оператор должен содержать правильный набор “частей речи”, иначе это не оператор вообще.
- **Процедура** – это наименьшая единица программного кода, на которую можно ссылаться по имени. Это также наименьшая единица программного кода, которая может выполняться независимо.
- VBA распознает два главных типа процедур: **Sub** и **Function**. Любая процедура содержит один или более операторов, помещенных между двумя специальными операторами: объявлением процедуры в начале и оператором завершения процедуры в конце (**End Sub** или **End Function**).
- **Модуль** – это именованная единица, состоящая из одной или нескольких процедур, а также объявлений, относящихся ко всем процедурам в модуле.

Алфавит языка VBA

- Для записи операторов, функций, имен, арифметических выражений используются:
- все прописные и строчные буквы латинского алфавита;
- арабские цифры;
- специальные знаки ! & ' \$? , . { } () [] = - + _ ^ % / ~ < > : ; .

Имя переменной

Имена переменных выбираются по следующим правилам:

- имена должны начинаться с буквы, но не с цифры;
- в имени допускается знак подчеркивания, все остальные знаки запрещены;
- не допускается, чтобы имена имели пробелы;
- длина имени не должна превышать 255 символов;
- имя не должно совпадать ни с каким ключевым словом, функцией или оператором VBA;
- все процедуры и модули должны иметь разные имена.

В VBA прописные и строчные буквы не различаются, но введенные прописные буквы сохраняются.

Примеры допустимых и недопустимых имен переменных приведены в таблице.

Допустимые имена	Недопустимые имена
A	Имя более 255 символов
Go4Ln	1p
SUMMA	P 1
P1	W?
S_1	Sub
Сумма	For

Оператор описания переменных

Каждую переменную перед использованием следует объявить. Для этого в VBA используется оператор описания переменных **Dim**, который описывает переменные и выделяет для них оперативную память, причем описывать нужно каждую переменную отдельно. Оператор описания переменных выглядит следующим образом:

Dim Имя переменной **As** Тип

Пример

Dim a As Integer, b As Long, c As Byte

Dim d As String, f As String

Если в разделе объявлений модуля (в начале модуля) поместить оператор **Option Explicit**, то при попытке использования предварительно не объявленной переменной VBA будет сообщать об ошибке.

Основные типы данных VBA

Тип данных	Содержимое переменной	Диапазон допустимых значений
Boolean	Логический тип данных	Истина (-1) или Ложь (0)
Byte	Достаточно малое целое число	От 0 до 255
Integer	Не слишком большое целое число	От -32768 до 32767
Long	Большое целое число	От -2147483648 до 2147483647
Single	Вещественное число Значение одинарной точности с плавающей запятой	От -3,402823E38 до -1,401298E-45 для отрицательных значений и от 1,401298E-45 до 3,402823E38 для положительных значений
Double	Вещественное число Значение двойной точности с плавающей запятой	От -1,79769313486231E308 до 1,79769313486232E308
String	Последовательность любых символов	От 0 до 65400 символов
Пользовательский Type	Группа переменных, используемых вместе как единое целое	Определяется пользователем

Знаки операций

В VBA операция представляет собой либо специальный символ, либо ключевое слово в выражении, которое комбинирует два значения с целью получения нового результата. VBA разделяет операции на три главные категории: арифметические, логические и операции сравнения.

Если выражение содержит знаки операций из двух или более категорий, то VBA выполняет операции из разных категорий в следующем порядке:

- арифметические;
- операции сравнения;
- логические операции.

Чтобы изменить последовательность выполнения операций, в выражении используют скобки. Внутри каждой категории операций тоже имеются правила порядка выполнения операций (табл. 13).

