Типы данных, переменные и константы.

Выражения в VBA. Использование функций VBA

- **Программными единицами в языке VBA являются процедуры и функции, которые принято хранить в модулях, либо в текстовых файлах. Количество процедур или функций в модуле не ограничивается. Чтобы программировать, требуется знать, какими типами данных оперирует язык VBA, операции и операторы для их обработки. Важно также знание стандартных функций для обработки данных различных типов.**

Создание процедуры по типу макроса

- Текст процедуры подчиняется следующему формату:

***Sub* <имя процедуры> ( )**

<тело процедуры>

End Sub

Ошибки синтаксиса

- ***Синтаксисом* (*syntax*) – называется определенный порядок слов и символов, который образует правильный оператор VBA. Некоторые из наиболее общих ошибок, с которыми вы сталкиваетесь во время написания или редактирования процедур VBA, – это *ошибки синтаксиса* (*syntax errors*), например, пропущенные запятые, кавычки, аргументы и так далее.**

Ошибки синтаксиса

- **Всякий раз, когда вы пишете новую строку кода или изменяете существующую, VBA *анализирует* (*parses*) строку, как только курсор вставки перемещается из новой или измененной строки.**
- ***Синтаксический анализ* (*parsing*) – так называется процесс разбивки оператора VBA на составляющие части и определение того, какие части строки являются ключевыми словами, переменными или данными. После выполнения анализа строки кода VBA компилирует эту строку кода.**

Ошибки синтаксиса

- **После выполнения анализа строки кода VBA компилирует эту строку кода.**
- ***Компиляция* (*compiling*) в VBA означает составление исходного кода в форме, которую VBA может непосредственно выполнять без необходимости снова анализировать код.**

Ошибки синтаксиса

- После того как VBA успешно завершит анализ и компиляцию строки кода в процедуре и не будет обнаружено никаких ошибок, выполнится цветовое кодирование различных частей строки
- Ключевые слова в Редакторе VB отображаются синим цветом, комментарии – зеленым, а данные или другие операторы отображаются в виде черного текста.
- Если, однако, VBA обнаруживает ошибку синтаксиса в строке в процессе анализа или компиляции, VBA отображает всю строку красным цветом и выводит на экран диалоговое окно с сообщением об ошибке.

- **При написании процедуры следует соблюдать требования синтаксиса VBA.**

- **В частности, в одной строке можно писать одну или несколько инструкций. Если их несколько, то их следует отделять друг от друга символом двоеточие ":".**
- **Если же требуется перенести часть инструкции на следующую строку, то следует в конце предыдущей строки ввести символ продолжения строки – это комбинация клавиш пробел и символа подчеркивания " _".**
- **Запись комментариев с строке начинается с символа апостроф (').**

- **Пример** простой процедуры, выводящей на экран слова ***Привет, мир***.

Наши действия по созданию процедуры:

1. Открывает книгу Excel. Открываем редактор Visual Basic, нажав ***Alt+F11***. В окне **Project** отмечаем книгу, в которой создадим новый модуль для записи текста процедуры.
2. Выбираем команду ***Insert /Module***. Visual Basic создает новый модуль и открывает для него окно ***Code***. Можно изменить имя модуля, дав ему содержательное имя. Для этого открываем окно свойств ***Properties***, в котором в текстовом поле ***Name*** вводим новое имя для модуля, например ***Приветствие***.

3. В окне Code вводим текст процедуры:

Sub Hello() *' начало процедуры – ее объявление*

MsgBox "Привет, мир" *' тело процедуры*

End Sub *' конец процедуры*

4. Для запуска процедуры следует курсор установить в ее теле и нажать на панели инстр

Run Sub/.

