

# Тема 2. Программирование на VBA

---



# Вопросы темы

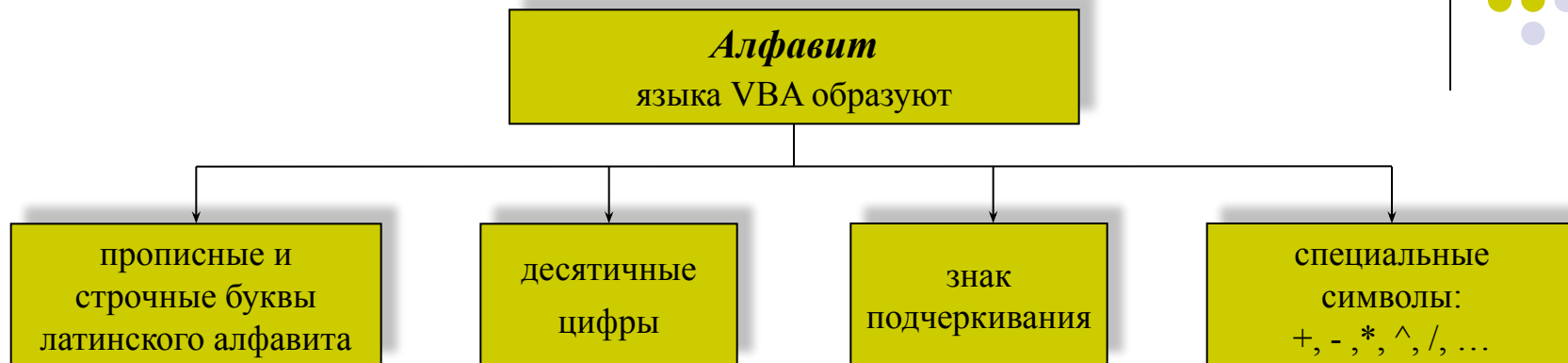


1. Базовые элементы языка VBA .
2. Организация ввода – вывода данных .
3. Программирование линейных алгоритмов.
4. Программирование разветвляющихся алгоритмов
5. Программирование циклических алгоритмов .

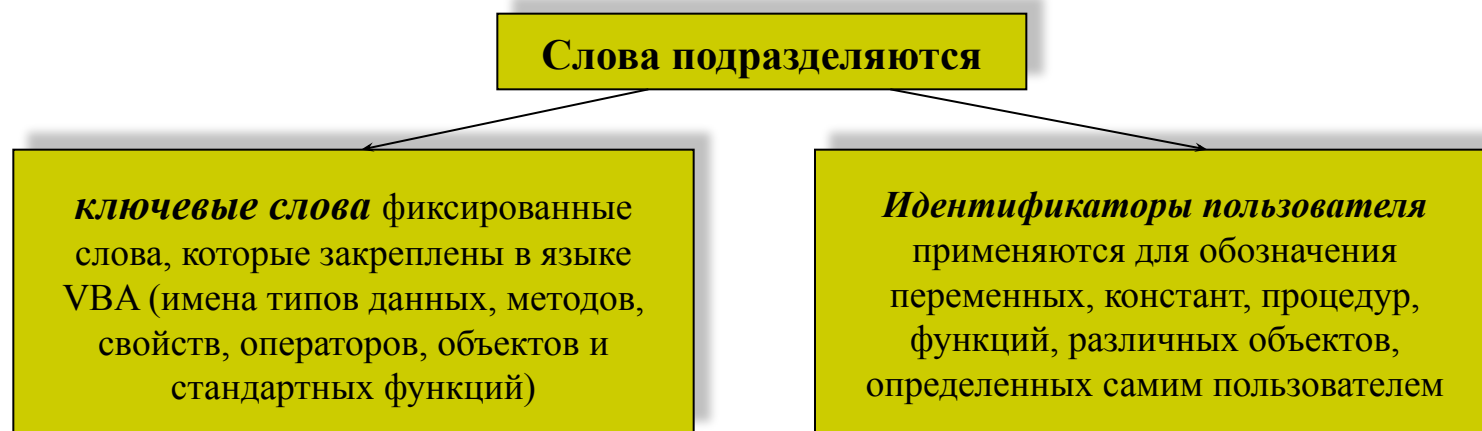
# Вопрос 1. Базовые элементы языка VBA .



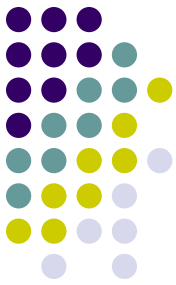
## 1.1. Алфавит, словарь, идентификаторы



**Слова** – неделимые последовательности знаков алфавита, отделенные друг от друга разделителями, и несущие определенный смысл в программе.



