

Лекция

Введение в язык VBA

Проект и его составляющие

- **Проект** – файл, в котором находится программа VBA (книга Excel).
- **Модуль** – именованная область внутри проекта, содержит **подпрограммы** и другие элементы программы.
- **Программа**
- **Подпрограмма** – именованная часть программы.
- **Процедура** – часть программы, имеющая имя и которую можно вызвать из другой подпрограммы.
- **Функция** – часть программы, предназначенная, как правило, для вычисления значения выражения;

Переменные и константы

- **Переменная** – область в оперативной памяти компьютера, которая имеет имя; способна содержать в себе информацию определенного типа.
- **Константа** - область в оперативной памяти компьютера, которая имеет имя, и содержит в себе информацию, которую нельзя изменить во время работы программы.

Объявление переменных и констант

Формат объявления переменных:

Dim *имя_переменной1* [**as** тип], *имя_переменной2* [**as** тип]

Если служебное слово **as** с типом опущены, то переменная имеет тип **Variant**.

Формат описания констант:

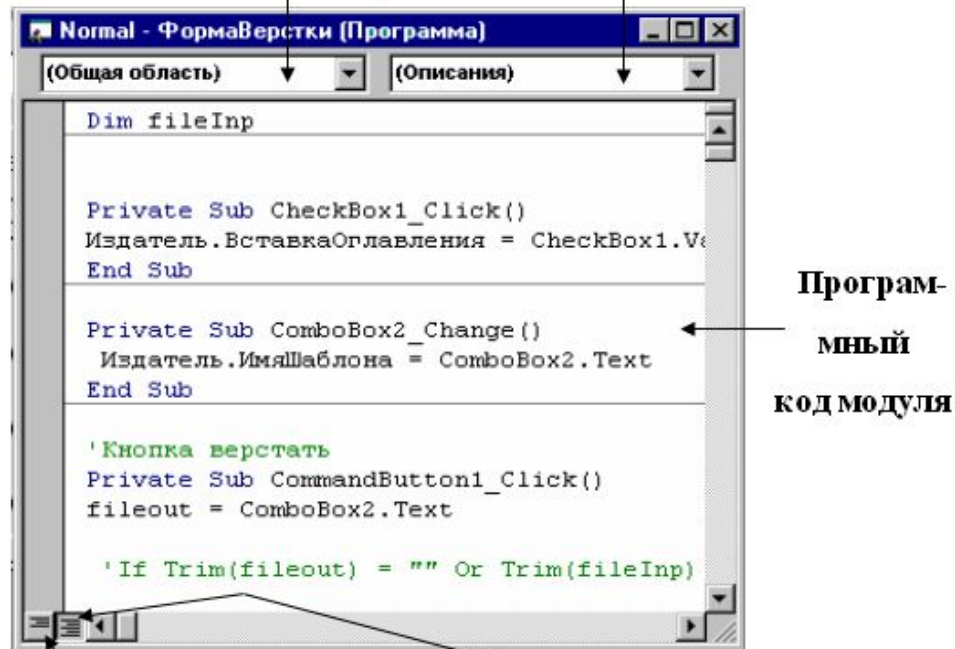
Const *Имя_константы1* [**as** тип] = Значение

Структура модуля

- **General (Общая область)** – область для описания глобальных переменных, а так же для объявления не событийных подпрограмм, состоит из подобластей:
- **Declaration (Описания)** - раздел описания глобальных переменных(расположен в верхней части модуля)
- **Раздел описания не событийных подпрограмм** – расположен под разделом описания.
- **Область событийных процедур объектов** – расположена под разделом описания.

Выбор областей модуля в окне редактора модуля

Выбор области или объекта	Общей или имени	Выбор имени раздела	Описания или не	Описания или событийной
		подпрограммы, а так же события	происходящего в	выбранном объекте



Програм-
мный
код модуля

● Кнопка,
уганавливающая режим
просмотра только текущей
подпрограммы

● Кнопка,
уганавливающая режим
просмотра всех подпрограмм

Объявление полей модуля

Private / Public *Имя_поля_модуля* [**as** тип]

Private / Public Const *Имя_поля_модуля* [**as** тип] = Значение

где

Private – поле используется только внутри текущего модуля.

