Лекция 5.

Ввод и вывод информации

РХТУ им. Д.И. Менделеева
Каф. ИКТ
Курс создал: ст. преп. А.М. Васецкий

2013 г

Встроенные диалоговые окна

В проектах VBA встречаются две разновидности диалоговых окон: окна сообщений (MsgBox) и окна ввода (InputBox).

Они встроены в VBA, и если их возможностей достаточно, то можно обойтись без проектирования диалоговых окон. Окно сообщений выводит простейшие сообщения для пользователя, а окно ввода обеспечивает ввод информации.

Функция InputBox

Выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа string, содержащее текст, введенный в поле.

Синтаксис:

InputBox (prompt [, title] [, default] [, xpos][, ypos] [, helpfile, context])

Аргументы функции InputBox

prompt – строковое выражение, отображаемое как сообщение в диалоговом окне. Строковое значение prompt может содержать несколько строк. Для разделения строк допускается использование символов chr(10), chr(13) и их комбинации

title – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения

default – строковое выражение, Отображаемое в поле ввода как используемое по умолчанию, если пользователь не введет другую строку. Если этот аргумент опущен, поле ввода изображается пустым

xpos – числовое выражение, задающее расстояние по горизонтали между левой границей диалогового окна и левым краем экрана. Если этот аргумент опущен, диалоговое окно выравнивается по центру экрана по горизонтали

ypos – числовое выражение, задающее расстояние по вертикали между верхней границей диалогового окна и верхним краем экрана. Если этот аргумент опущен, диалоговое окно помещается по вертикали примерно на одну треть высоты экрана

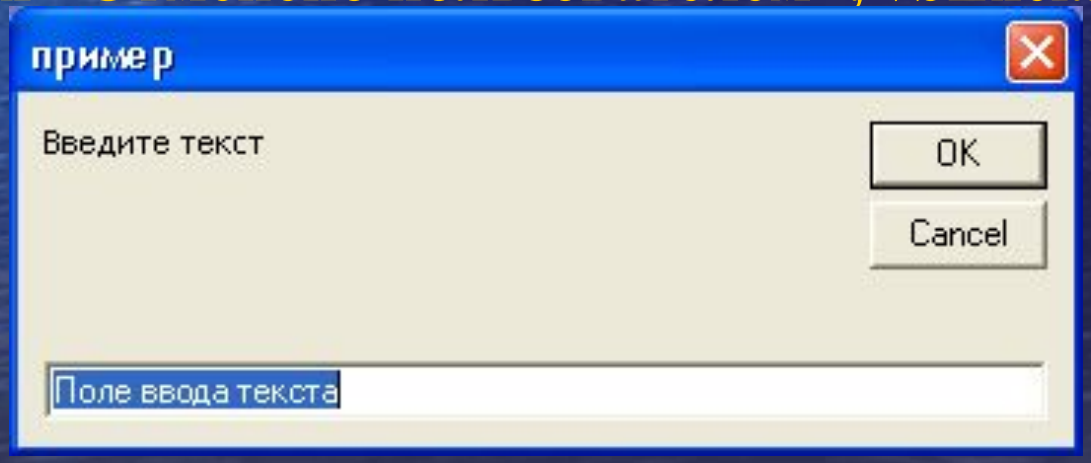
Аргументы функции InputBox (продолжение)

helpfile – строковое выражение, определяющее имя файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот аргумент указан, необходимо наличие также аргумента context

context – числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот аргумент указан, необходимо наличие также аргумента helpfile

Пример InputBox

```
10:
str = InputBox("Введите текст", "пример", "Поле ввода текста")

If Len(str) > 0 Then
   MsgBox "Введено:" & vbCr & str, vbExclamation + vbOKOnly
   Else
   i = MsgBox("Текст не введён!", vbCritical + vbRetryCancel)
   If i = vbRetry Then
      GoTo 10
   Else
      MsgBox "Отменено пользователем", vbExclamation + vbOKOnly
   End If
End If
```



Процедура MsgBox

Выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем, а затем возвращает значение типа Integer, указывающее, какая кнопка была нажата.

Синтаксис:

MsgBox (prompt [, buttons] [, title] [, helpfile, context])

Аргументы функции MsgBox

prompt – строковое выражение, отображаемое как сообщение в диалоговом окне

buttons – числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку и модальность окна сообщения. Значение по умолчанию этого аргумента равняется 0. Значения констант, определяющих число и тип кнопок используемого значка, приведены на следующих слайдах

title – строковое выражение, отображаемое в строке, заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения

helpfile – строковое выражение, определяющее имя файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот аргумент указан, необходимо указать также аргумент context

context – числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот аргумент указан, необходимо указать также аргумент helpfile

Значения аргумента buttons, определяющие кнопки

vbOKOnly	0	
vbOKCancel	1	
vbAbortRetryIgnore	2	
vbYesNoCancel	3	
vbYesNo	4	
vbRetryCancel	5	

Значения аргумента buttons, определяющие отображаемые информационные значки

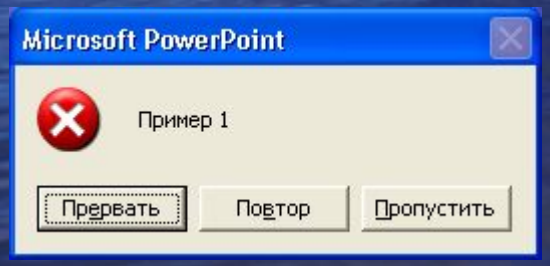
vbCritical	16	MsgBox "Пример", vbOKOnly + vbCritical
vbQuestion	32	MsgBox "Пример", vbOKOnly + vbQuestion
vbExclamation	48	MsgBox "Пример", vbOKOnly + vbExclamation
vbInformation	64	MsgBox "Пример", vbOKOnly + vbInformation 

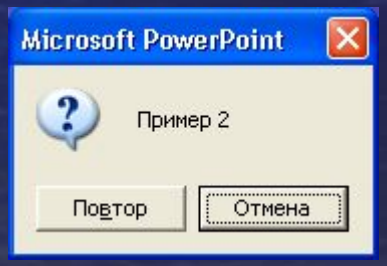
Значения аргумента buttons определяющие основную кнопку

