Лекция 6.

Работа с формами (часть 1)

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

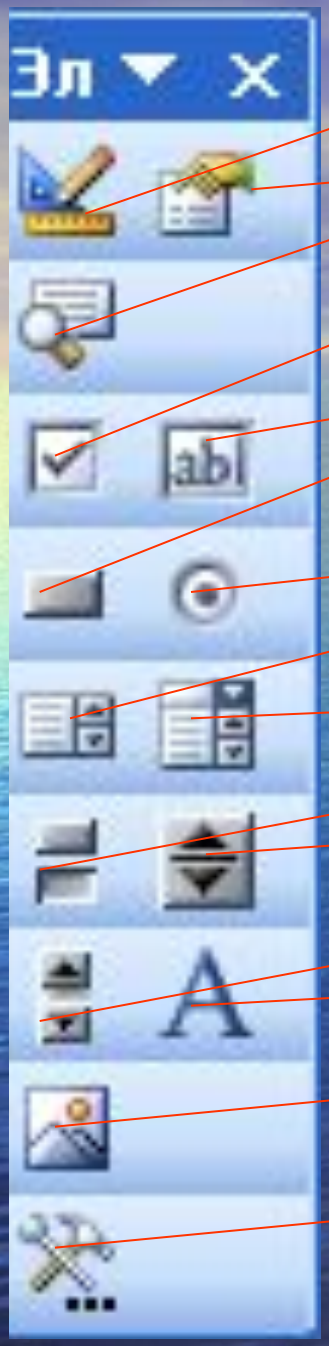
2013 г

Элементы управления

VBA обладает встроенным набором элементов управления. Используя этот набор и редактор форм не трудно создать любой пользовательский интерфейс, который будет удовлетворять всем требованиям, предъявляемым к интерфейсу в среде Windows

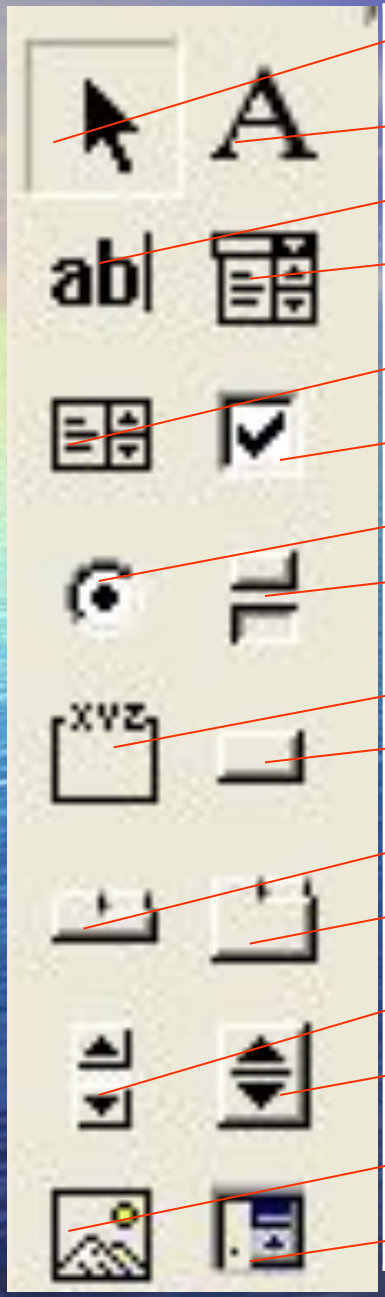
Создание элементов управления на рабочем листе или в форме как правило происходит на начальном этапе конструирования приложения. Большинство элементов управления можно располагать как на рабочем листе, так и в форме. Но существуют такие элементы управления, как RefEdit, Набор страниц и Набор вкладок, которые можно располагать только в форме

Панель инструментов в Excel



Режим конструктора	Свойства
Код	
Флажок (Checkbox)	Поле (Textbox)
Кнопка (Button)	Переключатель (OptionButton)
Список (ListBox)	Поле со списком (ComboBox)
Выключатель (ToggleButton)	Счётчик (SpinButton)
Полоса прокрутки (ScrollBar)	Надпись (Label)
Рисунок (Image)	
Другие элементы	

Панель инструментов VBA



Выделение объектов	Надпись (Label)
Поле (Textbox)	Поле со списком (ComboBox)
Список (ListBox)	Флажок (Checkbox)
Переключатель (OptionButton)	Выключатель (ToggleButton)
Рамка (Frame)	Кнопка (Button)
Набор вкладок (TabStrip)	Набор страниц (MultiPage)
Полоса прокрутки (ScrollBar)	Счётчик (SpinButton)
Рисунок (Image)	RefEdit

Событие – происходит, когда пользователь воздействует на элемент управления. Например, щелчок на командной кнопке инициирует событие Click (Щелчок), ассоциированное с данной кнопкой.

Процедуры обработки событий носят имена, в которых название элемента управления объединено с названием события с помощью символа подчеркивания. Например, процедура, которая выполняется после щелчка на кнопке Button1, называется Button1_Click.

Соглашение об именах

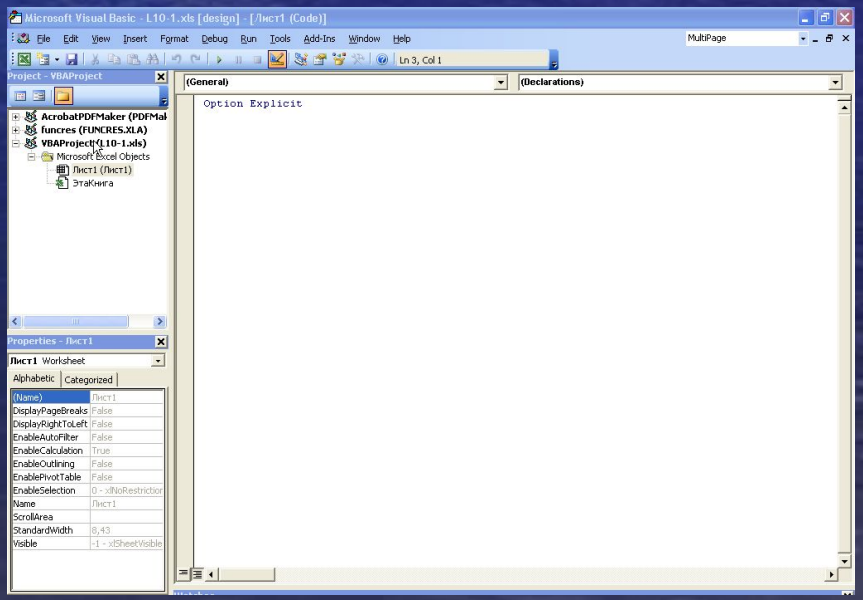
Управляющий элемент	Префикс	Пример имени
TextBox	txt	txtAccount
Label	lbl	lblInform
ComanandButton	cmd	cmdOK
ListBox	lst	lstNames
ComboBox	cbo	cboFirms
ScrollBar	scr	scrDown
SpinButton	spn	spnUp
OptionButton	opt	optChoice
CheckBox	chk	chkFlag
ToggleButton	Tgl	TglSwitch
Frame	fra	fraStatus
Image	img	imgBall
RefEdit	ref	refFun
MultiPage	mlt	mltPages
TabStrip	tab	tabTwoTabs
UserForm	frm	frmGame