Порядок выполнения операций в VBA

Знаки операций	Операции
Арифметические	
^	Возведение в степень
-	Отрицание
* или /	Умножение или деление
\	Вычисление целой части от деления
Mod	Вычисление остатка от деления
+ или -	Сложение или вычитание
Сравнение	
=	Равенство
<>	Неравенство (не равно)
<	Меньше
>	Больше
<=	Меньше или равно
>=	Больше или равно
Логические	
Not	Логическое НЕ
And	Логическое И
Or	Логическое ИЛИ

Математические функции

В VBA используются математические функции, знакомые по работе в электронных таблицах.

Математические функции VBA

Функция	Значение
Abs(число)	Абсолютное значение числа (модуль)
Atn(число)	Арктангенс числа
Cos(число)	Косинус числа
Exp(число)	Число e в степени, равной заданному числу (e^x)
Fix(число)	Целая часть числа
Int(число)	Целая часть числа. Функции Int и Fix по-разному действуют только на отрицательные числа: Int – возвращает ближайшее меньшее целое, а Fix – просто отбрасывает дробную часть числа
Log(число)	Натуральный логарифм числа
Rnd	Случайное число от 0 до 1
Sin(число)	Синус числа
Sqr(число)	Квадратный корень из числа
Tan(число)	Тангенс числа

Оператор присваивания

Оператор присваивания приписывает переменным или свойствам объектов конкретные значения. Такой оператор всегда состоит из трех частей: имени переменной или свойства, знака равенства и выражения, задающего нужное значение.

Имя переменной = Выражение

Пример

$A = 2$

$C = A * 3 + 4$

Ввод данных

Для составления программ необходимо в память ЭВМ ввести данные. Вводить данные можно тремя способами.

1. При помощи **оператора присваивания**.

Пример

$a = 2$

$b = 4.5$

2. При помощи **оператора ввода InputBox(“Сообщение”)**.

Данный оператор выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа String, содержащее текст, введенный в поле.

Для преобразования символа в число будем использовать функцию **Val (Строка)**, которая возвращает число, содержащееся в строке, как числовое значение соответствующего типа.

Если мы запишем $A = \text{Val}(\text{InputBox}(\text{“Введите } A\text{”}))$ и в поле ввода введем число 2, то переменной A присвоится значение числа 2.

Пример

A = InputBox (“Введите A”)

На экране появится диалоговое окно (рис. 1).

Если оператор оставить в таком виде, то переменной A присвоится значение символа “2”, а не числа 2.

