

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ «МАМИ»**

А.Н. Любин

**СОЗДАНИЕ  
ПОЛЬЗОВАТЕЛЬСКИХ ФОРМ  
В VISUAL BASIC FOR APPLICATIONS**

**Учебное пособие**  
по дисциплинам  
«Информатика» и «Информационные технологии»  
для студентов  
всех направлений и специальностей

Москва  
2012

УДК 681.3.06

*Разработано в соответствии с Государственным образовательным стандартом 2008 г. Для всех направлений и специальностей на основе примерных программ по дисциплинам «Информатика» и «Информационные технологии».*

Рецензенты: генеральный директор «Института информационных технологий», д.т.н., профессор В. Г. Зубков;  
профессор кафедры «Информационные системы и дистанционные технологии» МГТУ «МАМИ» к.т.н. А. И. Макаров.

## «СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ФОРМ В VISUAL BASIC FOR APPLICATIONS»

учебное пособие по дисциплинам «Информатика» и «Информационные технологии» для студентов всех направлений и специальностей: М., МГТУ «МАМИ», 2012

Кафедра

«Информационные системы и дистанционные технологии»

Пособие ориентировано на изучение темы « Пользовательские формы VBA в Microsoft Office», а именно в Excel и получение навыков в создании пользовательских форм при решении задач на ПК.

© А.Н. Любин

© МГМУ «МАМИ», 2012

## Введение

Язык программирования VBA (Visual Basic for Applications) объектно-ориентированный язык программирования, базируется на синтаксисе и операторах языка Visual Basic. VBA встроен в среду всех основных, популярных офисных приложений, и позволяет манипулировать их объектами, используя их методы и свойства.

Чаще всего основным офисным приложением, в котором пользователь создаёт свои приложения, является MS Excel. Поэтому в настоящем учебном пособии рассматривается программирование на VBA именно в Excel.

В VBA, как и в большинстве языков, любые приложения, обладающие пользовательским интерфейсом, создаются (программируются) на основе специальных средств – пользовательские формы или экранные формы. Именно этот раздел VBA, как наиболее интересный и трудно усваиваемый, рассматривается в данном учебном пособии. Предполагается, что читатель знаком с элементами структурного программирования на VBA. Пользовательские формы дают возможность пользователю создавать разнообразные диалоговые окна в разрабатываемых приложениях и размещать в окнах элементы управления.

В учебном пособии подробно рассмотрены примеры создания пользовательских форм и процедур к ним, предложены задания для самостоятельного выполнения. Пособие может использоваться, как в учебном процессе при изучении VBA, так и при самостоятельном обучении.

Примеры создания пользовательских форм, их процедур и изображения диалоговых окон, приведённые в пособии, выполнены в среде MS Office 2007.

# 1. Объекты, методы, свойства и события в VBA

VBA относится к языкам объектно-ориентированного программирования, поэтому знакомство с ним следует начать с понятия объекта.

**Объект** - основной элемент VBA Excel. В VBA объектами являются рабочая книга, рабочий лист и его составляющие, например:

Sheet - лист Excel;

Cell - ячейка;

Range - диапазон ячеек;

Application - приложение;

**UserForm** - пользовательская форма.

CommandButton –кнопка, и другие.

Работа с объектами возможна через его методы и свойства. Над объектами можно совершать различные действия.

**Методы** это действия, совершаемые над объектами. Например, ячейку или диапазон ячеек можно очистить (Clear), можно выбрать (Select), приложение закрыть (Quit), пользовательскую форму показать (Show) или скрыть (Hide). Название метода отделяется от названия объекта точкой:

**название объект . название метода**

Ниже приведены примеры использования методов применительно к объектам:

Range("B2:E2").Select - выбрать диапазон ячеек B2:E2;

Range("C1:C5").Clear - очистить диапазон ячеек C1:C5;

UserForm2.Hide - скрыть форму под номером 2;

UserForm5.Show - показать форму под номером 5;

Application.Quit - выйти из приложения.

**Свойства** служат для описания, задания характеристик объектов. Например, размера и цвета шрифта, положения формы на эк-

ране или состояние объекта (доступность, видимость), задания значений. Чтобы задать или изменить характеристику объекта, надо изменить значение его свойства, т.е. присвоить ему определенные значения.

Синтаксис задания значения свойства следующий:

**Объект.Свойство = Значение Свойства**

где:

**Объект** - обозначает имя объекта;

**Свойство** - имя свойства, которому присваивается значение, имя объекта отделяется от имени свойства точкой.

Ниже приведены примеры задания свойств объектам:

`Range("B1").Value = 2011` - поместить в ячейку B1 значение 2011.

`Range("A1:A12").Text = "Программа"` - поместить в диапазон ячеек A1:A12 текст Программа.

`Range("C2").Font.Size = 18` - в ячейке C2 установить размер шрифта 18.

Свойства: `Value` ( позволяет задать значение), `Text` (ввод текста), `Font.Size` (задание размера шрифта).

**Событие** представляет собой действие, например, щелчок мышью или нажатие клавиши, перемещение мыши или выход из программы, для которого можно запрограммировать отклик, т.е. реакцию объекта на произошедшее событие.

В языке программирования VBA для каждого объекта определен набор стандартных событий. Например, стандартное событие для объекта `CommandButton` (кнопка) - `Click` (щелчок мышью).

Если пользователь нажимает на кнопку, то это событие. На это событие должен быть отклик, т.е. выполнение какой-либо процедуры (программы). Такая процедура называется процедурой обработки события и имеет стандартное имя. Если такой отклик не создан,

т.е. не написана соответствующая процедура, то система не будет реагировать на это событие. Для каждого события можно написать процедуру, которая будет срабатывать именно тогда, когда это событие произойдет. На каждое событие пишется своя процедура, которая это событие обрабатывает, т. е. выполняет определенные действия, которые должны быть выполнены, когда это событие происходит. Особое значение понятие события имеет при написании процедур реакций пользовательской формы на изменения ее элементов.

## 2. Пользовательские формы

При работе в Excel, как и в большинстве, других приложениях, приходится сталкиваться с такими элементами интерфейса, как диалоговые окна. Диалоговые окна используются повсеместно для получения информации для ввода и вывода сообщений и данных.

Одним из объектов VBA, объектом визуализации являются **UserForm** - пользовательские формы.

Пользовательские формы **UserForm** - это диалоговые окна интерфейса процедур VBA. С их помощью пользователь может эффективно передавать данные в процедуры и получать результаты их работы.

Пользовательские формы дают возможность пользователю создавать диалоговые окна в разрабатываемых приложениях и размещать в окнах элементы управления.

В VBA есть две функции – MsgBox и InputBox, которые позволяют отображать простые диалоговые окна, не создавая пользовательскую форму. Эти окна можно видоизменять, использовать управляющие ими параметры, но они не имеют тех широких и эффективных возможностей, опций которые предоставляют пользовательские формы.

VBA предлагает обширные возможности, которые можно использовать при создании пользовательских диалоговых окон, для программирования элементов управления, размещаемых непосредственно на листах Excel, или для работы с диалоговыми окнами, встроенными в Excel. Дополнительные сведения об этих возможностях см. в разделе «Элементы управления, диалоговые окна и формы» в справочном руководстве разработчика для Excel 2007.

В последующих разделах мы подробно рассмотрим процесс создания пользовательских форм.

### Создание пользовательской формы

Чтобы создать форму, откройте редактор Visual Basic для этого нажмите (Alt + F11) или выберете на ленте вкладку «Разработчик», а на ней Visual Basic см. рис.1

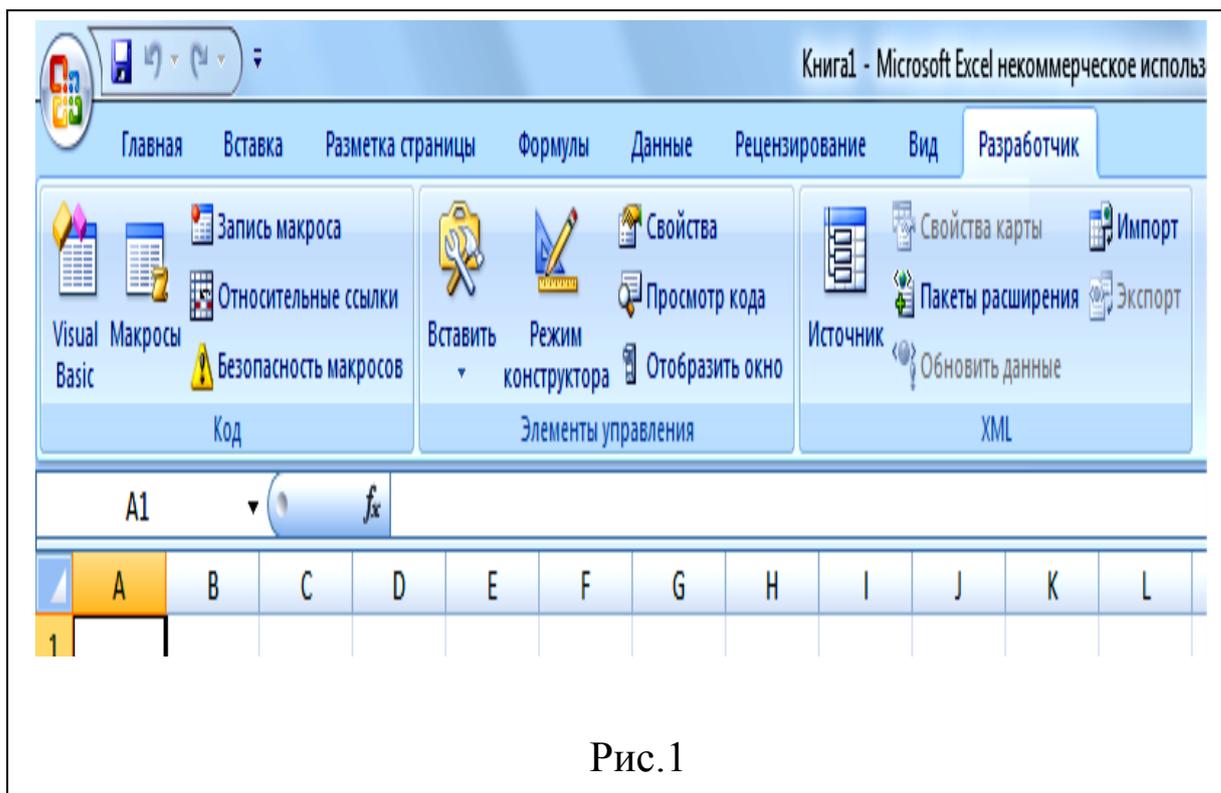


Рис.1

В открывшемся окне Microsoft Visual Basic см. рис. 2 выбрать на ленте вкладку Insert. В открывшемся меню выбрать UserForm.

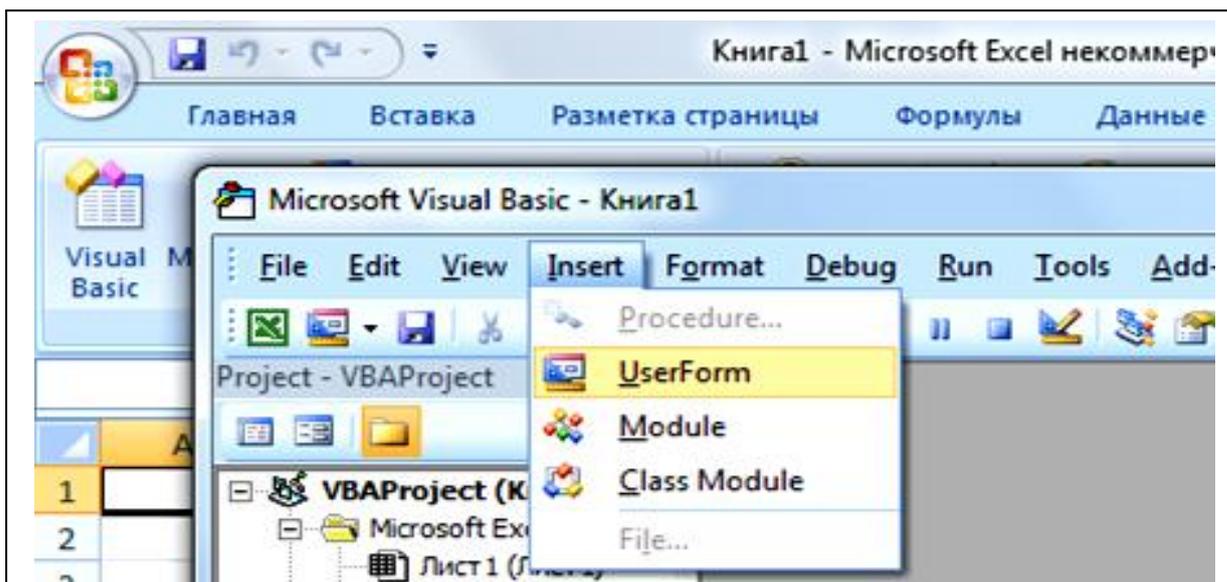


Рис. 2

На экране появилась пустая форма UserForm1 см. рис.3, инструментальная панель Toolbox и при нажатии правой кнопки мыши контекстное меню с предложением Properties – свойства формы.

