

Программирование на *Visual Basic for Application (VBA)*

В качестве приложения рассматриваем приложение Microsoft Excel

Язык и среда VBA . Редактор VBA .

- VBA (Visual Basic for Application) – это язык программирования, поддерживаемый всеми приложениями интегрированного пакета Microsoft Office, в состав которого входят такие популярные приложения, как Microsoft Access, Microsoft Excel, Microsoft PowerPoint, Microsoft Word и др.
- Редактор VBA представляет собой совокупность инструментальных средств, предназначенных для создания *модулей* программ, просмотра существующих модулей, создания и редактирования исходного кода программного модуля, создания пользовательских диалоговых окон и выполнения других задач.

Порядок действий при создании проекта VBA

- а) открыть приложение (Excel)
- б) из Excel 2007 нужно переключиться на ленту **Разработчик**, выбрать команду **Visual Basic**, расположенную в группе команд **Код** (рис. 1). (нажать клавиши <Alt>+<F11> – запуск редактора VBA)
- в) выполнить команды **Insert** ⇒ **Module** – вставка модуля для набора текста программы
- г) выполнить команды **View** ⇒ **Project Explorer** – будет вызван «Проводник» по работе с VBA

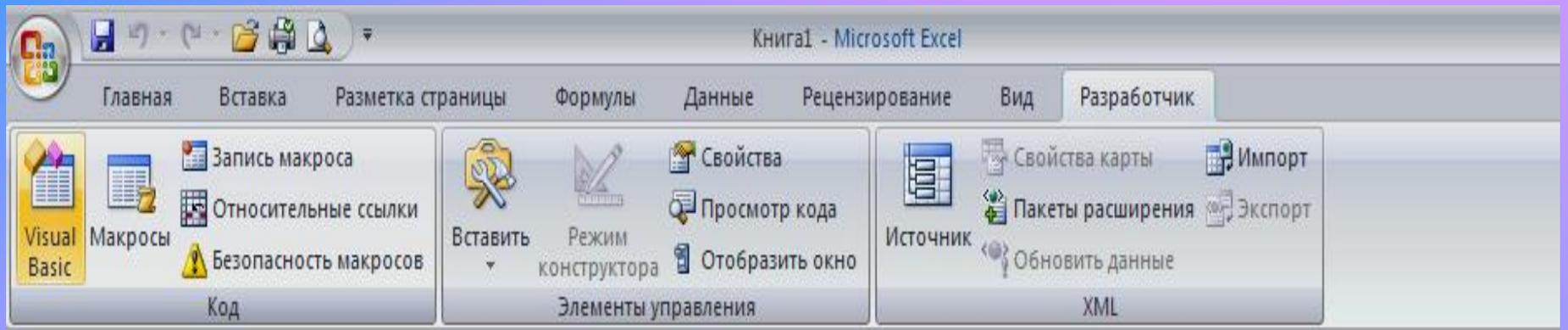


Рис. 1. Запуск Visual Basic из Excel 2007

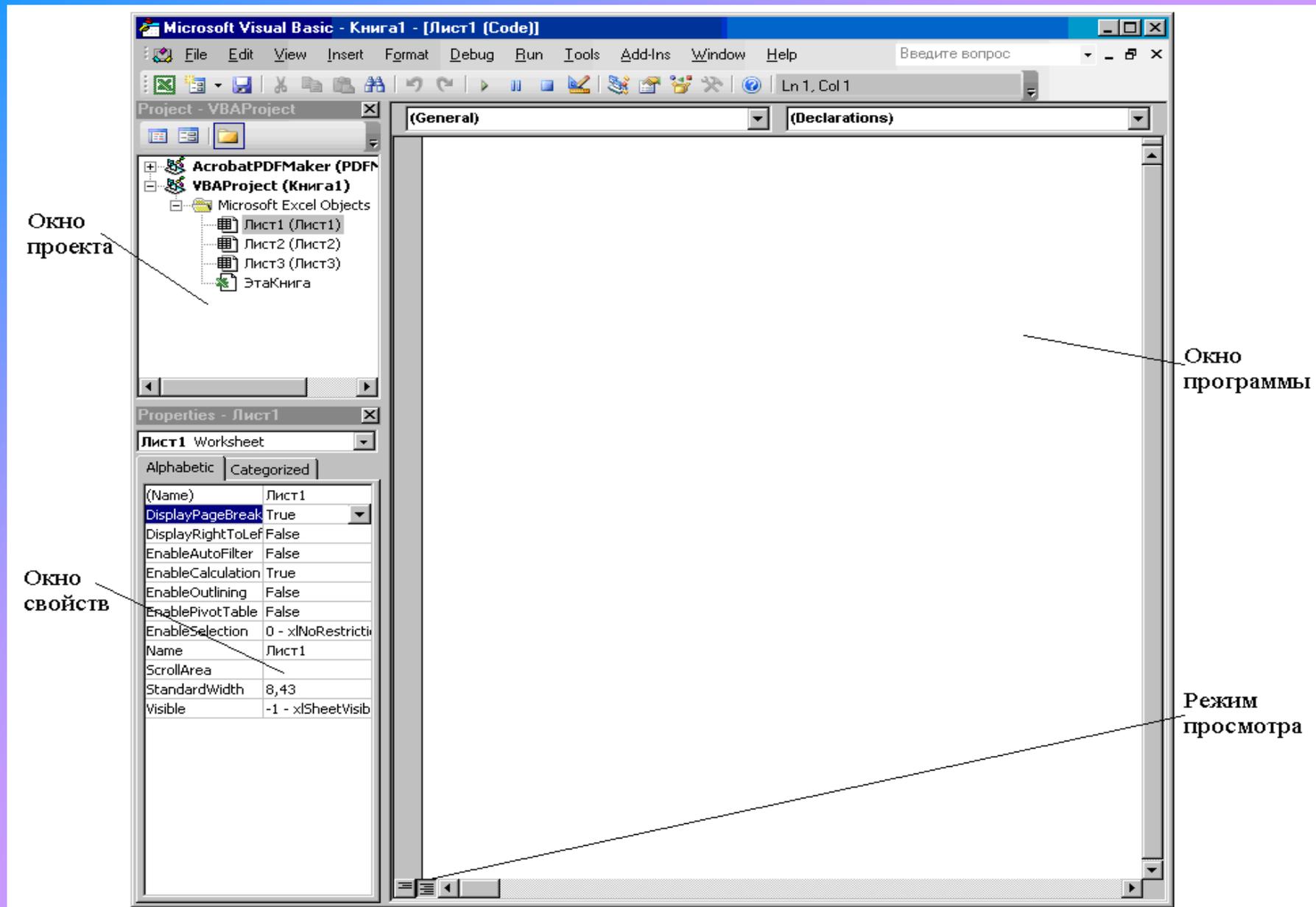


Рис. 2. Окно редактора Visual Basic в Microsoft Excel

Структура программ на VBA. Процедуры и функции пользователя

Программа VBA представляет собой совокупность процедур и функций, размещенных в зависимости от особенностей решаемой задачи, в одном или нескольких модулях. Каждый модуль имеет две области: общую область и область подпрограмм.

В общей области помещаются операторы описания переменных, которые являются общими для всех процедур и функций этого модуля.

В области подпрограмм помещается только код подпрограммы.

Определение процедур

Различают следующие типы процедур:

- процедуры обработки событий;
- процедуры макросов;
- ***процедуры пользователя***

Примечание. Необязательные параметры в структуре оператора помещаются в [квадратные скобки]

Альтернативные параметры разделяются вертикальной чертой |: A|B

Структура программы-процедуры (п/п)

Sub ИмяПроцедуры(СпискиПараметров)

<<Тело п/п>>

End Sub

Sub – ключевое слово, определяющее тип п/п.

ИмяПроцедуры – имя п/п (дает/придумывает сам пользователь – программист).