Создание пользовательской формы в VBA

Пользовательская форма является базой для создания пользовательского диалогового окна.

В окне Properties можно менять различные свойства формы.

На форме располагаются различные элементы управления.



Основные свойства объекта UserForm

Name	Имя пользовательской формы
Caption	Текст, отображаемый в строке заголовка формы
BackColor	Цвет фона формы
BorderStyle	Устанавливает тип границы
Picture	Указывает рисунок, отображаемый как фон формы
Left, Top	Возвращают местоположение верхнего левого угла формы в пунктах
Height, Width	Возвращают высоту и ширину формы в пунктах
StartUpPosition	Возвращает значение, определяющее положение формы при ее первом отображении на экране. Допустимые значения: • Manual (начальное значение не устанавливается), • CenterOwner (выравнивание по центру объекта, к которому принадлежит форма) • CenterScreen (выравнивание по центру экрана) • windows Default (положение верхнего левого угла экрана)

Основные методы и события объекта UserForm

Методы объекта UserForm	
Show	Отображает форму на экране
Hide	Закрывает форму
Move	Изменяет положение и размер формы
PrintForm	Печатает изображение формы
События объекта UserForm	
Initialize	Происходит при отображении формы на экране
Activate	Происходит, когда становится активным окном
Deactivate	Происходит, когда перестаёт быть активным окном
Terminate	Происходит при закрытии формы

Общие свойства элементов управления

Name	Имя элемента управления
Caption	Надпись, отображаемая при элементе управления
AutoSize	Допустимые значения: True (устанавливает режим автоматического изменения размеров элемента управления так, чтобы на нем полностью помещался текст, присвоенный свойству Caption) и False (в противном случае)
Visible	Допустимые значения: True (элемент управления отображается во время выполнения программы) и False (в противном случае)
Enabled	Допустимые значения: True (пользователь вручную может управлять элементом управления) и False (в противном случае)
Height, Width	Устанавливают геометрические размеры объекта (высоту и ширину)
Left, Top	Устанавливают координаты верхнего левого угла элемента управления, определяющие его местоположение в форме

Общие свойства элементов управления (продолжение)

ControlTipText	Устанавливает текст в окне всплывающей подсказки, связанной с элементом управления.
BackColor, ForeColor и BorderColor	Устанавливают цвет заднего и переднего плана элемента управления, также его границы
BackStyle	Устанавливает тип заднего фона
BorderStyle	Устанавливает тип границы. Допустимые значения: • fmBorderStyleSingle (граница в виде контура) • fmBorderStyleNone (граница невидима)
SpecialEffect	Устанавливает тип границы. Отличается от свойства BorderStyle тем, что позволяет установить несколько типов, но одного цвета. BorderStyle позволяет установить только один тип, но различных цветов

Общие свойства элементов управления (продолжение)

Picture (создание картинки)	Внедряет картинку на элемент управления. Например, на поверхности кнопки картинка отображается с помощью инструкции: CommandButton1.Picture = LoadPicture("c:\work\krug.jpg") Функция LoadPicture(ПолноеИмяФайла) считывает графическое изображение.
Picture (удаление картинки)	Иногда возникает необходимость ее удалить. Это достигается присвоением свойству Picture значения LoadPicture ("")
Tag	Используется для хранения дополнительной информации о форме или элементе управления.

Цвета для свойств BackColor, ForeColor, BorderColor

Константа	Значение	Цвет
vbBlack	0x0	Черный
vbRed	0xFF	Красный
vbGreen	0xFF00	Зеленый
vbYellow	0xFFFF	Желтый
vbBlue	0xFF0000	Синий
vbMagenta	0xFF00FF	Розовый
vbCyan	0xFFFF00	Голубой
vbWhite	0xFFFFFF	Белый

Общие методы элементов управления (Коллекция Controls)

Add	Позволяет добавить элемент управления во время выполнения программы
Move	Перемещает элемент управления
Zorder	Помешает объект до или после всех пересекающихся с ним объектов
Clear	Для MultiPage и TabStrip удаляет индивидуальные страницы или вкладки Для ListBox/ComboBox очищает все поля Для коллекции Controls удаляет всё, что добавлено методом Add. Нельзя удалять то, что создано в конструкторе!

Общие события элементов управления

Click	Происходит, когда пользователь выбирает элемент управления с помощью одинарного щелчка кнопкой мыши
DblClick	Происходит, когда пользователь выбирает элемент управления с помощью двойного щелчка кнопкой мыши
KeyPress	Происходит, когда пользователь нажимает любую клавишу на клавиатуре, кроме функциональных и клавиш управления курсором
Change	Происходит при изменении значения элемента управления
GotFocus, LostFocus	Происходит, когда элемент управления получает или теряет фокус
Error	Используется при уведомлении об ошибке

Элемент управления «Поле» (TextBox)

Text	Возвращает текст, содержащийся в поле
Visible	Допустимые значения: True (поле отображается во время выполнения программы) и False (в противном случае)
Enabled	Допустимые значения: True (пользователь непосредственно может вносить изменения в содержание поля) и False (в противном случае)
Multiline	Допустимые значения: True (устанавливается многострочный режим ввода текста в поле) и False (однострочный режим)
Wordwrap	Допустимые значения: True (устанавливается режим автоматического переноса) и False (в противном случае)
AutoSize	Допустимые значения: True (устанавливается режим автоматического изменения размера поля так, чтобы весь вводимый текст помещался в нем) и False (устанавливается фиксированный размер поля)

