

ОБРАБОТКА ОШИБОК

Типы ошибок:

- Синтаксические
- Ошибки выполнения (run-time)
- Логические (алгоритмические) ошибки

Отладка программы – процесс пошагового исполнения программы для проверки наличия логических (алгоритмических ошибок)

Обработка ошибок – это процесс отслеживания ошибок во время исполнения программы

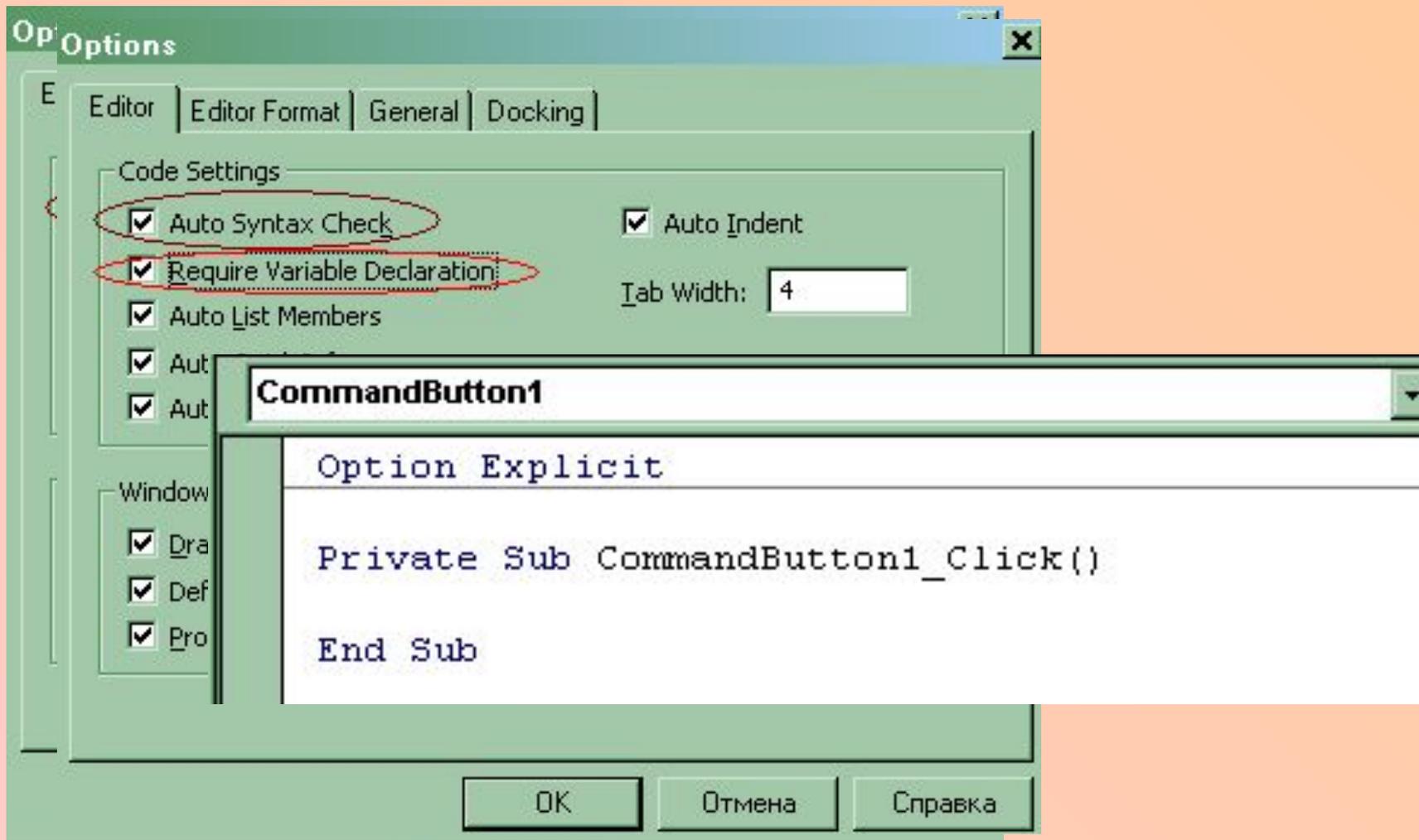
Синтаксические

Возникают из-за неправильного использования синтаксиса языка. (MsgBox вместо MsgBox)