## 1.2. Типы данных



### ТИПЫ ДАННЫХ определяют

множество значений,  
которые могут принимать  
объекты программ

совокупность операций,  
допустимых над этими  
объектами

объем выделяемой памяти и  
форму представления данных  
в ней

### Типы данных в VBA

Тип данных	Описание	Размер, байт	Диапазон
<i>Byte</i>	Байт	1	От 0 до 255
<i>Boolean</i>	Логический	2	<i>True</i> или <i>False</i>
<i>Integer</i>	Целое	2	От -32 768 до 32 767
<i>Long</i>	Длинное целое	4	От -2 147 473 648 до 2 147 483 647
<i>Single</i>	С плавающей точкой обычной точности	4	От -3,402823E38 до - 1,401298E-45 для отрицательных чисел От 1,401298E-45 до 3,402823E38 для положительных чисел
<i>Currency</i>	Денежный	8	От -922337230685477,5808 до 922337203685477,5807
<i>Date</i>	Даты и время	8	С 1 января 100 года до 31 декабря 9999 года
<i>Variant</i>	Числовые подтипы	16 байт + N байт (согласно типу)	Переменная любого числового типа

## 1.3. Переменные



**Переменными** называются элементы данных, значения которых при выполнении программы могут принимать различные значения в соответствии с указанным типом.



Синтаксис описания переменных:

*Dim* имяПеременной [*As тип*] [, имяПеременной [*As тип*] ...

Например,

*Dim A As Integer*

*Dim B As Single, C As Boolean*

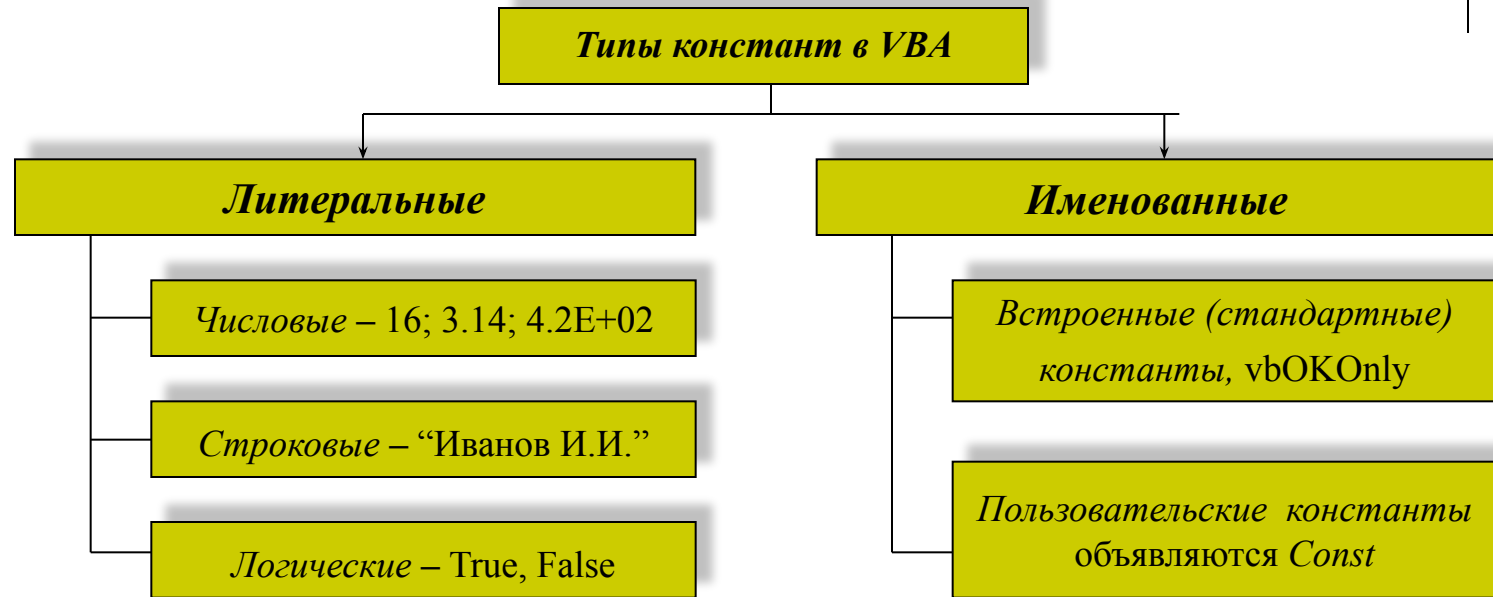
*Dim D*

Если тип опущен, то он равен  
***Variant***

## 1.4. Константы



**Константами** называются элементы данных, значения которых определены при их описании и в процессе выполнения программ не изменяются.