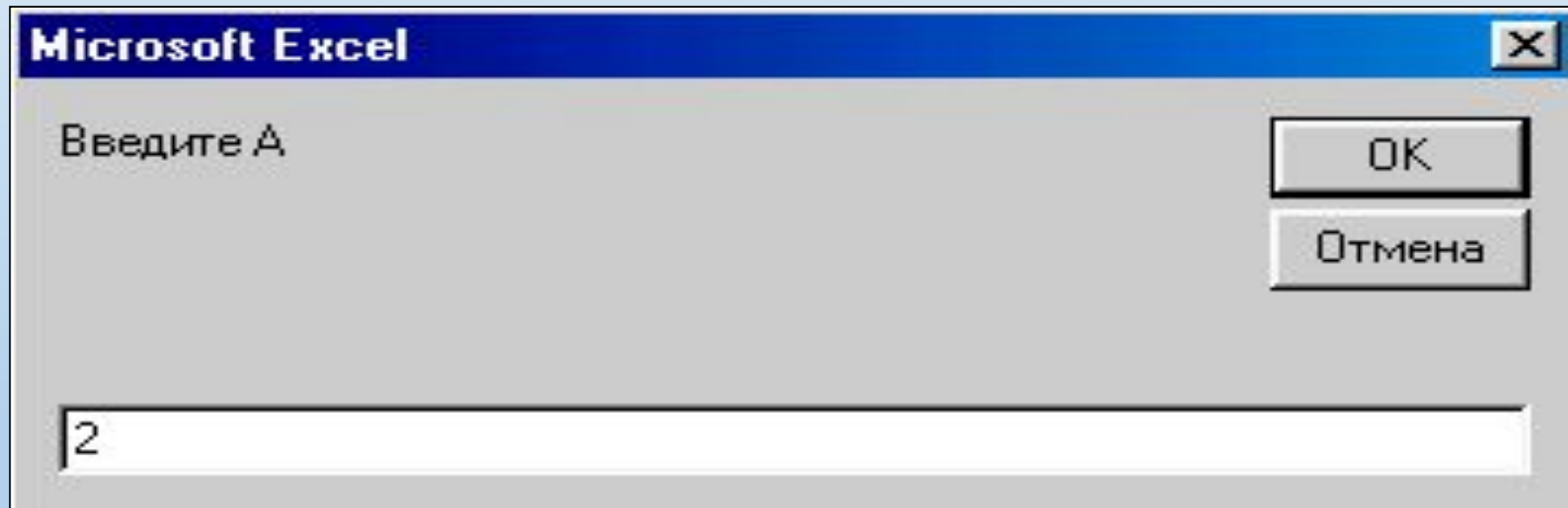


Рис. 1. Стандартное окно ввода с клавиатуры

3. Считывание данных с листа рабочей книги Excel

При составлении программы VBA, встроенном в электронных таблицах, есть возможность использовать ячейки для считывания или записи данных при помощи оператора

Cells (номер строки, номер столбца) в адресе соотв. ячейки
(англ. клетка, ячейка)

Пример

`A = Cells(1, 3)`

После выполнения этого оператора переменной *A* будет присвоено значение, которое хранится в ячейке, находящейся в первой строке (первая цифра) и в третьем столбце *C* (вторая цифра), т.е. в ячейке *C1* электронной таблицы.

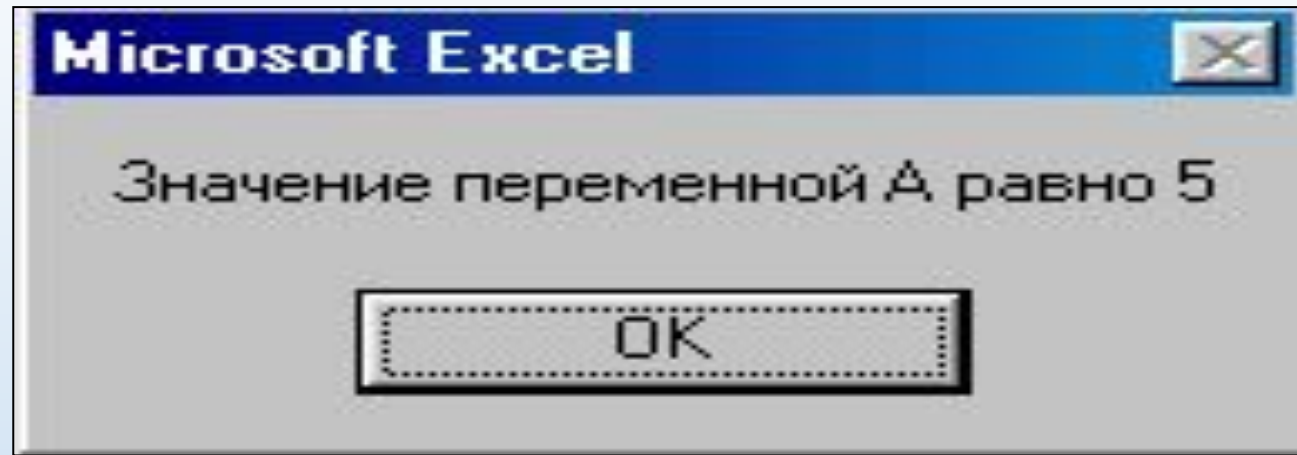


Рис. 2. Стандартное окно вывода на экран

2. Можно **вывести данные на рабочий лист**. Для этого нужно записать следующее:

Cells(строка, столбец) = значение

Пример

Cells(3, 2) = K
переменной K.