Возникают во время написания текста программы

VBA автоматически выявляет синтаксические ошибки при наборе программы

Tools - Options



Логические (алгоритмические) ошибки

О логических ошибках говорят тогда, когда программа выполняется не так, как ожидалось.

Вероятнее всего ошибку нужно искать в алгоритме.

Поэтому иногда логические ошибки называют алгоритмическими. Например, если в программе вычисления квадрата числа с контролем вводимых значений условие проверки значения написать неверно, то программа может запрашивать ввод правильных значений при правильно введенных значениях

Логические (алгоритмические) ошибки

Инструменты отладки программы

Точки останова (Toggle BreakPoint)

```
Private Sub CommandButton2_Click()  
Dim I As Integer  
For I = 2 To 10 Step 2  
    MsgBox I, vbExclamation, "Четные числа"  
Next I  
End Sub
```

Пошаговое выполнение программы
View – Toolbars – Debug



Продолжить

Break - прервать

Reset - сброс

Step Into
Шаг с заходом

Step over
Шаг с обходом

Step out
Шаг с выходом

Логические (алгоритмические) ошибки

- Режим останова – это состояние, в котором выполнение программы прервано
- Кнопка Запуск или Перейти/Продолжить позволяет продолжить выполнение программы
- Нажатие кнопки Break приводит к немедленному прекращению выполнения программы и делает окно модуля активным
- Кнопка Reset приводит к сбросу значений глобальных переменных

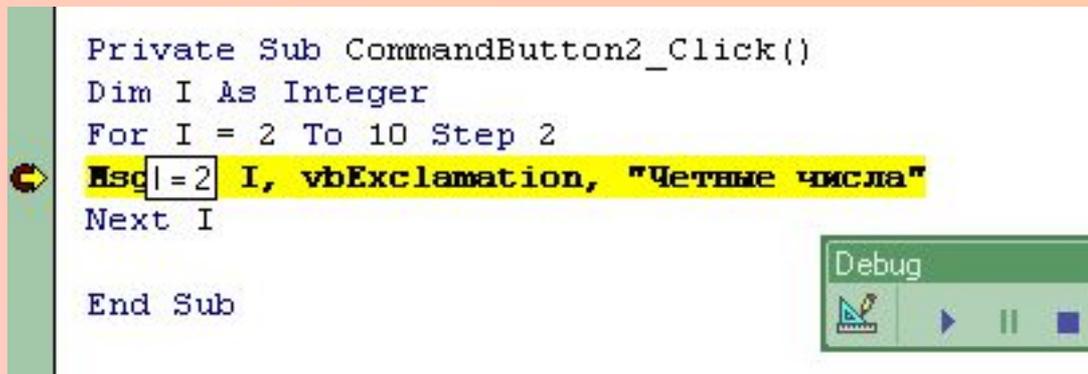
Управление ходом выполнения программы

- **Step Into (<F8>) (Шаг с заходом)** Выполняется текущий оператор после чего выполнение программы снова останавливается. Если текущий оператор содержит вызов процедуры (функции или подпрограммы) происходит передача управления в эту процедуру и текущим становится первый ее оператор.
- **Step Over(<Shift-F8>) (Шаг с обходом)** Выполняется аналогично, но только не происходит захода в вызываемую процедуру (она выполняется за один шаг). Останов происходит на следующем операторе главной (текущей) процедуры.
- **Step Out(<Ctrl-Shift-F8>)(Шаг с выходом)** Служит для выполнения всех строк процедуры, включая все вызываемые процедуры, в режиме останова.

