Лекция 8

Реализация приложений с использованием системных математических функций

В рассмотренных нами приложениях калькуляторов были использованы варианты вычислений с использованием арифметических операций. Рассмотрим ещё один пример калькулятора, в котором выполняются вычисления значений математических функций. Например, значения Sin(X), Cos(X) и Ln(X).

Проект окна пользовательской формы, реализующей это приложение, показан на рис.1.

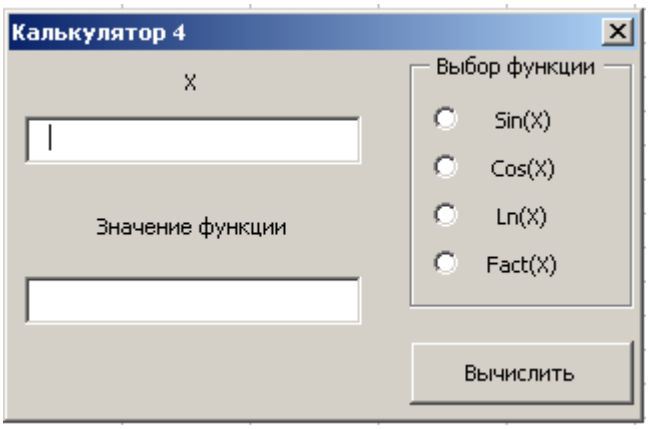

Рис.1. Проект окна формы "Калькулятор 4"

Тип элемента управления	Изменяемое свойство	Новое значение	Комментарий
UserForm (Форма)	Name	*FrmCalc4*	
	Caption	*Калькулятор 4*	
Label1 (Метка1)	Caption	*X*	
	TextAlign	*2 - fmTextAlignCenter*	
TextBox (Поле ввода)	Name	*TxtBoxX*	
Label2 (Метка1)	Caption	*Значение функции*	
	TextAlign	*2 - fmTextAlignCenter*	
Frame1	Caption	*Выбор функции*	
OptionButton1	Caption	*Sin(X)*	Для всех кнопок установим свойство TextAlign равным *2 - fmTextAlignCenter*
OptionButton2	Caption	*Cos(X)*	
OptionButton3	Caption	*Ln(X)*	
OptionButton4	Caption	*Fact(X)*	
CommandButton (Кнопка)	Name	*CmdEnter*	
	Caption	*Вычислить*	
TextBox (Поле ввода)	Name	*TxtResult*	
	Locked	*True*	блокируем ввод текста в поле ввода
	TextAlign	*3 - fmTextAlignRight*	

Ниже приведён текст процедуры, выполняемой при нажатии кнопки "Калькулятор 4" на листе Формы книги Excel.

```
Option Explicit
Private Sub CmdEnter_Click()
' Объявление переменных
   Dim X As Double
' переменной X присваивается значение, введённое в поле ввода TxtBoxX
If IsNumeric(TxtBoxX.Text) Then X = CDbl(TxtBoxX.Text) _
Else MsgBox ("Введённое значение X не число"): Exit Sub
' Sin(X)
   If OptionButton1.Value Then TxtResult.Text = Sin(X)
   If OptionButton1.Value Then Label2.Caption = "Sin(X)"
' Cos(X)
   If OptionButton2.Value Then TxtResult.Text = Cos(X)
   If OptionButton2.Value Then Label2.Caption = "Cos(X)"
' Ln(X)
If OptionButton3.Value And X <= 0 Then MsgBox ("Введённое значение X <= 0"): Exit Sub
If OptionButton3.Value Then TxtResult.Text = Application.WorksheetFunction.Ln(X)
Label2.Caption = "Ln(X)"
If OptionButton4.Value And X < 0 Then MsgBox ("Введённое значение X < 0"): Exit Sub
If OptionButton4.Value Then TxtResult.Text = Application.WorksheetFunction.Fact(X)
   Label2.Caption = "Fact(X)"
End Sub
```

В приведённом примере вычисляются значения четырёх функций ***Sin(X), Cos(X), Ln(X)*** и ***Fact(X).*** Первые две функции входят в набор встроенных функций языка программирования ***Visual Basic***. Функции ***Ln(X)*** и ***Fact(X)*** определены как функции, входящие в коллекцию функций рабочего листа – ***WorksheetFunction.*** Вызов такого рода функций реализуется через запись последовательности ключевых слов ***Application*** и ***WorksheetFunction,*** отделяемых друг от друга точкой. Ввод точки после ключевого слова ***Application*** обеспечивает выпадение списка, в котором имеется строка ***WorksheetFunction.*** Точка, введённая после этого слова, обеспечивает выпадение очередного списка, в котором имеются строки с именами функций ***Ln*** и ***Fact.***

Элементы управления. Методы, используемые в реализациях алгоритмов обработки

Элементы управления (ЭУ) - объекты, размещаемые в окнах пользовательских форм, обладают определёнными свойствами, которые могут быть заданы или изменены. Для задания и изменения свойств ЭУ, могут быть использованы различные методы.