‘ Выведет в ячейку B3 значение

Задача 1. Заданы два числа. Вычислить их сумму, произведение и частное.

Программный код

Option Explicit

Sub PR1() ' заголовок процедуры

Dim a As Integer, b As Integer, s As Integer, p As Integer ' описание переменных целого типа

Dim ch As Double ' описание переменных дробного типа

a = Val(InputBox("Введите A")) ' ввод первого числа

b = Val(InputBox("Введите B")) ' ввод второго числа

s = a + b ' вычисление суммы

MsgBox ("сумма=" & s) ' вывод суммы на экран

Cells(1,1)= "сумма=" & s) ' вывод суммы в ячейку A1

p = a * b ' вычисление произведения

MsgBox ("произведение=" & p) ' вывод произведения на экран

Cells(2,2)= ("произведение=" & p) ' вывод произведения в ячейку B2

ch = a / b ' вычисление частного

MsgBox ("частное=" & ch) ' вывод частного на экран

Cells(3,2)= "частное=" & ch ' вывод частного в ячейку B3

End Sub

- В программе можно писать комментарии – пояснения к вашей программе, которые предназначены для пользователя, а не для компьютера. Комментарии начинаются с символа ' (апостроф). Все, что написано в строке программного кода справа от апострофа, считается комментарием, в программном коде в редакторе VBA они окрашиваются в зеленый цвет.
- Если вы хотите разместить несколько операторов в одну строку, то для этого необходимо записать эти операторы через двоеточие.

Задача 2. Заданы целые числа a, b, c . Вычислить значение выражения

$$Y = \frac{\sqrt{a + b} + b^2}{(a + b + c)^3} \cdot \operatorname{tga}.$$

Программный код

```
Option Explicit
```

```
Sub PR2()
```

```
Dim a As Integer, b As Integer, c As Integer ' описание переменных
```

```
Dim y As Double
```

```
a = Val(TextBox("Введите A")) ' ввод a
```

```
b = Val(TextBox("Введите B")) ' ввод b
```

```
c = Val(TextBox("Введите C")) ' ввод c
```

```
y = (Sqr(a + b) + b ^ 2) / (a + b + c) ^ 3 * Tan(a) ' вычисление
```

```
MsgBox ("y=" & y) ' вывод результата
```

```
End Sub
```

Условный оператор

Условный оператор, или оператор условного перехода, служит для организации процесса вычислений в зависимости от какого-либо условия. Общий вид *полной формы* условного оператора:

If Условие **Then** Оператор1 **Else** Оператор2

Условие – это выражение логического типа.

Оно может быть простым или сложным.

Если в условном операторе <Условие> истинно, то выполняется <Оператор1>, в противном случае выполняется <Оператор2>.

Для записи условий могут быть использованы знаки логических отношений, используемых в электронных таблицах, они представлены в таблице 2

Таблица 2

Знаки сравнения

Операция	Название	Пример выражения
=	Равно	$A=B$
<>	Не равно	$A<>B$
>	Больше	$A>B$
<	Меньше	$A<B$
>=	Больше или равно	$A>=B$
<=	Меньше или равно	$A<=B$

Пример

$A \geq 10$

$B < 5$

$C \neq 0$

Сложные условия образуются из простых путем применения логических операций и круглых скобок.

Пример

$A > 10 \text{ And } A < 20$

$(B > 4 \text{ Or } B < 2) \text{ And } A > 5$

Операция	Название	Пример выражения
Not	Логическое отрицание	Not A
And	Логическое умножение И	A And B
Or	Логическое сложение ИЛИ	A Or B

В условном операторе допустимо использование блока операторов.

В этом случае условный оператор (полная форма) имеет вид

If Условие **Then**

 Блок операторов 1

Else

 Блок операторов 2

End If

Ветвь **Else** в условном операторе является необязательной. Такая форма условного оператора называется *неполной* и выглядит следующим образом:

1. **If** Условие **Then** Оператор

2. **If** Условие **Then**

 Блок операторов

End If

В условном операторе после слова **Then** или **Else** можно разместить несколько операторов в одну строку. В этом случае они должны быть разделены двоеточием.

