Лекция 6

1. Автоматизация проектов VBA

Реализации алгоритмов решения задачи табулирования функций с использованием средств ***VBA***, рассмотренные в предыдущих лекциях, мало ***отличаются*** от реализаций этих же алгоритмов средствами ***Excel***.

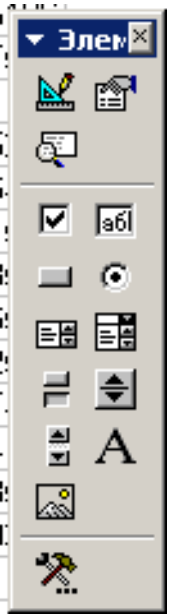
Возможности ***VBA*** позволяют создавать более эффективные проекты, благодаря включению в них средств для организации диалога пользователя с программной реализацией проекта – ***интерфейса между пользователем и программой***. К таким средствам можно отнести ***элементы управления***,

Набор элементов управления, которые могут быть помещены на рабочий лист книги ***Excel,*** размещается на панели инструментов (***ПИ***) "***Элементы управления***" (***ЭУ***), которая может быть открыта из меню "***Вид***" программы ***Excel*** при выборе в этом меню команды "***Панели инструментов ►***".

ПИ "***ЭУ***" разделена на три поля. На первом (верхнем) поле находятся кнопки, обеспечивающие переключение ***VBA*** в режим конструктора, открытие окна "***Свойства***" и открытие окна редактора с исходным текстом модуля активного листа книги.

На втором (нижнем) поле ***ПИ*** находятся кнопки, используемые для создания ***ЭУ*** на активном листе книги ***Excel.*** Это кнопки "***Флажок***" – *CheckBox*, "***Поле***" – *TextBox*, "***Кнопка***" – *CommandButton*, "***Переключатель***" - *OptionButton*, "***Список***" *ListBox*, "***Поле со списком***" - *ComboBox*, "***Выключатель***" - *ToggleButton*, "***Счётчик***" - *SpinButton*, "***Полоса прокрутки***" - *ScrollBar*, "***Надпись***" - *Label*, "***Рисунок***" - *Image*.

Кнопка "***Другие элементы…***", расположенная в третьем разд "***ЭУ***" открывает окно диалога, в котором находится список с числом специальных ***ЭУ***. Число таких элементов зависит от сделанного при установке программ комплекса ***Microsoft Office*** (у 202).



При нажатии на любую из кнопок, находящихся во втором или третьем полях ***ПИ*** "***ЭУ"*** программа ***Excel*** переводится в "***Режим конструктора***" – режим, обеспечивающий возможность создания на листе книги выбранного **Элемента Управления** и запись программы, реализующей функции этого ***ЭУ***.

Для проверки работоспособности созданного ***ЭУ*** следует нажать кнопку "***Выход из режима конструктора***". При этом система переводится в режим выполнения. В этом режиме взаимодействие с программами осуществляется на уровне пользователя, т.е. нажатием ***ЭУ*** пользователь может запустить программу, соответствующую этому ***ЭУ***, но не может внести какие-либо изменения в её текст.

В инструкции по выполнению лабораторной работы №8 дано описание процесса создания на листах книги ***Excel*** кнопок, обеспечивающих переход с одного листа книги на другой. Комментируя эту ***ЛР***, отмечу, что программы, запускаемые при щелчке ***ЛКМ*** по кнопке, могут быть реализованы двумя способами. Первый, это включение в процедуру ***CommandButton_Click()*** команды ***Sheets("Имя_листа_книги").Select,*** а второй – включение команды ***Worksheets("Имя_листа_книги").Activate.***

Процедуры, выполняемые при нажатии командных кнопок, помещаемых на листах книги ***Excel,*** позволяют организовать выполнение различных сервисных задач.

В качестве примеров такого рода задач, рассмотрим два примера:

- вывод сведений об авторе, выполненной работы, и
- вывод сведений о размерах и распределении памяти, имеющейся на Вашем компьютере.

