



# ОБРАБОТКА ОШИБОК

Лекция 4

# TRY...CATCH...FINALLY

**Try**

**Debug.WriteLine("Try")**

**Catch**

**Debug.WriteLine("Catch")**

**Finally**

**Debug.WriteLine("Finally")**

**End Try**

**Debug.WriteLine("Done Trying")**



# TRY...CATCH...FINALLY

- Try - В этот раздел помещается фрагмент программы, который может вызвать исключение
- Catch - Текст, помещенный в этот раздел, выполняется только при возникновении исключения.
- Finally - Текст внутри этого раздела выполняется после выполнения текста в разделе Try и/или разделе Catch. Обычно в этот раздел помещаются операторы, выполняющие высвобождение ресурсов, т.е. текст, который должен выполняться всегда, независимо от того, возникало ли исключение



# TRY...CATCH...FINALLY

- Выполнился фрагмент, включенный в раздел Try.
- 2. Так как не возникло никаких исключений, текст раздела Catch не выполнялся.
- 3. После окончания выполнения операторов в разделе Try выполнен текст раздела Finally.
- 4. После выполнения всех операторов в разделе Finally управление было передано оператору, следующему за End Try.



# РАБОТА С ИСКЛЮЧЕНИЯМИ

- Оператор **Catch** дает возможность определить переменную, содержащую ссылку на объект исключения.
- Используя этот объект, можно получить информацию об исключении.

***Catch objException As Exception*** - обрабатывает почти все исключения



ПРИМЕР:

*Dim intAnswer As Long*

*Try*

*LngAnswer=100/CInG(txtInput.Text)*

*MsgBox ("100 / " & txtInput.Text & " is " &  
LngAnswer)*

*Catch objException As*

*System.OverflowException*

*MsgBox("введите что-нибудь в textbox")*

*Catch objException As Exception*

*MsgBox("перехват других исключений")*

*End Try*



# ОБЪЕКТ ERR

- **Err** - это специальный объект Visual Basic, в котором при каждом возникновении ошибки времени исполнения сохраняется подробная информация по этой ошибке.
- Свойства объекта **Err**:
  - **Err.Number** содержит номер самой последней из ошибок времени исполнения,
  - **Err.Description** содержит краткое сообщение об этой ошибке.
  - **Err.Clear** этот метод стирает информацию о предыдущей ошибке
  - **Err.Raise** генерирует ошибку



## ПЕРЕЧЕНЬ ОШИБОК ВРЕМЕНИ ИСПОЛНЕНИЯ

Номер ошиб.	Сообщение об ошибке по умолчанию
5	Procedure call or argument is not valid (Недопустимый вызов процедуры или аргумент)
6	Overflow (Переполнение)
7	Out of memory (Недостаточно памяти)
9	Subscript out of range (Индекс массива выходит за его границы)
11	Division by zero (Деление на ноль)
13	Type mismatch (Несоответствие типов переменных)





## ГЕНЕРАЦИЯ - ИСКЛЮЧЕНИЙ

Try

Err.Raise(61) ' генерируем ошибку Disk Full

Catch When Err.Number = 61

MsgBox("Ошибка: Диск переполнен")

End Try



# ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ

## Отображение сообщений с помощью функции `MsgBox ()`

- Синтаксис: `MsgBox(prompt, [buttons], [title])`
  - **Prompt** — это текст, который будет отображаться в окне сообщения,
  - **buttons** определяет, какие будут кнопки в окне и какая пиктограмма,
  - **t i t l e** — что будет написано в заголовке окна.

### Класс





`MessageBox.show(prompt, [title, buttons, icon, defaultButton, option])`

*defaultButton* - какая кнопка активна

*Option* – расположение текста в окне



## ТИП ЗНАЧКА

ЧисКод1	Тип сообщения	Пиктограмма
16	Ошибка	
32	Вопрос	
48	Внимание	
64	Информация	



## ЗНАЧЕНИЯ КНОПОК

<b>ЧисКод2</b>	<b>Набор кнопок</b>
0	<i>ОК</i>
1	<i>ОК, Отмена</i>
2	<i>Стоп, Повторить, Пропустить</i>
3	<i>Да, Нет, Отмена</i>
4	<i>Да, Нет</i>
5	<i>Повторить, Отмена</i>



## ОБРАБОТКА НАЖАТИЯ КНОПКИ

- Выполнение программой определенных действий, в зависимости от того, какая была нажата кнопка, основано на использовании одной из логических конструкций выбора. Например:

```
If MsgBox("Хотите выполнить действия X?",  
MsgBoxStyle.YesNo Or  
MsgBoxStyle.Question) = MsgBoxResult.Yes  
Then  
    ' выполнение X происходит здесь'  
End If
```



## ОБРАБОТКА НАЖАТИЯ КНОПКИ

### *Обработка запроса на подтверждение*

(при нажатии кнопки ВЫХОД)

```
If MessageBox.Show("закрыть?", "close",  
MessageBoxButtons.OKCancel,  
MessageBoxIcon.Stop,  
MessageBoxDefaultButton.Button1,  
MessageBoxOptions.RightAlign) =  
DialogResult.OK Then Me.Close()  
End if
```



## СОБЫТИЕ НАЖАТИЯ КНОПКИ «ЗАКРЫТЬ»

```
Private Sub Form1_FormClosing(ByVal sender As  
Object, ByVal e As System.Windows.Forms.  
FormClosingEventArgs) Handles Me.FormClosing
```