Типы данных VBA

Название типа	Размер в байтах	Описание и диапазон значения
Byte	1	Целые положительные числа от 0 до 255
Integer	2	Целые числа от -32768 до 32767
Long	4	Длинные целые числа от -2147483648 до 2147483647

Название типа	Размер в байтах	Описание и диапазон значения
Single	4	Вещественные числа обычной точности с плавающей точкой. Отрицательные числа: от -3.402823E38 до -1.401298E-45. Положительные числа: от 1.401298E-45 до 3.402823E38
Double	8	Вещественные числа двойной точности с плавающей точкой. Отрицательные числа: от -1.7976931348623 2E308 до -4.940656458412 47E-324. Положительные числа: от 4.940656458412 47E-324 до 1.7976931348623 2E308

Название типа	Размер в байтах	Описание и диапазон значения
Currency	8	Числа, имеющие до 15 цифр до десятичной точки и 4 цифры после нее (денежные единицы). От -922337203685477.5808 до 922337203685477.5807
Boolean	2	Для хранения логических значений; может содержать только значения True (Истина) или False (Ложно)
Date	8	Для хранения комбинации информации о дате и времени. Диапазон дат может быть от 1 января 100 года до 31 декабря 9999 года. Диапазон времени от 00:00:00 до

Название типа	Размер в байтах	Описание и диапазон значения
String	Длина строки (один байт на один символ)	Используется для хранения текста. Может содержать от одного до (приблизительно) 65400 символов
Variant	16 байт + 1 байт/символ	Тип Variant может хранить любой другой тип данных.
Object	4	Используется для доступа к любому объекту, распознаваемому VBA. Сохраняет адрес объекта в памяти

Типы данных VBA

- VBA имеет шесть различных численных типов данных: **Byte**, **Integer**, **Long**, **Single**, **Double** и **Currency.** Численные типы данных используются для хранения (и манипулирования) чисел в различных форматах, в зависимости от конкретного типа.

Типы данных VBA

- Обычно VBA-программа (как и любые другие программы) «принимает» решения, проверяя, являются ли истинными различные условия. Для упрощения тестирования условий и обеспечения сохранения результатов такого тестирования в VBA имеется логический тип данных. Логические значения **True** и **False** называют *булевыми* (*Boolean*) значениями.

Типы данных VBA

- Любые текстовые данные, сохраняемые в программе VBA, называются *строками* (*strings*). Строки в VBA сохраняются с использованием типа данных **String.** Строка может содержать текстовые символы любых типов: буквы алфавита, цифры, знаки пунктуации или различные символы.

Типы данных VBA

- Тип данных **Variant** – это особый тип данных, который может сохранять любые типы, за исключением типа **Object.**
- VBA использует тип **Variant** для всех переменных, если только вы не объявляете явно тип переменной. Данные типа **Variant** принимают характеристики определенного типа, который они сохраняют в данный момент.
- Например, если данные типа **Variant** содержат строковые данные, **Variant** принимает характеристики типа String. Если данные типа **Variant** содержат численные данные, **Variant** принимает характеристики какого-либо численного типа, обычно – **Double**, хотя типы **Variant** могут также иметь характеристики типов **Integer**, **Long**, **Single** или **Currency**.

Типы данных VBA

- **Несмотря на то, что типы Variant удобны и избавляют от некоторой части работы при написании процедур, они требуют большего объема памяти, чем любой другой тип данных, за исключением больших строк.**
- **Кроме того, математические операции и операции сравнения над данными типа Variant выполняются медленнее, чем подобные операции над данными любого другого типа.**

Переменные

- **Имя переменной должно начинаться с буквы, за которой может следовать любая комбинация цифр и букв с символом подчеркивания, длиной не более 255 символов.**
- **Имена переменных в VBA не чувствительны к регистру букв, т.е. не имеет значения набрана ли буква в верхнем или нижнем регистре.**
- **В именах допускается кириллица.**

Переменные

- **Тип переменной определяется двумя способами:**

1) с помощью инструкции

- ***Dim < имя переменной> As <тип переменной>***

Переменные

2) добавлением в конце имени специального символа определения типа:

- **! – тип Single;**
- **@ – тип Currency;**
- **# – тип Double;**
- **$ – тип String;**
- **% – тип Integer;**
- **& – тип Long.**

Константы

- Различают **непоименованные** и **поименованные** константы.
- Непоименованные константы появляются в тесте программы непосредственным указанием некоторого значения.
- В числовых константах целая часть от дробной отделяется десятичной точкой.