Пример 1. Создание кнопки для вывода информации "Об авторе"

- На листе "***Оглавление***" книги, перейдя предварительно в режим конструктора, создадим командную кнопку. ***ЩПК*** по вновь созданной кнопке откроем контекстное меню. Выберем в этом меню строку "***Свойства***".

- В строку "***Caption***" (*Заголовок*) этого окна введём текст "***Сведения об авторе".***
- В строку "***Name***" (*Имя*) изменим стандартное имя кнопки, присвоенное ей при создании (***CommandButtoni***), на более информативное ***CmdAboutAuthor***.
- Выполнив двойной ***ЩЛК***, перейдём в окно редактора ***VBA*** на страницу модуля листа "***Оглавление***". В поле ввода этого листа появляются две строки заготовки процедуры, которая будет выполняться при нажатии на созданную нами кнопку.

```
Private Sub cmdAboutAuthor_Click()

End Sub
```

- Для вывода информации об авторе работы воспользуемся функцией ***MsgBox***, используемой в программах ***VBA*** для вывода окон с сообщениями программы. Запишем после заголовка процедуры строку с обращением к функции ***MsgBox***.

```
MsgBox "Работа выполнена студентом Ивановым И., гр. 1-П-1,. 2007г." _
    ,,"Сведения об авторе"
```

- Выйдем из режима конструктора и нажмём кнопку "Сведения об авторе" на листе книги. Результат выполнения этой процедуры показан на рис. 1.

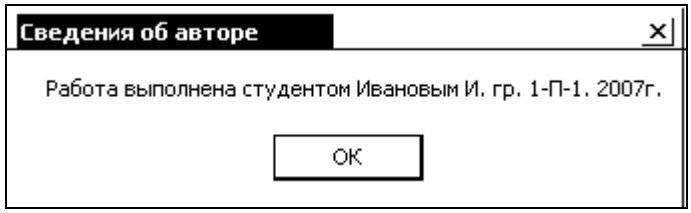


Рис. 1. ДО, выводимое на экран при нажатии кнопки "Сведения об авторе"

Синтаксис обращения к функции *MsgBox:*

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

Синтаксисом функции **MsgBox** определяются следующие именованные аргументы:

prompt – подсказка. Обязательный аргумент. Строковое выражение, отображаемое сообщением, выводимым в диалоговое окно.

buttons – кнопки. Необязательный аргумент. Числовое выражение - сумма, определяемая числом и типом кнопок, отображаемых в ДО **MsgBox,** типом используемой пиктограммы, идентификацией кнопки, активной по умолчанию и модальностью (the modality) окна. Если этот аргумент опущен, то его значение "по умолчанию" равно 0.

title – заголовок. Необязательный аргумент. Строковое выражение, отображаемое как текст в заголовке окна **MsgBox.** Если аргумент ***title*** опущен, то в заголовке окна **MsgBox** выводится название приложения.

Аргументы ***helpfile***, ***context*** определяются при создании для ДО **MsgBox** контекстной справки.

Пример 2. Создание кнопки для вывода сведений о распределении памяти ЭВМ

- На листе "***Оглавление***" книги, перейдя предварительно в режим конструктора, создадим командную кнопку. ***ЩПК*** по вновь созданной кнопке откроем контекстное меню. Выберем в этом меню строку "***Свойства***".
- В строку "***Caption***" (*Заголовок*) этого окна введём текст "***Сведения о распределении памяти ЭВМ***".
- В строку "***Name***" (*Имя*) изменим стандартное имя кнопки, присвоенное ей при создании (***CommandButtoni***), на более информативное ***CmdMemoryDistribution***.
- Выполним двойной ***ЩЛК*** по кнопке, в результате чего перейдём в окно редактора ***VB***, соответствующее модулю листа "***Оглавление***". В поле окна редактора появится заготовка новой процедуры с кодом

```
Private Sub CmdMemoryDistribution_Click()

End Sub
```

