

Лекция №3

**Процедуры и
операторы.
Переменные.**

Вопросы лекции:

1. Процедуры и операторы.
2. Переменные.
 - Типы переменных.
 - Объявление переменных.
 - Область видимости переменных.
 - Присвоение значений переменных.
 - Нулевые значения переменных.
 - Примеры использования переменных.

Процедуры и операторы

Процедура – это логически законченный блок программного кода. Процедура может принимать какие-либо входные значения и возвращать результат. Использование процедур сокращает объём программы, упрощает её структуру и делает легче отладку программы. Их можно использовать при создании других процедур.

Стандартные процедуры начинаются ключевым словом - оператором **Sub** и заканчиваются оператором **End**.

Процедуры состоят из **операторов** – наименьших жизнеспособных единиц программного кода, которые занимают, как правило, одну строку кода. Операторы осуществляют действия над **процедурами, переменными и константами**.

Переменная (variable) – это объём оперативной памяти, зарезервированный для временного хранения данных. Строго говоря, **переменная** (как контейнер для хранения данных) и хранящаяся в ней информация – это не одно и то же, и надо бы говорить: "информация, представляемая переменной", т. к. надо иметь в виду, что **переменная может не только содержать данные, но и стать командой**.

Каждая переменная имеет собственное имя и чтобы обратиться в программе к содержанию переменной достаточно указать её имя.

Переменные

Правила выбора имен для переменных

Для того чтобы сделать переменные простыми для понимания, следует давать им имена, имеющие определённое смысловое значение.

При выборе имени переменной должны выполняться следующие **правила:**

- имя переменной может содержать **любые** буквы, цифры и символ подчёркивания;
- первый символ в имени переменной должен быть **буквой или символом подчёркивания**;
- в имени переменной **должны отсутствовать** пробелы, знаки пунктуации и символы, используемые для математических операций и операций сравнения;
- имя должно быть **уникальным** внутри области видимости переменной;
- имя переменной **не должно быть** ключевым словом VB.
- Имена переменных "не чувствительны" к состоянию регистра, т. е. употребление в имени переменной прописных или строчных букв значения **не имеет**.

Типы переменных

Переменные могут хранить данные следующих типов:

Тип	Описание	Значение
Byte	1 байт, целое без знака	От 0 до 255
SByte	1 байт, целое со знаком	От -128 до 127
Short	2 байта, целое со знаком	От – 32768 до 32767
UShort	2 байта, целое без знака	От 0 до 65535
Integer	4 байта, целое со знаком	От -2 147 483 648 до 2 147 483 647
UInteger	4 байта, целое без знака	От 0 до 4 294 967 295
Long	8 байтов, целое со знаком	От – 9 223 372 036 854 775 808 до 9 223 372 036 854 775 807
ULong	8 байтов, целое без знака	От 0 до 18 446 744 073 709 551 615
Single	4 байта, с плавающей точкой (одинарной точности)	Отрицательные числа: от – 3.4028235E38 до -1.40129E-45 Положительные числа: От 1.40129E-45 до 3.4028235E38

Double	8 байтов, с плавающей точкой (двойной точности)	Отрицательные числа: от – 1.79769313486231570E308 до 4.940656458412E-324 Положительные числа: от 4.940656458412E-324 до 1.79769313486231570E308
Decimal	16 байтов, с фиксированной точкой	Как целое число со знаком: От – 79 228 162 514 264 337 593 543 950 335 до 79 228 162 514 264 337 593 543 950 335 Фиксированное дробное с точкой (28 разрядов после точки): от – 7.9228162514264337593543950335 до 7.9228162514264337593543950335
Date	8 байтов	Дата от 1 января 0001 года до 31 декабря 9999 года и время от 0:00:00 до 23:59:59
String	Строка символов	от 0 до 2 000 000 000 символов текста в кодировке Unicode (ориентировочно)
Char	2 байта, символы Unicode	65536 символов
Boolean	Различное	Два значения: True или False
Object	Различное	Хранятся данные любого типа

Типы переменных

Переменная типа **Boolean** принимает только два значения: **True** или **False**. При переводе числовых значений в логические 0 становится False, а 1 – True. Когда логические значения переводятся в числовые False становится 0, а True – 1. Начальное значение переменной типа Boolean присваивается (по умолчанию) - False.

Для хранения двоичных чисел используется либо переменная, либо массив чисел типа **Byte**.

Для хранения текстовых данных предназначены переменные типов **Char** и **String**. Первый из них хранит символ в кодировке Unicode, а второй – строку от 0 до 2 млрд. символов. **Строка** – это последовательность символов, заключённая в кавычки.