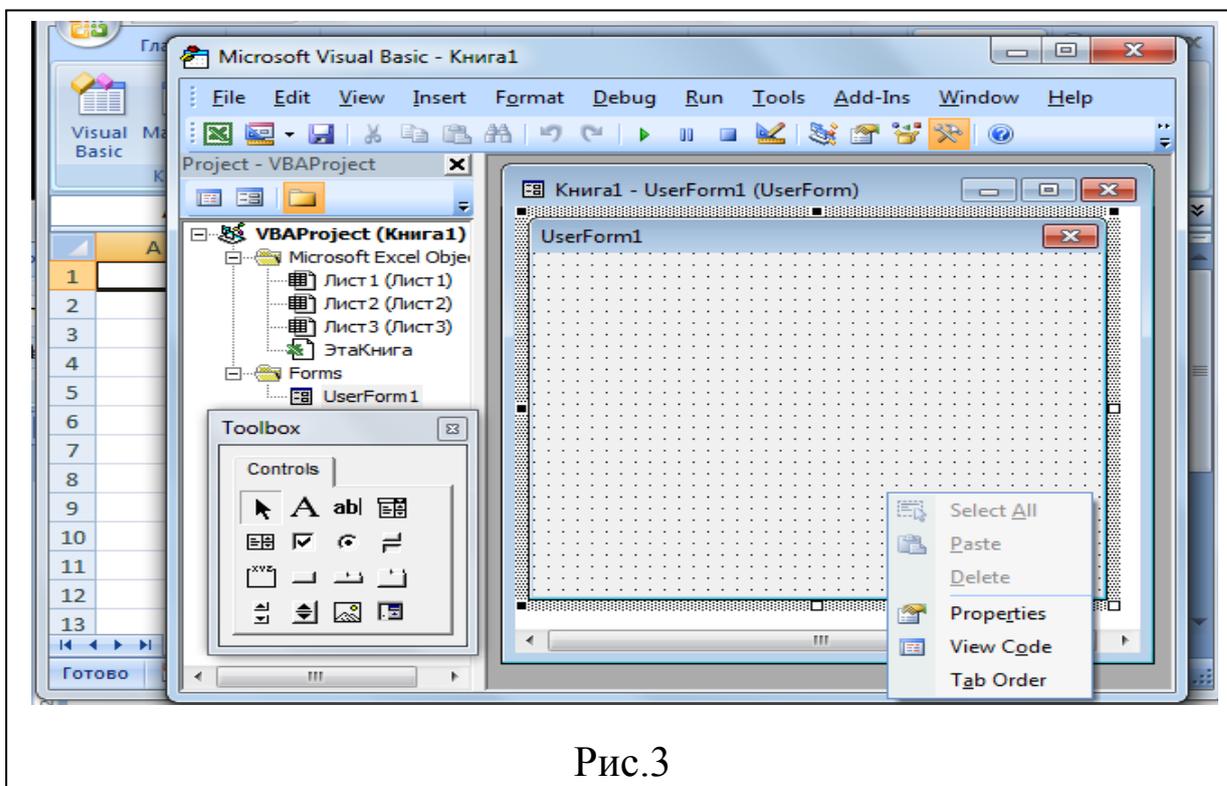


Рис.3

## 2.1. Элементы управления

На инструментальной панели Toolbox см. рис.3 расположены следующие элементы управления, которые являются объектами VBA : TextBox - поле, Label - надпись, CommandButton - кнопка, ListBox - список, ComboBox - поле со списком, ScrollBar - полоса прокрутки, SpinButton - счетчик, OptionButton - переключатель, CheckBox - флажок , ToggleButton - выключатель, Frame - рамка, Image - рисунок, RefEdit - RefEdit, MultiPage - набор страниц.

Любой из этих элементов управления, объектов можно разместить на форме. Например, можно поместить кнопку. Для этого мышью перетаскиваем на форму объект или элемент управления  под названием CommandButton, расположенный вторым слева в третьем ряду панели Toolbox. На форме появится кнопка CommandButton1 см. рис. 4.

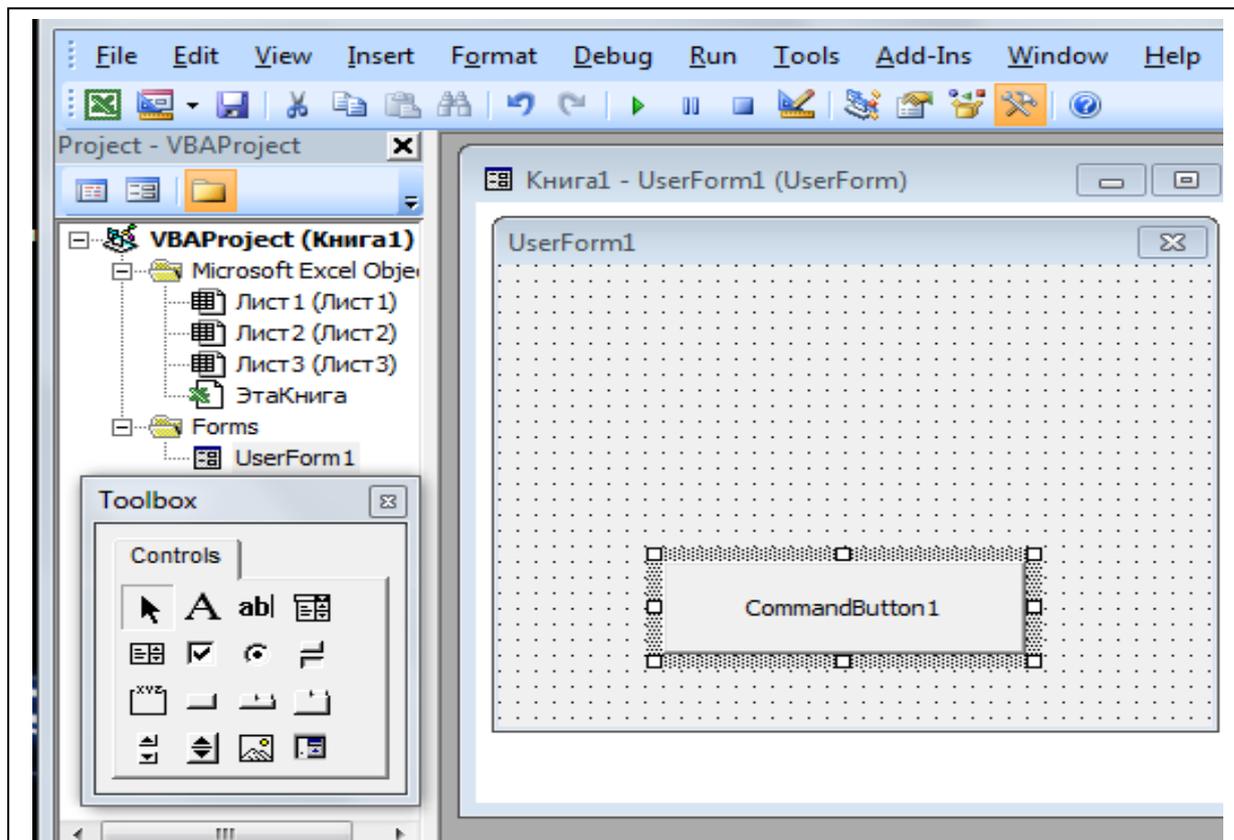


Рис. 4

## 2.2.Свойства элементов управления

Выбрав в контекстном меню см. Рис. 3 Properties- свойства формы, в открывшемся окне см. рис. 5, можно изменить заголовок элементов формы, в частности кнопки, её шрифт, цвет фона, добавить картинки, добавить и изменить многие другие свойства.

Наиболее часто используемые свойства объекта UserForm: Name - возвращает имя пользовательской формы, Caption - возвращает текст, отображаемый в строке заголовка формы, BackColor - возвращает цвет фона формы, BorderStyle - устанавливает тип границы, Picture - указывает рисунок, отображаемый как фон формы.

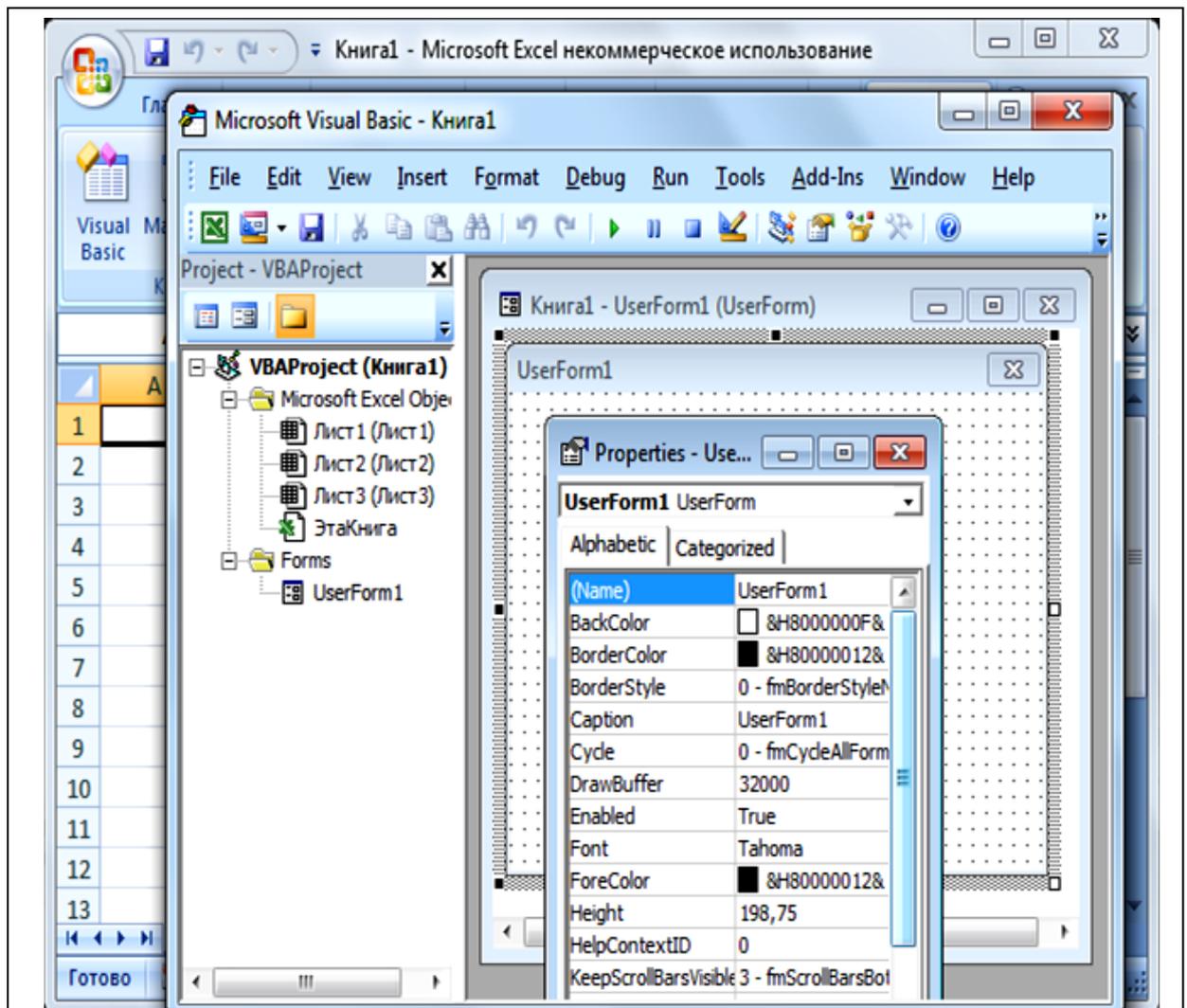


Рис.5

## 2.3. Связь пользовательской формы с процедурой

Осталось разобраться, как привязать к форме, а именно к созданной кнопке, программный код, процедуру. Выполняется это так: после двойного щелчка по кнопке откроется окно редактора VBA, где уже будет создана заготовка вашей процедуры обработки нажатия на кнопку см. рис. 6.

Теперь в тело этой процедуры можно вводить команды (операторы), которые будут выполняться, если событие в заголовке - Click мыши - произойдет. Это событие считается стандартным для кнопок, флажков, рисунков, рамок, переключателей и окон отображения текста. Для остальных элементов - полей ввода текста, полос прокрутки, счетчиков и др. - стандартным считается событие Change, то есть их изменение.

Список возможных событий для элемента можно получить, поставив курсор внутрь процедуры и выбрав, нужное из верхнего правого выпадающего меню. В верхнем левом выпадающем меню указаны все процедуры и функции программы реакции формы на события см. рис. 6.

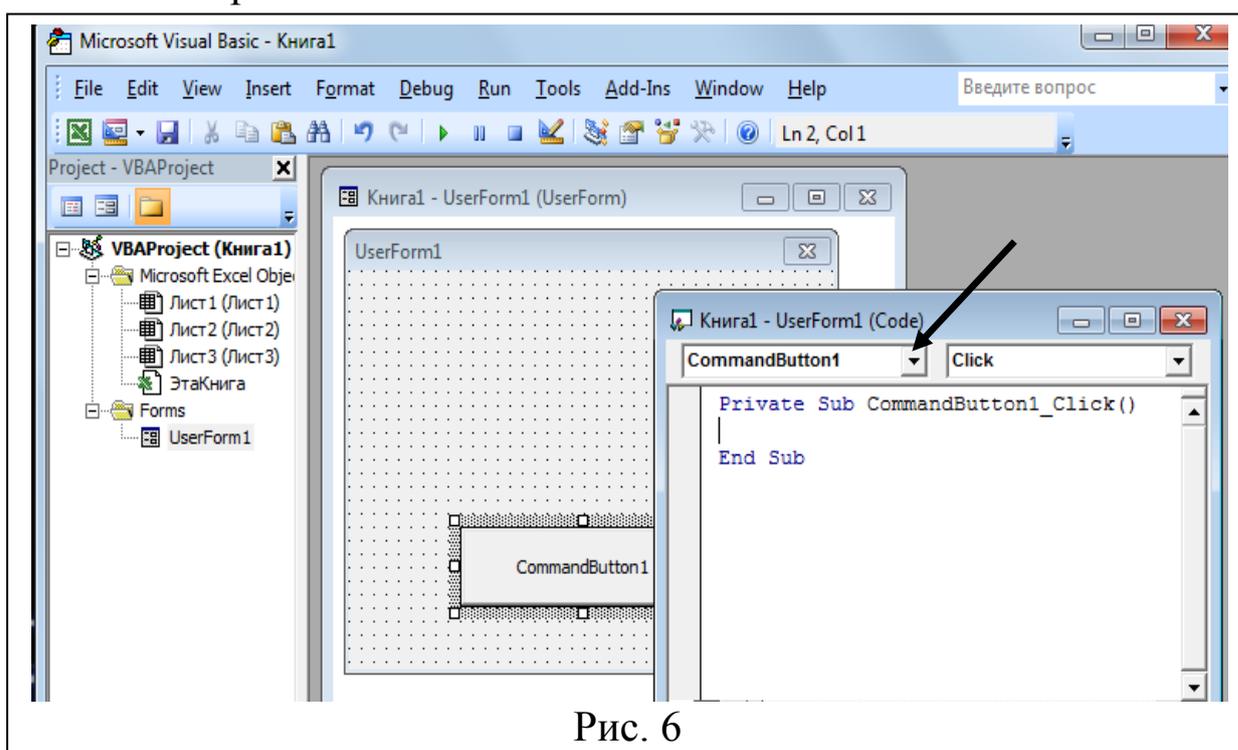


Рис. 6

### 3. Примеры создания пользовательских форм и написания процедур к ним

Разберём процесс создания пользовательских форм и написания процедур к ним на конкретных примерах.