- За первой строкой этой процедуры введём следующие команды:

```
'объявление трёх переменных типа Long
Dim S1 As Long, S2 As Long, S3 As Long
S1 = Application.MemoryFree
S2 = Application.MemoryUsed
S3 = Application.MemoryTotal
```

```
MsgBox "Размер памяти ЭВМ: " & S3 _
& " байт. Использовано: " & S2 _
& " байт. Свободно: " & S3 & "байт."
```

- Вернёмся на лист "***Оглавление***", выйдем из режима конструктора и нажмём кнопку "***Сведения о распределении памяти ЭВМ***". В результате выполнения созданной нами процедуры будет выведено окно сообщения ***MsgBox***:

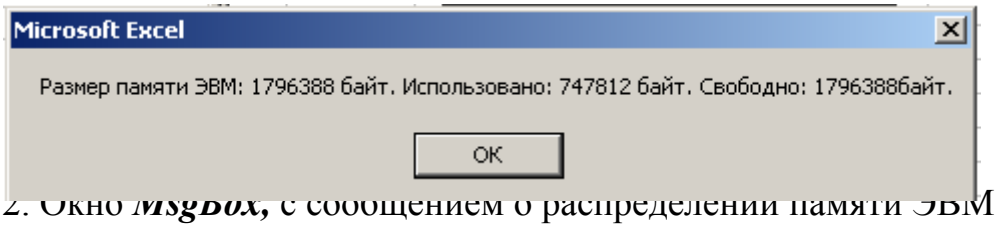
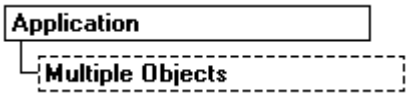


Рис. 2. Окно ***MsgBox,*** с сообщением о распределении памяти ЭВМ

Ключевое слово – ***Application (Приложение)***, представляет в данном случае приложение Microsoft Excel в целом.



В этом приложении определены ***Свойства*** (*Properties*): ***MemoryTotal***, ***MemoryUsed*** и ***MemoryFree***, которые позволяют определить размеры памяти ЭВМ, а также размеры используемой в данный момент памяти и свободной памяти.

Оператор ***MsgBox*** обеспечивает обращение к встроенной функции VBA, используемой для вывода диалогового окна с сообщением о размерах полной, использованной и свободной памяти ЭВМ (рис. 1).

Пример 3. Программа табулирования функции, запускаемая при нажатии кнопки

Рассмотрим пример программной реализации алгоритма табулирования разветвляющейся функции, в котором для организации ввода исходных данных используются ***ЭУ***, вызываемые непосредственно на лист книги ***Excel***.

- Вставим в книгу ***Excel*** новый лист. Переименуем его в ***"ТабКнопка"***, а затем, перейдя предварительно в режим конструктора, создадим на нём кнопку.

- ***ЩПК*** по вновь созданной кнопке откроем контекстное меню. Выберем в этом меню строку "***Свойства***".
- В строку "***Caption***" (*Заголовок*) этого окна введём текст "***Табулирование функции F(X)***".
- В строку "***Name***" (*Имя*) изменим стандартное имя кнопки, присвоенное ей при создании (***CommandButtoni***), на более информативное ***CmdFuncTab***.
- Выполним двойной ***ЩЛК*** по кнопке, в результате чего перейдём в окно редактора ***VB***, соответствующее модулю листа ***"ТабКнопка"***. В поле окна редактора появится заготовка новой процедуры с кодом

```
Private Sub CmdFunTab_Click()

End Sub
```

- Введём в строки, следующие после заголовка процедуры, тексты комментариев и операторы программы

```
' Объявление переменных X X0, Xк и Dx, определяющих аргумент,
' параметры изменения и шаг изменения аргумента функции
Dim X, X0, Xk, Dx As Double
' Объявление переменной sht как объекта Worksheet – рабочий лист
Dim sht As Worksheet
' Объявление переменных, определяющих функцию и её параметры
Dim Y, a, b As Double
' Объявление вспомогательных переменных
Dim i, i0 As Integer
' Определяем программное имя листа, на котором будут выполняться
' вычисления
Set sht = Worksheets("Lection6")
```