Пример

If A > 10 Then A = A + 1 : B = B + A Else A = A + 3 : B = B*3

If X > 0 Then y = log(X) : MsgBox(y)

Задача 1. Задано целое число x . Вычислить значение функции

$$Y = \begin{cases} \sqrt{x}, & \text{если } x > 0; \\ x^2, & \text{если } x \leq 0. \end{cases}$$

Программный код

```
Option Explicit
```

```
Sub PR3()
```

```
Dim x As Integer
```

```
Dim y As Double
```

```
x = Val(InputBox("Введите x"))
```

```
If x > 0 Then y = Sqr(x) Else y = x ^ 2
```

```
MsgBox ("y=" & y)
```

```
End Sub
```

Задача 2. Задано действительное число x . Вычислить значение функции

$$y = \frac{1}{\sqrt{x}}$$

Программный код

```
Option Explicit
Sub PR4()
Dim x As Double
Dim y As Double
x = Val(InputBox("Введите x"))
If x > 0 Then
    y = 1 / Sqr(x)
    MsgBox ("y=" & y)
Else
    MsgBox ("Решения нет")
End If
End Sub
```

Задача 3. Задано три действительных числа x , y , z . Вычислить самое большое из них.

Программный код

```
Option Explicit
```

```
Sub PR5()
```

```
Dim x As Double, y As Double, z As Double, max As Double
```

```
x = Val(InputBox("Введите x"))
```

```
y = Val(InputBox("Введите y"))
```

```
z = Val(InputBox("Введите z"))
```

```
If (x > y) And (x > z) Then
```

```
max = x
```

```
else
```

```
If (y > x) And (y > z) Then
```

```
max = y
```

```
else
```

```
If (z > x) And (z > y) Then
```

```
max = z
```

```
End If
```

```
End If
```

```
End If
```

```
MsgBox ("Максимум=" & max)
```

```
End Sub
```

Оператор выбора **Select Case**

Оператор выбора **Select Case** удобно использовать, когда в зависимости от значения некоторого выражения, имеющего конечное множество допустимых значений, необходимо выполнить разные действия.

Select Case Тестируемое выражение

Case Условие выбора 1

Блок операторов 1

Case Условие выбора 2

Блок операторов 2

.....

Case Условие выбора n

Блок операторов n

Case Else

Блок операторов

End Select

После каждого оператора **Case** может находиться произвольное количество других операторов, и все они будут выполняться, если условие оператора **Case** истинно.

Слово **Is**, используемое в программе, является ключевым словом VBA, обозначающим тестируемое выражение в операторе **Case**.

В операторе **Case** допустимо использовать составные условия, подобно тому, как это делается в условном операторе.

Пример

Case 5, 6, 9 to 10, 13, 14, Is>=16

В этом операторе проверяется, принадлежит ли тестируемое выражение отрезку от 9 до 10, или равняется одному из значений: 5, 6, 13, 14, или оно не меньше 16.

Задача. Вычислить размер комиссионных в зависимости от объема продаж.

Процент комиссионных приведен в таблице

Программный код

```
Option Explicit
```

```
Sub PR6()
```

```
Dim opr As Double, prem As Double
```

```
opr = Val(InputBox("Введите объем продаж"))
```

```
Select Case opr
```

```
Case 0 To 9999
```

```
    prem = 0.08 * opr
```

```
Case 10000 To 39999
```

```
    prem = 0.1 * opr
```

```
Case Is >= 40000
```

```
    prem = 0.14 * opr
```

```
End Select
```

```
MsgBox ("Комиссионные=" & prem)
```

```
End Sub
```

Объем продаж, тыс. руб.	Комиссионные, %
От 0 до 9999	8
От 10000 до 39999	10
40000 и более	14