#### 3.1. Пример 1. VBA программа «Дата и время»

**Задание.** Создать пользовательскую форму и написать процедуры к ней в которых реализуется ввод имени пользователя, вывод настоящей даты и времени и кнопка для завершения работы процедуры. VBA программу в целом назовём «Дата и время».

##### Создание пользовательской формы

1. Открыть или создать файл для работы.
2. Перейти в редактор Visual Basic (Сервис > Макрос > редактор Visual Basic).
3. Открыть пользовательскую форму (Insert> UserForm) см. рис.7.

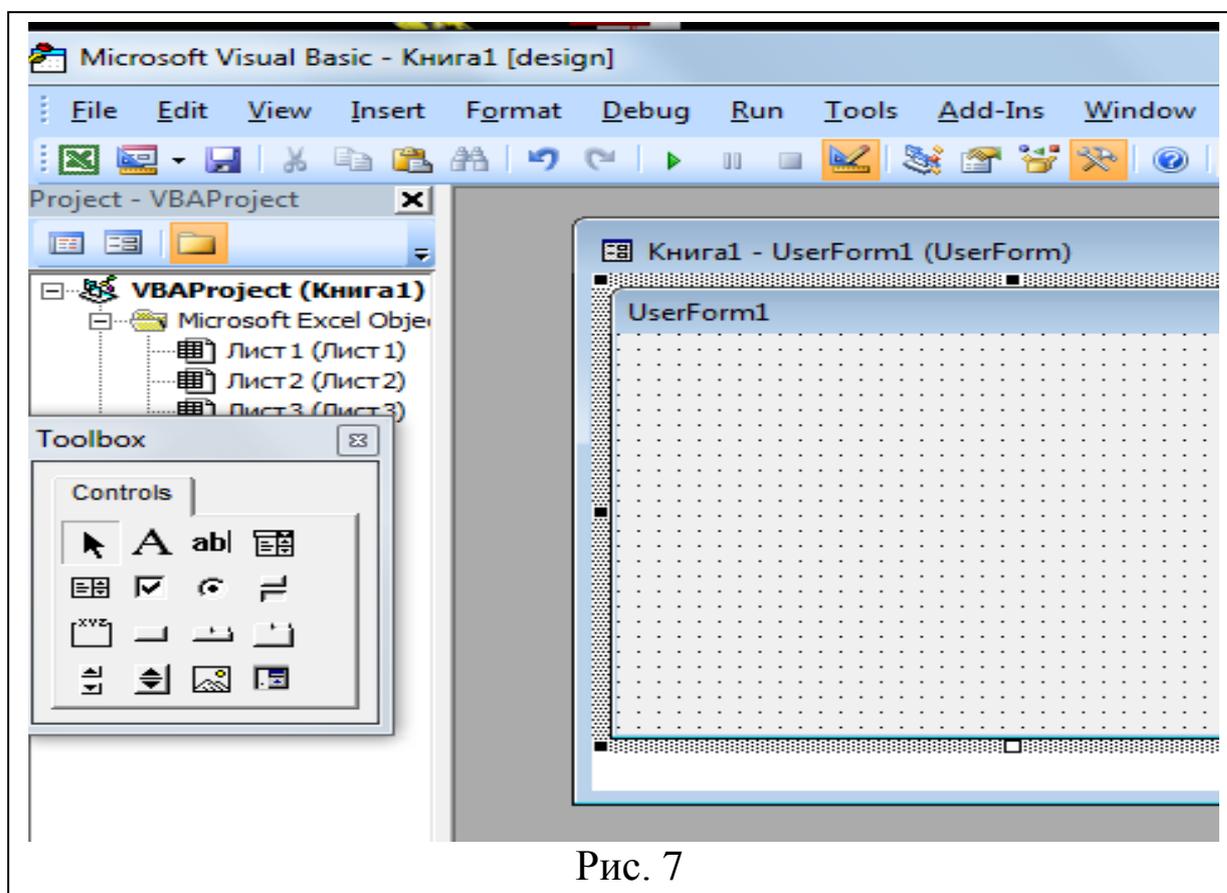


Рис. 7

4. На панели элементов управления выбрать пиктограмму  Label

– надпись см. рис.8.

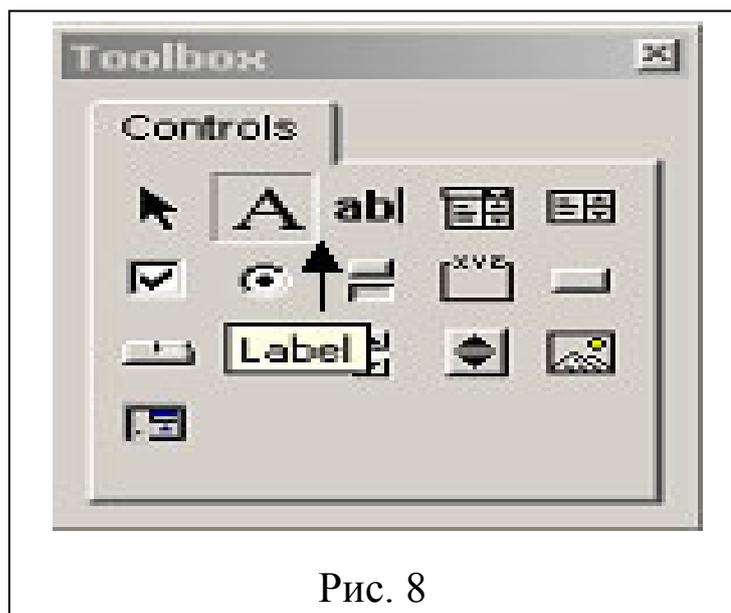


Рис. 8

5. Мышью перетащить пиктограмму Label на пользовательскую форму см. рис. 9.

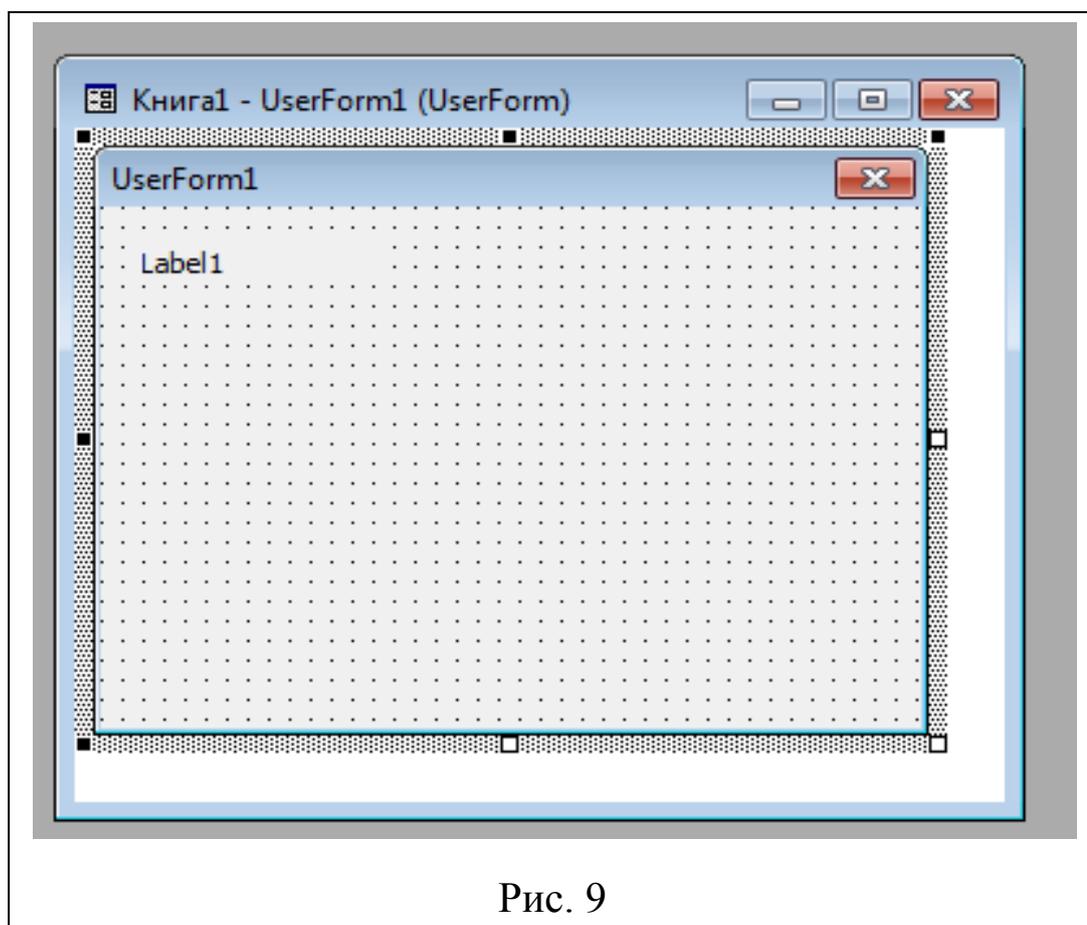


Рис. 9

6. Щёлкнуть мышью на поле Label. Стереть надпись Label и набрать «**Введите своё имя ->**» см. рис.10.

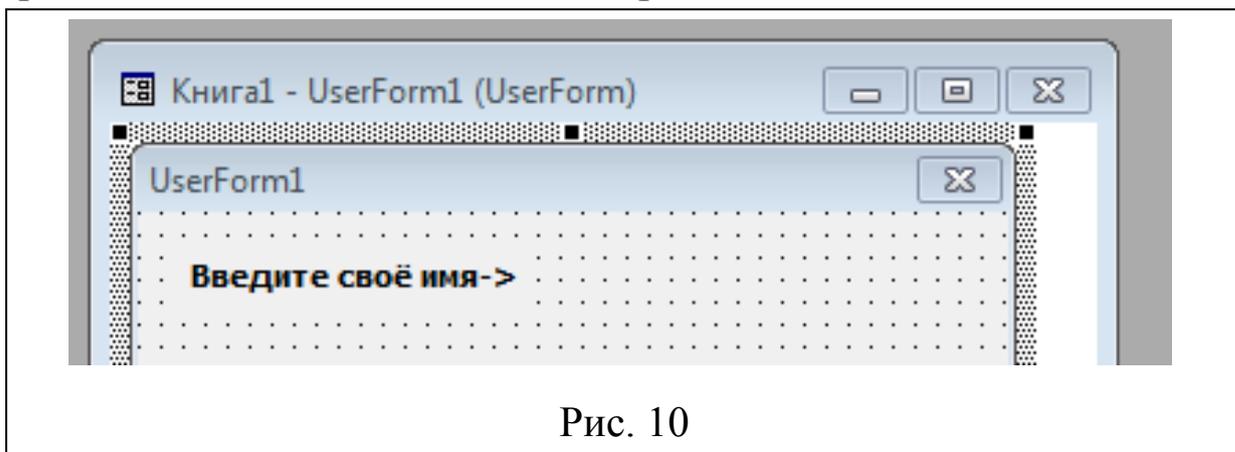


Рис. 10

7. Изменить UserForm1- название формы. Для этого использовать в контекстном меню Properties свойство Caption, которое возвращает указанный в нём текст, а именно «Хотите узнать дату и время?», в заголовок формы см. рис.11. Поле заголовка предварительно должно быть активизировано щелчком мыши.