Элемент управления «Поле» (TextBox) (продолжение)

ScrollBars	Устанавливает режим отображения в поле полос прокрутки. Допустимые значения: • fmScrollBarsNone (не выводить полос прокрутки) • fmScrollBarsHorizontal (выводить горизонтальную полосу прокрутки) • fmScrollBarsVertical (выводить вертикальную полосу прокрутки) • fmScrollBarsBoth (выводить горизонтальную и вертикальную полосы прокрутки)
SelLenght, SelStart и SelText	Эти свойства характеризуют выделенный в поле фрагмент текста (длина, начало и сам фрагмент текста соответственно)
MaxLength	Устанавливает максимальное допустимое количество вводимых в поле символов. Если это свойство равно 0, то нет ограничений на вводимое количество символов.

Элемент «Надпись» (Label)

Caption	Возвращает текст, отображаемый в надписи
Visible	Допустимые значения: True (поле отображается во время выполнения программы) и False (в противном случае)
Multiline	Допустимые значения: True (устанавливается многострочный режим ввода текста в поле) и False (однострочный режим)
AutoSize	Допустимые значения: True (устанавливается режим автоматического изменения размера поля так, чтобы весь вводимый текст помещался в нем) и False (устанавливается фиксированный размер поля)
Wordwrap	Допустимые значения: True (устанавливается режим автоматического переноса) и False (в противном случае)

Элемент «Кнопка» (CommandButton)

Caption	Возвращает текст, отображаемый на кнопке
Cancel	Допустимые значения: True (устанавливаются отменяющие функции для кнопки, т. е. нажатие клавиши <Esc> приводит к тем же результатам, что и нажатие кнопки) и False (в противном случае)
Visible	Допустимые значения: True (кнопка отображается во время выполнения программы) и False (в противном случае)
Enabled	Допустимые значения: True (разрешено нажатие кнопки пользователем) и False (в противном случае)
Accelerator	Назначает клавишу, при нажатии на которую одновременно с клавишей <Alt> происходит запуск действий, связанных с кнопкой. Например: CommandButton1.Accelerator = "C"

Элемент «Кнопка» (CommandButton)

Picture	Внедряет на поверхность кнопки картинку. Например, CommandButton1.Picture = LoadPicture ("c:\work\pic.bmp") Функция LoadPicture (ПолноеИмяФайла) Считывает графическое изображение. Аргумент ПолноеИмяФайла указывает полное имя графического файла
Default	Задает кнопку по умолчанию, т. е. устанавливает ту кнопку, для которой действия, связанные с ней, будут выполняться при нажатии клавиши <Enter>

Пример использования кнопок и полей. Код в модуле

```
Option Explicit

Public str As String 'вводимая строка
Public dInp As Double 'вводимое число

Public Sub inpTxt()
'ввод текста в текстовое поле
dInp = 1.5

'При такой записи будет записываться десятичная точка
'в качестве разделителя
'При считывании это вызовет ошибку
UserForm1.TextBox2.Value = dInp * 0.5

UserForm1.Show

MsgBox "Считана строка: " & str & vbCr & "Считано число: " _
& dInp, vbInformation

End Sub
```

Пример использования кнопок и полей. Форма и её код

```
Option Explicit
Private Sub CommandButton1_Click()
Dim strI As String, i As Byte
str = TextBox1.Value 'считывание строки
'В таком виде это может вызвать ошибку, если в качестве разделителя
'использовалась десятичная точка:
'dInp = TextBox2.Value
'Поэтому используем текстовую строку:
strI = TextBox2.Value
strI = Replace(strI, ".", ",") 'Заменяем десятичный разделитель
If IsNumeric(strI) Then 'Проверка число/не число
    dInp = CDbl(strI) 'преобразуем строку в число
    Hide 'убирает форму с экрана при правильном вводе
Else
    MsgBox "некорректное число в поле", vbCritical
End If
End Sub
```

```
Private Sub UserForm_Initialize()
With UserForm1
    .TextBox1.Value = "Текстовое поле"
    .Label1.Caption = "Введите текст"
    .Label1.Font.Bold = True
    .Label1.ForeColor = vbBlue
    .Label2.Caption = "Введите число"
    .Label2.Font.Bold = True
    .Label2.ForeColor = vbRed
End With
End Sub
```

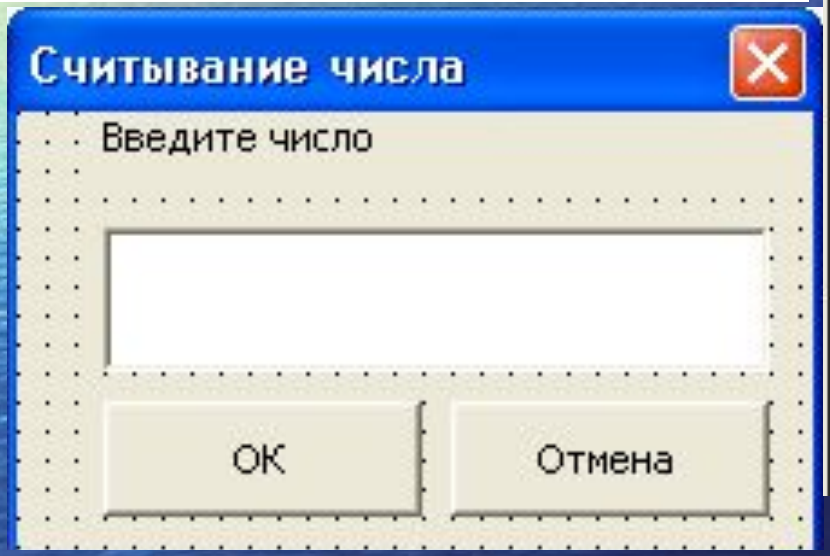
```
Private Sub CommandButton2_Click() 'Нажатие кнопки Отмена
UserForm1.Hide 'убирает форму с экрана после нажатия кнопки
End Sub
```

Считывание чисел с разделителем целой и дробной частей

Текстовое поле **TextBox** характеризуется тем, что при выводе числа (например, в формате **Double**) в это поле десятичным разделителем в нём становится точка (**.**). При считывании значения **Value** этого поля напрямую в переменную того же типа (**Double**) возникает **ошибка**, т.к. разделителем там является запятая (**,**)