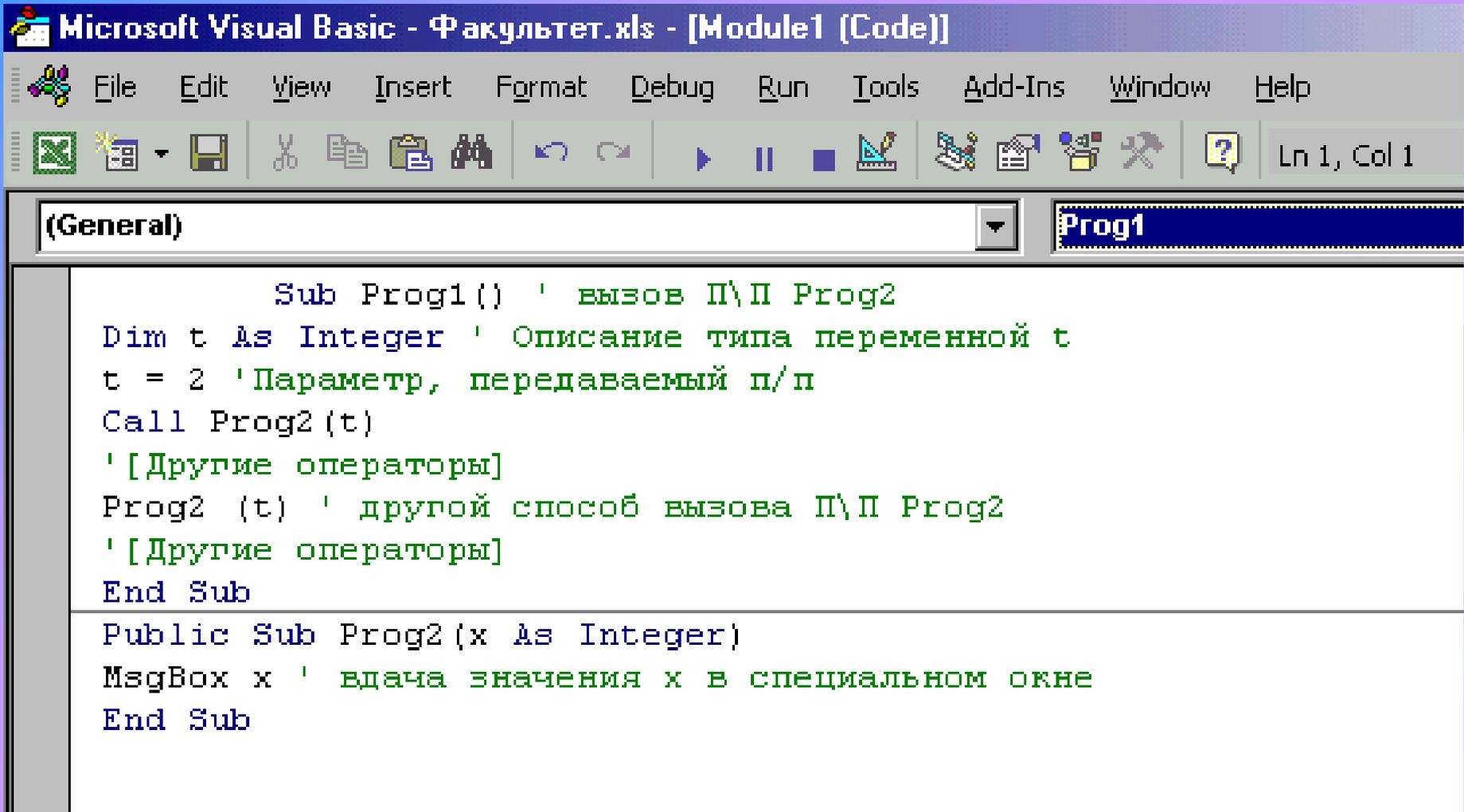
СписокПараметров служит для передачи процедуре исходных данных для вычислений (может отсутствовать). Он состоит из элементов списка, разделенных запятыми и имеет следующий синтаксис:

ИмяЭлемента As ТипДанных

ИмяЭлемента – идентификатор; *As* – ключевое слово (как); *ТипДанных* – тип данных элемента списка.

Процедура пользователя может быть вызвана из другой п/п оператором **Call** или указанием ее имени.

Пример программы-процедуры



The screenshot shows the Microsoft Visual Basic IDE with the following code in the code window:

```
Sub Prog1() ' вызов П\П Prog2
Dim t As Integer ' Описание типа переменной t
t = 2 ' Параметр, передаваемый п/п
Call Prog2 (t)
' [Другие операторы]
Prog2 (t) ' другой способ вызова П\П Prog2
' [Другие операторы]
End Sub

Public Sub Prog2 (x As Integer)
MsgBox x ' вдача значения x в специальном окне
End Sub
```

Определение функций

Функция – это программа, которая выполняет действия в пределах своего блока и возвращает единственное значение.