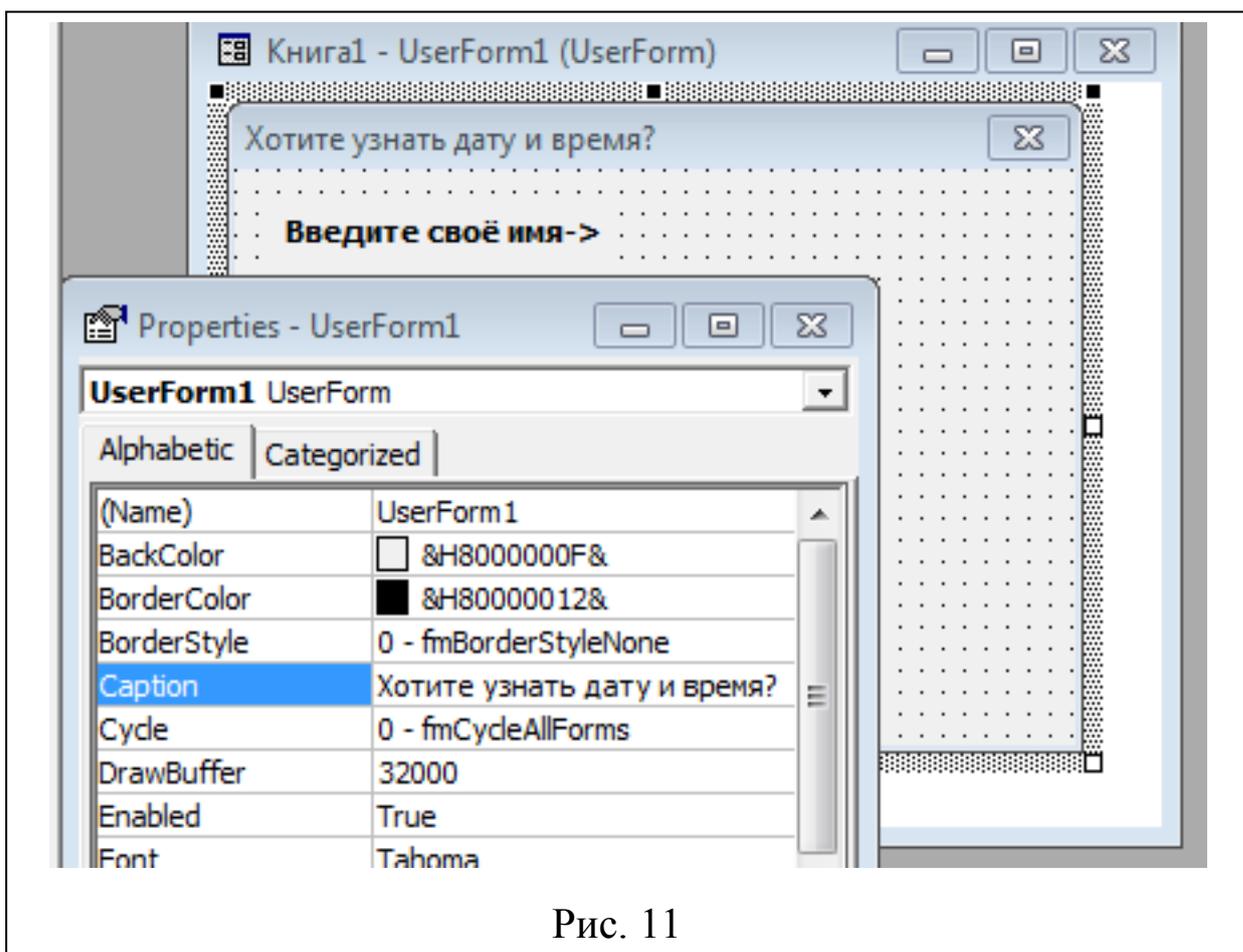


Рис. 11

8. На панели элементов управления выбрать пиктограмму  - TextBox- поле для ввода имени пользователя при работе процедуры. На форме появиться окно для ввода, которое надо расположить в нужном месте и придать нужную форму см. рис. 12.

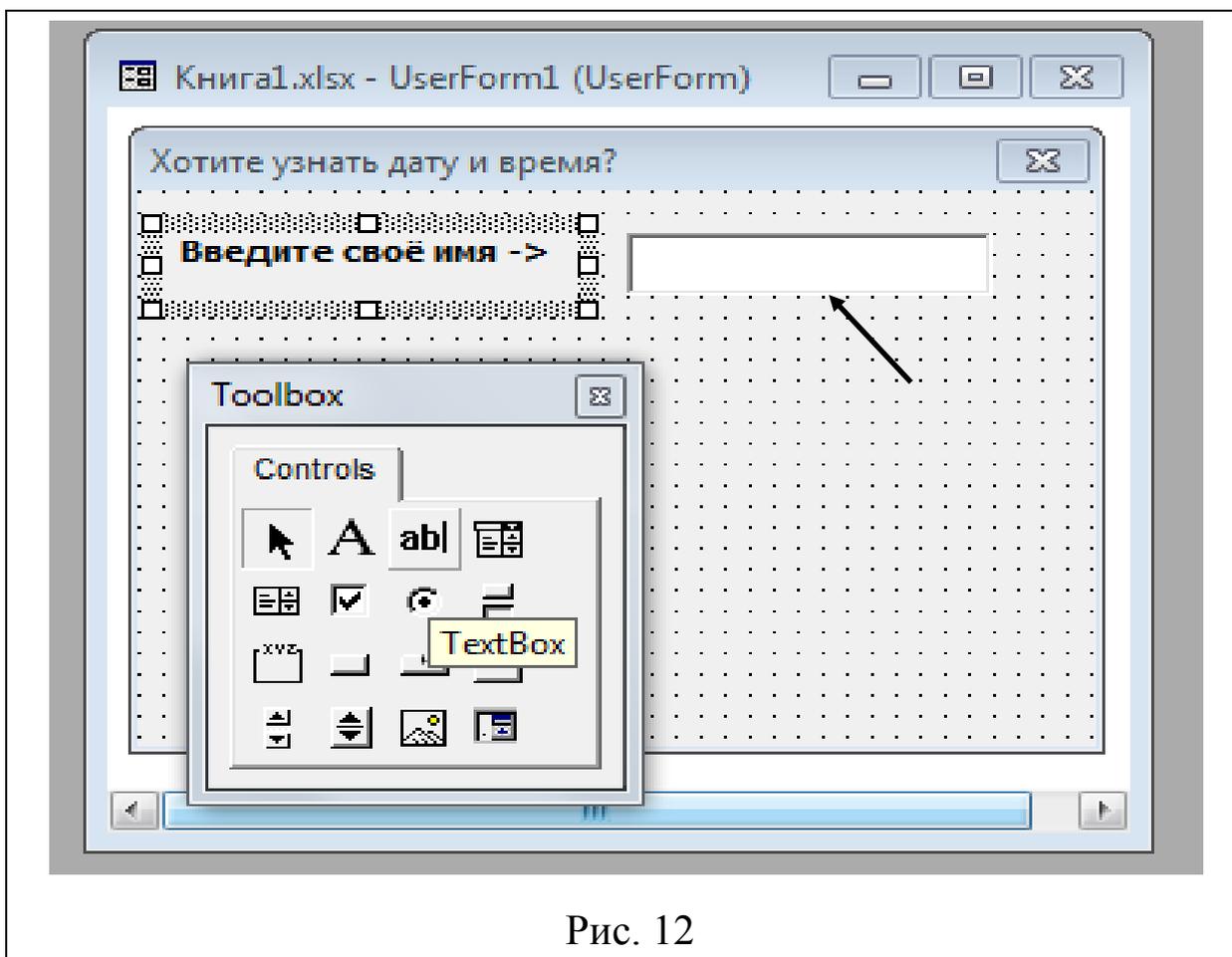


Рис. 12

9. Расположить на форме кнопку для ввода. Для этого на панели элементов выбрать пиктограмму  CommandButton – кнопка, и перетащить её мышью в выбранное место на форме. Придать кнопке желаемую форму см. рис. 13.

10. На полученной кнопке ввести надпись «Для получения ответа нажмите здесь». Для этого щелчком мыши активизируем кнопку и осуществляем ввод указанного текста см. рис. 14.

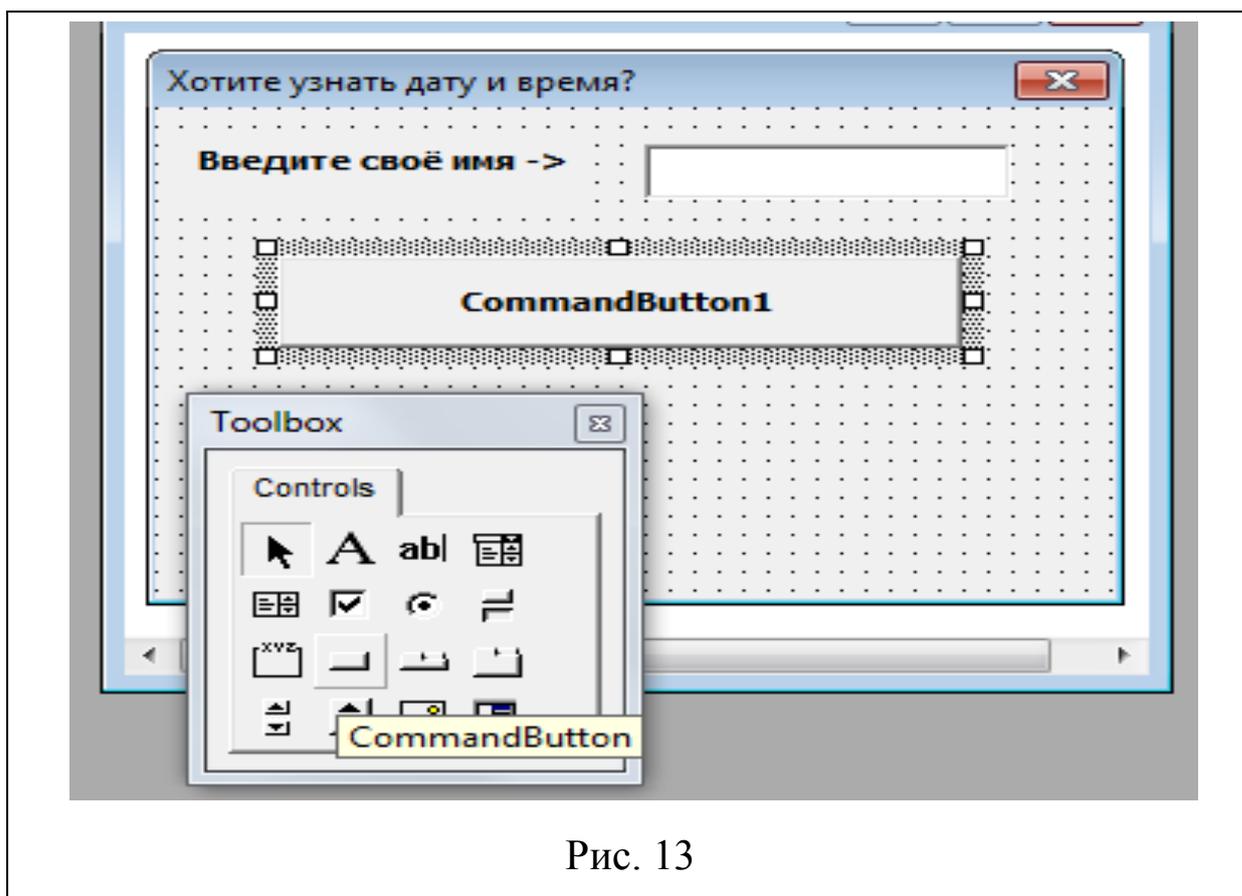


Рис. 13

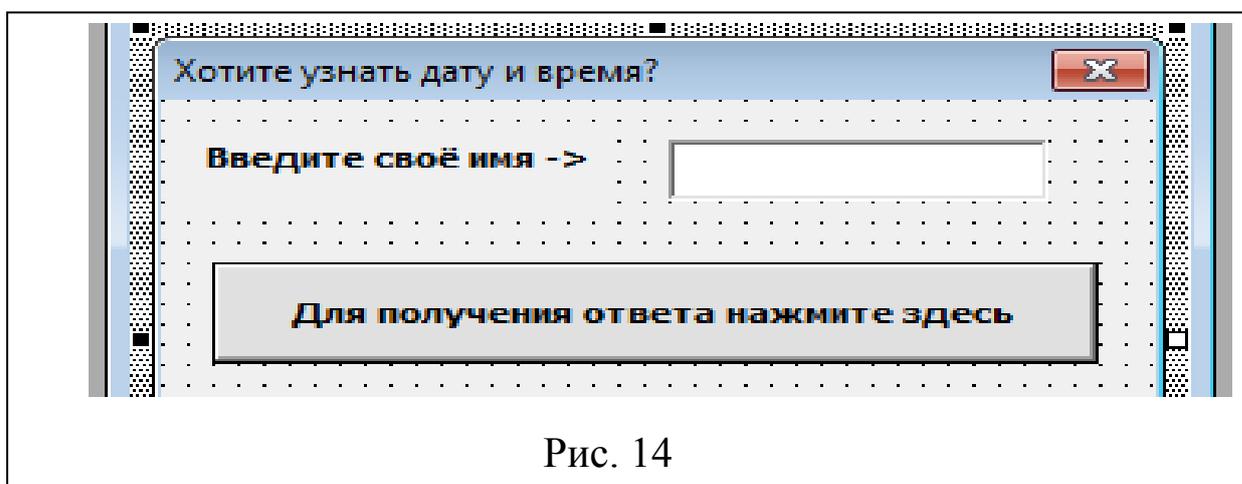


Рис. 14

11. Теперь надо ввести поле для вывода ответа, который будет сформирован в процедуре. Для этого используем уже известную пиктограмму **A** - Label, расположенную на панели элементов управления. Перетаскиваем мышью пиктограмму на форму, располагаем её в нужном нам месте и придаем ей нужную форму см. рис. 15



12. Используя контекстное меню Properties можно придать, кнопке например, желаемый цвет или оттенок с помощью свойства BackColor. На рис. 15 кнопке придан легкий оттенок, а полю для вывода придали белый цвет.

13. Для завершения работы процедуры расположим на форме, аналогично тому, как это было сделано выше, ещё одну кнопку, но с надписью «Выход» см. рис. 16.



## Процедуры для созданной пользовательской формы

Далее рассмотрим, как написать и привязать к созданной пользовательской форме процедуру или процедуры для решения поставленной задачи. Двойной щелчок по форме позволяет открыть окно редактора VBA, где будут предложены заготовки для написания процедур см. рис. 17.