Разрешение проблем считывания

```
Option Explicit
Public dVar As Double 'целевая переменная
Sub ReadFig()
'Считывание чисел вне зависимости от
' десятичного разделителя
Randomize
dVar = Rnd() 'Задаём случайное число
UserForm8.Show
MsgBox "Считано число" & vbCr & _
dVar, vbInformation
End Sub
```




```
Option Explicit

Private Sub CommandButton1_Click()
Dim Str As String 'промежуточная переменная
Dim i As Byte
Str = Replace(TextBox1.Value, ".", ",")
If IsNumeric(Str) Then
    dVar = CDbl(Str) 'можно и Val(Str)
Else
    MsgBox "Введено нечисловое значение"
End If
Hide
End Sub

Private Sub CommandButton2_Click()
Hide
End Sub

Private Sub UserForm_Initialize()
TextBox1.Value = dVar
End Sub
```

Список (Listbox)

Применяется для хранения списка значений. Пользователь может выбрать одно или несколько значений

Основные свойства элемента ListBox	
ListIndex	Возвращает номер текущего элемента списка. **Нумерация элементов списка начинается с нуля**
ListCount	Возвращает число элементов списка
TopIndex	Возвращает элемент списка с наибольшим номером
ColumnCount	Устанавливает число столбцов в списке
TextColumn	Устанавливает столбец в списке, элемент которого возвращается свойством Text
Enabled	Допустимые значения: True (запрещен выбор значения из списка пользователем) и False (в противном случае)
Text	Возвращает выбранный в списке элемент

Основные свойства элемента ListBox (продолжение)

List	Возвращает элемент списка, стоящий на пересечении указанных строки и столбца. Синтаксис: **List(row, column)**
RowSource	Устанавливает диапазон, содержащий элементы списка
Control Source	Устанавливает диапазон (ячейку), куда возвращается выбранный элемент из списка
MultiSelect	Устанавливает способ выбора элементов списка. Допустимые значения: **fmMultiSelectsingle** - выбор только одного элемента **fmMultiSelectMulti** - разрешен выбор нескольких элементов посредством либо щелчка, либо нажатием клавиши <Пробел> **fmMultiSelectExtended** - разрешено использование клавиши <Shift> при выборе ряда последовательных элементов списка)

Основные свойства элемента ListBox (продолжение)

Selected	Допустимые значения: True (если элемент списка выбран) и False (в противном случае). Используется для определения выделенного текста, когда свойство MultiSelect имеет значение fmMultiSelectMulti или fmMultiSelectExtended
ColumnWidths	Устанавливает ширину столбцов списка." Синтаксис: ColumnWidths = String • string – строка, устанавливающая ширину столбцов
ColumnHeads	Допустимые значения: True (выводятся заголовки столбцов раскрывающегося списка) и False (в противном случае)
ListStyle	Допустимые значения: • fmListstylePlain – выбранный элемент из списка выделяется цветом) • fmListstyleOption – перед каждым элементом в списке располагается флажок и выбор элемента из списка соответствует установке этого флажка

Основные свойства элемента ListBox (продолжение)

MatchEntry	Выводит первый подходящий элемент из списка при наборе его имени на клавиатуре. Допустимые значения: • fmMatchEntryNone - (режим вывода подходящего элемента в списке отключен) • fmMatchEntryFirstLetter - (вводит подходящий элемент по набранной первой букве. В этом случае, предпочтительно, чтобы элементы списка были бы упорядочены в алфавитном порядке) • fmMatchEntryComplete - (вводит подходящий элемент по полному набранному имени)
BoundColumn	Устанавливает тип, возвращаемый свойством Value. Если свойство BoundColumn равно 0, то свойство value возвращает индекс выбранной строки, т. е. в этом случае оно действует как свойство ListIndex. • Если свойство BoundColumn принимает значение из диапазона от 1 до количества столбцов в списке, то свойство Value возвращает элемент из выбранной строки, стоящий в столбце, определенном свойством BoundColumn

Основные методы элемента ListBox

Clear	Удаляет все элементы из списка
RemoveItem	Удаляет из списка элемент с указанным номером. Синтаксис: **RemoveItem (index)** • **index** – номер удаляемого из списка элемента
AddItem	Добавляет элемент в список. Синтаксис: **AddItem, ([item [, varIndex)])** • item – элемент (строковое выражение), добавляемый в список • **VarIndex** – номер добавляемого элемента

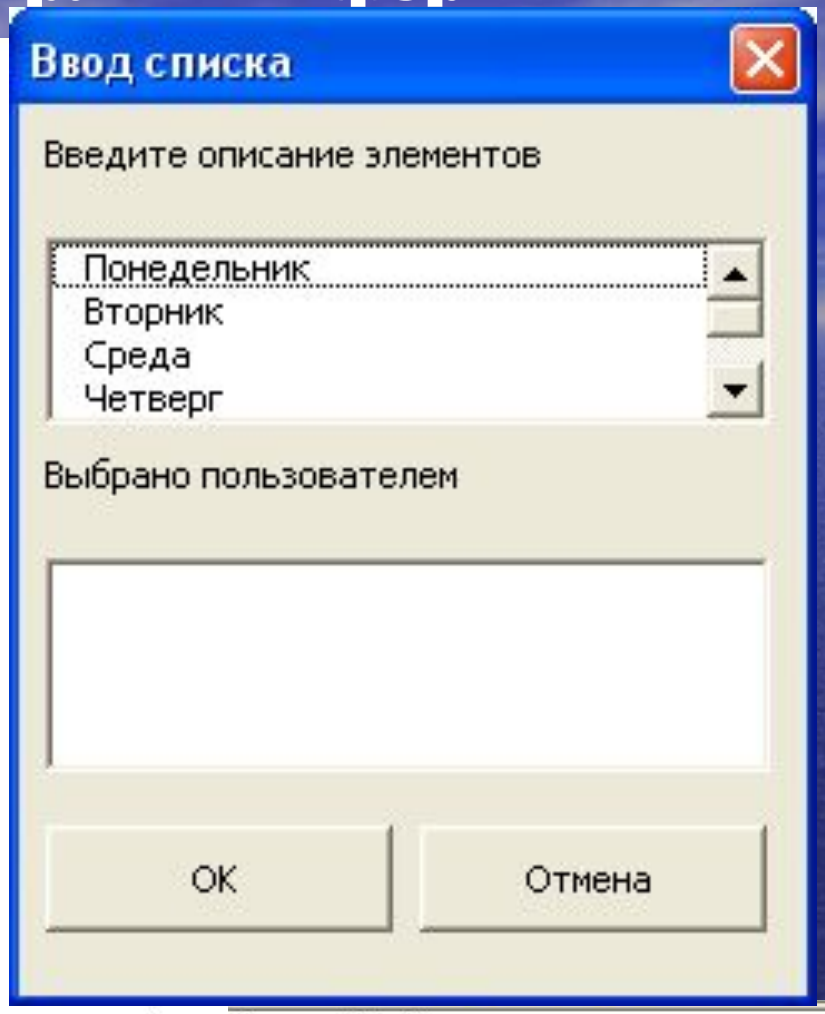
Примеры последовательного заполнения списка