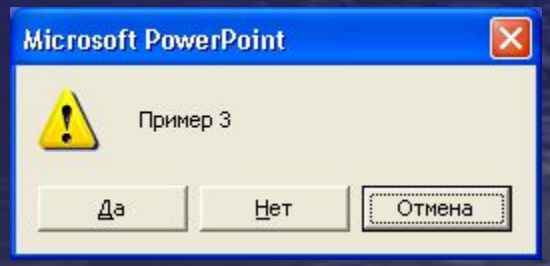
Константа	Значение	Номер основной кнопки
vbDefaultButton1	0	1
vbDefaultButton2	256	2
vbDefaultButton3	512	3
vbDefaultButton4	768	4

Примеры:

1. MsgBox "Пример 1", vbAbortRetryIgnore + vbCritical + vbDefaultButton1
2. MsgBox "Пример 2", vbRetryCancel + vbQuestion + vbDefaultButton2
3. MsgBox "Пример 3", vbYesNoCancel + vbExclamation + vbDefaultButton3







Возвращаемые значения окна MsgBox

Константа	Значение	Нажатая кнопка
vbOK	1	OK
vbCancel	2	Отмена (Cancel)
vbAbort	3	Прервать (Abort)
vbRetry	4	Повторить (Retry)
vbIgnore	5	Пропустить (Ignore)
vbYes	6	Да (Yes)
vbNo	7	Нет (No)

Пример 1.

```
i = MsgBox("Нажмите одну из клавиш",
vbYesNoCancel + vbInformation + vbDefaultButton1)
```

Пример 2.

```
i = MsgBox("Нажмите одну из клавиш",
vbAbortRetryIgnore + vbCritical + vbDefaultButton2)
```

Типы файлов в VBA

Файл последовательного доступа	Рассматривается как последовательность строк произвольной длины, разделенных специальными символами. Чтение и запись в файл производится построчно
Файл произвольного доступа	Состоит из записей фиксированной длины и размер записи указывается при его открытии. Это позволяет локализовать любую запись в файле по ее номеру
Бинарный файл	Является частным случаем файла произвольного доступа. Размер записи в бинарном файле считается равным 1 байту

Открытие файла

Open Открывает файл для работы с ним.
Синтаксис: **Open Путь For Режим [Access Доступ] [Блокировка] As [#] НомерФайла [Len=Длина]**
Путь – строковое выражение, указывающее имя файла
Режим – устанавливает режим работы с файлом.
Допустимые значения: Append, Binary, Input, Output или Random
Доступ – устанавливает операции, разрешенные с файлом.
Допустимые значения: Read, Write или Read Write
Блокировка – устанавливает операции, разрешенные с открытым файлом другим процессам.
Допустимые значения: Shared, Lock Read, Lock Write и Lock Read Write
номерФайла – допустимый номер файла (**1-255**). Параметру НомерФайла предшествует символ #. Значение номерФайла нельзя изменять, пока файл открыт.
длина – число, ≤32 767 (байт). Для файлов, открытых в режиме Random, это значение является длиной записи. Для файлов с последовательным доступом это значение является числом буферизуемых символов

Закрытие файла

Close	Завершает операции ввода/вывода с файлом, открытым с помощью инструкции open. Эта инструкция очищает буфер и указывает операционной системе обновить таблицу размещения файлов. **Каждый файл по завершении работы с ним должен быть закрыт, иначе это может привести к частичной потере информации.** Синтаксис: Close [СписокНомеровФайлов] Аргумент СписокНомеровФайлов может представлять один или несколько номеров файлов. При этом используется следующий синтаксис, где номерФайла представляет любой допустимый номер файла: [[#] номерФайла] [, [#] номерФайла]… Если опустить список аргументов, то будут закрыты все открытые файлы

Прочие функции для работы с файлами

Reset	Закрывает все активные файлы, открытые с помощью инструкции open, и записывает содержимое всех буферов файлов На диск, открытых с помощью инструкции Open
FreeFile	Функция возвращает доступный номер, который может использоваться в инструкции Open

```
Sub FilOpen()
'Открывает файл для ввода данных из него.
'Файл должен существовать
Open "File.inp" For Input As #1

'Открывает файл для бинарного вывода
Open "Filebin.txt" For Binary Access Write As #2

'Открывает файл для вывода
'Другим процессам разрешено пользоваться этим файлом для
'операций ввода/вывода
Open "FileOutput.txt" For Output Shared As #3
Close
End Sub
```

File System Objects (FSO)

FSO позволяет открывать файлы как текстовой поток и читать и писать в них **object.OpenTextFile(filename[, iomode[, create[, format]]])**

- **filename** – полный путь к файлу
- **iomode** – режим ввода/вывода
- **create** – Логическое значение, показывающее будет ли создан новый файл, если файл filename не найден (False – по умолчанию)
- **format** – формат открываемого файла. По умолчанию - ASCII

Значения констант для открытия файлов через FSO

iomode принимает следующие значения:
ForReading = 1 – открывает файл для чтения
ForAppending =8 – открывает файл для добавления

format принимает следующие значения:
TristateUseDefault =–2 для открытия файла используются системные установки
TristateTrue =-1 – открывает файл в кодировке Unicode.
TristateFalse =0 – открывает файл в кодировке ASCII

Открытие файла, как объекта FSO
Метод OpenTextFile

```
Sub OpenTextFileTest()
Const ForReading = 1, ForWriting = 2
Const ForAppending = 8
Const TristateFalse =0
Dim fs, f
Set fs=CreateObject("Scripting.FileSystemObject")
Set f = fs.OpenTextFile("c:\testfile.txt",
ForAppending,TristateFalse)
   f.Write "Hello world!"
   f.Close
End Sub
```

OpenAsTextStream метод

Метод открывает файл и делает возможным чтение/запись в него

***object*.OpenAsTextStream([*iomode*, [*format*]])**

- *object* – файловый объект
- *iomode* – режим записи/чтения
- *format* – формат файла

Значения констант OpenAsTextStream

iomode принимает следующие значения:
ForReading = 1 – открывает файл для чтения
ForWriting = 2 – открывает файл для записи. Если файл существовал, то его содержимое предварительно стирается
ForAppending = 8 – открывает файл для добавления

format принимает следующие значения:
TristateUseDefault = –2 для открытия файла используются системные установки
TristateTrue = -1 – открывает файл в кодировке Unicode.
TristateFalse = 0 – открывает файл в кодировке ASCII