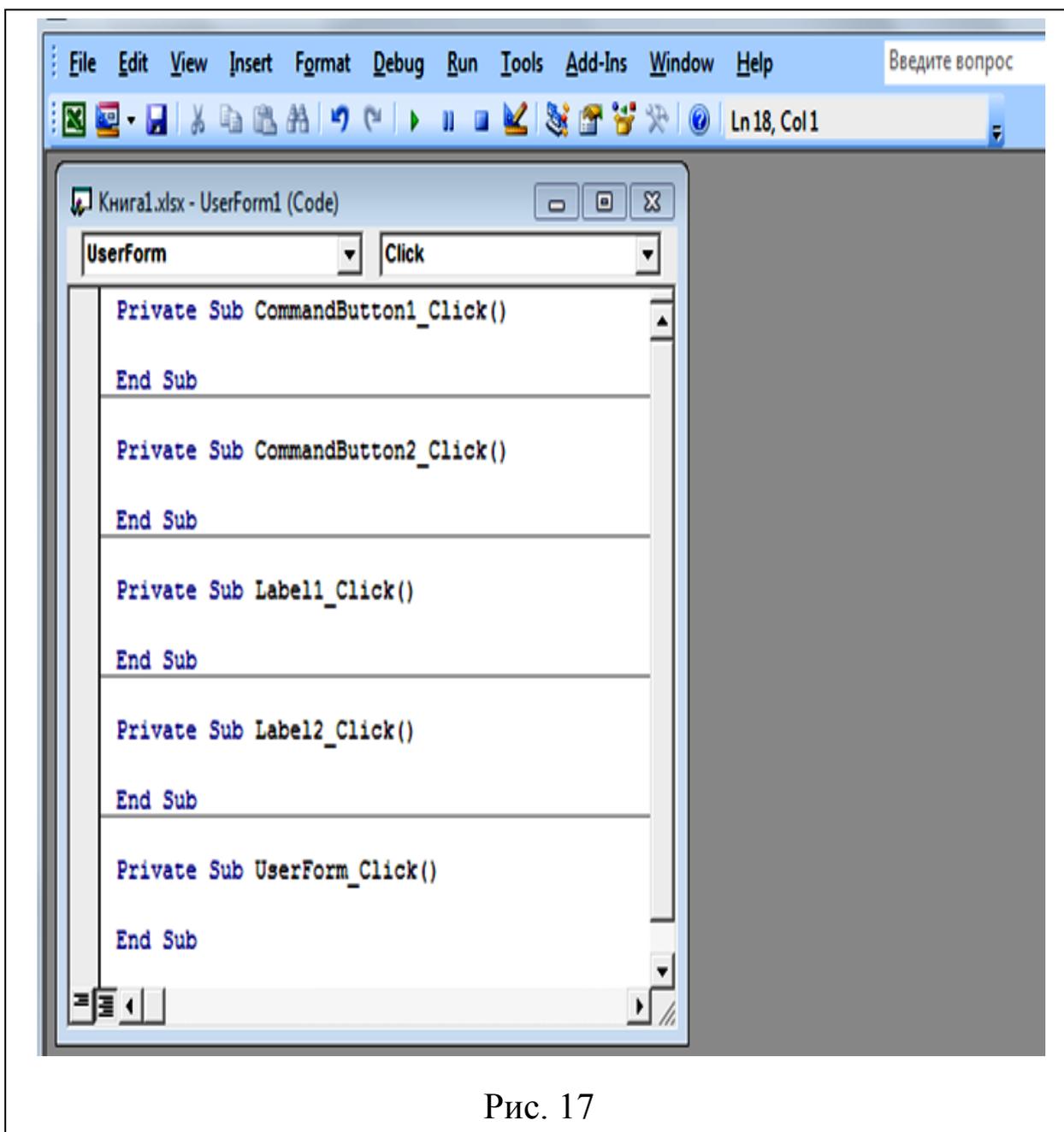


Рис. 17

Теперь в процедуры можно вводить операторы, которые будут выполняться, если событие в заголовке - Click мыши - произойдет.

Используем заготовку для процедуры `CommandButton1_Click()`, задаем в процедуре переменную под названием **Имя**, описываем её как строковую переменную (`Dim Имя As String`). Переменной **Имя** присваиваем текст, который будет вводиться в поле элемента управления `TextBox1`, указывая при этом параметр `.Text` или `.Value`. В процедуре это выглядит так: `Имя = TextBox1.Text` см. рис.18.

Далее в поле элемента управления `Label2` вводится текст, с приветствием по имени пользователя процедуры и указанием настоящей даты и времени. В процедуре это задаётся следующим образом:

```
Label2.Caption = "                ЗДРАВСТВУЙТЕ," & Имя & "!" & _  
vbLf & vbLf & " СЕГОДНЯ " & Format(Now, "dddddd,hh ч. mm мин.")
```

см. рис. 18. Все действия процедуры `CommandButton1_Click()` будут выполняться в случае нажатия кнопки `CommandButton1`, которую мы ввели на созданной форме см. рис.13.

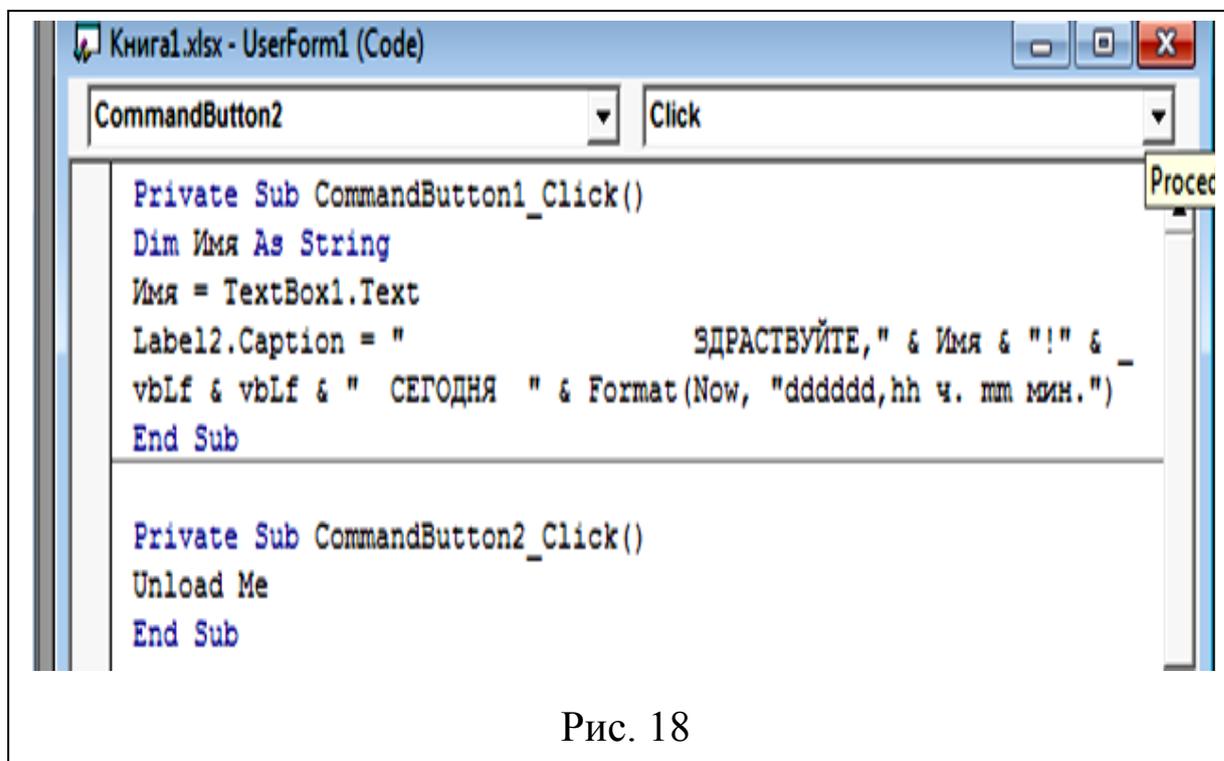


Рис. 18

В процедуре `CommandButton2_Click()` используется оператор **Unload Me** который закрывает форму и процедуры при нажатии кнопки «Выход», которую мы ввели на созданной форме см. рис. 16.

### **Работа пользовательской формы и её процедур**

Запускаем процедуру. Появится созданная пользовательская форма. В окне для ввода имени зададим имя, например « Александр », и нажмём клавишу для вывода результата работы процедуры. Получим следующий результат см. рис.19.

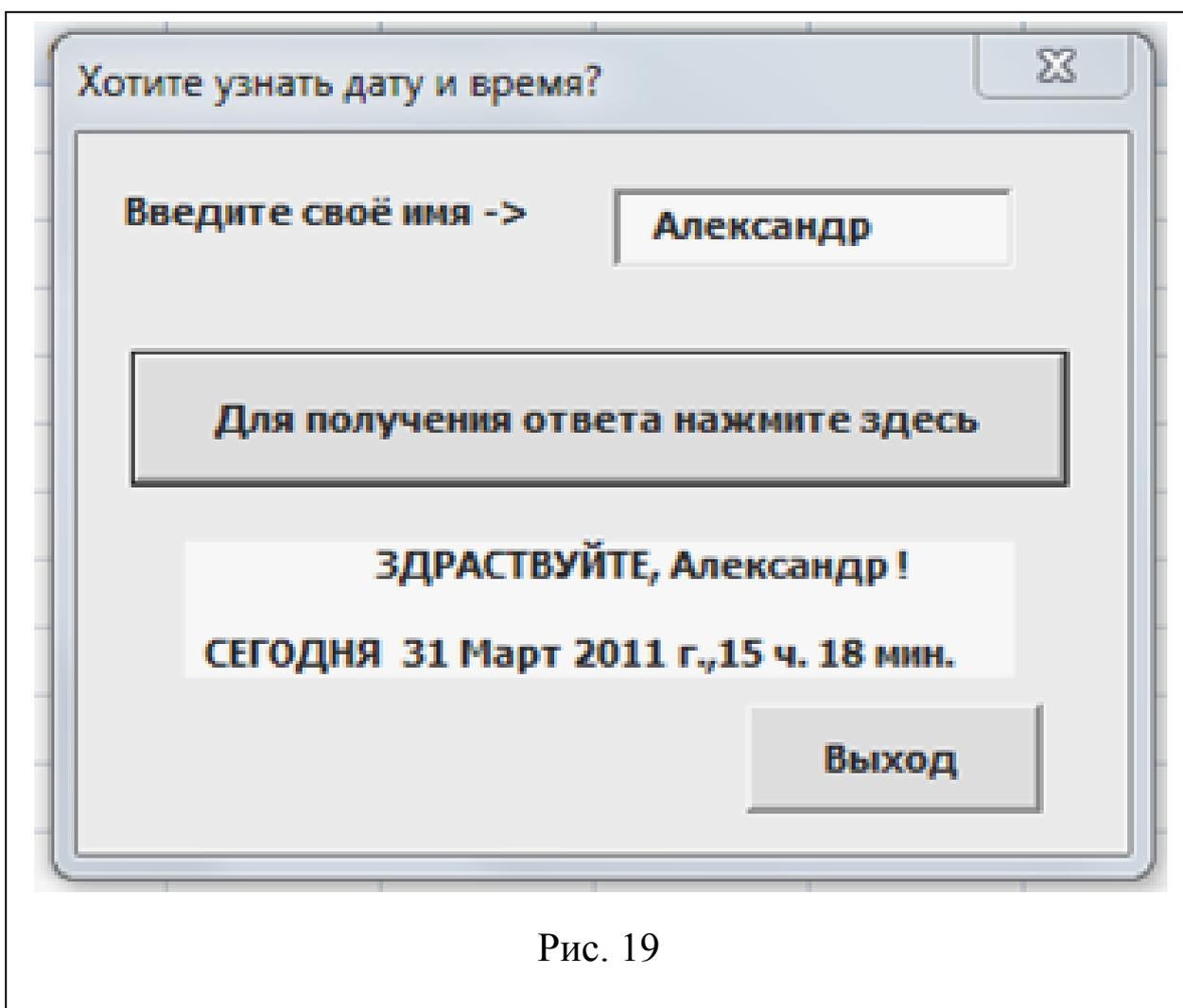


Рис. 19

## 3.2. Пример 2. VBA программа «Автомобиль года»

В данном примере разберём, каким образом используя процедуру заполнять определёнными записями поле со списком ComboBox. Каким образом выбрать из списка нужную запись, наименование и передать его в другую предварительно созданную форму. Как организовать переход из одной формы в другую и как изменить шрифт и цвет текста. Разберём в данном примере как организовать запуск программы с помощью кнопки созданной на рабочем листе Excel.

**Задание.** Создать программу, в которой реализуются пользовательские формы и процедуры к ним. Программу назвать «Автомобиль года».

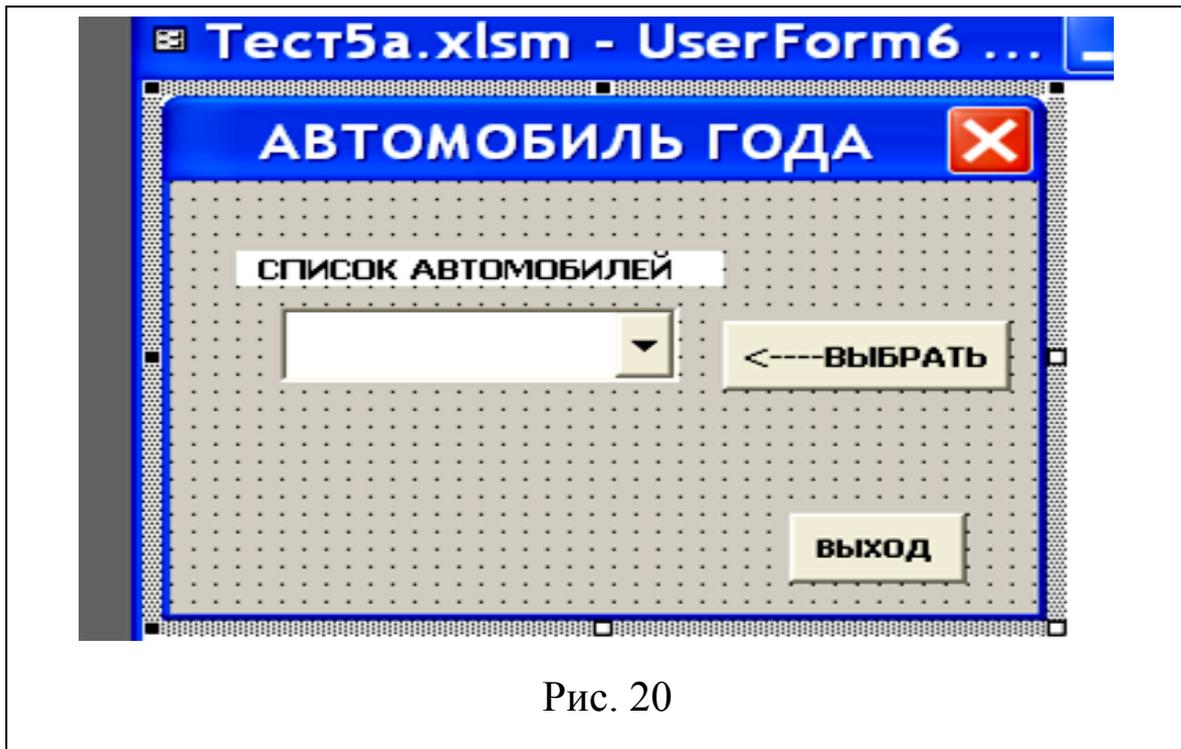
В программу включить две пользовательские формы. В первой форме с помощью элемента управления Label организовать надпись «Список автомобилей», а под ней посредством элемента управления ComboBox организовать поле со списком. В поле со списком через процедуру ввести записи в виде наименований десяти любых марок автомобилей. Кроме того, на первой форме разместить две кнопки. Одна кнопка с надписью «← ВЫБРАТЬ », имеется в виду выбор названия марки автомобиля, среди тех, которые введены в поле со списком ComboBox. Вторая кнопка «ВЫХОД».