Отслеживание значений переменных и свойств объектов

1 Если выполнение программы приостановлено – можно подвести указатель мыши к переменной или свойству

```
Private Sub CommandButton2_Click()  
Dim I As Integer  
For I = 2 To 10 Step 2  
MsgBox I, vbExclamation, "Четные числа"  
Next I  
End Sub
```



Local Window

Immediate Window

Watch Window

2 Использование специальных окон (меню **View**)

Local Window -окно- локальных переменных

Immediate Window – окно отладки

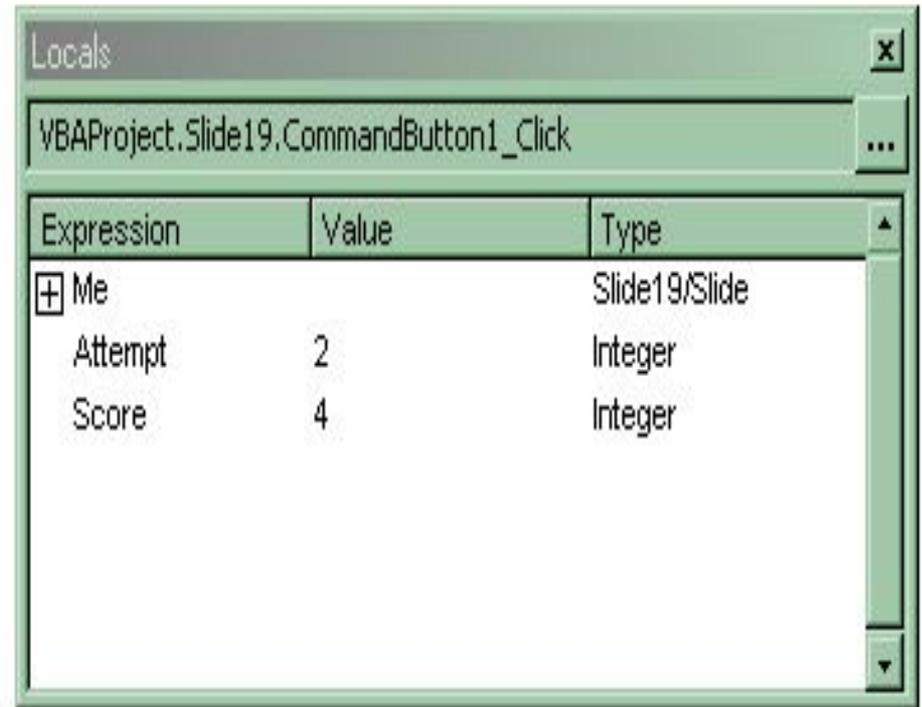
Watch Window – окно контрольных переменных



Local Window

Позволяет наблюдать за значением всех локальных переменных

```
Private Sub CommandButton1_Click()  
Dim Attempt As Integer  
Dim Score As Integer  
Randomize  
Score = Int(6 * Rnd + 1)  
Attempt = 1  
While Score < 6  
    Attempt = Attempt + 1  
    Score = Int(6 * Rnd + 1)  
Wend  
MsgBox "Вы победили на броске " & Attempt  
  
End Sub
```



