



Бесплатная электронная книга

УЧУСЬ

Visual Basic 6

Free unaffiliated eBook created from
Stack Overflow contributors.

#vb6

.....	1
1: Visual Basic 6	2
.....	2
Examples.....	2
.....	2
,	2
2:	3
Examples.....	3
if / else	3
.....	3
Do Loop.....	3
.....	4
3:	5
Examples.....	5
.....	5
.....	5
.....	6
.....	6
4: VB6 Windows 10	8
Examples.....	8
.....	8
5:	9
.....	9
.....	9
.....	9
Examples.....	9
.....	9
.....	12

Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [visual-basic-6](#)

It is an unofficial and free Visual Basic 6 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Visual Basic 6.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с Visual Basic 6

замечания

В этом разделе представлен обзор того, что такое vb6, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в vb6 и ссылаться на связанные темы. Поскольку документация для vb6 является новой, вам может потребоваться создать начальные версии этих связанных тем.

Examples

Установка или настройка

Подробные инструкции по настройке или установке vb6.

Привет, мир

```
' A "Hello, World!" program in Visual Basic.  
Module Hello  
  Sub Main()  
    MsgBox("Hello, World!") ' Display message on computer screen.  
  End Sub  
End Module
```

Прочитайте Начало работы с Visual Basic 6 онлайн: <https://riptutorial.com/ru/vb6/topic/3439/начало-работы-с-visual-basic-6>

глава 2: Основной синтаксис

Examples

if / else

```
If condition Then
    code to execute if true
ElseIf condition Then
    code
Else
    code to execute if conditions are both false
End If
```

для цикла

```
For I as Integer = 1 To 10 Step 1
    code to execute
Next
```

Шаг является необязательным, и шаг 1 по умолчанию. Шаг подсказывает, как считать, поэтому -1 будет вычитать каждый раз каждый раз, а шаг 5 будет добавлять 5 каждый раз через цикл.

В случае, если цикл необходимо остановить, тогда можно использовать оператор `Exit For`, как в приведенном ниже примере;

```
Dim iIndex as integer

For I as Integer = 1 To 10 Step 1

    Debug.Print I
    iIndex = I * 10

    If iIndex > 90 Then
        Exit For
    End If

Loop
```

Здесь вместо того, чтобы печатать от 1 до 10, он остановится на уровне 9, так как условие говорит о прекращении процесса, когда `iIndex` достигает 90.

Do Loop

Другим распространенным типом цикла в Visual Basic является `DO loop`, который будет запускать часть кода непрерывно, пока не будет сказано остановить. Напротив некоторых других циклов, в которых индексы используются для остановки процесса, в этом

конкретном цикле следует остановиться.

Простой пример, иллюстрирующий цикл, выглядит следующим образом

```
Dim iIndex1 As Integer
iIndex1 = 1

Do
    Debug.Print iIndex1
    iIndex1 = iIndex1 + 1

    If iIndex1 = 10 Then
        Exit Do
    End If
Loop
```

Вышеприведенный фрагмент кода примет индекс, инициализированный до 1 и увеличит его. `Debug.Print` поможет распечатать индекс для установки цикла. В каждом цикле код проверяет, достиг ли индекс 10, и если и только если условие истинно, будет выполнен `Exit Do`, который остановит цикл.

Выбор случая

```
Dim number As Integer = 8
Select Case number
    Case 1 To 5
        Debug.WriteLine("Between 1 and 5, inclusive")
        ' The following is the only Case clause that evaluates to True.
    Case 6, 7, 8
        Debug.WriteLine("Between 6 and 8, inclusive")
    Case 9 To 10
        Debug.WriteLine("Equal to 9 or 10")
    Case Else
        Debug.WriteLine("Not between 1 and 10, inclusive")
End Select
```

Прочитайте Основной синтаксис онлайн: <https://riptutorial.com/ru/vb6/topic/9389/основной-синтаксис>

глава 3: переменные

Examples

Переменные типы

Существуют разные типы переменных для разных целей. В Visual Basic 6 доступны следующие типы переменных:

- массив
- логический
- Байт
- валюта
- Дата
- двойной
- целое число
- Долго
- не замужем
- строка
- Вариант

Вы объявляете переменную с помощью ключевого слова `Dim` :

```
Dim RandomNumber As Integer
```

Если вы не укажете тип переменной, переменная будет по умолчанию `Variant` :

```
Dim Foo
```

эквивалентно

```
Dim Foo As Variant
```

ЛОГИЧЕСКИЙ

Boolean - самый простой тип переменной, поскольку он может содержать только одно из двух значений: True или False.

```
Foo = True  
Bar = False
```

Булевы могут использоваться для управления потоком кода:

```
Dim Foo as Boolean
Foo = True

If Foo Then
    MsgBox "True"
Else
    MsgBox "False"
End If
```

целое число

Целое число представляет собой числовой тип данных и может содержать 16-разрядное значение со знаком (от -32768 до +32767). Если вы знаете, что переменная будет содержать только целые числа (например, 9), а не дробные числа (например, 5.43), объявите ее как целочисленный (или длинный) тип данных.

```
Dim RandomNumber As Integer
RandomNumber = 9
```

Целые числа обычно используются в качестве счетчиков в `For...Next` loop:

```
Dim Counter As Integer

For Counter = 0 to 2
    MsgBox Counter
Next Counter
```