В VBA различают следующие виды функций:

- **функции пользователя**
- функции классов.

Структура подпрограммы- функции (п/ф)

Function **ИмяФункции(СпискиПараметров)** **As** **ТипДанных**

<<Тело п/ф>>

ИмяФункции=Возвращаемое значение

End Function

Function – ключевое слово, указывающее на то, что это функция;

ИмяФункции – имя функции (придумывает сам программист);

СписокПараметров служит для передачи процедуре исходных данных для вычислений (может отсутствовать).

As ТипДанных – оператор, определяющий тип данных возвращаемого значения.

ВозвращаемоеЗначение – значение, возвращаемой функцией.

Пример вызова п/ф

Ракультет.xls - Module2 (Code)

general)

Func

```
Sub Prog3 ()  
Dim u, t As Integer ' Описание типа переменных u, t  
t = 2 ' Параметр, передаваемый п/ф  
y = Func(t)  
' [Другие операторы]  
MsgBox (y)  
End Sub  
  
Public Function Func(x As Integer) As Integer  
z = x ^ 2 + x + 5  
Func = z  
End Function
```

Переменные и константы

Правила выбора имён переменных

Данные различных типов представляются в VBA в виде переменных или констант. *Переменная* – это величина, значение которой может изменяться в программе. Соответственно, *константа* – это величина, значение которой в программе изменяться не будет (константы можно изменять только редактированием исходного кода VBA).

При выборе имени переменной в VBA необходимо соблюдать следующие правила:

- имя переменной должно начинаться с буквы алфавита (допускается только латиница);
- после первой буквы имени переменной может стоять любая комбинация цифр, букв или символов подчеркивания, или нижнее тире (_);
- имена переменных не должны содержать пробелы, знаки препинания или любые другие *специальные символы* (№, #, \$, /, & и т.п) или символы математических операций (=, +, −, /);
- имена переменных могут быть любой длины, но не должны превышать 255 символов, но следует иметь в виду, что VBA учитывает только первые 31 символ;
- имя переменной *не должно дублировать* определенные ключевые слова языка VBA;
- имя переменной должно быть уникальным в рамках его *области действия*, то есть в пределах процедуры или модуля, в котором данная переменная объявляется.

Примеры правильных имен:

InputValue, Адрес_Организации, Номер1,
Номер2, A12B; C45d

Примеры неправильных имен:

2Week,
_Номер,
MsgBox,
Number 1.

Советы|Правила задания имен

- Имя должно отражать назначение переменной или программы – *мнемоническое* имя:

Пример1: $F = m \cdot a$ или $V_{\text{шар}} = \frac{4}{3} \cdot \pi \cdot R^3$

Пример2: Стоимость = Тариф*Кол-во_дней

- Лучше использовать строчные (маленькие) буквы для имен переменных. Прописными обычно обозначают константы
- **Важно помнить:** VBA не различает строчные и прописные буквы!!!

$R = a / \text{Sqr}(3)$: $r = a / (2 * \text{Sqr}(3))$

Значение R будет «забито» значением r

Типы констант. Представление чисел в форме с плавающей точкой

- Константой называется некоторая величина, не изменяющая своего числового или символьного значения в течение выполнения всей программы.

Имеются два типа констант:

- – числовые конст. - представляют собой положительные или отрицательные числа двух видов:
 - а) целочисленные, представляющие собой положительные и отрицательные числа и ноль, при их записи не должна использоваться десятичная запятая, например:

-100 2 0 +458

Типы констант. Представление чисел в форме с плавающей точкой

б) вещественные константы – это числа, записанные в форме с плавающей точкой.

Записываются

с помощью

10

(вместо

то

с плавающей точкой

как $2.5E-02$

Здесь 2.5 – мантисса, E , D – десятичная экспонента одинарной или двойной (D) точности.