Значение даты заключается в знаки # (решётка) в формате #месяц /день/год#. Переменной типа Date присваивается (по умолчанию) начальное значение – **12:00 1 января 0001 года**.

Для хранения целых чисел со знаком служат переменные типа Sbyte, Short, Integer, Long. а для чисел без знака - переменные типа Byte, UShort, UInteger, ULong.

Переменные типа **Decimal** применяются в сложных вычислениях, когда недопустимо появление ошибок округления, которые могут возникнуть при обработке чисел с плавающей точкой.

Типы переменных

Преобразование чисел из одних типов данных в другие может быть неявным, когда определённое значение присваивается переменной **автоматически**.

В случае явного преобразования используется конвертирование с помощью методов класса **System.Convert**.

Тип данных **Object** может хранить различные данные и изменять их тип во время выполнения программы.

При разработке программ в зависимости от типа данных переменных можно использовать префиксы, что уменьшает вероятность появления синтаксических ошибок, а именно:

Тип данных	Префикс	Пример
Byte	byt	bytРисунок
Short	shr	shrГод
Integer	int	intРанг
Long	lng	lngCount
Single	sng	sngLength
Double	dbl	dblSum
Decimal	dec	decЦена
String	str	strИмя
Date	dtm	datДеньПобеды
Boolean	bln	blnЦвет
Object	obj	objСотрудник

Объявление переменных (variable declaration)

Самый простой способ создания переменной – это использование ее с оператором (**=**), который называется **оператором присваивания**.

Сохранение значения данных в переменной называется **присваиванием переменной**. Например: **MyVar = 25**, т. е., в ячейку под именем MyVar заносится значение 25.

Создание переменной присваивания ей значения без объявления называется **неявным объявлением переменной** (implicit variable declaration).

Явное объявление переменной (explicit variable declaration) обеспечивается операторами объявления (ключевыми словами): **Dim**, **Private**, **Static** и **Public**, которые имеют следующий синтаксис.

Dim имяПеременной [As типДанных] [= Значение]
Private имяПеременной [As типДанных] [= Значение]
Static имяПеременной [As типДанных] [= Значение]
Public имяПеременной [As типДанных] [= Значение]

Объявление переменных (variable declaration)

Как видно из синтаксиса, объявление и присвоение переменной значения может осуществляться в одной строке, но чаще эти операции разделяются.

Наиболее часто для объявления переменных используется ключевое слово **Dim** (Dimension – размер), создающее пространство для хранения данных, на которое ссылается переменная.

Явное объявление переменной имеет следующие преимущества:

- **ускоряется** выполнение кода;
- уменьшается **количество ошибок** в результате неправильного написания имени переменной;
- код становится более **читабельным и понятным**;
- можно **нормализовать** выделение имени переменной прописными буквами, при этом за основу всегда берется **объявленное** имя.

По умолчанию компилятор VB устанавливает режим явного объявления переменных (Option Explicit On).

Для того чтобы отменить этот режим, нужно в начале программного кода ввести оператор **Option Explicit Off** или выделить в окне **Обозреватель решений** соответствующий проект, выбрать в контекстном меню пункт **Свойства**, затем - вкладку **Компиляция** и

Область видимости переменных

Глобальные переменные объявляются **внутри проекта** (в классе) оператором **Public**. Это означает, что такая переменная имеет доступ в проекте **без каких-либо ограничений**. Такая переменная **не может быть** объявлена **внутри** процедуры.

Переменная, объявленная с ключевым словом **Private**, доступна **только в пределах контекста**, в котором объявлена. Она может быть объявлена внутри модуля или класса, но **не внутри** процедуры.

Переменная, имеющая видимость **внутри модуля**, создаётся в разделе объявлений модуля с объявлением, как правило, оператором **Dim**, но также и оператором **Private**.

Если же переменная объявлена оператором **Dim** внутри **процедуры**, то она становится доступной только в пределах этой процедуры. Такая переменная называется **локальной**.

Локальные переменные, или переменные уровня процедуры, объявляются **только** операторами **Dim** или **Static**.

Если переменная при объявлении указана как **Static**, то она **сохраняет** своё прежнее значение после завершения процедуры, в которой была объявлена. Она **не может быть** объявлена **вне** процедуры.