В файле "***Миникалькулятор***" я предложил Вам рассмотреть самостоятельно, тексты двух вариантов реализации кальуляторов. В окна этих приложений включены такие ***ЭУ*** как ***Label*** (*Метка*), ***TextBox*** (*Поле*

ввода), ***Frame*** (*Рамка*), ***OptionButton*** (*Переключатель*), ***CommandButton*** (*Кнопка*) и ***CheckBox*** *(Флажок).*

Рассмотрим ещё одно приложение, в котором используется ещё один новый для нас ***ЭУ*** – *Список* (***ListBox***).

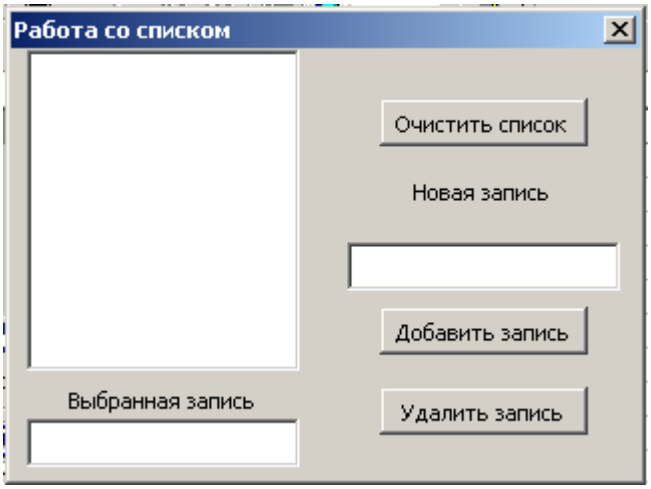

Рис. 2. Проект окна формы приложения "Работа со списком"

В окно этой формы включены кнопки, метки для полей ввода, поля ввода и поле списка. Рассмотрим таблицу, в которой отметим все изменяемые свойства формы и ЭУ, размещённых в окне этой формы.

Тип элемента управления	Изменяемое свойство	Новое значение	Комментарий
UserForm (Форма)	Name	*FrmListBox1*	
	Caption	*Работа со списком*	
ListBox (Список)	Name	*LstItems*	
CommandButton (Кнопка)	Name	*CmdClearList*	
	Caption	*Очистить список*	
Label (Метка)	Name	*LblNewRecord*	
	Caption	*Новая запись*	
TextBox (Поле ввода)	Name	*TxtNewRecord*	
CommandButton (Кнопка)	Name	*CmdAdd*	
	Caption	*Добавить запись*	
CommandButton (Кнопка)	Name	*CmdDelete*	
	Caption	*Удалить запись*	
Label (Метка)	Name	*LblLstRecord*	
	Caption	*Выбранная запись*	
TextBox (Поле ввода)	Name	*TxtRecord*	
	Locked	*True*	!!! блокирует ввод с клавиатуры

Алгоритм работы приложения "Работа со списком" и реализующие его процедуры

Элементы управления, размещённые на поле формы, обеспечивают ввод записи в текстовое поле, добавление записи в поле списка при нажатии

кнопки ***"Добавить запись"*** и очистку текстового поля, удаление записи из списка при нажатии кнопки ***"Удалить запись"*** и полную очистку списка при нажатии кнопки ***"Очистить список".***

Для внесения новой записи в поле ЭУ ***ListBox*** (*Список*) эта запись предварительно вводится в ***Поле ввода*** (*TextBox*) ***TxtNewRecord.*** Используя свойство ***.Text*** , определяющее или устанавливающее значение текста в ЭУ **TextBox.**

Текст этой записи присваивается как значение строковой переменной ***rec.***

При нажатии кнопки ***"Добавить запись"*** выполняется процедура ***CmdAdd_Click,*** записывающая текст из поля ввода в поле списка ***LstItems,*** создавая тем самым новую запись в списке. Для наглядности к тексту создаваемой записи добавляется её порядковый номер. Ниже приводится текст этой процедуры.

```
' Процедура, работающая при нажатии кнопки "Добавить запись"
Private Sub CmdAdd_Click()
   Static iCount As Long ' счётчик числа записей в списке
   Dim rec As String ' объявление строковой переменной для записи
   iCount = iCount + 1
   rec = TxtNewRecord.Text ' запись - текст, вводимый в поле TextBox
  ' формируется запись, к тексту которой добавляется номер
  ' помещаем запись в список
   LstItems.AddItem(rec) & "  " & CStr(iCount) ' добавляемая запись
  ' очищаем поле ввода после включения записи в список
   TxtNewRecord.Text = ""
End Sub
```

При необходимости любая выбранная в списке запись может быть из него удалена. Для этой цели в число ЭУ, размещенных в поле формы, включена кнопка ***CmdDelete*** с заголовком "***Удалить запись***". При нажатии на эту кнопку выполняется процедура ***CmdDelete_Click().***

```
Private Sub CmdDelete_Click()
    With LstItems ' оператор, определяющий объект, свойства и
                  ' методы которого используются и изменяются
        If .ListCount >= 1 Then _  ' в списке есть записи?
        ' Убеждаемся, что элемент выбран
            If .ListIndex = -1 Then Exit Sub ' список пуст. Конец
           .RemoveItem.ListIndex ' удаляем выбранную запись
        End If
    End With ' завершающий оператор объявления объекта
End Sub
```