Попытка присвоить целочисленному значению значение меньше -32768 или больше 32767 приведет к ошибке во время выполнения:

```
Dim MyNumber As Integer
MyNumber = 40000 'Run-time error '6': Overflow
```

строка

Строковая переменная может содержать пустой текст, символ, слово или текст переменной длины. Строковое значение должно содержаться в кавычках (").

```
Dim Fruit as String
Fruit = "Banana"
```

Если вам нужны кавычки внутри строкового литерала, вы используете две последующие кавычки ("").

```
Dim Quote as String
Quote = "Bill says: ""Learn VB!"""
```


Прочитайте переменные онлайн: <https://riptutorial.com/ru/vb6/topic/7511/переменные>

глава 4: Установка VB6 в Windows 10

Examples

Мастер установки

Мастер установки Visual Studio 6.0

По умолчанию следующие пакеты не устанавливаются должным образом в Windows 10:

- Visual Studio 6 Enterprise
- Visual Studio 6 Professional
- Visual Basic 6 Enterprise
- Visual Basic 6 Professional

Чтобы установить вышеуказанные пакеты, вам нужно будет либо выполнить множество настроек, либо взломать реестр, либо воспользоваться фантастическим [мастером установщика Visual Studio 6.0](#) от Giorgio Brausi.

Перед запуском вам понадобятся следующие элементы:

- Оригинальные компакт-диски и клавиши Visual Studio / Basic Program
- Оригинальные компакт-диски MSDN
- Visual Studio Service Pack 6
- [Мастер установки Visual Studio 6.0](#)
- В Windows 10 build 1511 или более поздней версии вам понадобятся права администратора.

[Мастер](#) проведет вас через необходимые шаги для успешной установки Visual Basic 6.

Обратите внимание, что установка серверных приложений в настоящее время невозможна.

Прочитайте [Установка VB6 в Windows 10 онлайн](#): <https://riptutorial.com/ru/vb6/topic/4230/установка-vb6-в-windows-10>

глава 5: Функциональные процедуры

Вступление

Функция представляет собой ряд операторов, заключенных в выражения «Функция» и «Конечная функция».

Функция выполняет действие и возвращает управление вызывающему. Когда он возвращает управление, он также возвращает значение вызывающему коду.

Вы можете определить функцию в классе, структуре и модуле. По умолчанию оно открыто. Это означает, что вы можете вызывать его из любого места приложения, имеющего доступ к классу, структуре или модулю, в котором вы его определили.

Синтаксис

- [Модификаторы] Функция Name_Of_The_Function [(Arg_List)] Как Return_Type
- [Заявления]
- Конечная функция

замечания

- В этом примере используются два модификатора функций: Public & Private. Эти модификаторы определяют область действия функции.
- Функции с частной областью можно вызывать только из исходного файла, из которого они были определены. В нашем случае его можно вызвать с помощью модуля. И не может вызываться за пределами модуля.
- Функции с общедоступной областью можно вызывать как снаружи, так и внутри модуля. Просто мы можем сказать: «Мы можем назвать это где угодно в программе».
- Модификатор по умолчанию функции является общедоступным.
- По умолчанию аргументы функции передаются по ссылке (в отдельной теме это будет подробно объяснено).

Examples

Создание и вызов функции

Этот пример использует стандартный проект EXE с добавлением файла модуля.

- Создайте новый проект «Стандартный EXE». Таким образом, форма будет добавлена в проект по умолчанию.

- Добавить файл модуля в проект
- Поместите кнопку команды в форму
- Создать командную кнопку Нажмите «Событие».

Код модуля

Создал две функции в модуле. Одна из них - публичная функция (FnAdd). Он принимает два аргумента Integer val_1 & val_2. Он возвращает целое число. Эта функция добавляет два аргумента и возвращает значение вызывающему. Перед добавлением два аргумента проходят процесс в другой функции. Это частная функция. Характеристика / Правила публичной и частной функции, описанные в разделе «Примечания».

```
Public Function FnAdd(val_1 As Integer, val_2 As Integer) As Integer

'Calling private function
val_1 = FnMultiplyBy5(val_1)

'Calling private function
val_2 = FnMultiplyBy5(val_2)

'Function return statement
FnAdd = val_1 + val_2

End Function
```

Ниже приведена Частная функция модуля. Он принимает один целочисленный аргумент val. Он возвращает целое число. Эта функция умножает значение 5 на аргумент и возвращает результат вызывающему.

```
Private Function FnMultiplyBy5(Val As Integer) As Integer

'Function return statement
FnMultiplyBy5 = Val * 5

End Function
```

Код формы

В командной строке нажмите «Событие». Здесь мы вызываем функцию Public Public «FnAdd»,

```
Private Sub Command1_Click()
Debug.Print FnAdd(3, 7)
End Sub
```

Результат в окне Immediate

50

Прочитайте Функциональные процедуры онлайн: <https://riptutorial.com/ru/vb6/topic/9227/>

кредиты

S. No	Главы	Contributors
1	Начало работы с Visual Basic 6	Community , Scheffer
2	Основной синтаксис	Nadeem_MK , Talal Abdoh , user7491506
3	переменные	BlueEel
4	Установка VB6 в Windows 10	ThunderFrame
5	Функциональные процедуры	Jeet