Пример

```
Sub TextStreamTest()
    Const ForReading = 1, ForWriting = 2, ForAppending = 3
    Const TristateUseDefault = -2, TristateTrue = -1, TristateFalse = 0
    Dim fs, f, ts, s
    Set fs = CreateObject("Scripting.FileSystemObject")
    fs.CreateTextFile "test1.txt"              'Create a file
    Set f = fs.GetFile("test1.txt")
    Set ts = f.OpenAsTextStream(ForWriting, TristateUseDefault)
    ts.Write "Hello World"
    ts.Close
    Set ts = f.OpenAsTextStream(ForReading, TristateUseDefault)
    s = ts.ReadLine
    MsgBox s
    ts.Close
End Sub
```

FSO. Список папок

```
Private Sub GetFolders()
Dim FSO As Object, Drive As Object, Folders As Object
Dim Folder As Object
    ' формируем ссылку на объект FileSystemObject
    Set FSO = CreateObject("Scripting.FileSystemObject")
    ' получаем доступ к диску D:\
    Set Drive = FSO.GetFolder("D:\")
    ' фрмируем коллекцию каталогов
    Set Folders = Drive.SubFolders
    With ListBox1 'устанавливаем свойства списка
        .ColumnHeads = False
        .MultiSelect = fmMultiSelectSingle
    End With
    For Each Folder In Folders     'Начинаем перебор коллекции
        ListBox1.AddItem Folder.Name
    Next
End Sub
```

Функции по работе с файлами

ChDir	Изменяет текущую папку. Синтаксис: **ChDir путь**
ChDrive	Изменяет текущий диск. Синтаксис: **ChDrive диск**. Например, ChDrive "D"
CurDir	Функция возвращает текущую папку
Dir	Возвращает строку, представляющую имена файла и папки, соответствующие определённому критерию или метку диска Синтаксис: **Dir[(Путь[, атрибуты])]** **Путь** – путь к файлу/папке **Атрибуты** – атрибуты файла/папки (См. ниже)
FileAttr	Возвращает значение типа Long, представляющее режим файла, открытого с помощью инструкции open. Возвращаемые значения: **1** (для режима input), **2** (output), **4** (Random), **8** (Append) и **32** (Binary). Синтаксис: **FileAttr (НомерФайла, Тип)** **НомерФайла** – допустимый номер файла **Тип** – число, указывающее характер возвращаемых данных. Если тип установлен равным 1 , то функция FileAttr возвращает значение, указывающее режим работы файла

Функции по работе с файлами (продолжение)

GetAttr	Возвращает значение типа integer, определяющее атрибуты файла, каталога или папки. Значение, возвращаемое функцией GetAttr, является суммой констант, приведенных ниже. Синтаксис: GetAttr (путь)
SetAttr	Устанавливает атрибуты файла. Синтаксис: SetAttr pathname, attributes Атрибуты в аргументе attributes определяются как сумма констант из таблицы ниже.

Константы атрибутов файла

Константа	Значение	Описание
vbNormal	0	Обычный
vbReadOnly	1	Только чтение
vbHidden	2	Скрытый
vbSystem	4	Системный
vbDirectory	16	Каталог или папка
vbArchive	32	Файл был изменен после последнего резервирования

Функции по работе с файлами (продолжение)

FileCopy	Копирует файл. Синтаксис:FileCopy source, destination Аргументы: • source – строковое выражение, указывающее имя копируемого файла • destination – строковое выражение, указывающее имя результирующего файла. Аргумент destination может содержать имя каталога или папки и диска
FileDateTime	Функция возвращает дату и время последнего изменения файла. Синтаксис: FileDateTime(путь)

Функции по работе с файлами (продолжение)

Kill	Удаляет существующий файл. Синтаксис: Kill путь В аргументе путь допустимо использование символов (*) и (?) для удаления нескольких файлов по маске.
MkDir	Создает новую папку. Синтаксис: MkDir путь
RmDir	Удаляет существующую папку. Синтаксис: RmDir путь

Примеры. Функция DIR

```
Sub Get_All_File_from_Folder()
    Dim sFolder As String, sFiles As String
    'диалог запроса выбора папки с файлами
    With Application.FileDialog(msoFileDialogFolderPicker)
        If .Show = False Then Exit Sub
        sFolder = .SelectedItems(1)
    End With
    sFolder = sFolder & IIf(Right(sFolder, 1) = Application.PathSeparator, _
    "", Application.PathSeparator)
    'отключаем обновление экрана, чтобы наши действия не мелькали
    Application.ScreenUpdating = False
    sFiles = Dir(sFolder & "*.xls*")
    Do While sFiles <> ""

        Workbooks.Open sFolder & sFiles 'открываем книгу
        'действия с файлом
        'Запишем на первый лист книги в ячейку A1 - test
'Правда здесь есть вероятность ошибки, если первый лист имеет тип Chart
        ActiveWorkbook.Sheets(1).Range("A1").Value = "test"
        'Закрываем книгу с сохранением изменений
        ActiveWorkbook.Close True 'если поставить False - книга будет
        'закрыта без сохранения
        sFiles = Dir 'Ищем следующий файл
    Loop
    'возвращаем ранее отключенное обновление экрана
    Application.ScreenUpdating = True
End Sub
```

Ввод данных в файл последовательного доступа. Оператор Write

Записывает неформатированные данные в файл последовательного доступа. В отличие от инструкции Print, инструкция Write вставляет запятые между элементами и заключает строки в кавычки по мере записи их в файл.

Синтаксис:

Write #НомерФайла, [СписокВывода]

- НомерФайла – номер файла
- СписокВывода – выражение или список выражений, записываемых в файл.

Данные, записанные с помощью инструкции Write, обычно считываются из файла с помощью инструкции Input

Ввод данных в файл последовательного доступа. Оператор Print

Записывает форматированные данные в файл последовательного доступа. Синтаксис:

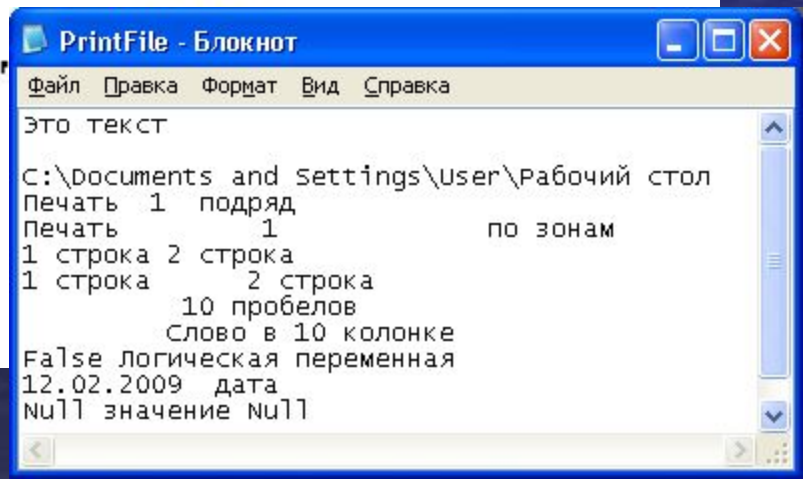
Print #НомерФайла, [СписокВывода]

- номерФайла – номер файла
- списокВывода – выражение (или список выражений), записываемое в файл.