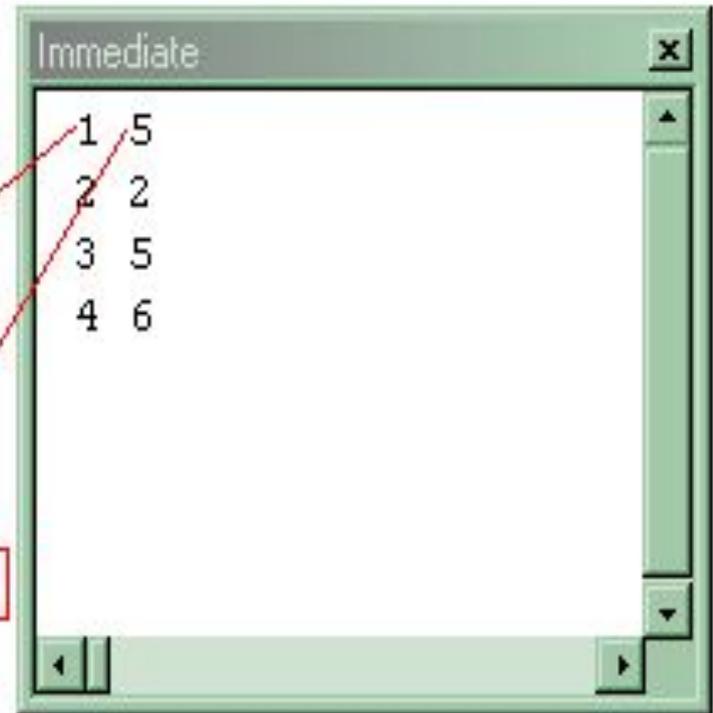
Expression	Value	Type
Me	Slide19/Slide	Slide19/Slide
Attempt	2	Integer
Score	4	Integer

Watch Window

- Watch Window (окно контрольных выражений) позволят наблюдать за выбранными разработчиком переменными или свойствами

Immediate Window

```
Private Sub CommandButton1_Click()  
Dim Attempt As Integer  
Dim Score As Integer  
Randomize  
Score = Int(6 * Rnd + 1)  
Attempt = 1  
While Score < 6  
  
    Score = Int(6 * Rnd + 1)  
    Debug.Print Attempt & " " & Score  
    Attempt = Attempt + 1  
  
Wend  
MsgBox "Вы победили на броске " & Attempt  
End Sub  
|
```



Ошибки выполнения (run-time)

Возникают, например, тогда, когда программа запущена на исполнение и пытается выполнить операцию, которую не может выполнить операционная система

Ошибки выполнения

Перехват ошибок во время выполнения

Microsoft Visual Basic

Run-time error '13':

Type mismatch

Continue

End

Debug

Help

Создание обработчика ошибок
включает в себя три шага:

1. установка перехвата ошибок
2. написание кода обработки ошибок
3. создание выхода из обработчика
ошибок

Внутри процедуры можно корректно обрабатывать возникающие ошибки при помощи оператора **On Error**

При возникновении любой ошибки при выполнении программы VBA ищет оператор **On Error**. Если - находит, то ошибка обрабатывается. Если - не находит, то программа прерывается на ошибочном операторе, и выдается системное сообщение об ошибке.

On Error Go To Метка - если в процедуре возникает ошибка , то происходит перехват ошибки и управление передается в начало обработчика ошибок. Начало обработчика задается меткой.

On Error Resume Next передается управление оператору следующему за обработчиком ошибок (игнорирование ошибок)

On Error Go To 0 – отключение обработчика ошибок.

Оператор **Err** содержит информацию об ошибке

Некоторые его свойства **ERR**

Err.Number номер ошибки

(**Err.Number=0** ошибка отсутствует)