Синтаксис инструкции *Const*:

*Const* ИмяКонстанты [*As тип*] = выражение

Например,

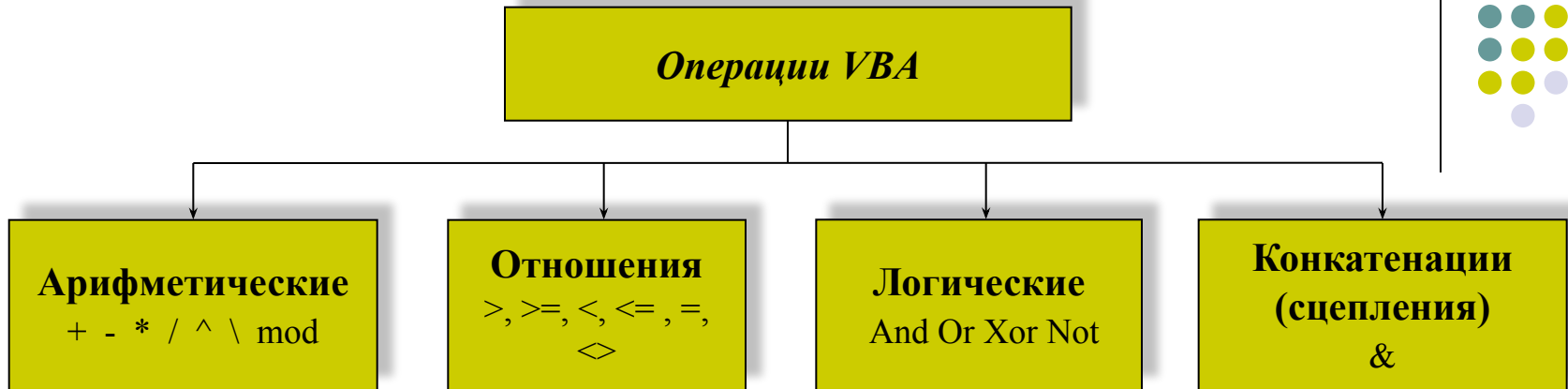
*Const Year As Integer=2003*

*Const Plan=129*

*Const NameGrup = “MT - 21”*

*Const Flag = True*

## 1.5. Операции и операторы VBA



## 1.6. Встроенные функции VBA

Ключевое слово, синтаксис	Функция
<i>Atn (x)</i>	<i>Арктангенс числа</i>
<i>Cos (x)</i>	<i>Косинус угла</i>
<i>Sin (x)</i>	<i>Синус угла</i>
<i>Tan (x)</i>	<i>Тангенс угла</i>
<i>Exp(x)</i>	<i>Возведение числа e в степень x</i>
<i>Log (x)</i>	<i>Логарифм натуральный</i>
<i>Sqr (x)</i>	<i>Квадратный корень числа</i>
<i>Abs (x)</i>	<i>Абсолютное значение числа</i>

## 1.7. Приоритеты операций



Приоритет	Оператор (операция)
1	Вызов функции и скобки
2	$\wedge$ (возведение в степень)
3	- (смена знака)
4	*, / (умножение, деление)
5	$\backslash$ (целочисленное деление)
6	<i>Mod</i> (остаток от целочисленного деления)
7	+, - (сложение, вычитание)
8	&, + (слияние строк)
9	=, <, >, <=, >= (операции отношения)
10	<i>Not</i> (логическое отрицание)
11	<i>And</i> (логическое И)
12	<i>Or</i> (логическое ИЛИ)
13	<i>Xor</i> (исключающее ИЛИ)

## 1.8. Оператор присваивания

**Оператор присваивания** осуществляет присваивание результата вычисления некоторого выражения переменной или константе.

Синтаксис:

*<идентификатор> = <выражение>*



## 1.9. Структура программы

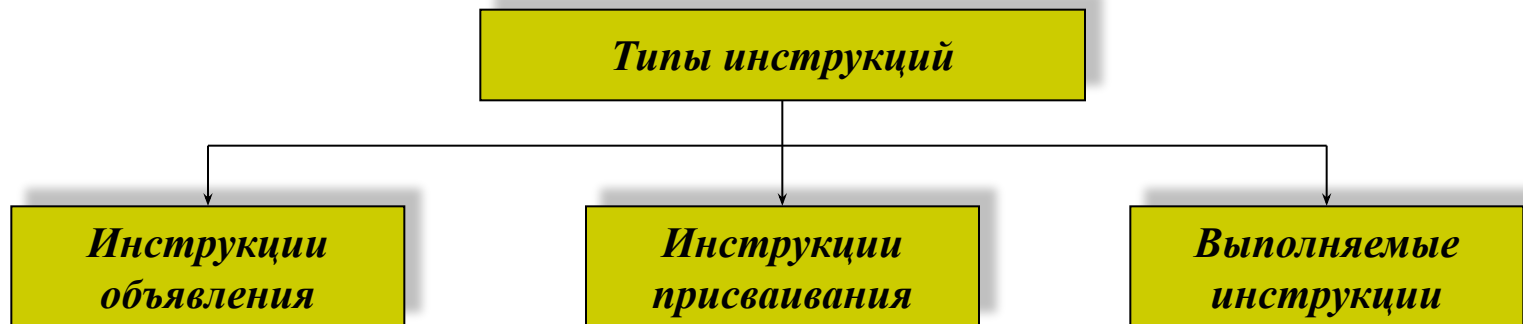


**Проект** – это совокупность нескольких элементов. Основными из них являются:

- приложение (в нашем случае это Excel);
- среда разработки VBA (редактор VBA);
- совокупность модулей, в которых записывается программный код (совокупность программных единиц – процедур проекта).