В аргументе СписокВывода разделителем списка выводимых выражений является ";" (данные выводятся подряд) или "," (данные выводятся по зонам). Кроме того, в аргументе СписокВывода допускается использование функций Spc и Tab:
Spc(n) – используется для вставки n пробелов в файл
Tab(n) – устанавливает курсор в столбец с номером n

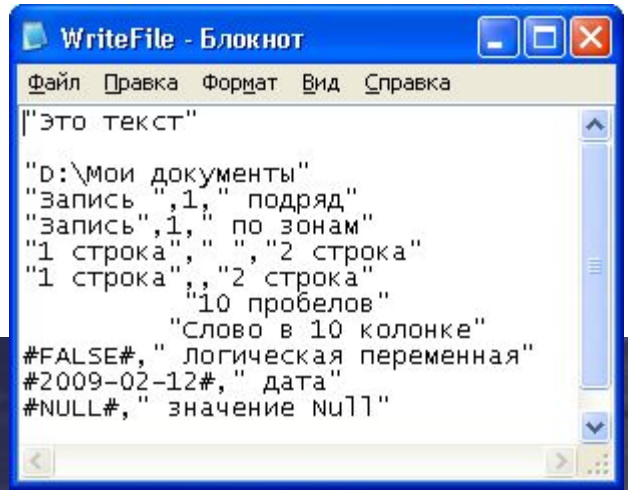
Пример использования оператора Print

```
Sub PrintFun()
'Открываем файл для записи
Open "PrintFile.txt" For Output As #1
'Вывод в файл текстовой строки
Print #1, "Это текст"
Print #1,     'пустая строка.
Print #1, CurDir 'Текущая папка (по умолчанию папка исходного файла)
Print #1, "Печать "; 1; " подряд"      ' Печать подряд
Print #1, "Печать", 1, " по зонам"     ' Печать в 2 зоны
Print #1, "1 строка"; " "; "2 строка"       'Строки с пробелом
Print #1, "1 строка"; Tab; "2 строка"       'Строки с табулятором
Print #1, Spc(10); "10 пробелов"
Print #1, Tab(10); "Слово в 10 колонке"

Dim MyBool, MyDate, MyNull, MyError
MyBool = False: MyDate = #2/12/2009#: MyNull = Null

Print #1, MyBool; " Логическая переменная"
Print #1, MyDate; " дата"
Print #1, MyNull; " значение Null"

Close #1
End Sub
```



Пример применения оператора Write

```
Sub WriteFun()
'Открываем файл для записи
Open "WriteFile.txt" For Output As #1
'Вывод в файл текстовой строки
Write #1, "Это текст"
Write #1,     'пустая строка.
Write #1, CurDir 'Текущая папка (по умолчанию папка исходного файла)
Write #1, "Запись "; 1; " подряд"      ' Запись подряд
Write #1, "Запись", 1, " по зонам"     ' Запись в 2 зоны
Write #1, "1 строка"; " "; "2 строка"       'Строки с пробелом
Write #1, "1 строка"; Tab; "2 строка"       'Строки с табулятором
Write #1, Spc(10); "10 пробелов"
Write #1, Tab(10); "Слово в 10 колонке"

Dim MyBool, MyDate, MyNull, MyError
MyBool = False: MyDate = #2/12/2009#: MyNull = Null

Write #1, MyBool; " Логическая переменная"
Write #1, MyDate; " дата"
Write #1, MyNull; " значение Null"

Close #1
End Sub
```



Ввод данных из файла последовательного доступа. Оператор Input

Input #	Считывает данные из открытого файла последовательного доступа и присваивает их переменным. Данные, считываемые с помощью инструкции input #, обычно записываются в файл с помощью инструкции Write #. Синтаксис: Input #НомерФайла, СписокПеременных НомерФайла – номер файла СписокПеременных – список переменных, которым следует присвоить значения, считанные из файла. Переменные в списке разделяются запятыми
Input	Возвращает значение типа string, содержащее символы из файла, открытого в режиме input или Binary. Функция input считывает данные, записываемые в файл с помощью инструкции Print # или Put. Синтаксис: **Input (Число, [#] НомерФайла)** **Число** задает число возвращаемых символов. Если аргумент Число равен 1, то производится посимвольное считывание данных.

Ввод данных из файла последовательного доступа

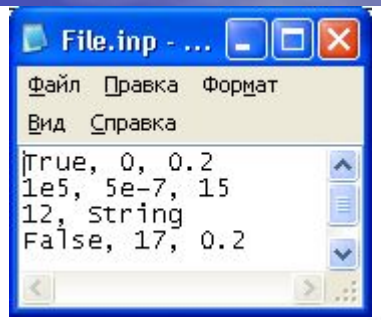
Line Input #	Считывает строку из открытого файла последовательного доступа и присваивает ее переменной типа string. Данные, считываемые с помощью инструкции Line input #, как правило, записываются в файл с помощью инструкции Print #. Синтаксис: **Line Input #НомерФайла, ИмяПеременной** НомерФайла – номер файла ИмяПеременной – имя переменной типа Variant или String
EOF	Функция возвращает значение True при достижении конца файла. Синтаксис: **EOF (НомерФайла)** При последовательном считывании информации из файла часто используется следующий цикл: Do While Not EOF(1) Loop или, эквивалентный цикл: While Not EOF (1) Wend Прим. Для бинарных файлов вместо этой функции используется LOF

Пример использования оператора Input

```
Dim MyChar
Open "FILE.tst" For Input As #1
‘Цикл до конца файла
Do While Not EOF(1)
‘Считываем 1 символ
   MyChar = Input(1, #1)
‘Вывод в окно Immediate
   Debug.Print MyChar
Loop
Close #1
```