В процедурах реализовать событие, которое должно произойти при нажатии кнопки «← ВЫБРАТЬ». Событие состоит в выборе и передаче выбранного названия автомобиля во вторую пользовательскую форму, а именно в организованное на ней поле элемента управления TextBox. Название выбранного автомобиля вывести с размером шрифта 20 и красным цветом. В поле, отведённом для названия формы ввести текст « Лучший автомобиль года»

## Создание первой пользовательской формы

Создание для данного примера его первой пользовательской формы с именем UserForm6 производится аналогично тому, как это делалось в разобранный выше примере 1.

Первая пользовательская форма будет иметь следующий вид см. рис. 20.

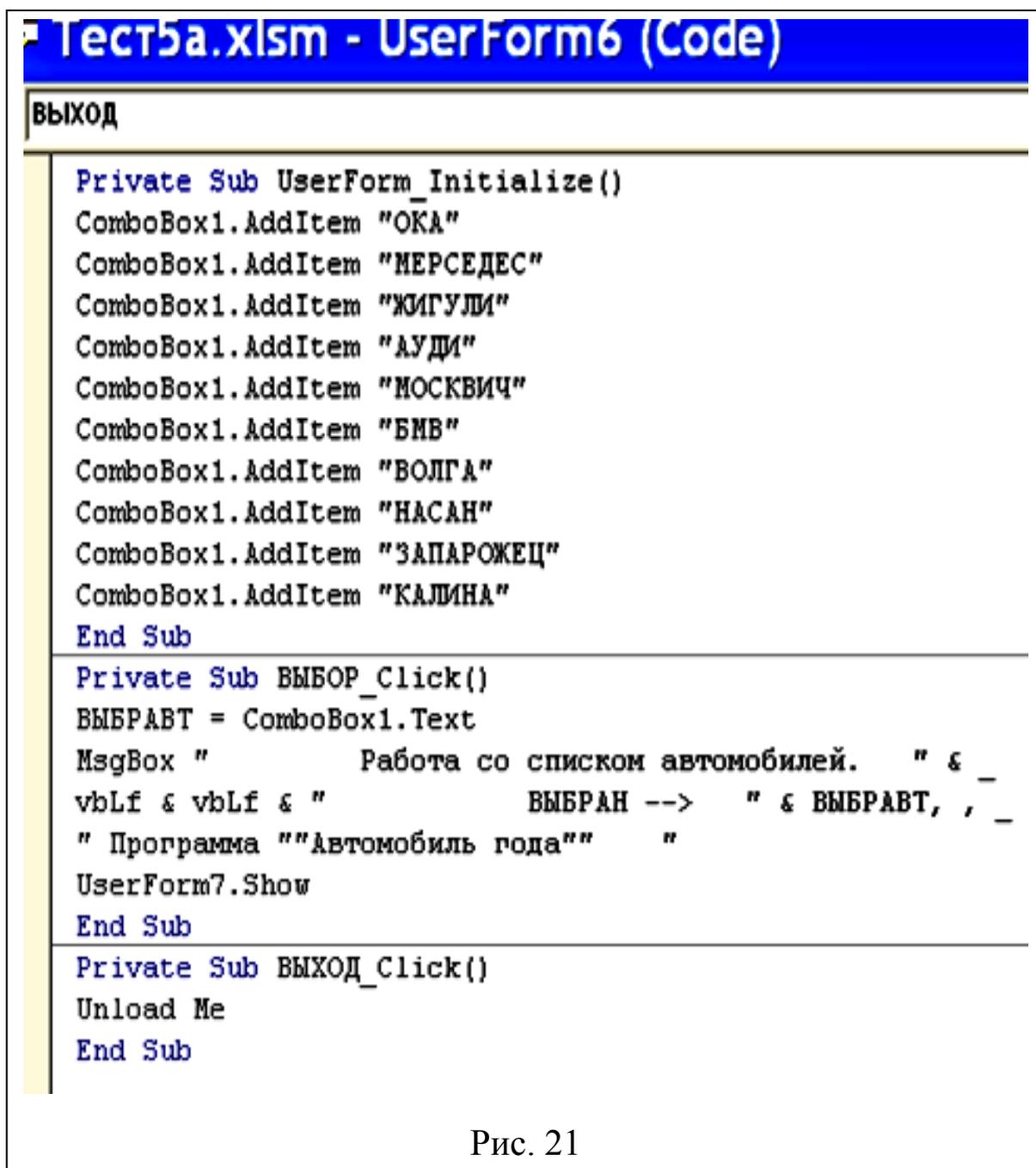


## Процедуры для первой пользовательской формы

Процедуры к первой созданной форме будут иметь вид, представленный на рис. 21. В первой процедуре используется метод AddItem, который служит для добавления в список записей или заполнения списка, обычно он помещается в обработчик события Initialize.

Метод Show служит для перехода, просмотра указанной формы. Например, UserForm7.Show означает переход и просмотр формы с именем UserForm7. В нашем случае это имя второй пользовательской формы рассматриваемого примера, которую нам предстоит создать см. рис. 22.

Следует обратить внимание на то, что предлагаемые системой стандартные имена процедур и элементов управления можно менять на другие. Это позволяет сделать процедуры более удобными для чтения и понимания, особенно когда их большое количество. Так в нашем случае предлагаемые имена процедур CommandBatton1\_Click() и CommandBatton2\_Click() заменены соответственно на имена ВЫБОР\_Click() и ВЫХОД\_Click().



```
Тест5а.xlsm - UserForm6 (Code)
ВЫХОД

Private Sub UserForm_Initialize()
ComboBox1.AddItem "ОКА"
ComboBox1.AddItem "МЕРСЕДЕС"
ComboBox1.AddItem "ЖИГУЛИ"
ComboBox1.AddItem "АУДИ"
ComboBox1.AddItem "МОСКВИЧ"
ComboBox1.AddItem "БМВ"
ComboBox1.AddItem "ВОЛГА"
ComboBox1.AddItem "НАСАН"
ComboBox1.AddItem "ЗАПАРОЖЕЦ"
ComboBox1.AddItem "КАЛИНА"
End Sub

Private Sub ВЫБОР_Click()
ВМБРАВТ = ComboBox1.Text
MsgBox "          Работа со списком автомобилей.  " & _
vbLf & vbLf & "          ВМБРАН -->  " & ВМБРАВТ, , _
" Программа ""Автомобиль года""  "
UserForm7.Show
End Sub

Private Sub ВЫХОД_Click()
Unload Me
End Sub
```

Рис. 21

## Создание второй пользовательской формы

Создание для данного примера второй пользовательской формы с именем UserForm7 производится аналогично тому, как это делалось в разобранный выше примере 1.

Вторая пользовательская форма будет иметь следующий вид см. рис. 22.

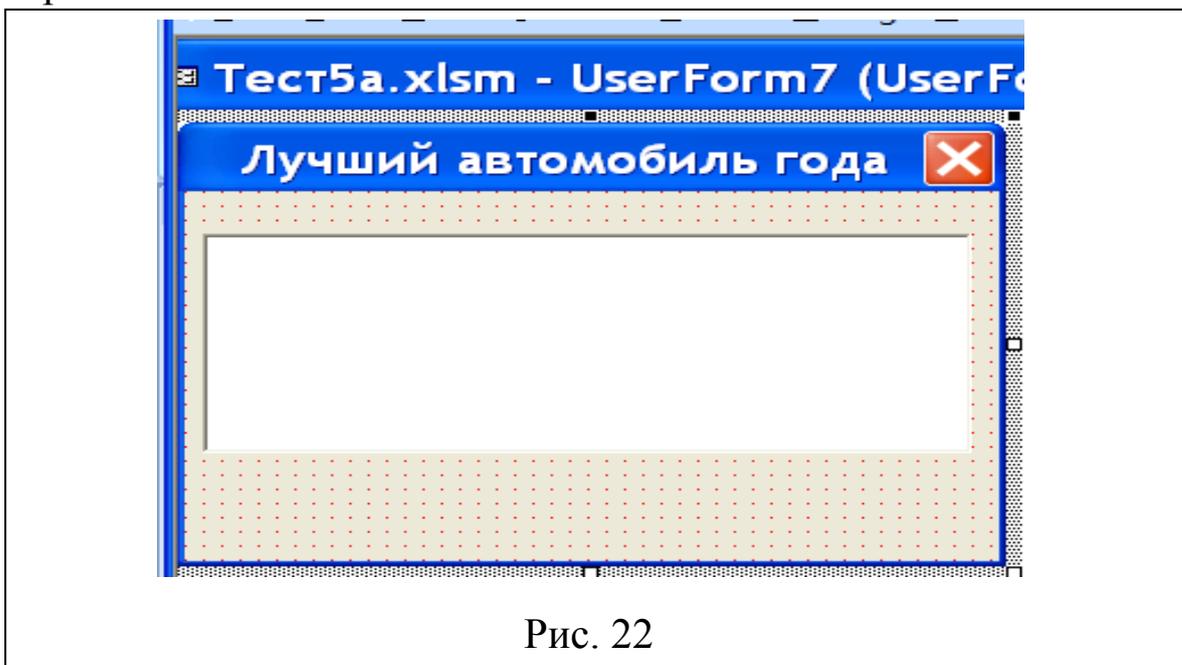


Рис. 22

## Процедура для второй пользовательской формы

В данной процедуре название элемента управления TextBox1 заменено на название ЛучАвто. В поле этого элемента управления передаётся запись в виде названия выбранного автомобиля в первой форме UserForm6 см. рис.23.

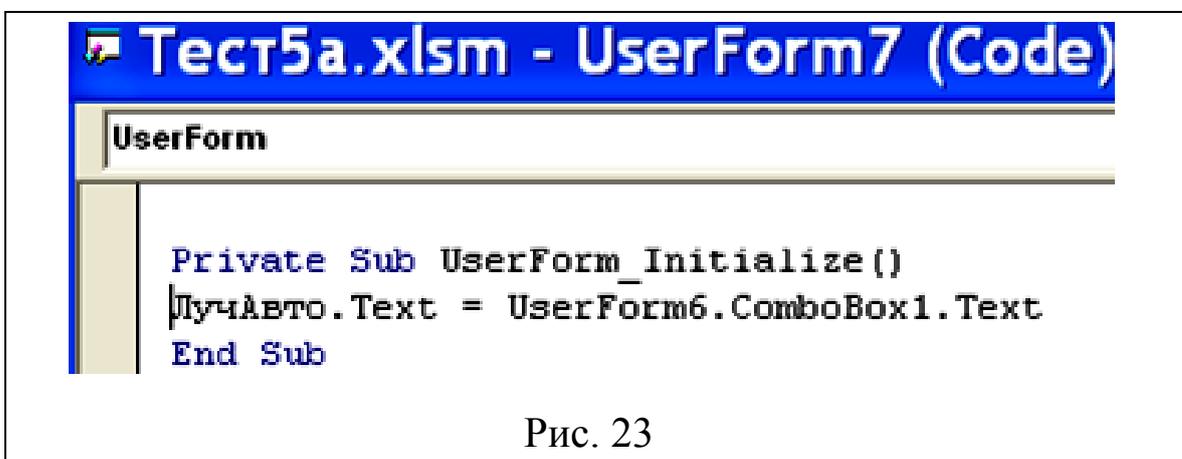


Рис. 23

## Работа пользовательских форм и их процедур

Созданная программа состоит из двух пользовательских форм и их процедур. Запуск программы можно произвести как в предыдущем примере или с рабочего листа Excel.

### Запуск программы с рабочего листа Excel

Для удобства можно организовать запуск программы с рабочего листа Excel. Для этого надо на рабочем листе Excel создать кнопку, при нажатии которой происходило бы событие, связанное с запуском процедур первой пользовательской формы UserForm6. Процесс создание такой кнопки аналогичен тому, как это делается для пользовательской формы. На ленте вкладок Excel выбрать вкладку «**Разработчик**», затем «**Вставить**». В открывшемся окне «**Элементы управления формы**» выбрать элемент управления «**Кнопка**» см. рис. 24.