Например: $1,5 \times 10^{-3} \Rightarrow 1.5E-3=0.15E-02=15E-04=0.0015$

$150000 \Rightarrow 1.5E+05$

$0,0000254 \Rightarrow 2.54E-05$

В теле программы в качестве разделителя целой и дробной части используется ТОЧКА.

Если данные считываются с листа EXCEL, то на листе вещественные дроби пишутся через ЗАПЯТУЮ!

Типы данных языка Visual Basic

Используемые данные имеют разные типы: числовые и символьные.

- Числовые данные в свою очередь могут быть целыми и дробными. Например, 328 – это целое число (целый тип данных), 12,0385 – это дробное число

- Символьные данные представляют собой набор произвольных символов, принадлежащих языку Бейсик, заключенных в кавычки, например:

“alpha”, “beta”, “Луна”, «введите», «результат» – это символьный тип данных (в VBA этот тип называют также строковый тип данных).

Тип	Хранимая информация	Занимаемая память	Диапазон значений
Целочисленные типы			
<u>Byte</u>	Целые числа	1 байт	От 0 до 255.
<u>Integer</u>	Целые числа	2 байт	От -32 768 до 32 767.
<u>Long</u>	Длинное целое	4 байт	От -2 147 483 648 до 2 147 483 647. или +/- 2.1E9

Дробные числа

<u>Single</u>	Используется для хранения <i>дробного</i> числа (8 десятичных знаков после запятой).	4 байта	Отрицательные числа: от $-3,402823E38$ до $-1,401298E45$ Положительные числа: от $1,401298E-45$ до $3,402823E38$
<u>Double</u>	Используется для хранения <i>дробных</i> чисел с повышенной точностью (16 десятичных знаков после запятой)	8 байт	Отрицательные числа: от $-179,769313486232D+308$ до $-4,94065645841247D-324$ Положительные числа: от $4,94065645841247D-324$ до $1,79769313486232D+308$

Строковые типы

<u>String</u>	текстовая информация (строка) переменной длины	10 байт + 1 байт на каждый символ	От 0 до приблизительно 2 миллиардов символов.
<u>String*n</u>	строка постоянной длины в n символов	Длина строки	От 1 до приблизительно 600 тыс.

Типы Variant

<u>Variant</u> (числовые подтипы)	Значения любого из перечисленных типов данных. Этот тип данных является типом данных <u>По умолчанию</u> (если в программе отсутствует указание типа какой-либо переменной, то VBA приписывает ей тип <u>Variant</u> и выделяет 16 байт)	16 байт для чисел; 22 байт + 1 байт на каждый символ строки	Любое числовое или строковое значение вплоть до границ диапазона для типа <u>Double</u> .
--------------------------------------	--	---	---

Тип данных *Currency* (8 байт)

Это число с фиксированной точкой, т.е., десятичная точка всегда находится в одном и том же месте - справа от точки всегда имеются четыре цифры. Этот тип данных используется при денежных вычислениях, когда требуется высокая точность.

Currency хранит числа в диапазоне от -922337203685477,5808 до 922337203685477,5807.

Тип данных *Boolean* (2 байта)

Используется для хранения **логических** значений (переменная имеет только два возможных значения - *True* (Истина) или *False* (Ложь). Логические значения *True* и *False* называются булевскими значениями.

Тип данных *Date* (8 байт)

Хранение даты и времени.

Даты от 1.01.100 до 31.12.9999

Время от 00:00:00 до 23:59:59

Задание типа данных переменной

Типы данных в VBA объявляются в программе с помощью ключевых слов (в табл. см.выше).

Объявлять тип переменным не обязательно (в этом случае VBA сам автоматически присваивает им универсальный тип **Variant**).

Преимущества при объявлении типа переменных:

1. объявленные переменные ускоряют выполнение кода программ (при необъявленной переменной среда VBA тратит некоторое время на анализ переменной, чтобы определить ее конкретный тип);
2. объявленные переменные повышают эффективность кода. Переменные типа **Variant** могут занимать гораздо больше памяти, чем переменные определенных типов (=> можно значительно сократить объем памяти, требуемый для программы в целом);
3. программный код с переменными объявленного типа легче читать и понимать.

Объявить переменную – значит заранее сообщить программе о ее существовании.