**Процедура** – представляет собой последовательность инструкций.

**Инструкция** – это синтаксически полный компонент программы, представляющий собой операцию, описание или определение.



Синтаксис объявления процедуры *Sub* имеет следующий вид:

```
Sub <ИмяПроцедуры> ([<Список параметров>])  
    <Тело процедуры>  
End Sub
```

## 1.10. Правила оформления кода



Рекомендаций при программировании на VBA:

1. В соответствии с правилами хорошего стиля программирования **размещать инструкции описания в начале процедуры.**
2. **Использовать комментарии**, т.е. пояснения к фрагменту текста процедуры.

*программирование линейных алгоритмов*

3. **Перенос строк кода.** Длинные инструкции можно размещать в нескольких строках. Для этого используют *признак продолжения строки*, состоящей из двух символов: пробела и подчеркивания ( ).

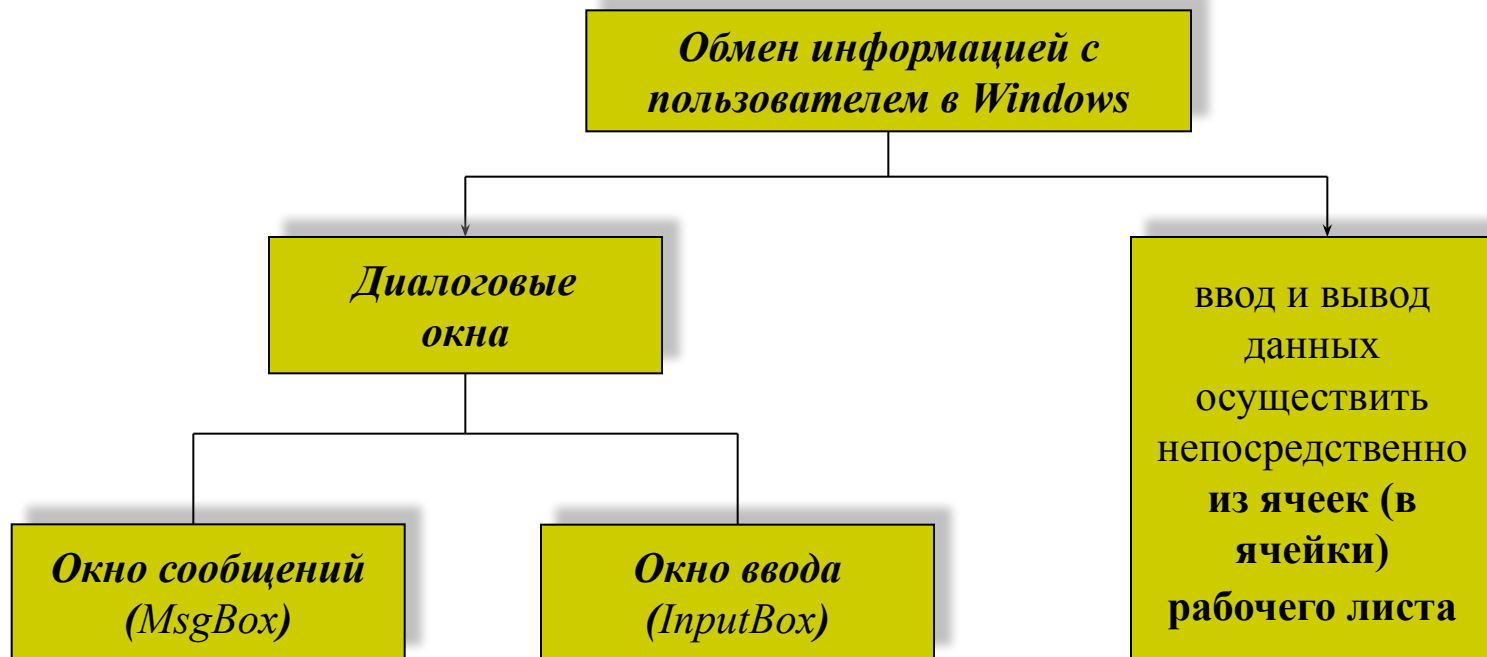
$$y = 2 * Sqr(x+3) - (log(x^2) + 2) \_ \\ / (sin(x-1))$$

4. **Расположение нескольких операторов в одной строке.** Использование знака двоеточия (:) позволяет разместить несколько операторов (инструкций) на одной строке.

$$x=x+a: y=x-b$$

5. **Выделение логических уровней кода отступами.** (отступы делаются пробелами)

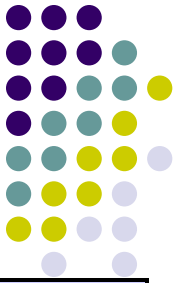
## Вопрос 2. Организация ввода – вывода данных .