Константы

- Допускается экспоненциальная форма записи числовых констант, например запись 6.1435E2 определяет число 614.35.
- Строковые константы заключаются в кавычки, например, "Это строковая константа".
- Константы даты обрамляются знаками #, например, #18/10/04#.
- Введены две логические константы – True и False.

Константы

- Поименованные константы объявляются либо с явным указанием типа, либо без явного указания типа, соответственно инструкциями:
- ***Const <имя>As <тип> = <значение>***
- ***Const <имя> = <значение>***

Выражения в VBA. Использование функций VBA

- ***Выражение* (*expression*) – это значение или группа значений, выражающая отдельное значение.**

- **Каждое выражение *вычисляется до* (или имеет результатом) отдельного значения.**

- **Выражения состоят из одной или более следующих частей: константы, переменные, знаки операций, массивы, элементы массива, функции.**

Арифметические операции

- ***Арифметические операции*** позволяют выполнить все стандартные арифметические действия:
- \+ – операция сложения, например, *A1+A2* складывает *A1* и *A2*;
- – – операция вычитания, например, *A1–A2* вычитает *A2* из *A1*;
- * – операция умножения, например, *A1*A2* умножает *A1* и *A2*;

Арифметические операции

- / – операция деления, например, *A1/A2* делит *A1* на *A2*;
- \ – операция целочисленного деления, например, *A1\ A2* делит *A1* на *A2*, отбрасывая дробную часть;
- **Mod** – операция деления по модулю, например, *A1 Mod A2* делит *A1* на *A2*, возвращая остаток от деления. Остаток – целое число;
- ^ – операция возведения в степень, например, *A1^ A2* возводит *A1* в степень *A2*.

Операции сравнения

Операции сравнения иногда называют также *операциями отношения*.

Результат операции сравнения имеет тип ***Boolean*:**

- = – равно. Синтаксис: *E1=E2*. True, если E1 равно E2, False – в противном случае;
- < – меньше. Синтаксис: *E1<E2*. True, если E1 меньше чем E2, False – в противном случае;

Операции сравнения

- > – больше. Синтаксис: *E1>E2*. True, если E1 больше чем E2, False – в противном случае;
- <= – больше или равно. Синтаксис: *E1 >= E2*. True, если E1 больше чем E2 или рано E2, False – в противном случае;
- <> – не равно. Синтаксис: *E1<>E2*. True, если E1 не равно E2, False – в противном случае

Логические операции

- ***Логические операции*** употребляются для построения сложных логических выражений, где в качестве операндов используются значения типа **boolean** или число, которое можно преобразовать к этому типу. Результат логической операции имеет тип *Boolean*:
- ***And – Конъюнкция***. Выражение *A1 And A2* истинно, если истинны оба его операнды;
- ***Or – Дизъюнкция***. Выражение *A1 Or A2* истинно, если истинен хотя бы один его операнд;
- ***Not – Отрицание***. Выражение *Not A1* истинно, если значение *A1*ложно и наоборот.

- Тип ***String*** допускает единственную операцию слияния, обозначаемую знаком амперсанд "&", которую называют также ***конкатенацией*** строк

- ***& – Конкатенация***. Значение выражения *A1 & A2* есть строка, полученная присоединением к строке *A1* строки *A2*.

Функции VBA и Excel

Встроенные функции VBA по назначению можно разделить на несколько категорий:

- математические функции;
- функции преобразования данных;
- функции даты и времени;
- функции взаимодействия с пользователем;
- строковые функции;
- функции управления файловой системой;
- другие функции.

Математические функции

- **Abs(x)** – абсолютное значение x;
- **Atn(x)** – arctg(x) – в радианах;
- **Cos(x)** – косинус угла x;
- **Exp(x)** – e^x;
- **Fix(x)** – возвращает целую часть числа *x*. Если $x < 0$, то $Fix(x) \geq x$;
- **Int(x)** – возвращает целую часть числа *x*. Если $x < 0$, то $Int(x) \leq x$;
- **Log(x)** – ln(x);

Математические функции

- **Round(x, [d])** – округление числа *x* до *d* десятичных знаков. Если аргумент *d* опущен, производится округление *x* до целого числа;
- **Sgn(x)** – функция знака
- **Sin(x)** – синус угла x;
- **Sqr(x)** – $\sqrt{x}$
- **Tan(x)** – tg(x).