Пример

```
Sub FileInp()
Dim IM(1 To 3) As Integer
Dim RL(1 To 3) As Double
Dim FL(1 To 2) As Boolean
Dim St(1 To 5) As String

St(3) = "D:\Мои Документы-AMVAS\AMVAS-4_Work\Educate-Programming\Samples"
ChDir (St(3)) 'изменяем путь
St(4) = CurDir
Open "File.inp" For Input Access Read As #1
St(2) = EOF(1)
'Do While Not EOF(1)
    Input #1, FL(1), IM(1), RL(1)
    Input #1, RL(2), RL(3), IM(2)
    Input #1, IM(3), St(1)
    Line Input #1, St(2)
'Loop
Close #1
End Sub
```

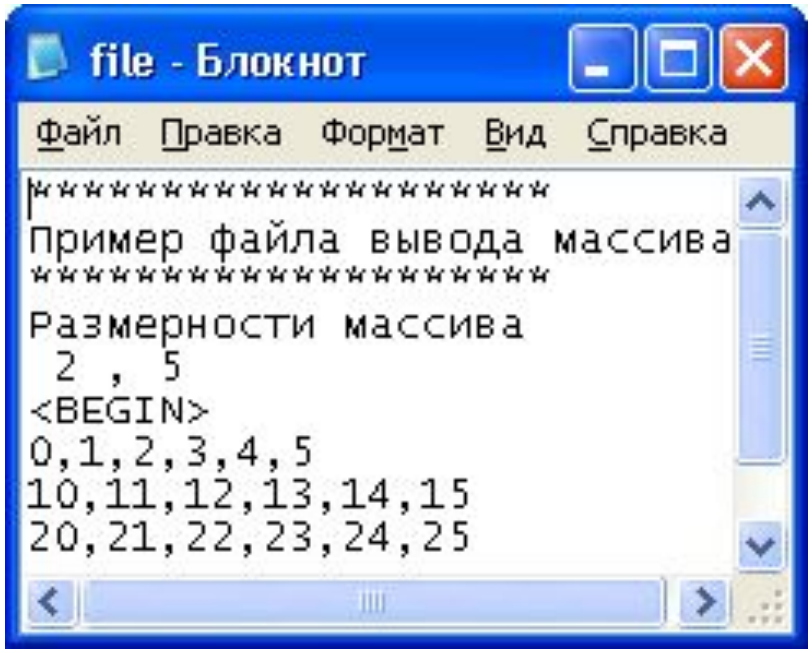


fl	
fl(1)	True
fl(2)	False
im	
im(1)	0
im(2)	15
im(3)	12
rl	
rl(1)	0,2
rl(2)	100000
rl(3)	0,0000005

St	
St(1)	"String"
St(2)	"False, 17, 0.2"
St(3)	"D:\Мои Документы-AMVAS\AMVAS-4_Work\Educate-Programming
St(4)	"D:\Мои Документы-AMVAS\AMVAS-4_Work\Educate-Programming

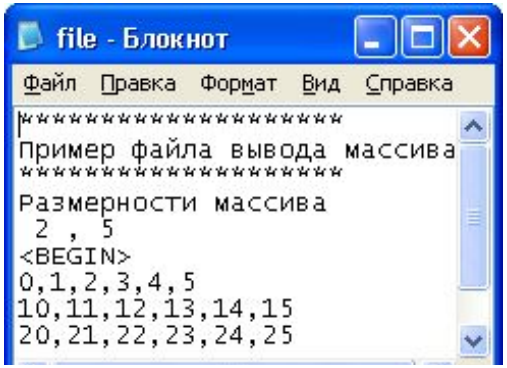
Пример. Вывод массива в файл

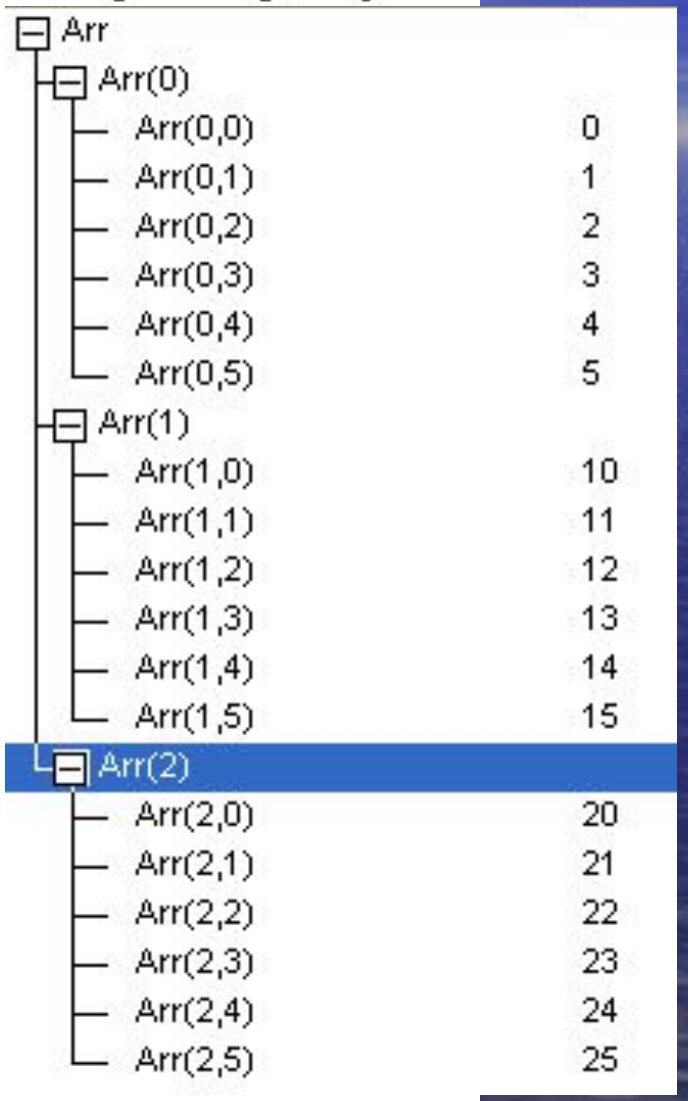
```
Sub ArrOutput()
Const M As Byte = 2, N As Byte = 5
Const cDir = "D:\Мои Документы-AMVAS\AMVAS-4_Work\Educate-Programming\Samples"
Dim Arr(M, N) As Integer
Dim i As Byte, j As Byte
Dim str As String

ChDir (cDir)
Open "file.out" For Output As #1
Print #1, String(20, "*")
Print #1, "Пример файла вывода массива"
Print #1, String(20, "*")
Print #1, "Размерности массива"
Print #1, M; ","; N
Print #1, "<BEGIN>"
For i = 0 To M
    str = ""
    For j = 0 To N
        Arr(i, j) = i * 10 + j
        str = str & CStr(Arr(i, j)) 'Заполнение строки
        If j < N Then str = str & "," 'добавление разелителя
    Next j
    Print #1, str
Next i
Print #1, "<END>"
Close #1
End Sub
```