Объявить тип переменной можно **явно и неявно.**

Для **явного** объявления переменной и ее типа используется оператор **Dim**

Dim переменная ***[As тип]***

- Одновременно с объявлением переменной после ее имени можно записать ключевое слово *As*, после которого задается тип переменной (см.табл.)

Примеры правильного синтаксиса для объявлений типа переменных:

Dim a, b, sum, z12 As Integer 'переменной a, b, sum присвоится тип данных по умолчанию Variant, переменной z12 присваивается тип целое.

Dim Vuz As String 'Luna соответствует тип строковый

Dim Counter2 As Integer

Dim Profit As Single, Message As String

При **неявном** объявлении тип переменной задается добавлением суффикса *символа определения типа* в конец имени переменной (см. табл. 2), например:

MyVar% = 25 ' соответствует типу Integer

CelsiusTemp! = 15,35 ' соответствует типу Single

Title\$ = "Excel" ' соответствует типу String

Хотя неявное объявление переменных может показаться более удобным, с ним связаны некоторые проблемы. Когда переменные объявляются неявно, существует риск нечаянно создать новую переменную, когда на самом деле, необходимо использовать уже существующую переменную, или использовать существующую переменную, когда надо создать новую.

После объявления переменной независимо от того, объявляется ли эта переменная явно или неявно, эта переменная сохраняет тот же самый тип столько времени, сколько она существует. Нельзя переобъявить переменную или переопределить ее тип.

Встроенные математические функции

Функции (аргументы)	Функции (аргументы) на VBA	Возвращает/ действие
1	2	3
$ x $	Abs (x)	Возвращает абсолютное значение x
$\arctg(x)$	Atn (x)	Возвращает арктангенс x , как угол в радианах
$\cos(x)$	Cos (x)	Косинус угла x , где x – это угол, измеряемый в радианах
$\sin(x)$	Sin (x)	Возвращает синус угла; x – это угол, измеряемый в радианах
\sqrt{x}	Sqr (x)	Возвращает корень квадратный из x . VBA отображает ошибку исполнения, если x – отрицательное
$\tg(x)$	Tan (x)	Возвращает тангенс угла; x – угол в радианах
e^x	Exp (x)	Возвращает константу e , возведенную в степень x (e – это основание натурального логарифма)
$\log(x)$	Log (x)	Возвращает натуральный логарифм x

1	2	3
	<p><u>Sgn</u> (x)</p>	<p>Возвращает знак числа: -1, если <u>x</u> отрицательное; 1, если <u>x</u> положительное; 0, если <u>x</u> равно 0</p>
	<p><u>Int</u> (x)</p>	<p>Возвращает целую часть <u>x</u>. <u>Int</u> не округляет число, а отбрасывает любую дробную часть. Если <u>x</u> является отрицательным, <u>Int</u> возвращает ближайшее отрицательное целое меньшее или равное <u>x</u></p>
	<p><u>Fix</u> (x)</p>	<p>Возвращает целую часть <u>x</u>. <u>Fix</u> не округляет число, а отбрасывает любую дробную часть. Если <u>x</u> является отрицательным, <u>Fix</u> возвращает ближайшее отрицательное целое, большее или равное <u>x</u></p>

Примеры.

Пусть $a = -12.1$; $b = -12.56$; $c = 12.1$; $d = 12.56$

Тогда:

- $\text{Int}(a) \Rightarrow -13$; $\text{Int}(b) \Rightarrow -13$;
- $\text{Int}(c) \Rightarrow 12$; $\text{Int}(d) \Rightarrow 12$
- $\text{Fix}(a) \Rightarrow -12$; $\text{Fix}(b) \Rightarrow -12$;
- $\text{Fix}(c) \Rightarrow 12$; $\text{Fix}(d) \Rightarrow 12$

функция **FIX** равносильна отбрасыванию десятичных знаков числа.

- $\text{Sgn}(a) \Rightarrow -1$; $\text{Sgn}(c) \Rightarrow 1$
- $\text{Abs}(a) \Rightarrow 12.1$; $\text{Abs}(c) \Rightarrow 12.1$

**Математические выражения. Порядок
вычисления математических выражений.
Запись математических выражений.**

- Математические выражения в широком смысле – это выражения (формулы), в которых могут присутствовать операции вычисления арифметических, алгебраических, тригонометрических и других математических выражений, а также логические операции сравнения (больше, меньше и т.п.).