## 2.1. ОКНО СООБЩЕНИЯ. Стандартная процедура *MsgBox*

Сокращенный синтаксис процедуры *MsgBox*:

*MsgBox* сообщение [, атрибуты] [, заголовок]

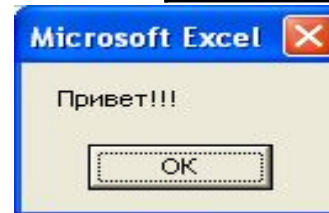


Значения параметра <i>Buttons</i> процедуры и функции <i>MsgBox</i> , определяющие отображаемые кнопки в окне сообщения		
Константа	Значение	Отображаемые кнопки
<i>vbOKOnly</i>	0	<b>ОК</b>
<i>vbOKCancel</i>	1	<b>ОК, Отмена</b>
<i>vbAbortReplyIgnore</i>	2	<b>Стоп, Повтор, Пропустить</b>
<i>vbYesNoCancel</i>	3	<b>Да, Нет, Отмена</b>
<i>vbYesNo</i>	4	<b>Да, Нет</b>
<i>vbReplyCancel</i>	5	<b>Повтор, Отмена</b>

Значения параметра <i>Buttons</i> процедуры и функции <i>MsgBox</i> , определяющие отображаемые информационные значки в окне сообщения		
Константа	Значение	Значок сообщения
<i>vbCritical</i>	16	
<i>vbQuestion</i>	32	
<i>vbExclamation</i>	48	
<i>vbInformation</i>	64	

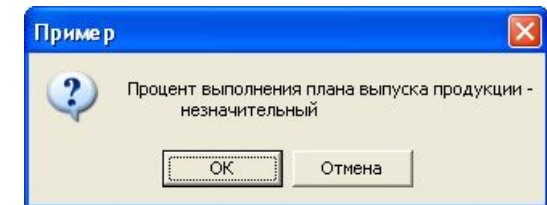
### Например

```
Sub Pr2_1()
    ' Простое информационное сообщение
    MsgBox "Привет!!!"
End Sub
```



### Например

```
Sub Pr2_3()
    ' Вывод сообщения с двумя кнопками ОК и Отмена в сочетании с
    ' предупреждающим информационным значком. Кнопка ОК является основной
    MsgBox "Процент выполнения плана выпуска продукции -" & _
        Chr(13) & "    незначительный", vbOKCancel + _
        vbQuestion, "Пример"
End Sub
```



## 2.2. ОКНО ВВОДА. Стандартная функция *InputBox()*



Сокращенный синтаксис:

***InputBox*** (*сообщение* [, *заголовок*] [, *умолчание*])

*сообщение* – Строковое выражение, отображаемое как сообщение в диалоговом окне.

Может содержать несколько строк. Максимальная длина строки 1024 символа.

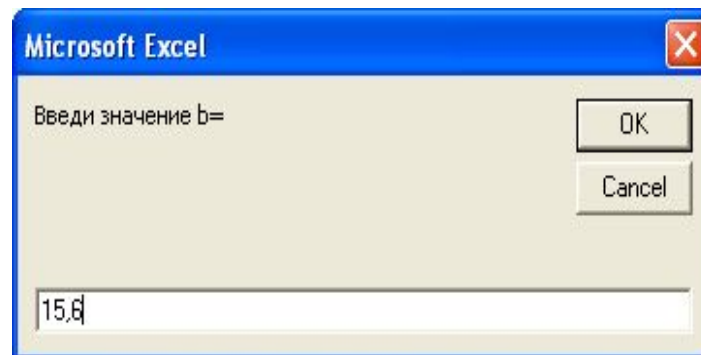
*заголовок* - Строковое выражение, отображаемое в строке заголовка диалогового окна.

Если этот параметр опущен, то в строку заголовка помещается имя приложения.

*умолчание* – Строковое выражение, отображаемое в поле ввода. Используется по умолчанию, если пользователь не введет другую строку. Если параметр опущен, то поле ввода изображается пустым.

Например,

$b = \mathbf{InputBox}$ ("Введи значение b=", "Пример 3.5")