Пример. Ввод массива из файла

```
Sub ArrInput()

Const cDir = "D:\Мои Документы-AMVAS\AMVAS-4_Work\Educate-Programming\Samples"
Const stStart = "<BEGIN>"
Dim M As Byte, N As Byte
Dim i As Byte, j As Byte
Dim Arr() As Integer
Dim str As String * 10

ChDir (cDir)
Open "file.out" For Input As #1

For i = 1 To 4
    Input #1, str ' считывание строк комментариев
Next i
Input #1, M, N 'считывание размерностей
ReDim Arr(M, N)

Do
    Input #1, str
    str = Mid(str, 1, 7)
Loop While Trim(str) <> stStart

For i = 0 To M
    For j = 0 To N
        Input #1, Arr(i, j)
    Next j
Next i
Close #1
End Sub
```





Работа с файлом произвольного доступа

Put

Записывает содержимое переменной в файл произвольного доступа.

Синтаксис: **Put [#] НомерФайла, [НомерЗаписи] , ИмяПеременной**

- НомерФайла – номер файла
- НомерЗаписи – номер записи (режим Random) или номер байта (режим Binary), с которого следует начать запись. Если аргумент НомерЗаписи опущен, то записывается на то место, где был установлен указатель после выполнения последней инструкции Get или Put, либо куда он переведен с помощью функции Seek
- ИмяПеременной – имя переменной, содержащей данные, которые следует записать в файл

Get

Читает данные из открытого файла произвольного доступа в переменную. Синтаксис:

Get [#] НомерФайла, [НомерЗаписи], ИмяПеременной

- НомерФайла – номер файла
- НомерЗаписи – номер записи (режим Random) или номер байта (режим Binary), с которого следует начать чтение
- ИмяПеременной – имя переменной, в которую следует поместить считанные данные

Работа с файлом произвольного доступа

Seek	Функция возвращает значение типа Long, определяющее текущее положение указателя чтения/записи внутри файла, открытого с помощью инструкции Open. Синтаксис: Seek(НомерФайла)
LOF	Функция возвращает значение типа Long, представляющее размер файла в байтах, открытого с помощью инструкции Open. Для определения размера закрытого файла следует использовать функции FileLen. Синтаксис: LOF(НомерФайла)
FileLen	Возвращает значение типа Long, содержащее размер файла в байтах. Синтаксис: FileLen(Путь)
Loc	Возвращает текущую позицию указателя в файле. Синтаксис: Loc(НомерФайла)

Пример записи

```
Type Record ' Пользовательский тип.
    ID As Integer
    Name As String * 20
End Type

Dim MyRecord As Record, RecordNumber
‘Открываем для прямой записи
Open "TESTFILE" For Random As #1 Len = Len(MyRecord)
For RecordNumber = 1 To 5 ' 5 оборотов цикла
    MyRecord.ID = RecordNumber ' Заполняем поля
MyRecord.Name = "My Name" & RecordNumber
‘Записываем записи в файл
    Put #1, RecordNumber, MyRecord
Next RecordNumber
Close #1
```

Пример чтение из файла

```
‘Открываем тестовый файл
Open "TESTFILE" For Random As #1
Len = Len(MyRecord)
Position = 3 ' номер записи.
' читаем третью запись
Get #1, Position, MyRecord
Close #1
```

Выбор файла пользователем

Если приложению необходимо получить от пользователя имя файла, то это можно сделать несколькими путями.

Например через GetOpenFilename объекта Application.

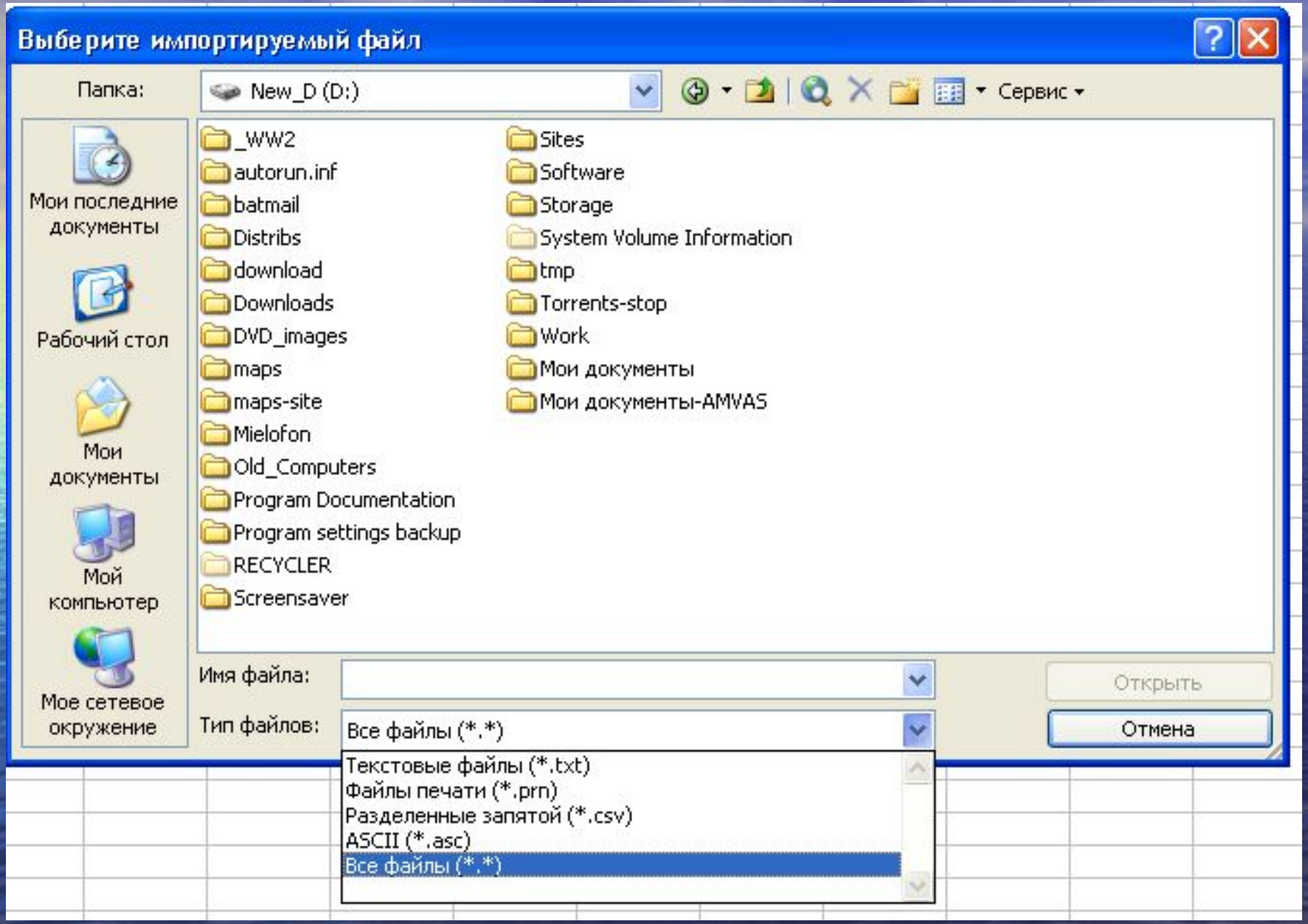
Более общим приёмом является использование метода FileDialog.