General) ListBoxFi

```
Option Explicit
'здесь собираем данные о выделенных элементах списка
Public flDays(6) As Boolean
Sub ListBoxFill()

Dim str(6) As String, strDays(6) As String
Dim i As Byte, idCount As Byte
'Заполнение начальных значений
str(0) = "Понедельник"
str(1) = "Вторник"
str(2) = "Среда"
str(3) = "Четверг"
str(4) = "Пятница"
str(5) = "Суббота"
str(6) = "Воскресенье"

idCount = 0 'нет выделенных дней
'Заполнение значений на форме
With UserForm2
    .Label1.Caption = "Введите описание элементов"
'     .Label1.AutoSize = True
    .Label1.ForeColor = vbBlack
    .Label2.Caption = "Выбрано пользователем"
    .Label2.AutoSize = True
    .ListBox1.Clear
    .ListBox2.Clear 'очистка списка вывода
    .ListBox2.Enabled = False 'защита от юзера
    .ListBox1.Enabled = True 'Разрешен выбор
End With
```

Продолжение кода программы и формы

```
'Последовательное заполнение элементов списка
With UserForm2.ListBox1
    For i = 0 To 6
        .AddItem str(i)
        flDays(i) = False
    Next i
    .ListIndex = 0
    .MultiSelect = fmMultiSelectExtended
End With

UserForm2.Show

'подсчёт выделенных дней
'Код специально сделан _неоптимальным_
'можно было обойтись без промежуточных массивов
With UserForm2
    If .ListBox1.ListCount > 0 Then 'что-то выбрано
        For i = 0 To .ListBox1.ListCount - 1
            flDays(i) = .ListBox1.Selected(i)
            If flDays(i) Then 'Найден выделенный день
                strDays(idCount) = str(i)
                .ListBox2.AddItem strDays(idCount)
                idCount = idCount + 1
            End If
        Next i
        .Label1.Caption = "Выбор закончен"
        .Label1.ForeColor = vbRed
        .ListBox1.Enabled = False 'Выбор закончен. Запрет на выбор
        .ListBox2.Enabled = True 'можно просматривать весь список
        .ListBox2.MultiSelect = fmMultiSelectSingle
        .Show
    End If
End With
End Sub
```



CommandButton2

```
Option Explicit

Private Sub CommandButton1_Click()
UserForm2.Hide
End Sub

Private Sub CommandButton2_Click()
UserForm2.Hide
End Sub
```

Ввод списком

Переделанный фрагмент кода из предыдущего примера

```
With UserForm2
    If .ListBox1.ListCount > 0 Then 'что-то выбрано
        For i = 0 To .ListBox1.ListCount - 1
            flDays(i) = .ListBox1.Selected(i)
            If flDays(i) Then 'Найден выделенный день
                strDays(idCount) = str(i)
 '              .ListBox2.AddItem strDays(idCount)
                idCount = idCount + 1
            End If
        Next i
        .ListBox2.List = strDays 'вывод списком
        .Label1.Caption = "Выбор закончен"
        .Label1.ForeColor = vbRed
        .ListBox1.Enabled = False 'Выбор закончен. Запрет на выбор
        .ListBox2.Enabled = True 'можно просматривать весь список
        .ListBox2.MultiSelect = fmMultiSelectSingle
        .Show
    End If
End With
```

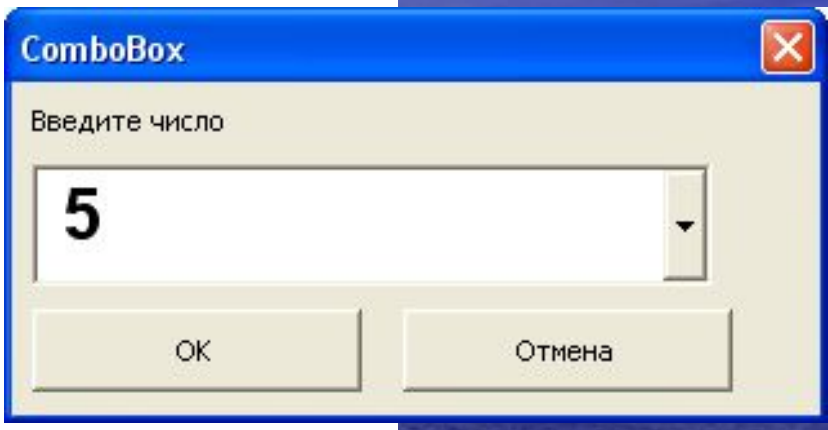
Поле со списком (ComboBox)

Методы и свойства объекта ComboBox во многом аналогичны таковым для ListBox. Уникальные методы приведены ниже:

DropButtonStyle	Устанавливает вид раскрывающегося списка. Допустимые значения: •**fmDropButtonStylePlain** (кнопка без символов) •**FmDropButtonStyleArrowDisplays** (кнопка со стрелкой) •**FmDropButtonStyleEllipsis** (кнопка с эллипсом) •**FmDropButtonStyleReduce** (кнопка с линией)
ListRows	Устанавливает число элементов, отображаемых в раскрывающемся списке
MatchRequired	Допустимые значения: True (в поле ввода раскрывающегося списка нельзя ввести значения, отличные от перечисленных в списке, т. е. в поле со спискам отключается функция поля ввода) и False (в противном случае)
MatchFound	Допустимые значения: True (среди элементов раскрывающегося списка имеется элемент, совпадающий с вводимым в поле ввода раскрывающегося списка) и False (в противном случае)