Public – поле используется во всех модулях проекта.

Const – значение поля невозможно изменить во время работы программы.

примечание:

В модуле макросов **DIM** – альтернативное объявление для изменяемого **Public** поля.

Обращение к полю модуля из другого модуля

Имя_Модуля1.Имя_Поля

примечание:

Если поле описано в модуле макросов и в других модулях нет одноименных полей, допускается обращаться к полю просто по имени, без указания модуля.

Типы переменных и констант

<i>Имя типа</i>	<i>Разновидность Типа</i>	<i>Диапазон значений</i>	<i>Размерность</i>
Byte	Байт	От 0 до 255	1 байт
Integer	Короткое целое	От -32768 до 32767	2 байта
Long	Длинное целое	От -2147483648 до 2147483647	4 байта
Single	Вещественное одинарной точности	От $\pm 1,5 \cdot 10^{-45}$ до $\pm 3,4 \cdot 10^{38}$	4 байта
Double	Вещественное двойной точности	От $\pm 5,0 \cdot 10^{-324}$ до $\pm 1,7 \cdot 10^{308}$	8 байт
Currency	Денежный	От -922 337 203 685 477,5808 до 922 337 203 685 477,5807	8 байт

Типы переменных и конст.(продол.)

Boolean	Логический	True или False	2 байта
Date	Дата	Хранит календарную дату и(или) время.	8 байт
String	Строковый	Хранит текстовую информацию	От 0 до 65535 байт
String * N	Строка фиксированной длины	Хранит текстовую информацию	N байт
Variant	Универсальный	Заменяет любой простой тип, также структурированный и объектный тип.	
Object	Объектный	Перекрывается более современным типом Variant	

Оператор присваивания

Имя_переменной = Выражение

Приоритет математических операций в VBA

<i>Название операции</i>	<i>Написание знака в Visual Basic</i>	<i>Пример написания в Visual Basic</i>	<i>Пример написания в математике</i>
Возведение в степень	^	A^B	A^B
Изменение знака числа на обратный	-	-A	-A
Умножение и деление	*, /	A*B/C	$A \bullet B / C$
Деление нацело	\	A \ B	$A \text{ div } B$
Получение остатка от деления	Mod	A mod B	$A \text{ mod } B$
Сложение и вычитание	+, -	A+B-C	$A+B-C$

Математические функции

<i>Название функции</i>	<i>Написания функции в Visual Basic</i>	<i>Пример написания в математике</i>
Модуль числа	ABS(x)	x
Экспонента	Exp(x)	E ^x
Натуральный логарифм	Log(x)	Ln(x)
Логарифм А по основанию В	Log(A)/Log(B)	Log _B A
Квадратный корень	SQR(x)	√x
Синус, косинус, Тангенс, арктангенс (аргументы в радианах!)	Sin(x), Cos(x), Tan(x), Atn(x)	Sin x, Cos x, Tg x, Arctg x

Пример строковых выражений

$L = \text{"Visual "}$

$M = \text{"Basic"}$

$K = L + M$

Переменной K присвоится значение "Visual Basic"

Строки можно складывать как знаком $+$, так и знаком $\&$. Причем знаком $\&$ можно приплюсовывать к строковому значению числовое без всякого преобразования:

$K = \text{"Excel"} \& 2007$

Переменной K присвоится текстовое значение "Excel2007"

Функции для работы со строками

<i>Имя Функции</i>	<i>Назначение</i>
MID(Строка, N)	Выдает символы из указанной строки, начиная с символа имеющего порядковый номер N, и кончая последним символом в строке.
MID(Строка,N,N1)	Выдает N1 символов из указанной строки, начиная с символа имеющего порядковый номер N.
Len(Строка)	Выдает длину строки в байтах (и не только строки)
Ltrim(Строка)	Выдает строку из строки аргумента, удаляя из нее все левые пробелы
Rtrim(Строка)	Выдает строку из строки аргумента, удаляя из нее все правые пробелы
Trim(Строка)	Выдает строку из строки аргумента, удаляя из нее все левые, и правые пробелы
INSTR(Строка1, Строка2)	Ищет Строку2 в Строке1, функция выдает значение 0, если указанная строка не найдена, или номер символа, откуда начинается вложение Строки2 в Строку1
Ucase(Строка)	Переводит символы букв строки в верхний регистр
Lcase(Строка)	Переводит символы букв строки в нижний регистр
Left(Строка, N)	Выдает из строки N первых символов
Right(Строка, N)	Выдает из строки N последних символов

Примеры логических выражений, знаки отношения

$A = 10$

$B = (A > 100)$

Переменная B будет иметь значение False (Ложь), т.к. значение переменной A не превышает число 100.