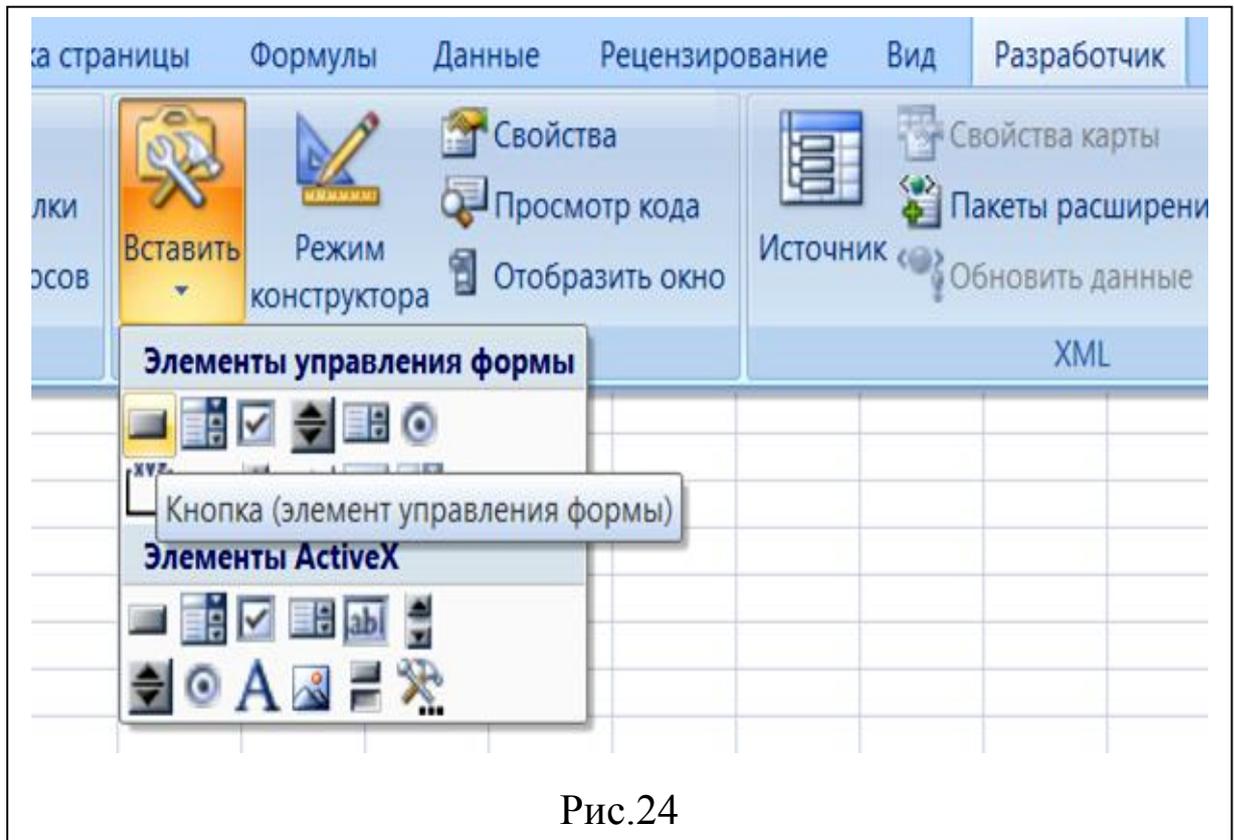


Рис.24

После нажатия на элемент управления «**Кнопка**», на рабочем листе появится маркер «+». Далее нажав правую кнопку мыши и смещая её установить нужные размеры кнопки при этом откроется окно с названием «**Назначить макрос объекту**» см. рис 25.

В поле «**Имя макроса**» можно ввести его новое имя или оставить предложенное «**Кнопка6\_Щелчок**» см. рис. 25. Введём новое имя «**Кнопка1\_Щелчок**». Нажать кнопку «**Создать**». Откроется окно для ввода кода см. рис 26. В это окно введём код вызова пользовательской формы UserForm6, а именно **UserForm6.Show**.

На кнопке с названием « **Кнопка 6** » см. рис. 25 введём новое название, согласно условию задания, а именно «**Запуск программы «Автомобиль года »** » см. рис. 27

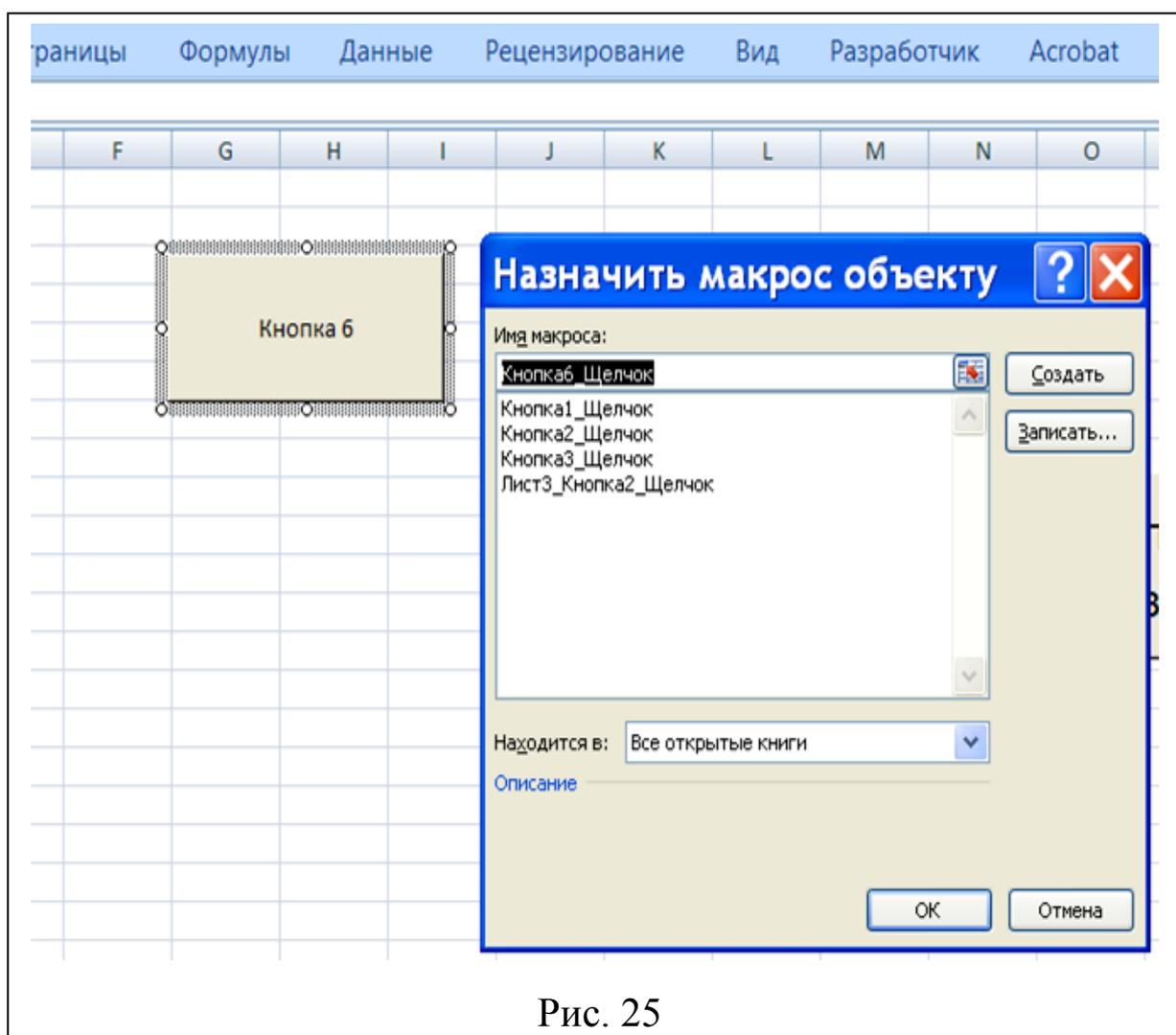


Рис. 25

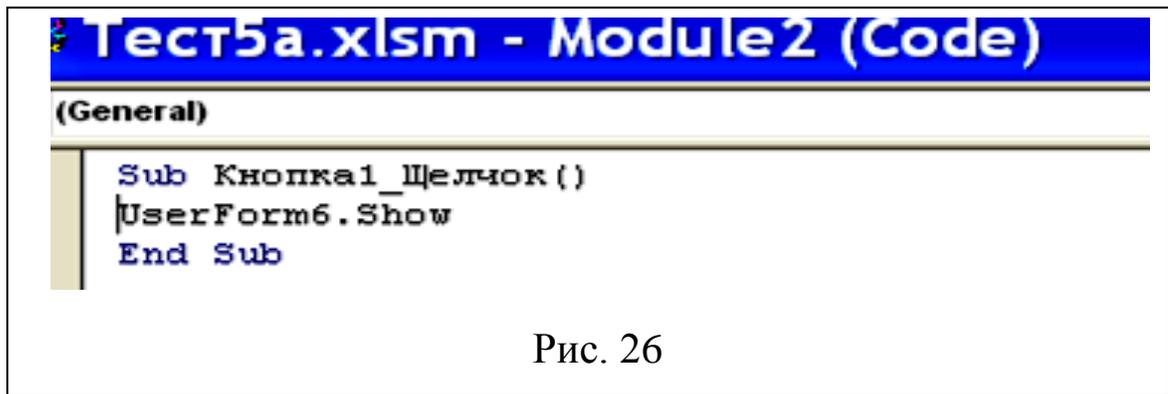


Рис. 26

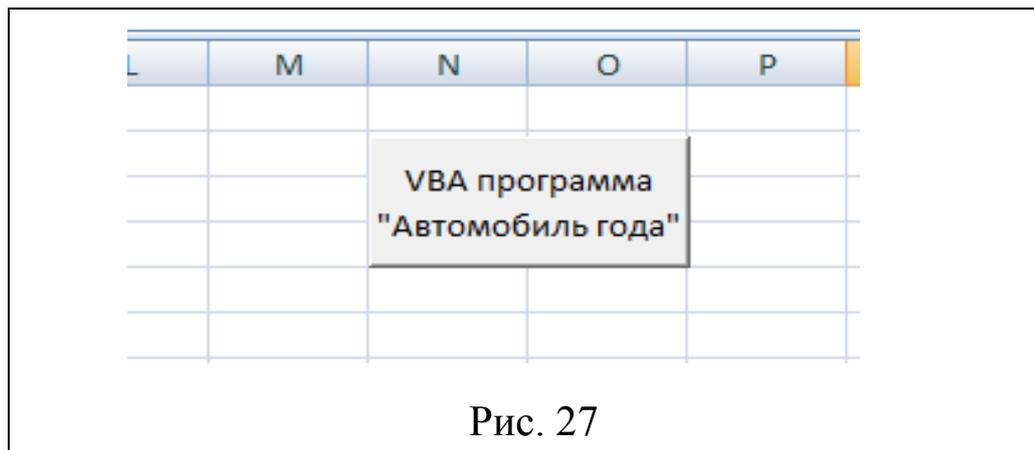


Рис. 27

Теперь чтобы запустить программу надо нажать на кнопку « **VBA программа «Автомобиль года»** » см. рис. 27. На рабочем листе появится первая созданная форма см. рис.28.

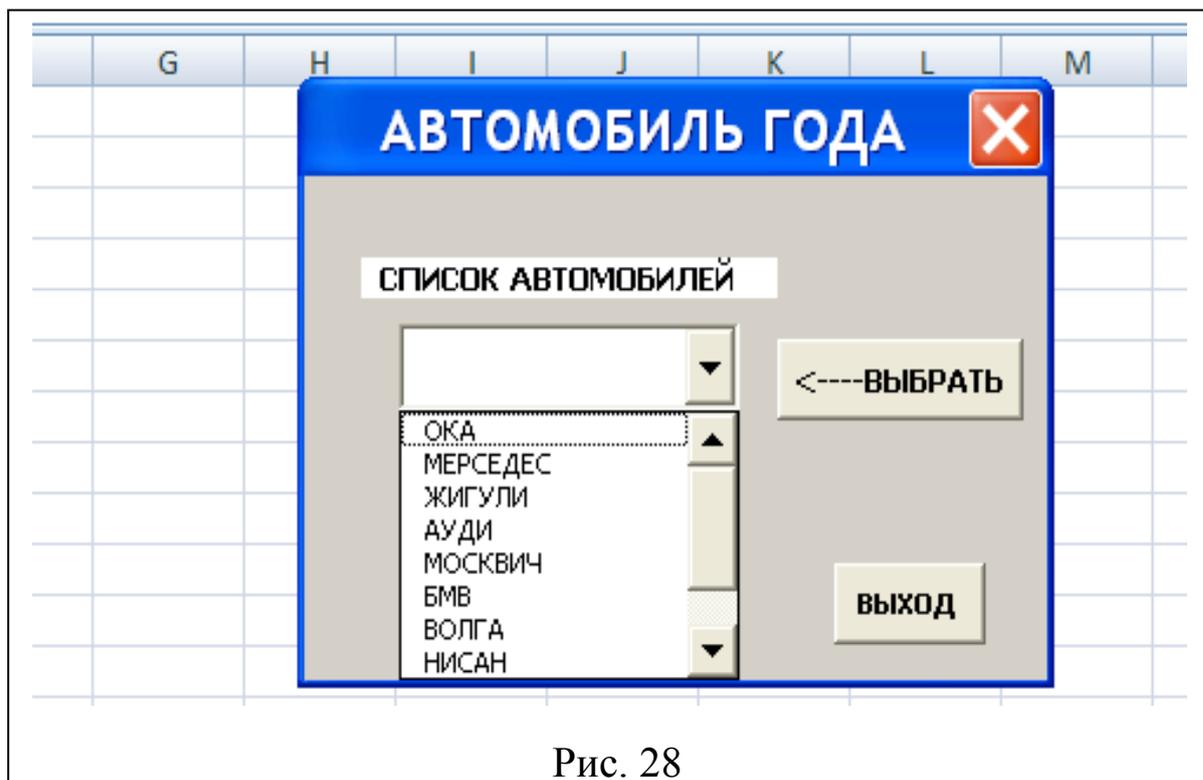


Рис. 28

Из приведённого списка курсором выбираем нужную марку автомобиля, например «ВОЛГА» и нажимаем левую кнопку мыши. Пользовательская форма примет следующий вид, см. рис. 29.