Пример заполнения поля со списком

```
Option Explicit
Sub ComboboxFill()
'подпрограмма заполняет и считывает значения из
'поля со списком
Const N As Byte = 10
Dim dIni(N) As Double, dRes(N) As Double
Dim dFin As Double
Dim i As Byte
Dim FlEq As Boolean 'флаг соответствия

For i = 0 To 5
   dIni(i) = i * 5 'заполняем начальный массив
Next i

UserForm5.Label1.Caption = "Введите число"
With UserForm5.ComboBox1
    .Clear
    .MatchRequired = False 'можно вводить другие значения
    .Font.Name = "Arial" 'Форматируем текст
    .Font.Bold = True
    .Font.Size = 20
'     .List = dIni приводит к тому, что элементы с 6 до 10 будут 0
    For i = 1 To 5
        .AddItem dIni(i)
    Next i
End With

10:
UserForm5.Show
```




```
Option Explicit

Private Sub CommandButton1_Click()
UserForm5.Hide
End Sub

Private Sub CommandButton2_Click()
UserForm5.Hide
End Sub
```

Пример заполнения поля со списком

```
With UserForm5.ComboBox1
    On Error GoTo ErrHandler1 'обработка ошибока ввода
    FlEq = .MatchFound
    If .Value = "" Then 'проверка, что что-то введено
        i = MsgBox("Не введено число", vbExclamation + vbOKCancel)
        If i = vbCancel Then Exit Sub
        GoTo 10
    End If
    dFin = .Value 'присвоение введённого значения
    For i = 0 To .ListCount - 1
        'считывание массива из списка
        dRes(i) = .List(i)
    Next i
    If (Not FlEq) Then
    'числа нет в списке. Добавляем его
        .AddItem dFin
         MsgBox "Число принято", vbInformation
        If .ListCount >= N Then 'лимит чисел исчерпан
            MsgBox "Поле заполнено", vbInformation
            Exit Sub
        End If
    End If
End With
'Если добавили элемент  возвращаемся к заполнению списка
If FlEq Then
    i = MsgBox("Такое число уже есть", vbExclamation + vbOKCancel)
    If i = vbOK Then GoTo 10 'продолжаем заполнение поля
    Exit Sub 'Иначе - выход
End If
GoTo 10
ErrHandler1: 'Обработчик ошибок
 MsgBox "Handler: ввели неправильное значение"
 Resume 10
End Sub
```

Приёмы работы с Listbox и Combobox

Обычно требуется в зависимости от выбранного пункта произвести какую-либо операцию. Данный пример демонстрирует компактный код для выбора цвета.

```
Private Sub ComboBox1_Change()
Dim Colors() As Variant
Dim i As Byte
'Array("Красный", "Жёлтый", "Зелёный", "Синий", "Чёрный", "Белый")
'Массивы обязательно должны иметь одинаковую размерность
'Заполняем массив констант.
Colors = Array(vbRed, vbYellow, vbGreen, vbBlue, vbBlack, vbWhite)
'Обрабатываем Combobox
i = Me.ComboBox1.ListIndex 'просто промежуточная переменная для читабельности
Label1.ForeColor = Colors(i) 'присваиваем цвет по индексу выбранного пункта в combobox
End Sub
```

```
Private Sub UserForm_Initialize()
Dim mColors() As Variant
Dim i As Byte
mColors = Array("Красный", "Жёлтый", "Зелёный", "Синий", "Чёрный", "Белый")
With Me.ComboBox1
    For i = 0 To UBound(mColors) 'добавляем цвета в Combobox
        Me.ComboBox1.AddItem mColors(i)
    Next i
    .ListIndex = 0 'Устанавливаем выбраныний по умолчанию пункт
    .Font.Size = 32 'Устанавливаем размер шрифта
    .MatchRequired = True 'Запрещаем пользователю ввод своего значения
End With
End Sub
```

Рисунок (Image)

Создаётся с помощью кнопки **Рисунок** (Image). Используется для отображения графических файлов в формате bmp, cur, gif, ico, jpg и wmf.

Некоторые свойства рисунка:

AutoSize	Допустимые значения: True (рисунок автоматически изменяет размер для того, чтобы отобразить изображение целиком) и False (в противном случае)
Picture	Задает отображаемый графический файл. Используется с функцией LoadPicture. Синтаксис: Picture = LoadPicture (ПолноеИмяФайла) • ПолноеИмяФайла – полное имя отображаемого графического файла
PictureTiling	Допустимые значения: True (объект покрывается мозаикой из рисунка) и False (в противном случае)

Рисунок (продолжение)

Picture SizeMode	Устанавливает масштабирование рисунка. Допустимые значения: •fmpictureSizeModeClip (не помещающиеся в границах объекта части рисунка обрезаются) •fmPictureSizeModeStretch (рисунок масштабируется так, чтобы он занимал всю поверхность объекта) •fmPictureSizeModeZoom (рисунок масштабируется с сохранением относительных размеров так, чтобы он помещался целиком внутри объекта)
Picture Alignment	Устанавливает расположение рисунка внутри объекта. Допустимые значения: •fmPictureAlignmentTopLeft (в верхнем левом углу) •fmPictureAlignmentTopRight (в верхнем правом углу) •fmPictureAlignmentCenter (в центре) •fmPictureAlignmentBottomLeft (в нижнем левом углу) •fmPictureAlignmentBottomRight (в нижнем правом углу)

Пример

```
Sub Picture1()
'выбираем файл
With Application.FileDialog(msoFileDialogOpen)
          .AllowMultiSelect = False 'только 1 файл
          .Filters.Clear
          'фильтр по типам. не работает в Excel!
          .Filters.Add "Рисунки", "*.gif; *.jpg; *.jpeg; *.BMP", 1
          .Filters.Add "Все файлы", "*.*", 1
          .Show
          If .SelectedItems.Count = 1 Then
             strPath = .SelectedItems(1) '.InitialFileName
          Else
             MsgBox "Файл не выбран", vbCritical
             Exit Sub
          End If
End With
UserForm10.Image1.Picture = LoadPicture(strPath) 'загрузка рисунка
UserForm10.Show
End Sub
```

```
Sub Picture2()
'более старый но более надёжный вариант
Dim Filt As String
Filt = "Графика (*.JPG), *.jpg, (*.GIF), *.gif, (*.BMP), *.BMP"
strPath = Application.GetOpenFilename(FileFilter:=Filt, _
Title:="Выберите нужный файл и кликните кнопку", _
MultiSelect:=False)

If strPath = "False" Then GoTo 100
UserForm10.Image1.Picture = LoadPicture(strPath) 'загрузка рисунка
UserForm10.Show
100:
End Sub
```