## 2.3. Ввод/вывод данных с/на рабочий лист Excel



Например,

$x = \text{Cells}(5,3).Value$

или

$x = \text{Cells}(5,3)$

или

$x = \text{Range}("C5")$

Переменной  $x$  присваивается значение ячейки C5

Например,

*Sub Pr2\_4 ()*

*Dim a As Byte, b As Byte, x As Integer, y As Single*

*a = Cells (2, 3): b = Cells (3, 3): x = Cells (4, 3)*

*y = (x + 3) ^ 2 + (2 \* a - 3 \* b) / (x ^ 2 - 2.8)*

*Cells(6, 1) = "Значение функции:"*

*Cells (7, 3) = y*

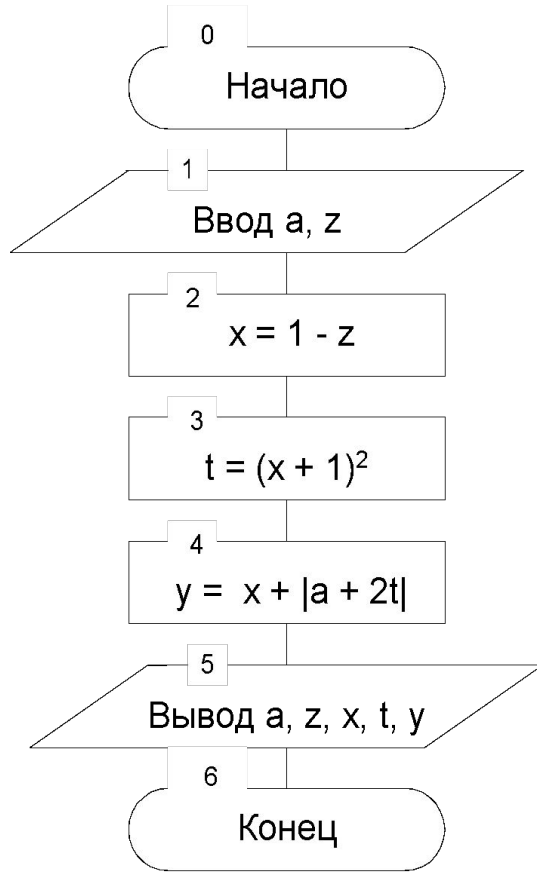
*End Sub*

	A	B	C	D	E	F
1	Исходные данные:					
2		a=	13			
3		b=	57			
4		x=	108			
5						
6	Значение функции:					
7			12320,99			
8						
9						

## Вопрос 3. Программирование линейных алгоритмов.



**Пример** Вычислить значение функции  $y = x + |a+2t|$ ,  
где  $x = 1 - z$ ,  $t = (x + 1)^2$



```
Sub Pr3_1 ()
```

```
Dim y As Single, x As Single, a As Single
```

```
Dim t As Single, z As Single
```

```
a = InputBox("Введи значение a")
```

```
z = InputBox("Введи значение z")
```

```
x = 1 - z
```

```
t = (x + 1) ^ 2
```

```
y = x + Abs (a + 2 * t)
```

```
MsgBox "Исходные данные:" & Chr(13) & _
```

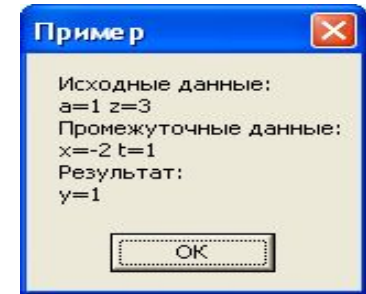
```
"a=" & a & " z=" & z & Chr(13) & _
```

```
"Промежуточные данные:" & Chr(13) & _
```

```
"x=" & x & " t=" & t & Chr(13) & _
```

```
"Результат:" & Chr(13) & "y=" & y, "Пример"
```

```
End Sub
```

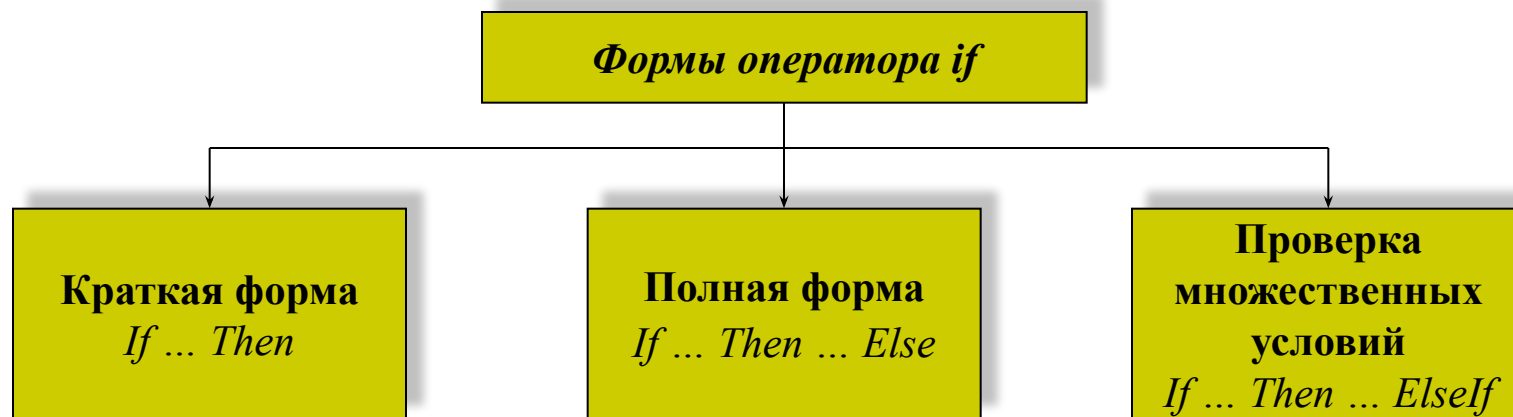


# Вопрос 4. Программирование разветвляющихся алгоритмов.



## 4.1. Условный оператор *If*

Оператор условного перехода (*условный оператор*) *If* реализует разветвляющиеся процессы