Использование InputBox возможно, но не рекомендуется в силу большой вероятности ошибок и неудобства для пользователя.

Пример кода GetOpenFileName

```
Option Explicit

Sub GetImportFileName()
Dim Filt As String
Dim FilterIndex As Integer
Dim FileName As Variant
Dim Title As String
' Настройка списка фильтров
    Filt = "Текстовые файлы (*.txt),*.txt," & _
           "Файлы печати (*.prn),*.prn," & _
           "Разделенные запятой (*.csv),*.csv," & _
           "ASCII (*.asc),*.asc," & _
           "Все файлы (*.*),*.*"
' По умолчанию используется фильтр *.*
    FilterIndex = 5
' Заголовок окна
    Title = "Выберите импортируемый файл"
'Получение имени файла
    FileName = Application.GetOpenFilename _
    (FileFilter:=Filt, FilterIndex:=FilterIndex, Title:=Title)
' При отмене выйти из окна
    If FileName = False Then
        MsgBox "Файл не выбран"
        Exit Sub
    End If
' Отображение полного имени и пути
MsgBox "Вы выбрали " & FileName
End Sub
```

Диалоговое окно



FileDialog

```
Sub ShowFileDialog()
Dim Fd As FileDialog, i As Integer, btCancel As Integer
Dim FileNames() As String
'Задаём тип диалогового окна
Set Fd = Application.FileDialog(msoFileDialogOpen)
'Получение имени файла
With Fd
    With .Filters                                  'Добавляем фильтры
        .Clear                                     'Очищаем старые фильтры
        .Add "Текстовые файлы", "*.txt"
        .Add "Файлы печати", "*.prn"
        .Add "Все файлы", "*.*"
    End With
    .AllowMultiSelect = True                       'Разрешаем выбор нескольких файлов
    .Title = "Выберите файлы"                       'Назначаем заголовок окна
    .ButtonName = "ВЫБОР"                          'Надпись на кнопке
    .FilterIndex = .Filters.Count                  'По умолчанию выбраны "все файлы"
    .InitialView = msoFileDialogViewDetails        'Задан вид по умолчанию
    btCancel = .Show                               'Показываем форму пользователю
    If btCancel <> 0 Then
            ReDim FileNames(1 To .SelectedItems.Count) 'Переразмериваем
            For i = 1 To .SelectedItems.Count
            'Здесь содержится полные пути к выбранным файлам
                FileNames(i) = .SelectedItems.Item(i)
            Next i
        Else
        ReDim FileNames(1 To 1)                      'Переразмериваем
            FileNames(1) = ""                         'Пользователь выбрал "Отмена"
    End If
    For i = 1 To UBound(FileNames)
        MsgBox "Выбран файл: " & FileNames(i)
    Next i
End With
End Sub
```

Выбор папки

Когда требуется выбрать папку, то можно использовать метод BrowseForFolder объекта Application

```
Sub FolderOpen()
    Dim objShell, objFolder, objFolderItem
    Dim strPath As String

    On Error GoTo Problem
    Set objShell = CreateObject("Shell.Application")
    Set objFolder = objShell.BrowseForFolder(0, "Выберите папку:", &H4000, "")
    Set objFolderItem = objFolder.Self
    strPath = objFolderItem.Path
    MsgBox strPath
    Exit Sub
Problem:
    MsgBox "Выбор отменен"
End Sub
```

Выбор папки при помощи FileDialog

```
Sub PickFolderDialog()
'Выбор папки
Dim Fd As FileDialog, i As Integer, btCancel As Integer
Dim Folder As String

'Задаём тип диалогового окна
Set Fd = Application.FileDialog(msoFileDialogFolderPicker)
'Получение имени папки
With Fd
    .AllowMultiSelect = False                  'Запрещаем выбор нескольких папок
    .Title = "Выберите папку"                  'Назначаем заголовок окна
    .ButtonName = "ВЫБОР"                     'Надпись на кнопке
    .InitialView = msoFileDialogViewDetails 'Задан вид по умолчанию
    btCancel = .Show                          'Показываем форму пользователю
    Folder = ""
    If btCancel <> 0 Then Folder = .SelectedItems.Item(1) 'Выбранный путь
End With

MsgBox "Выбрана папка: " & Folder
End Sub
```

Процедуры обработки ошибок

При составлении приложений важно предусмотреть, чтобы программа анализировала возможные ошибки, возникающие при ее выполнении по вине пользователя, и информировала. При этом возможно два подхода:

- Предотвращение ошибок: программно анализировать вводимые или вычисляемые данные и в случае, если они могут приводить к ошибке, обеспечить, чтобы программа информировала пользователя о необходимости корректного задания данных.
- Обработка ошибок: в случае появления ошибки, перехватить ее, обработать и программно откликнуться на возникшую ошибку.

При создании приложений надо сочетать оба подхода, применяя в каждом конкретном случае и для каждой возможной ошибки тот, который кажется разработчику наиболее эффективным.