Проанализируем более подробно текст этой процедуры. Оператор ***With LstItems*** указывает, что в дальнейшем имя объекта ***LstItems*** при действиях, определяемых как свойства или методы этого объекта, может быть опущено. Это указание действует вплоть до завершающего оператора ***End With.*** Таким образом, свойства ***.ListCount*** и ***.ListIndex*** определяются для объекта ***LstItems.*** Свойство ***ListCount*** объекта ***LstItems*** определяет число записей в списке - объекте ***LstItems,*** а свойство ***ListIndex*** число выбранных в данный момент записей списка.

Вторая команда в этой процедуре

If .ListCount >= 1 Then _

проверяет наличие в списке хотя бы одного элемента. Если в списке нет записей, то значение **ListCount** равно 0, а значение **ListIndex** равно –1 выполнение процедуры завершается.

Если же запись с номером ***.ListIndex*** в списке реально существует и она выбрана, то она удаляется из списка методом объекта ***LstItems .RemoveItem.***

В приложение "Работа со списком" включена процедура ***CmdClear_Click()***, обеспечивающая возможность полной очистки списка при нажатии кнопки ***"Очистить список".*** Текст этой процедуры приведён ниже.

```
' Процедура, работающая при нажатии кнопки "Очистить список"
        Private Sub CmdClear_Click()
          LstItems.Clear
        End Sub
```

Для объекта ***LstItems*** , соответствующего списку, выполняется метод ***Clear***, удаляющий все объекты из списка.

В форму рассматриваемого нами приложения включено ещё одно поле ввода (***TextBox***) отмеченное как "Выбранная запись". В этом поле отображается запись, выбранная в списке ***LstItems***. Для того, чтобы запретить ввод текста в это поле с клавиатуры при создании этого ЭУ мы определяем его свойство Locked как Истина (True).

Для того, чтобы сделать кнопку не выводимой на печать, надо в окне Properties определить значение свойства PrintObject как FALSE !!!

Список с несколькими столбцами

Число столбцов в списке определяется значением свойства ***ColumnCount.*** Кроме этого свойства при работе с такими списками следует установить значение свойства ***ColumnWidth***. Это свойство для каждого столбца определяется цифрой, отделяемой от следующей цифры точкой с запятой.

Ниже на рис. 3 приведен вид окна пользовательской формы, в которое выводится таблица, содержащая два столбца: столбец значений аргумента X и столбец значений функции F(X).

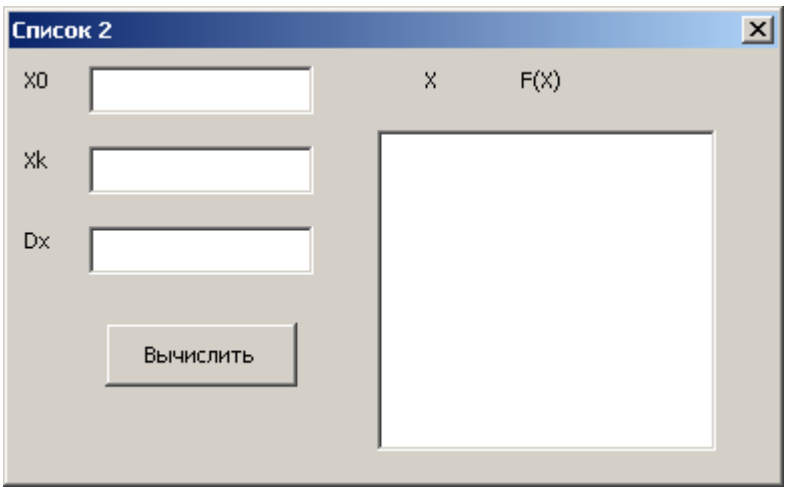

Рис. 3.

В этом окне

```
Option Explicit
Private Sub UserForm_Initialize()
   With LstTab
   .ColumnCount = 2
   .ColumnWidths = "40;60"
   End With
End Sub
Private Sub CommandButton1_Click()
   Dim X0 As Double, Xk As Double, Dx As Double
   Dim X  As Double, F As Double, i As Integer, k As Integer
   Dim Values()
   If IsNumeric(TextBox1.Text) Then X0 = CDbl(TextBox1.Text)
   If IsNumeric(TextBox2.Text) Then Xk = CDbl(TextBox2.Text)
   If IsNumeric(TextBox3.Text) Then Dx = CDbl(TextBox3.Text)
   k = CInt((Xk - X0) / Dx)
   If (k < 1) Then MsgBox "Проверьте исходные данные": Exit Sub
     ReDim Values(k, 1)
   i = 0
   For X = X0 To Xk Step Dx
    F = Sin(X)
    'F = Format(F, "#.####")
    Values(i, 0) = X
    Values(i, 1) = F
    i = i + 1
   Next X
   LstTab.List = Values
End Sub
```