### Примеры

*If x > 5 Then y = 3 \* x + 1.5*

*If a > 5 Then  
    b = b + a  
Else  
    b = b - a  
End If*

*If g > 10 Then  
    z = 5  
ElseIf g < 5 Then  
    z = 1  
End If*



## 4.2. Оператор выбора *Select Case*

Для реализации выбора нескольких альтернатив можно использовать оператор *Select Case*

**Например**, реализовать вывод сообщения об оценке по полученным балам

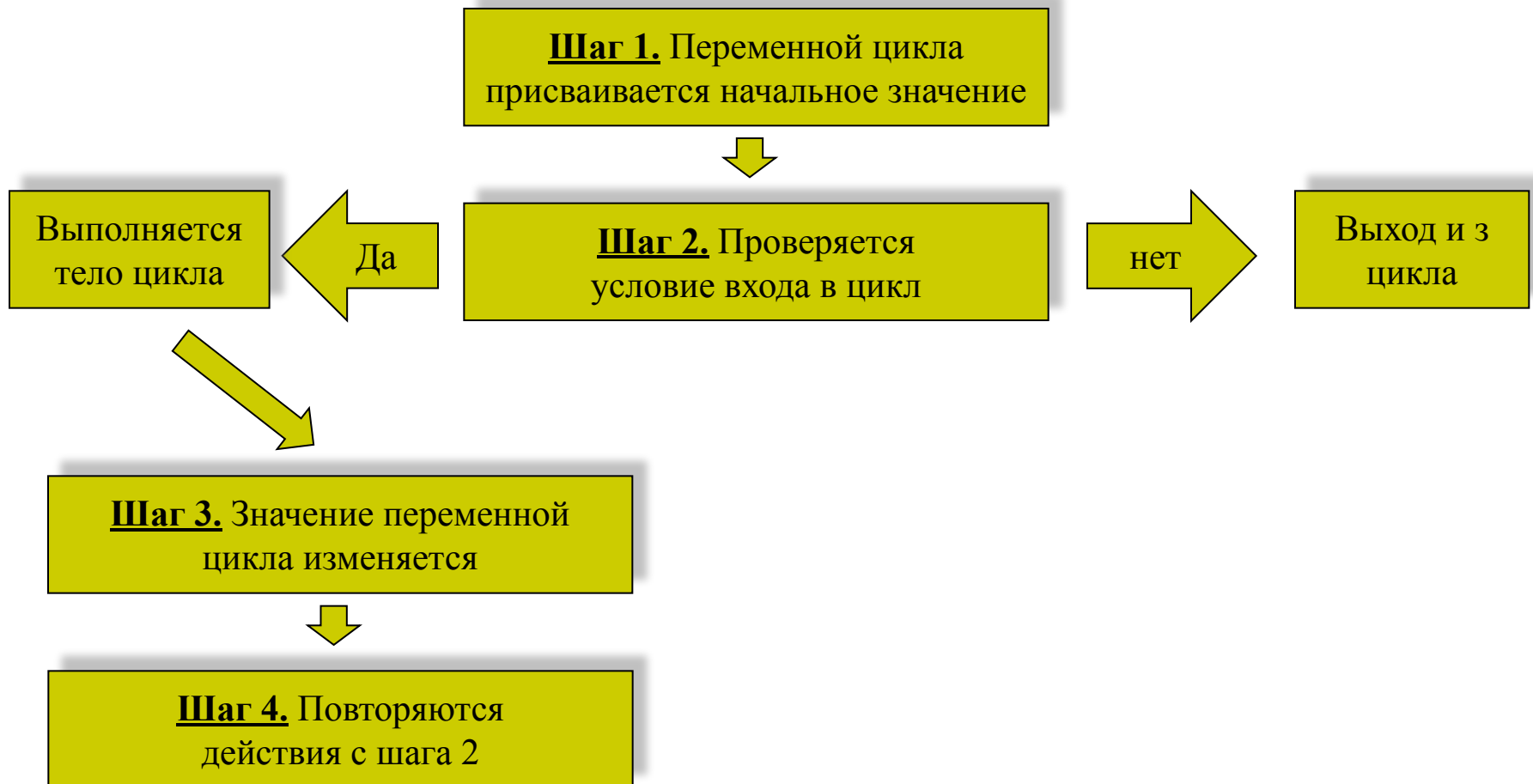
```
Select Case Range("A3")
  Case Is >=90
    MsgBox "Вы получаете оценку 5"
  Case 75 To 89
    MsgBox "Вы получаете оценку 4"
  Case 60 To 74
    MsgBox "Вы получаете оценку 3"
  Case 35 To 59
    MsgBox "Вы получаете оценку 2"
  Case Else
    MsgBox "Вас необходимо отчислить"
End Select
```