Правила старшинства операций

- VBA поддерживает следующее старшинство операций от большего к меньшему;

- ^ – возведение в степень;
- *, / – умножение, деление;
- \ – операция целочисленного деления;
- Mod – операция деления по модулю;
- +, – – сложение, вычитание;
- & – конкатенация;
- <, <=, >, >=, =, <>– операции сравнения;
- *Not* – отрицание;
- *And* – конъюнкция;

Функции взаимодействия с пользователем

- **InputBox(Prompt, [Title], [Default], [XPos], [YPos], [HelpFile], [Context]) As String** – ввод данных.
- Здесь обязательным аргументом является только ***Prompt*** – строка, используемая для подсказки о вводимой информации.
- Остальные аргументы могут быть опущены, так как не являются обязательными.

Функции взаимодействия с пользователем

- Назначение их следующее:
- ***Title*** – заголовок диалогового окна для ввода информации;
- ***Default*** – значение сроки ввода по умолчанию;
- ***XPos, YPos*** – координаты левого верхнего угла диалогового окна, соответственно горизонтальное и вертикальные расстояния в твипах. Один твип равен 1/20 точки, которая составляет 1/72 дюйма;
- ***HelpFile*** – имя файла помощи в операционной системе Windows;
- ***Context*** – числовое выражение определяющее тематический раздел в файле помощи;

Функции взаимодействия с пользователем

- **MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context]) As VbMsgBoxResult** – вывод данных. Здесь обязательным аргументом является только ***Prompt*** – строка выводимой информации.
- Остальные аргументы могут быть опущены, так как не являются обязательными.

Функции взаимодействия с пользователем

- Назначение ***Title, HelpFile*** **и** ***Context***– те же, что и в функции InputBox.

- Аргумент ***Buttons*** является числовым выражением, определяющим тип кнопок в диалоговом окне вывода информации. По умолчанию значение *vbOKOnly* определяет активную кнопку *ОК*.

Присвоение значений переменным

- Переменная получает свое значение применением к ней оператора присваивания, который обозначается символом равно "=". В общем случае синтаксис следующий
- ***<переменная> = <выражение>***

Совместимость типов данных. Автоматическое преобразование данных

- Не все типы данных совместимы друг с другом, и нельзя использовать несовместимые типы данных в одном и том же выражении.
- Например, не имеет смысла арифметическое сложение строки с числом, так как такое выражение не является значащим и VBA не может его оценить.

Совместимость типов данных.

- Очень важно контролировать и знать тип выражения, потому что если выражения содержат несовместимые типы, VBA выдает ошибку времени исполнения – ошибку ***несовпадения типов*** **(*type-mismatch*).**

Автоматическое преобразование типов

- ***Преобразование числовых типов*** сводится к получению самого точного значения (имеется в виду количество значащих цифр).
- Например, если переменные в выражении типа **Single** и **Integer**, то результат выражения будет иметь тип **Single**, потому что точность типа **Single** выше.
- Если числовой переменной присваивается значение выражения, точность которого выше, то VBA округлит его до той точности, которая присуща переменной.
- Например, если переменной типа **Integer** присвоить значение выражения типа **Double**, значение выражения будет округлено до целого.

Автоматическое преобразование типов

- Преобразование строк и чисел выполняется по следующим правилам:

1)выражение ***Число+строка***

  - выдает сообщение о несоответствии типов, если число имеет числовой тип, а строка тип String;
  - выполняется конкатенация (сцепление), если число имеет тип Variant, а строка тип String; результат – тип String;
  - выполняется арифметическое сложение, если число имеет числовой тип, а строка – тип Variant; результат – числовой тип;

2) ***Число + "3"***

- всегда выполняется арифметическое сложение; результат – числовой тип;

Автоматическое преобразование типов

*3)**3 + строка***

- выполняется арифметическое сложение, если строка имеет числовой тип или тип Variant и в ней содержится текст, который можно интерпретировать как число; результат – числовой тип;
- выдает сообщение о несоответствии типов, если текст в строке нельзя интерпретировать как число;

4)Число & строка

- выполняется конкатенация; результат – тип String.