Err.Description – строка, содержащая описание ошибки

Err.Source - строковое выражение ,
показывающее источник , где произошла ошибка

После успешной обработки ошибки можно ошибку сбросить **Err.Clear**

Возврат из обработчика ошибок

Resume – возвращает управление оператору, создавшему ошибку

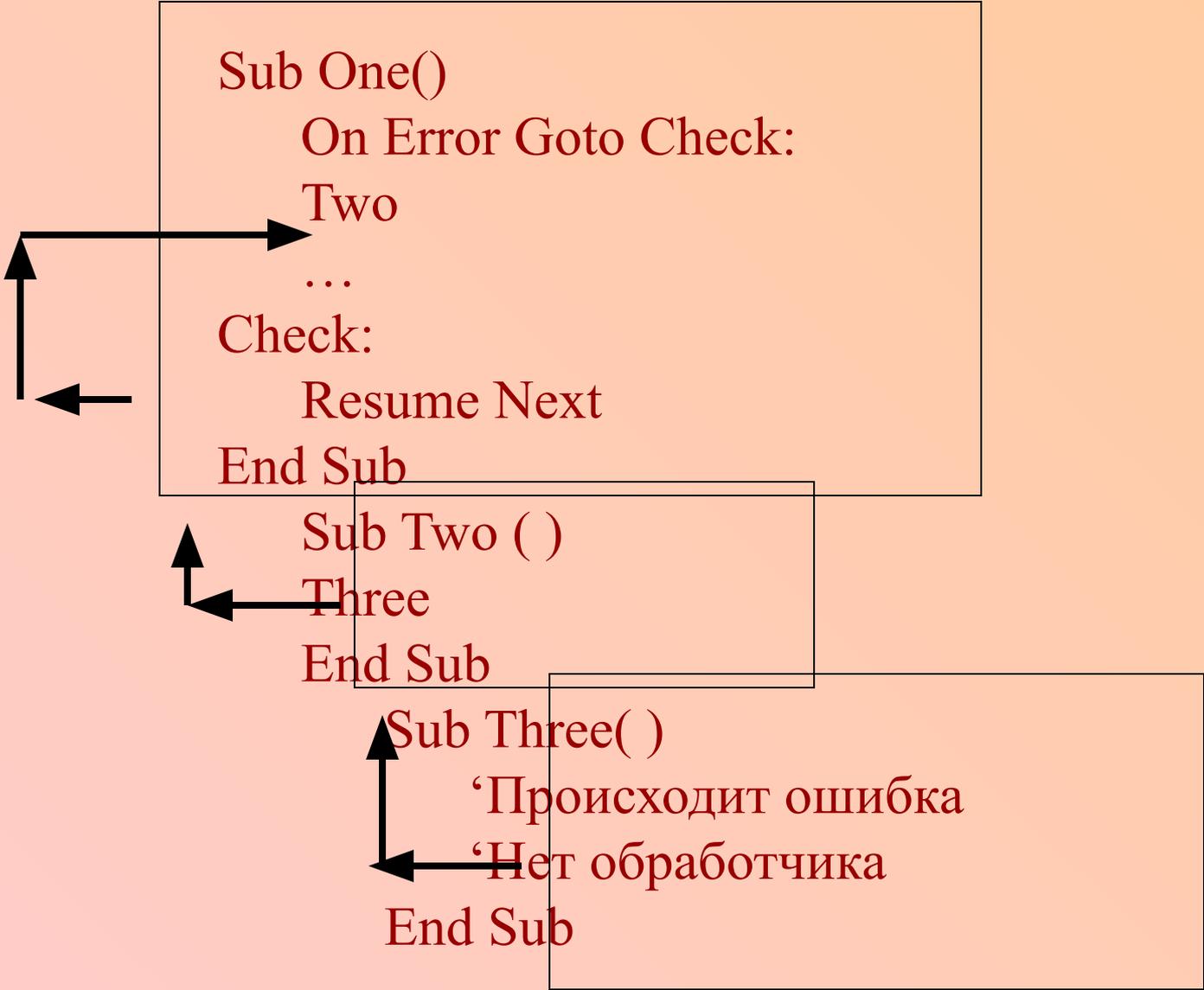
(позволяет повторить операцию, вызвавшую ошибку после ее исправления)

Resume Next – передает управление оператору за ошибочным

Resume Метка - передает управление оператору с меткой

Цепочка ВЫЗОВОВ

```
Sub One()  
  On Error Goto Check:  
  Two  
  ...  
Check:  
  Resume Next  
End Sub
```



```
Sub Two ( )  
  Three  
End Sub
```

```
Sub Three( )  
  'Происходит ошибка  
  'Нет обработчика  
End Sub
```

On Error GoTo CheckError

'Ошибка происходит здесь

Exit Sub

CheckError:

If Err=13 Then

MsgBox "ошибка типа данных "

Resume Next

Else

MsgBox "ошибка другая "

Resume

End If