## Вопрос 5. Программирование циклических алгоритмов.



Принцип работы циклических процессов



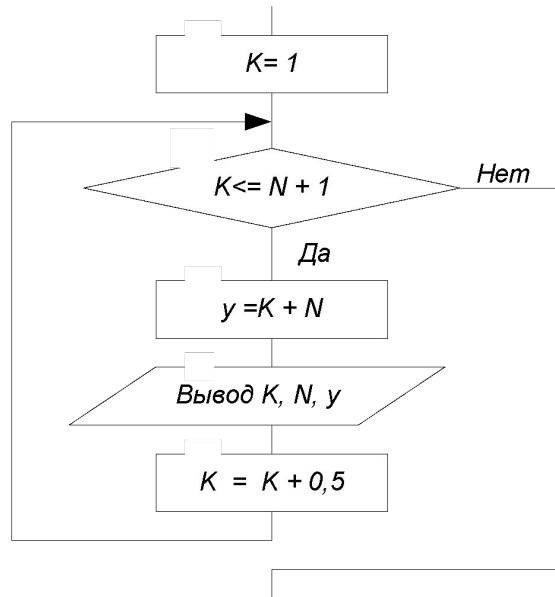
## 5.1. Оператор цикла *While...Wend*



Синтаксис:

```
While <условие>  
    <Повторяющиеся действия>  
Wend
```

### Пример



$K = 1$

*While*  $k \leq N + 1$

$y = k + N$

`MsgBox "При k=" & k & " и N=", & N_`

`& " значение y=" & y`

$k = k + 0.5$

*Wend*

## 5.2. Оператор цикла *Do...Until*



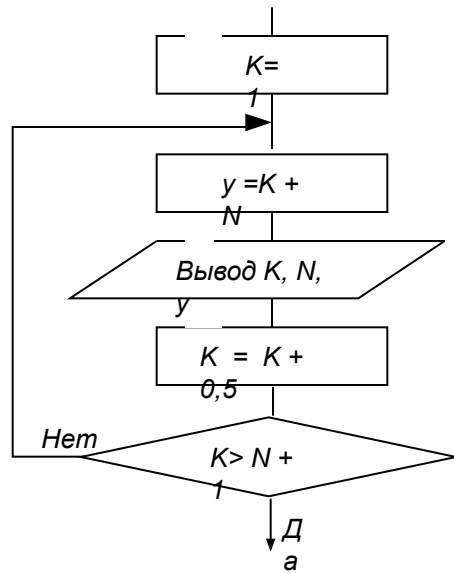
Синтаксис:

*Do*

*<Повторяющиеся действия>*

*Loop Until <условие>*

### Пример



$k=1$

*Do*

$y = k + N$

`MsgBox "При k=" & k & " и N=", & N_`

`& " значение y=" & y`

$k = k + 0.5$

*Loop Until  $k > N + 1$*

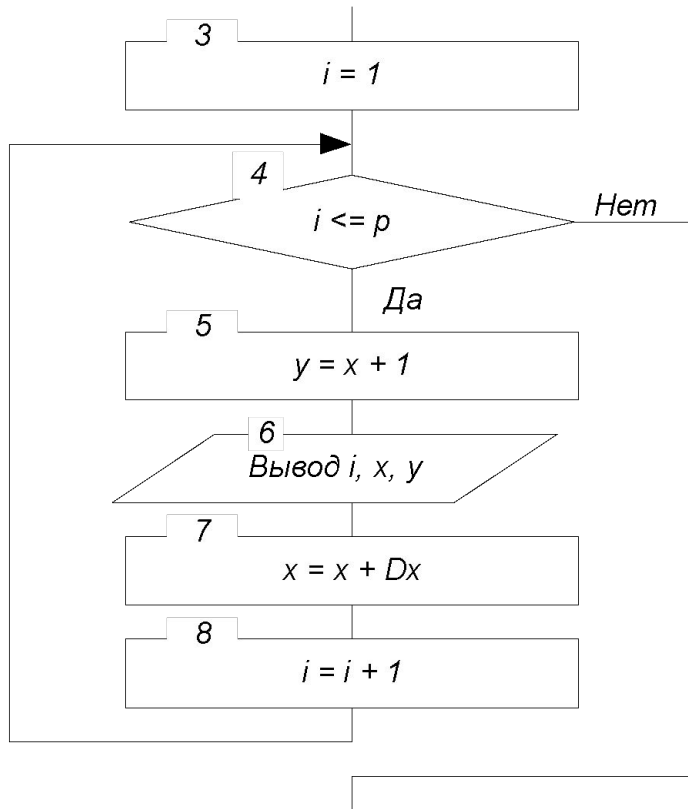
### 5.3. Оператор цикла *For...Next*



Синтаксис:

*For* Счетчик=НачЗначение *To* КонЗначение [*Step* Приращение]  
<Повторяющиеся действия>  
*Next* [Счетчик]

#### Пример



*For*  $i=1$  *To*  $p$

$y = x + 1$

*Cells*  $(1+i, 1)=i$ : *Cells*  $(1+i, 2)=x$ : *Cells*  $(1+i, 3)=y$

$x = x + Dx$

*Next*