При вычислении сложных выражений VBA следуют таким правилам:

- все операции выполняются последовательно;
- части выражения, заключенные в круглые скобки, всегда вычисляются в первую очередь. Если выражение имеет внутренние скобки, то применяются правила - последовательно от внутренних выражений в скобках к внешним;
- конкретные операции выполняются в зависимости от иерархии операторов (см. табл.); если операторы имеют равный уровень приоритета, они вычисляются в порядке слева направо;
- если в выражении присутствуют арифметические и логические операции, то сначала выполняются арифметические, и только потом логические.

Операции VBA

Математические операции.

Операции сравнения.

Оператор/ операция	Комментарии
1	2
$\cos x \Rightarrow$ $\text{Cos}(x)$	Стандартные функции
$x^2 \Rightarrow x^{\wedge}2$	Возведение в степень, наивысший приоритет
$-3 \Rightarrow -3$	Унарный минус (присвоение знака числу)
$\underline{a} \underline{b} \Rightarrow a * b,$ $\frac{a}{b} \Rightarrow a / b$	Умножение и деление имеют равные приоритеты; они вычисляются по мере появления в выражении слева направо
$\underline{a} \backslash b$	Вычисляет результат целочисленного деления первого математического выражения (a) на второе (b). Перед вычислением значение каждого выражения округляется до целых по правилам математики.
$\underline{a} + \underline{b} \Rightarrow \underline{a + b}$ $\underline{a} - \underline{b} \Rightarrow \underline{a - b}$	Сложение и вычитание имеют равный приоритет; они вычисляются по мере появления в выражении слева направо

1	2
&	Сложение строк
$<, >, =,$ $\underline{a \leq b} \Rightarrow a \underline{\leq} b$ $\underline{a \geq b} \Rightarrow a \underline{\geq} b$ $\underline{a \neq b} \Rightarrow a \underline{\neq} b$	Все операторы сравнения имеют равные приоритеты и вычисляются по мере появления в выражении слева направо. Для группирования сравнения используют круглые скобки
Not, And, Or, <u>Xor</u> , <u>Eqv</u> , Imp	Логические операции

Дополнительные, не представленные в таблице тригонометрические функции можно выводить из базовых математических функций. Например, если необходимо вычислить котангенс угла, для его нахождения можно использовать известную формулу $1/\text{Tan}(x)$.

Пример записи выражения:

$z = a^{x^y}$	<code>z = a ^ (x ^ y)</code>
$y = \frac{a+b+3}{\frac{a+1}{b+2}}$	<code>y = (a + b + 3) / ((a + 1) / (b + 2))</code>
$y = \sqrt[7]{\lg x}$	<code>y = (Log(x) / Log(10)) ^ (1/7)</code>
$y = \sqrt[3]{\sin^2 x}$	<code>y = Sin(x) ^ (2/3)</code>
$y = e^{\sqrt{ \cos x }}$	<code>y = Exp(<u>Sqr</u>(Abs(Cos(x))))</code>
$y = \sin^2 x$	<code>y = sin(x) ^ 2</code>
$y = \sin x^2$	<code>y = sin(x^2)</code>

Математические выражения в программе записываются в одну строку (не более 255 символов). Проверять длинные записи на предмет количества скобок становится легче, если знать правило: в любом выражении количество левых скобок должно быть равно количеству правых.

Целочисленное деление (\)

- *Целочисленное деление* (ц/д) отличается от деления с плавающей точкой тем, что **его результатом всегда есть целое число без дробной части.**
- Оба операнда в выражении ц/д должны быть *численными* выражениями.
- Перед выполнением операции ц/д VBA **округляет каждый операнд** до числа типа *Integer* или *Long* (такой же тип имеет и результат ц/д).
- Для удобного вычисления операции ц/д деление производят «столбиком» до тех пор, пока остаток не станет меньше делителя

Например, выражения $27 \setminus 6$ и $27.3 \setminus 5.9$, а также $22 \setminus 5$ и $24 \setminus 5$ будут иметь один и тот же результат = 4, в то время как обычное деление даст разные результаты:

$27/6 = 4,5$; $27.3/5.9 \approx 4,6$; $22/5 = 4,4$; $24/5 = 4,8$.

$$27.3 \setminus 5.9 = 27 \setminus 6$$

$$\begin{array}{r|l} 27 & 6 \\ \hline 24 & \textcircled{4} \\ \hline 3 & \end{array}$$

результат ц/д 47

Деление по модулю (Mod)

- *Деление по модулю* дополняет целочисленное деление. В делении по модулю выражение **возвращает только остаток операции деления как целое**:
- Например: $22 \text{ Mod } 5 = 2$;
 $24 \text{ Mod } 5 = 4$
 $25 \text{ Mod } 5 = 0$.
- Остальные свойства деления по модулю идентичны целочисленному делению.

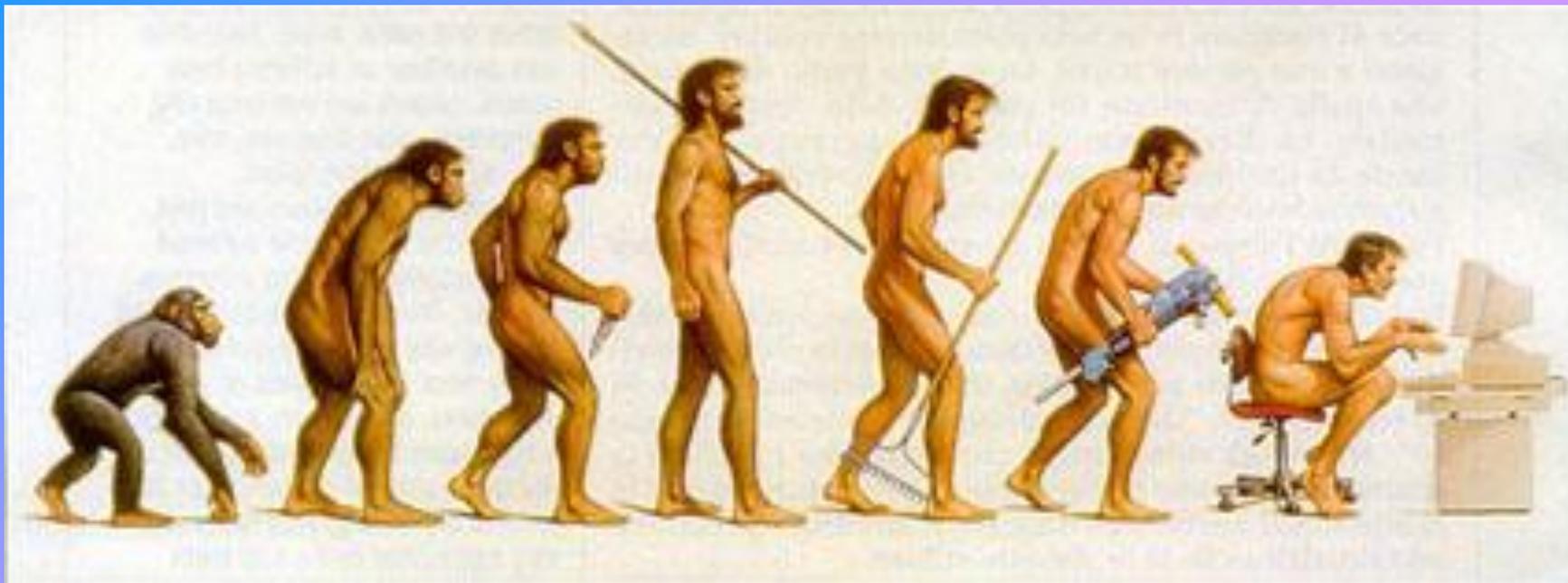
$$22.5 \text{ Mod } 4.51 \rightarrow 22 \text{ Mod } 5 = 2$$

$$\begin{array}{r|l} 22 & 5 \\ \hline 20 & 4 \\ \hline \end{array}$$

2 — остаток от ц/д

Вывод для алгоритмизации некоторых задач:

- Операцию ц/д $a \text{ Mod } b$ можно использовать для определения кратности числа a числу b .
- Если число a кратно числу b , то остаток от их ц/д равен нулю.



Лекция окончена!
Спасибо за внимание!!!