Пример

```
Option Explicit
Private Sub CommandButton1_Click()
Hide
End Sub
Private Sub OptionButton1_Click()
Image1.PictureSizeMode = fmPictureSizeModeClip
Image1.PictureTiling = False
End Sub
Private Sub OptionButton2_Click()
Image1.PictureSizeMode = fmPictureSizeModeStretch
Image1.PictureTiling = False
End Sub

Private Sub OptionButton3_Click()
Image1.PictureSizeMode = fmPictureSizeModeZoom
Image1.PictureTiling = False
End Sub

Private Sub OptionButton4_Click()
Image1.PictureSizeMode = fmPictureSizeModeZoom
Image1.PictureTiling = True
End Sub

Private Sub UserForm_Initialize()
Const FileName As String = "T_46cut.JPG"

With Image1 'Установка свойств по умолчанию
    .Picture = LoadPicture(strPath)
    .PictureSizeMode = fmPictureSizeModeZoom
    .AutoSize = False
    .BorderStyle = fmBorderStyleNone
    .PictureTiling = False 'мозаика отключена
End With
OptionButton3.Value = True 'по умолчанию
End Sub
```

Ввод с листа Excel

```
Option Explicit
'здесь собираем данные о выделенных элементах списка
Public flDays(6) As Boolean
Sub ListBoxFill2()

Dim str(6) As String, strDays(6) As String
Dim i As Byte, idCount As Byte

idCount = 0 'нет выделенных дней
'Заполнение значений на форме
With UserForm2.ListBox1
    .Clear
    .ColumnCount = 1 'количество колонок в списке
    .Enabled = True 'Разрешен выбор
    .RowSource = "A1:A7"
End With
With UserForm2
    .Label1.Caption = "Введите описание элементов"
    .Label1.ForeColor = vbBlack
    .Label2.Caption = "Выбрано пользователем"
    .Label2.AutoSize = True
    .ListBox2.Clear 'очистка списка вывода
    .ListBox2.Enabled = False 'защита от юзера
End With

'Последовательное заполнение элементов списка
With UserForm2.ListBox1
    For i = 0 To 6
        flDays(i) = False
        str(i) = .List(i)
    Next i
    .ListIndex = 0
    .MultiSelect = fmMultiSelectExtended
End With
```

Microsoft Excel - L10-1.xls

Файл Правка Вид Вставка

Обычный Arial Cyr

Англо-Русский Военный

A14

	A	B
1	понедельник	13,45
2	вторник	12,17
3	среда	20
4	четверг	25
5	пятница	-13,5
6	суббота	-17
7	воскресенье	-15,7
8		

```
UserForm2.Show

'подсчёт выделенных дней
'Код специально сделан _неоптимальным_
'можно было обойтись без промежуточных массивов

With UserForm2
    If .ListBox1.ListCount > 0 Then 'что-то выбрано
        For i = 0 To .ListBox1.ListCount - 1
            flDays(i) = .ListBox1.Selected(i)
            If flDays(i) Then 'Найден выделенный день
                strDays(idCount) = str(i)
                .ListBox2.AddItem strDays(idCount)
                idCount = idCount + 1
            End If
        Next i
        .Label1.Caption = "Выбор закончен"
        .Label1.ForeColor = vbRed
        .ListBox1.Enabled = False 'Выбор закончен. Запрет на выбор
        .ListBox2.Enabled = True 'можно просматривать весь список
        .ListBox2.MultiSelect = fmMultiSelectSingle
        .Show
    End If
End With
End Sub
```

Ввод массива. Элемент управления ListBox

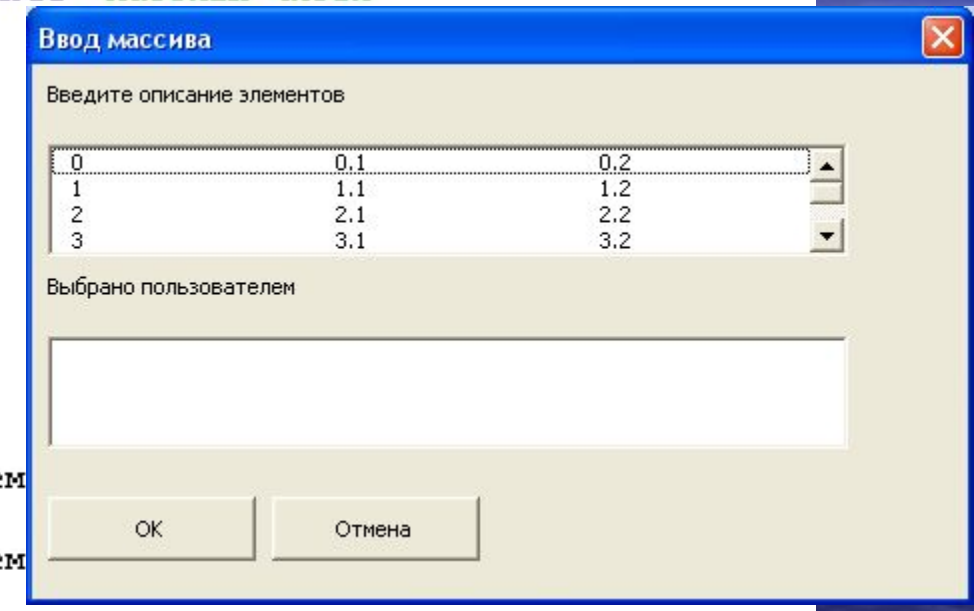
```
Option Explicit
'здесь собираем данные о выделенных элементах списка
Public flDays(6) As Boolean
Sub ListBoxFill2()

Dim str(6) As String, strDays(6) As String
Dim dNum(6) As Double 'массив чисел
Dim i As Byte, idCount As Byte

idCount = 0 'нет выделенных дней
'Заполнение значений на форме
With UserForm3.ListBox1
     .ColumnCount = 2 'количество колонок в списке
     .Clear
     .Enabled = True 'Разрешен выбор
     .RowSource = "A2:B8"
     .ColumnHeads = True 'Заголовки разрешены
End With

With UserForm3.ListBox2
     .Enabled = False 'защита от юзера
     .ColumnCount = 2 'количество колонок в списке
     .ColumnHeads = False 'Заголовки запрещены
     .Clear 'очистка списка вывода
End With

With UserForm3
     .Label1.Caption = "Введите описание элементов"
     .Label1.ForeColor = vbBlack
     .Label2.Caption = "Выбрано пользователем"
     .Label2.AutoSize = True
End With
```