<i>Название знака</i>	<i>Знак в VBA</i>	<i>Знак в математике</i>
Равно	=	=
Неравно	<>	≠
Больше	>	>
Меньше	<	<
Меньше, либо равно	<=	≤
Больше, либо равно	>=	≥

Приоритеты операций

Приоритет	Операция
1	Выражение, заключенное в скобки ()
2	Вызов функции
3	Степень (^)
4	- (смена знака)
5	(*) и (/)
6	(\) деление нацело
7	Mod(остаток от деления нацело)
8	(+) и (-)
9	>, <, >=, <=, <>, =
10	Not
11	And
12	Or

Примеры логических выражений, логические функции в порядке приоритета

<i>Название и функция</i>	<i>Имя функции</i>	<i>Пример</i>	<i>Значение B</i>
<i>Логическое отрицание.</i>	Not	$A=10$ $B = \text{Not } A > 5$	$B = \text{False}$
<i>Логическое “И”, логическое умножение.</i>	AND	$A=10$ $C=3$ $B = A > 5 \text{ And } C < 3$	$B = \text{False}$
<i>Логическое “ИЛИ”, логическое сложение.</i>	OR	$A=10$ $C=3$ $B = A > 5 \text{ or } C < 3$	$B = \text{True}$

Функции преобразование типов

<i>Имя Функции</i>	<i>Назначение</i>
CSTR(Число)	Преобразует число в строку
Val(Строка)	Преобразует строку в число, если в указанной строке нет символов цифр, данная функция выдает 0.
CHR(КодСимвола)	Преобразует код указанного символа в его строковое значение
ASC(Строка)	Выдает код первого символа в строке аргумента
Cint(ВещественноеЧисло)	Преобразует вещественное число в короткое целое путем округления числа, при спорном округлении, функция округляет число до ближайшего четного числа.
CLng(ВещественноеЧисло)	Преобразует вещественное число в длинное целое путем округления числа, при спорном округлении, функция округляет число до ближайшего четного числа.
CDate(СтроковоеВыражение)	Преобразует строковое выражение в формат Data

Условный оператор

If *условие* Then *операторы_если_истина* Else *операторы_если_ложь*

Блочная форма

If *условие-1* Then

операторы

Elseif *условие-2* Then

операторы

Else

операторы

End If

Оператор выбора

Select Case *выражение*

[Case *список_выражений-n*
инструкции-n]

...

[Case Else
инструкции_else]

End Select

Операторы цикла

ЦИКЛЫ со счетчиком
(итерационные)

For...Next

1. For i= 1 To 10 Step 2

2. S= 0

For i= 1 To 10

S = S + i

If S > 6 Then

ExitFor

End If

Next i

и

ЦИКЛЫ с условием
(параметрические)

Цикл с предусловием	Цикл с постусловием
--------------------------------	--------------------------------

ЦИКЛЫ С УСЛОВИЕМ

Условие задается ключевым словом **While** или ключевым словом **Until**. Оба они могут быть с предусловием или с постусловием.

Цикл с предусловием	Цикл с постусловием
Do While Until <условие> <Тело цикла> Loop	Do <Тело цикла> Loop While Until <условие>

ЦИКЛЫ С УСЛОВИЕМ

$$\sum_{i=1}^{10} i^2$$

Конструкция **Do ... While** : делать пока выполняется условие.

S = 0

i = 1

Do While i <= 10

S = S + i ^ 2

i = i + 1

Loop

ЦИКЛЫ С УСЛОВИЕМ

$$\sum_{i=1}^{10} i^2$$

Конструкция **Do ... Until** : делать до тех пор, пока не будет выполнено условие.

S = 0

i = 1

Do Until i > 10

S = S + i ^ 2

i = i + 1

Loop