- e.Cancel = True
- If MsgBox("закрыть?") = MsgBoxResult.Ok Then \_  
e.Cancel = False

```
End Sub
```



## ДИАЛОГОВЫЕ ОКНА

- определить результат диалога для одной или нескольких кнопок, т. е. что должно произойти при нажатии той или иной кнопки.
- указать, каким образом вызывается форма (`ShowDialog`).
  - форма будет отображена модально
  - если пользователь щелкает по кнопке, у которой свойство  **DialogResult**  выставлено на возвращение некоторой величины, то форма сразу же закрывается, а возвращенная величина считается результатом вызова `ShowDialog`





## ПРИМЕР

```
Dim objDialog As New formdial()  
If ObjDialog.ShowDialog=DialogResult.OK Then  
    MsgBox("You clicked OK.")  
Else  
    MsgBox("«You clicked Cancel.»")  
End If  
objCustomDialogBox = Nothing
```



## Ввод данных

- создать форму с управляющими средствами для перехвата данных;
- использовать функцию `InputBox ()` для получения данных от пользователя.

▣ *`InputBox(prompt, [ title ], [defaultresponce])`*  
**defaultresponce** текст, который будет появляться в окне ввода по умолчанию.

*`strResult = InputBox("How many eggs do you want to order?", "Order eggs", "12")`*



## СОБЫТИЯ КЛАВИАТУРЫ

- ▣ **KeyDown** - Происходит при нажатии клавиши,
- ▣ **KeyPress** - Происходит при нажатии клавиши (нажать и отпустить),
- ▣ **KeyUp** - Происходит, если отпустить нажатую клавишу

В событиях клавиатуры параметр *e* содержит данные о том, какая была нажата клавиша.

Информацию о том, какая была нажата клавиша, можно узнать из свойств **KeyChar** параметра *e*.



## ПРИМЕРЫ

```
Private Sub Form1_KeyDown(ByVal sender As Object,  
ByVal e As System.Windows.Forms.KeyEventArgs) Handles  
Me.KeyDown
```

```
    If e.KeyCode = Keys.A Then Label1.BackColor =  
        Color.Blue  
End Sub
```

```
Private Sub Form1_KeyPress(ByVal sender As Object,  
ByVal e As System.Windows.Forms.KeyPressEventArgs)  
Handles Me.KeyPress
```

```
    a = Label1.Left  
    If e.KeyChar = "d" Then a = a + 10  
    Label1.Left = a  
End Sub
```

```
Private Sub Form1_KeyUp(ByVal sender As Object, ByVal e  
As System.Windows.Forms.KeyEventArgs) Handles  
Me.KeyUp
```

```
    If e.KeyCode = Keys.B Then Label1.Text = "hhjhj"  
End Sub
```



## ПРИМЕРЫ

- Private Sub Form1\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles Me.KeyPress
  
- a = Label1.Left
- If e.KeyChar = "d" Then a = a + 10
- Label1.Left = a
  
- If e.KeyChar = "k" Then
- e.Handled = True
- Label1.BackColor = Color.Brown
- End If
  
- End Sub



## ПРИМЕР

Когда `Handled` имеет значение `TRUE`, то `Visual Basic` пропускает данное нажатие клавиши.

перехватывает ввод всех букв "К", независимо от их регистра:

- ▣ **`If e.KeyChar.ToLower(e.KeyChar) = "k" Then`  
`e.Handled = True`**
- ▣ **`End If`**



# СОБЫТИЯ МЫШИ

Названия событий	Описание
<b>MouseEnter</b>	Происходит, когда указатель мыши фиксирует управляющий элемент
<b>MouseMove</b>	Происходит, когда указатель мыши двигается над управляющим элементом
<b>HouseHover</b>	Происходит, когда указатель мыши останавливается над управляющим элементом
<b>MouseDown</b>	Происходит, когда указатель мыши находится над управляющим элементом и при этом нажата кнопка мыши
<b>Mouseup</b>	Происходит, когда указатель мыши находится над управляющим элементом и при этом отпускается кнопка мыши
<b>MouseLeave</b>	Происходит, когда указатель мыши "отходит" от управляющего элемента



## ПЕРЕТАСКИВАНИЕ ОБЪЕКТА

- Private Sub Form1\_MouseMove(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Me.MouseMove
- If e.Button = MouseButton.Right Then
- Label1.Left = e.X
- Label1.Top = e.Y
- End If
  
- End Sub





## РИСОВАНИЕ

```
Private Sub Form1_MouseDown()
```

```
    If e.Button = MouseButton.Left Then
```

```
        x1 = e.X
```

```
        y1 = e.Y
```

```
        I = True
```

```
    End If
```

```
End Sub
```

```
Private Sub Form1_MouseUp()
```

```
    i = False
```

```
End Sub
```



## РИСОВАНИЕ

```
Private Sub Form1_MouseMove()
```

```
    Dim graf As Graphics = MyClass.CreateGraphics
```

```
    If i Then
```

```
        x2 = e.X
```

```
        y2 = e.Y
```

```
        graf.DrawLine(New Pen(Color.Blue, 3), x1, _ y1,  
x2, y2)
```

```
        x1 = x2
```

```
        y1 = y2
```

```
    End If
```

```
End Sub
```