Функция перехвата ошибок On Error

On Error производит перехват ошибки. Устанавливает, что программа должна делать в случае появления ошибки.

Допустимы следующие синтаксисы.

- Синтаксис 1: **On Error GoTo строка**

Активизирует подпрограмму обработки ошибок, начало которой определяется обязательным аргументом строка, значением которого может быть либо метка строки, либо номер строки.

- Синтаксис 2: **On Error Resume Next**

Указывает, что при возникновении ошибки происходит передача управления на инструкцию, непосредственно следующую за инструкцией, где возникла ошибка.

- Синтаксис 3: **On Error GoTo 0**

Отключает любой активизированный обработчик ошибок в текущей процедуре.

Инструкция Resume

Обеспечивает процедуре возможность продолжить работу после обработки ошибки. Допустимы следующие синтаксисы.

- Синтаксис 1: Resume

  После обработки ошибки управление передается той инструкции, в которой произошла ошибка.

- Синтаксис 2: Resume строка

  После обработки ошибки управление передается инструкции, определенной аргументом строка. Значением этого аргумента может быть любая метка строки или номер строки.

- Синтаксис 3: Resume Next

  После обработки ошибки управление передается инструкции, следующей за инструкцией, в которой произошла ошибка.

Инструкция Exit

Инструкция Exit останавливает выполнение процедуры.
Допустимые синтаксисы:

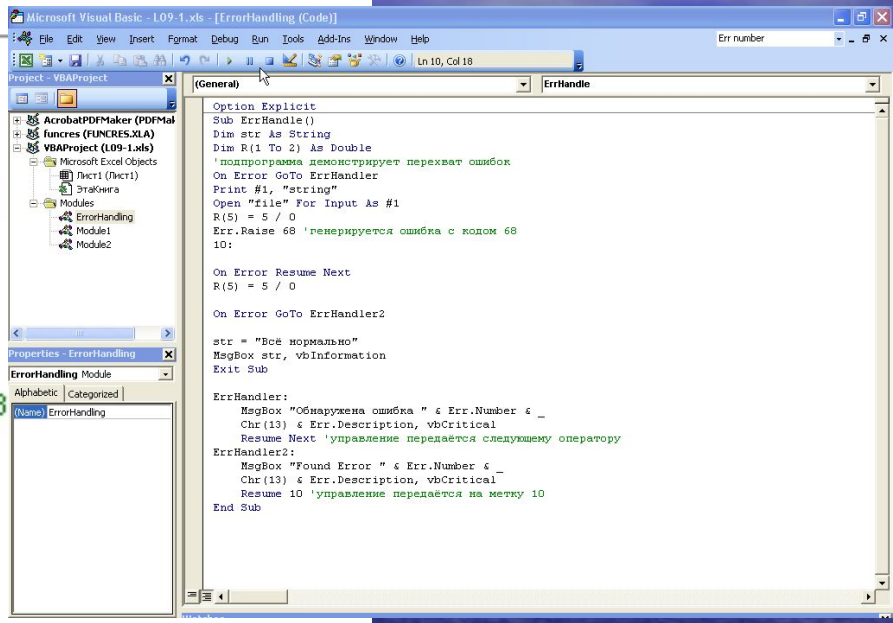
- Exit Sub
- Exit Function
- Exit Property

Свойства объекта Err

Number	Возвращает код ошибки
Source	Имя текущего проекта VBA
Description	Возвращает строковое выражение, содержащее текст сообщения об ошибке
HelpFile	Полное имя (включая диск и путь) файла справки VBA
HelpContext	Контекстный идентификатор файла справки VBA, соответствующий ошибке с кодом, указанным в свойстве Number
LastDLLError	Содержит системный код ошибки для последнего вызова библиотеки динамической компоновки (DLL)

Пример перехвата ошибок

```
Option Explicit
Sub ErrHandle()
Dim str As String
Dim R(1 To 2) As Double
'подпрограмма демонстрирует перехват ошибок
On Error GoTo ErrHandler
Print #1, "string"
Open "file" For Input As #1
R(5) = 5 / 0
Err.Raise 68 'генерируется ошибка с кодом 68
10:

On Error Resume Next
R(5) = 5 / 0

On Error GoTo ErrHandler2

str = "Всё нормально"
MsgBox str, vbInformation
Exit Sub

ErrHandler:
    MsgBox "Обнаружена ошибка " & Err.Number & _
    Chr(13) & Err.Description, vbCritical
    Resume Next 'управление передаётся следующему оператору
ErrHandler2:
    MsgBox "Found Error " & Err.Number & _
    Chr(13) & Err.Description, vbCritical
    Resume 10 'управление передаётся на метку 10
End Sub
```



Коды ошибок

Код	Сообщение
3	Return без вызова процедуры
5	Аргумент превышает допустимое значение
6	Переполнение
7	Не хватает памяти
9	Индекс за пределами диапазона
10	Массив недоступен для переразмеривания.
11	Деление на 0
13	Несоответствие типов
14	Слишком большая строка

Коды ошибок

Код	Сообщение
16	Слишком сложное выражение
17	Невозможно выполнить операцию
18	Прерывание, вызванное пользователем
20	Оператор Resume без наличия ошибки
28	Недостаточно стека
35	Функция, подпрограмма или свойство не определены
52	Неправильное имя файла или идентификатор.
53	Файл не найден

Коды ошибок

Код	Сообщение
54	Неверный режим работы с файлом
55	Файл уже открыт
57	Ошибка ввода-вывода
58	Файл уже существует
59	Некорректная длина записи
61	Диск заполнен
62	Попытка чтения из файла, после достижения его конца
63	Некорректный номер записи
67	Слишком много файлов

Коды ошибок

Код	Сообщение
68	Устройство недоступно
70	Доступ к ресурсу закрыт
71	Диск не готов
74	Попытка перемещения файла на другой диск при помощи оператора Name
75	Ошибка файла/пути
76	Путь не найден
91	Объектная переменная или With блок не заданы
92	Вход внутрь цикла For извне

Коды ошибок

Код	Сообщение
93	Некорректно задана строка в операторе Like
94	Некорректное использование Null
320	Некорректные символы в имени файла
321	Некорректный формат файла
322	Невозможно создать временный файл
325	Некорректный формат файла ресурсов
335	Невозможен доступ к системным ресурсам

СПАСИБО ЗА ВНИМАНИЕ!