Область видимости переменных

Присваивая имена переменным с учётом области их действия, для уменьшения ошибок можно использовать префиксы:

Область видимости переменной	Оператор объявления	Префикс	Пример записи
Глобальная внутри проекта	Public	g	gdtmСтарт
Локальная внутри модуля (модульный уровень)	Dim, Private	m	msngДлина
Локальная внутри процедуры	Dim, Static	Нет префикса	strФамилия

При наличии в одной области видимости нескольких переменных с **одним именем** для вычислений используется переменная с **большей степенью локализации**, например:

```
Public intA As Integer
```

```
Module Module1
```

```
    Dim intA As Integer
```

```
Sub Main()
```

```
    Dim intA As Integer
```

```
    intA = 10 'используется локальная переменная из процедуры Sub Main
```

```
    End Sub
```

```
End Module
```

Область видимости переменных

С помощью одного оператора можно объявлять ряд переменных, через запятую, например:

Private Успех As Boolean, Цена As Single

Dim Фамилия, Имя, Отчество As String, Сумма As Double

Указание типа данных (As типДанных) переменной не является обязательным, и, если тип данных не указан, то VB назначит переменной тип значения, которое ей **присваивается**.

Если тип данных при объявлении переменной не указан, и переменной не присвоено какое-нибудь начальное значение, то ей назначается тип **Object**.

Предпочтительным является объявление переменных с указанием типа данных, что снижает вероятность синтаксических ошибок при программировании.

При выполнении программы принципиальное значение имеет область видимости используемых переменных. Попытка применения переменной, которая **не действует** в данном месте программы, приводит к **синтаксической ошибке**.

Присвоение значения или инициализация переменной

Присвоение значения переменной обеспечивается с помощью оператора присвоения (=), при этом аргумент (имя переменной) инициализируется:

- конкретным значением;
- другой переменной;
- константой;
- созданной функцией;
- стандартной функцией VB;
- комплексным выражением.

Например:

```
sngРезультат = sngRest + 346
```

```
strName = "Фирсов" & ": " & strКоманда
```

Нулевые значения переменных

Каждый раз, когда создается переменная, ей по умолчанию приписывается некоторое **начальное** значение означающее, что в переменной "ничего не хранится":

- строки инициализируются строками нулевой длины (" ");
- числа инициализируются нулевым значением (0);
- даты – значением "30 декабря 1899 года". Это значение визуализируется как #12:00:00 AM#.

Если у логической переменной типа Boolean отсутствует определённое значение, то для присвоения ей нулевого значения используется структура **Nullable**.

Например, следует объявить переменную типа Boolean, которая может принимать нулевое значение:

Dim Дети As Nullable (Of Boolean)

По умолчанию при объявлении переменной типа Boolean она принимает значение **False**.

Нулевые значения переменных

Чтобы определить, содержит ли переменная определённое значение или нет, используется свойство **HasValue**.

Если переменной присваивается значение нулевого значения (Nullable), то необходимо провести проверку на его существование:

```
If Дети.HasValue Then  
    MsgBox("Есть дети")  
Else  
    MsgBox("Детей нет")  
End If
```

При этом в начальный момент будет генерироваться сообщение: "Детей нет".

Когда переменная "Дети" примет значение **True** (т. е. в программу перед проверкой условия Дети.HasValue вводится оператор: Дети = True), свойство HasValue также приобретает значение True, и при проверке высвечивается сообщение: "Дети есть".

Если же переменной необходимо присвоить неопределённое значение, то используется константа **Nothing** ("Ничего не известно" или неопределённость), и оператор будет выглядеть:

```
Дети = Nothing
```

Таким образом, результатом логической операции может быть или **False**, или **True**, или **Nothing**.

Примеры использования переменных

```
Dim strПриветствие As String      'используется всеми процедурами модуля
Sub ПриветМир( )
strПриветствие = "Привет, мир!"
MsgBox (strПриветствие, , "Окно приветствия")
End Sub

Sub ПриветПётр( )
strПриветствие = "Привет, Пётр!"
MsgBox (strПриветствие, , "Второе сообщение")
End Sub
```

Объявление переменной strПриветствие в процедурах не обязательно, т. к. она объявлена в **разделе объявлений** модуля и является общей для всех процедур модуля. Но в каждой процедуре будет действовать своё, **локальное** значение переменной. После имени процедуры (в скобках) записываются необходимые аргументы, например, имя файла, к которому относится данная процедура. По умолчанию скобки присутствуют **всегда**.

Например, в структуре оператора MsgBox, визуализирующего результат выполнения процедуры, используются три аргумента:

- первый – содержание сообщения (переменная strПриветствие);
- второй – указание вида окна сообщений MsgBox. Если этот аргумент опускается, т. е., между запятыми – пробел, то в окно (по умолчанию) встраивается только **кнопка ОК**;
- третий – заголовок окна сообщения. Так как заголовок - это текст, то его заключают в кавычки.