- Сначала, используя свойство ***Range*** для листа ***sht***, очистим диапазон столбцов ***A:D***, а затем, используя это же свойство, заполним ячейки заголовка таблицы Для этого запишем текст инструкции:

```
' используя метод Clear, очищаем столбцы листа sht
sht.Range("A:F").Clear

sht.Range("A1").Value = "Автоматизированное заполнение таблицы"
sht.Range("A4").Value = "№"
sht.Range("b4").Value = "X"
sht.Range("c4").Value = "Y"
sht.Range("A2").Value = "a="
sht.Range("C2").Value = "b="
sht.Range("A3").Value = "X0="
sht.Range("C3").Value = "Xk="
```

sht.Range("E3").Value = "Dx="

Метод ***Clear*** (*Очистить*) используется для очистки ячеек в диапазоне ***Range***, определяемом столбцами от ***A*** до ***F***.

Свойство ***Value*** (*Значение*) для объекта ***Worksheet***, позволяет определить значения в ячейках таблицы, определяемых свойством ***Range.***

В ячейки, значения которых определяются свойством ***Value,*** записываются текстовые константы, определяющие имена параметров, которые должны быть введены в соответствующие ячейки для формирования заголовка документа.

- Для того, чтобы организовать интерфейс пользователя с программой воспользуемся встроенной функцией ***VBA MsgBox.***

MsgBox("Выполните ввод необходимых исходных данных")

При обработке оператора ***MsgBox*** открывается окно, в котором выводится информационное сообщение (рис. 3). После нажатия кнопки ***OK*** в ДО ***MsgBox*** организуем ввод данных, необходимых для выполнения табулирования функции.

Рис. 3.

- Числовые значения, определяющие диапазон и шаг изменения аргумента, а также параметры табулируемой функции можно ввести, использовав для этого, последовательность обращений к функции ***VBA InputBox.*** Рассмотрим оператор для определения значения ***X0.***

X0 = Val(InputBox("Введите значение X0","X0","0"))

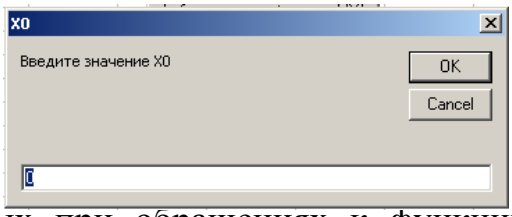
При выполнении этого оператора на рабочее поле, выполняемой программы, выводится ДО ***InputBox*** (см. рис. 4)**.**

В рабочее поле ДО выводится подсказка, определяемая значением первого параметра оператора ***InputBox:***

"Введите значение X0".

В заголовке ДО отображается значение второго параметра ***- "X0"***, а в поле ввода – значение третьего параметра ***"0"*** - значение, вводимое "по умолчанию" (см. рис. 4).

Рис. 4.



- Значения параметров вводимых при обращениях к функции ***InputBox*** имеют тип ***string*** (*строка*). Для того чтобы в дальнейшем при выполнении программы можно было использовать значения, введённые средствами ***InputBox***, необходимо преобразовать к одному из числовых типов. Так в приведённом выше операторе значение, введённое в поле ввода окна ***InputBox***, должно быть преобразовано к типу переменной ***X0*** – ***Double***. Аналогичные операторы записываются для ввода параметров ***Xk, Dx, a, b.***

```
Xk= Val(InputBox("Введите значение Xk","Xk","1"))
Dx = Val(InputBox("Введите значение Dx","Dx","0.1"))
a  =  Val(InputBox("Введите значение a","a","1"))
b  =  Val(InputBox("Введите значение b","b","1"))
```

- Для отображения введённых значений параметров в ячейках создаваемой таблицы дополним каждую из вышеприведённых инструкций с обращением к функции ***InputBox*** ещё одной инструкцией, записывающей введённое значение в ячейку таблицы. Ниже приводится фрагмент программы, в котором реализованы операции ввода исходных данных и операции записи введённых значений в ячейки таблицы.

```
X0 = Val(InputBox("Введите значение X0", "X0", "0"))
     sht.Range("b3").Value = X0
```

Конец 6-й лекции