Ввод массива (продолжение)

```
'Последовательное заполнение элементов списка
With UserForm3.ListBox1
    For i = 0 To 6
        flDays(i) = False
        str(i) = .List(i, 0) 'берем из 1-й колонки текст
        dNum(i) = .List(i, 1) 'берем из 2-й колонки числа
    Next i
    .ListIndex = 0 'переводим на 1-ю позицию
    .MultiSelect = fmMultiSelectExtended
End With

UserForm3.Show
'подсчёт выделенных дней
'Код специально сделан _неоптимальным_
'можно было обойтись без промежуточных массивов
With UserForm3
    If .ListBox1.ListCount > 0 Then 'что-то выбрано
        For i = 0 To .ListBox1.ListCount - 1
            flDays(i) = .ListBox1.Selected(i)
            If flDays(i) Then 'Найден выделенный день
                strDays(idCount) = str(i)
                .ListBox2.AddItem strDays(idCount) 'Добавляем строку в список
                .ListBox2.List(idCount, 1) = dNum(idCount) 'Добавляем в следующий столбец
                idCount = idCount + 1
            End If
        Next i
        .Label1.Caption = "Выбор закончен"
        .Label1.ForeColor = vbRed
        .ListBox1.Enabled = False 'Выбор закончен. Запрет на выбор
        .ListBox2.Enabled = True 'можно просматривать весь список
        .ListBox2.MultiSelect = fmMultiSelectSingle
        .Show
    End If
End With
End Sub
```

Ввод массива как массив

```
Option Explicit
Sub ListBoxFill3()
Const N As Byte = 9, M As Byte = 3
Dim i As Byte, idCount As Byte, j As Byte
Dim dIni(N, N) As Double, dRes(N, N) As Double 'массивы чисел

'Заполнение исходного массива
For i = 0 To N
    For j = 0 To M
        dIni(i, j) = i + j / 10
    Next j
Next i

idCount = 0 'нет выделенных дней
'Заполнение значений на форме
With UserForm4
    .Label1.Caption = "Введите описание элем
    .Label1.ForeColor = vbBlack
    .Label2.Caption = "Выбрано пользователем
    .Label2.AutoSize = True
End With

With UserForm4.ListBox1
    .ColumnCount = M 'количество колонок в списке
    .Clear
    .Enabled = True 'Разрешен выбор
    .ColumnHeads = False 'Заголовки запрещены
    .List = dIni 'Присвоение списку значений в массиве
    .MultiSelect = fmMultiSelectExtended 'Разрешение выбора нескольких элементов
End With
```



Ввод массива как массив (продолжение)

```
With UserForm4.ListBox2
    .Enabled = False 'защита от юзера
    .ColumnCount = M 'количество колонок в списке
    .ColumnHeads = False 'Заголовки запрещены
    .Clear 'очистка списка вывода
End With

UserForm4.Show
'подсчёт выделенных дней
'Код специально сделан _неоптимальным_
'можно было обойтись без промежуточных массивов
With UserForm4
    If .ListBox1.ListCount > 0 Then 'что-то выбрано
        For i = 0 To .ListBox1.ListCount - 1 'по строкам
            If .ListBox1.Selected(i) Then 'Найден выделенный день
                dRes(idCount, 0) = dIni(i, 0) 'добавляем 1-й столбец в результирующий массив
                .ListBox2.AddItem dRes(idCount, 0) 'Добавляем строку в список
                 For j = 1 To M 'идём по стобцам
                   dRes(idCount, j) = dIni(i, j) 'добавляем j-е столбцы в конечный массив
                   .ListBox2.List(idCount, j) = dRes(idCount, j) 'Добавляем в следующий столбец
                  Next j
                idCount = idCount + 1 'индекс конечного массива увеличивается
            End If
        Next i
        .Label1.Caption = "Выбор закончен"
        .Label1.ForeColor = vbRed
        .ListBox1.Enabled = False 'Выбор закончен. Запрет на выбор
        .ListBox2.Enabled = True 'можно просматривать весь список
        .ListBox2.MultiSelect = fmMultiSelectSingle
        .Show
    End If
End With
End Sub
```

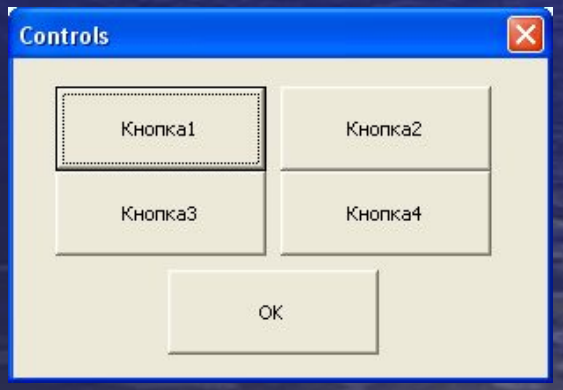
Перебор элементов. Коллекция Controls

Иногда требуется произвести однотипные действия со множеством элементов управления. Для их производства удобно задействовать коллекцию **Controls**

Пример

Изменение начертания шрифта на всех кнопках на жирный и обратно

```
Private Sub CommandButton5_Click()
Dim i As Integer, j As Integer

For i = 0 To Me.Controls.Count - 1
    j = InStr(1, Me.Controls.Item(i).Name,
"CommandButton", vbTextCompare)
    If j <> 0 Then
    'Имя элемента управления говорит о его
типе
    Me.Controls.Item(i).Font.Bold =
Not Me.Controls.Item(i).Font.Bold
    End If
Next i
End Sub
```



СПАСИБО ЗА ВНИМАНИЕ!