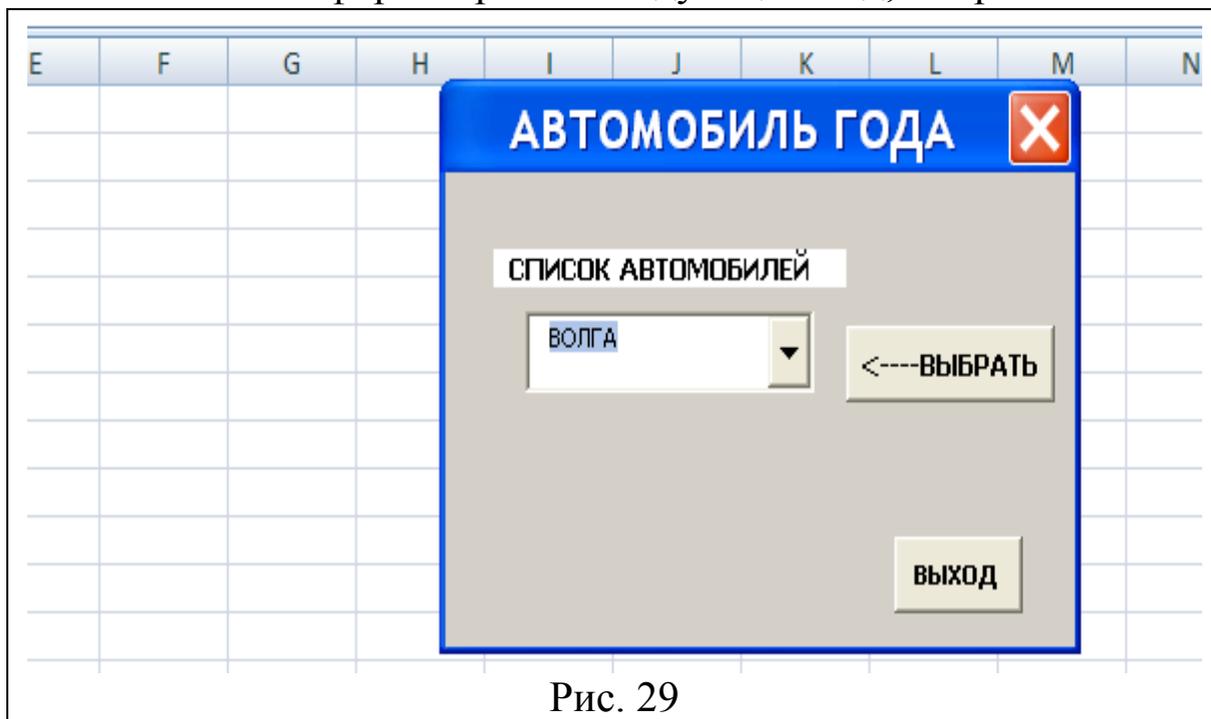


Рис. 29

Далее «нажать кнопку» «← ВЫБРАТЬ», для этого подвести к ней курсор и нажать левую кнопку мыши. Происходит запуск в работу второй пользовательской формы. Вторая пользовательская форма будет выглядеть следующим образом, см. рис. 30.

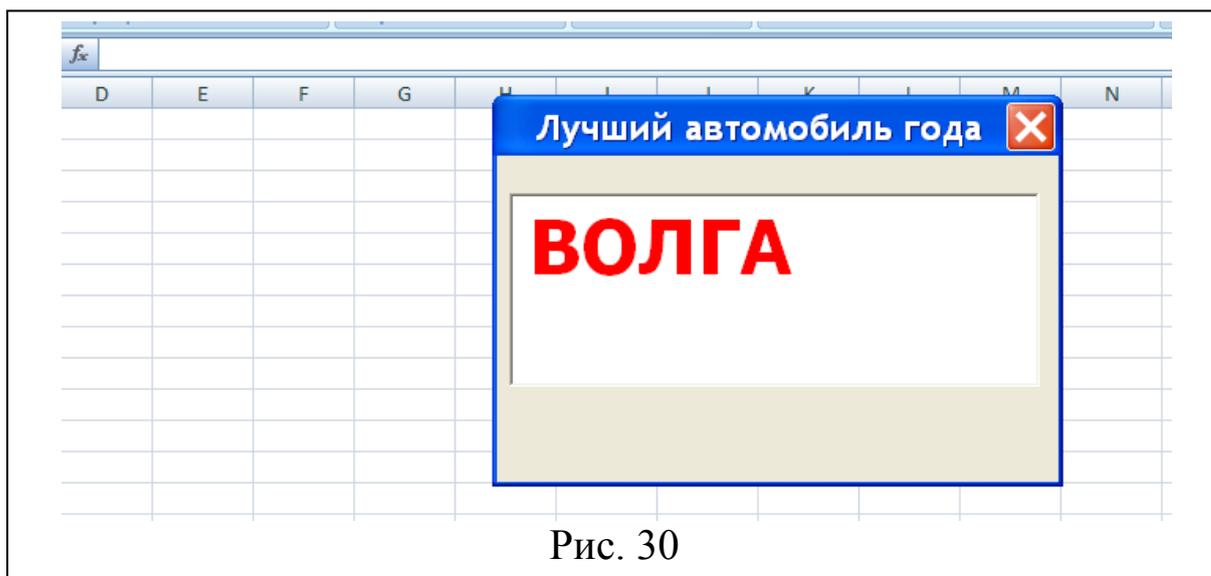


Рис. 30

Таким образом, все условия, сформулированные в задании примера 2, выполнены.

### 3.3. Пример 3. VBA программа «Остаток дней»

**Задание.** Создать пользовательскую форму и процедуры к ней в которых реализуется ввод даты, до которой требуется узнать количество оставшихся дней, например до Вашего дня рождения. Организовать вычисление и вывод количества оставшихся дней до указанной даты, создать кнопки ввода данных, повторения и завершения работы программы. VBA программу назовём «Остаток дней». Предусмотреть запуск программы с листа Excel.

#### Создание пользовательской формы и процедур

Порядок создания пользовательской формы аналогичен примерам 1 и 2. Пользовательская форма для примера 3 может быть выполнена, например, в следующем виде, см. рис. 31.

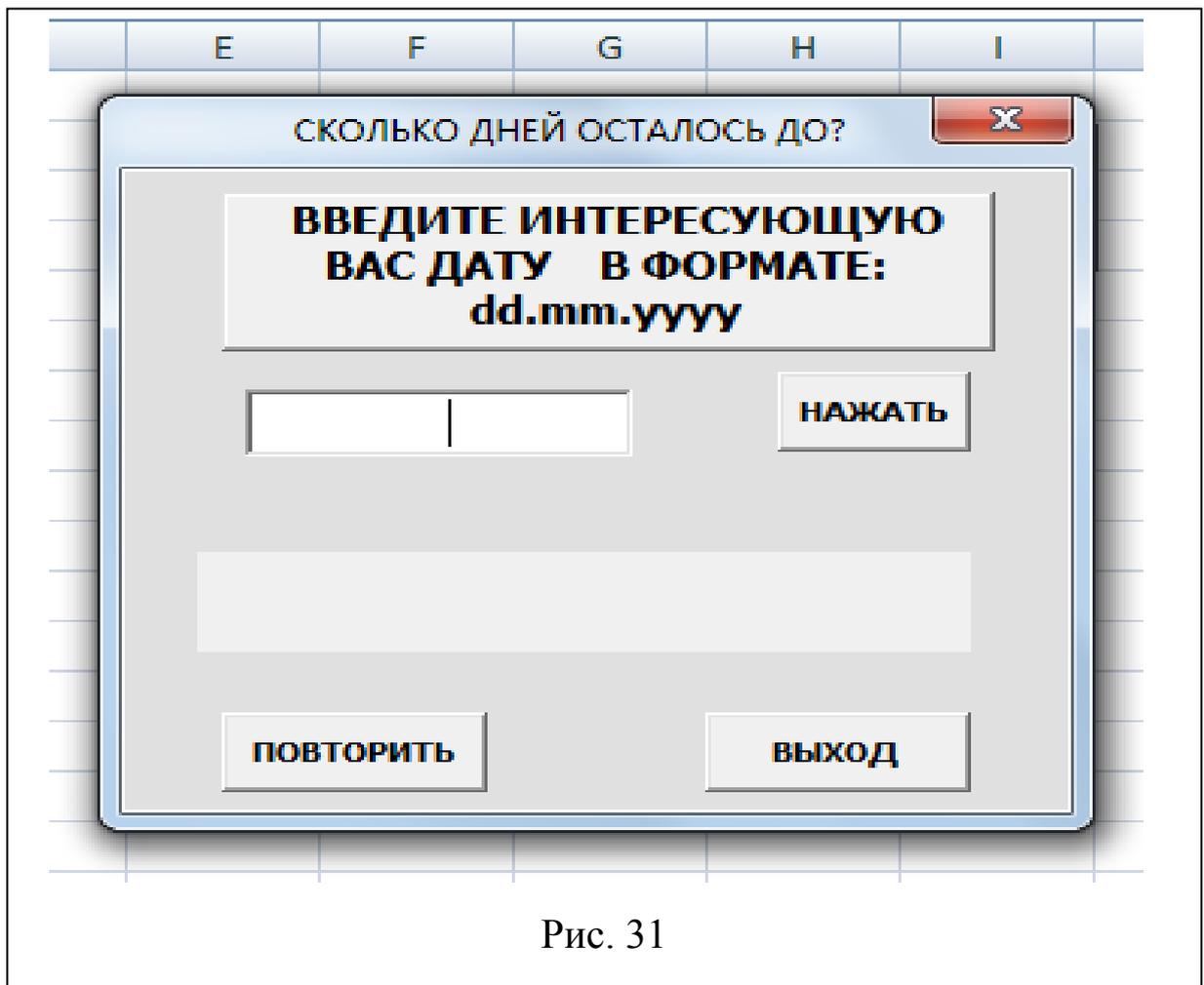
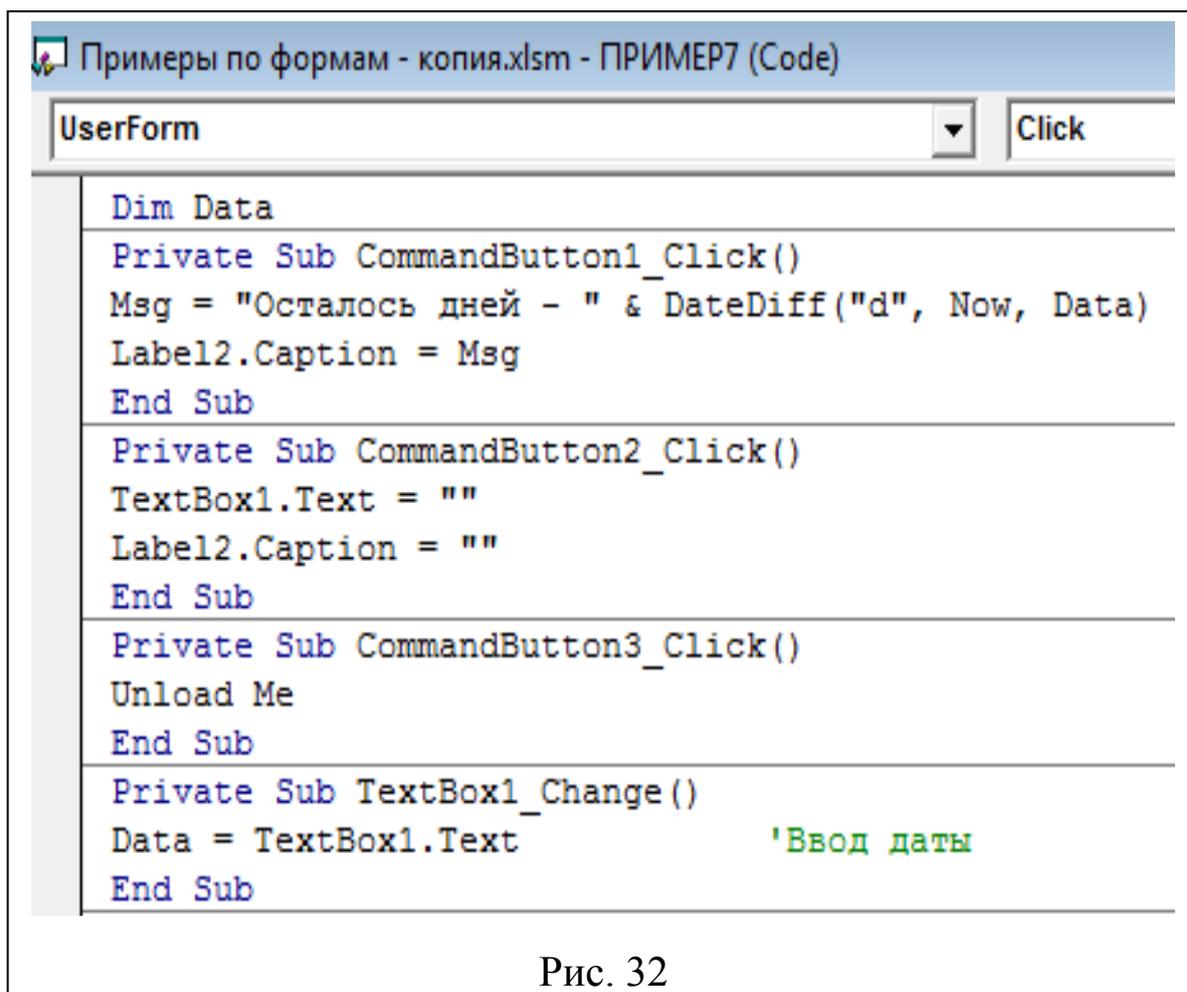


Рис. 31

Процедуры для данной пользовательской формы будут иметь вид, представленный на рис. 32.



```
Примеры по формам - копия.xlsm - ПРИМЕР7 (Code)
UserForm Click
Dim Data
Private Sub CommandButton1_Click()
Msg = "Осталось дней - " & DateDiff("d", Now, Data)
Label2.Caption = Msg
End Sub
Private Sub CommandButton2_Click()
TextBox1.Text = ""
Label2.Caption = ""
End Sub
Private Sub CommandButton3_Click()
Unload Me
End Sub
Private Sub TextBox1_Change()
Data = TextBox1.Text 'Ввод даты
End Sub
```

Рис. 32

Пользовательская форма для запуска vba программы с листа Excel, т.е. для вызова на лист Excel пользовательской формы представленной на рис. 31, может быть выполнена, например, как на рис.33. Здесь пользовательская форма представлена в виде кнопки (Кнопка10) с надписью «VBA программа « Остаток дней» ».

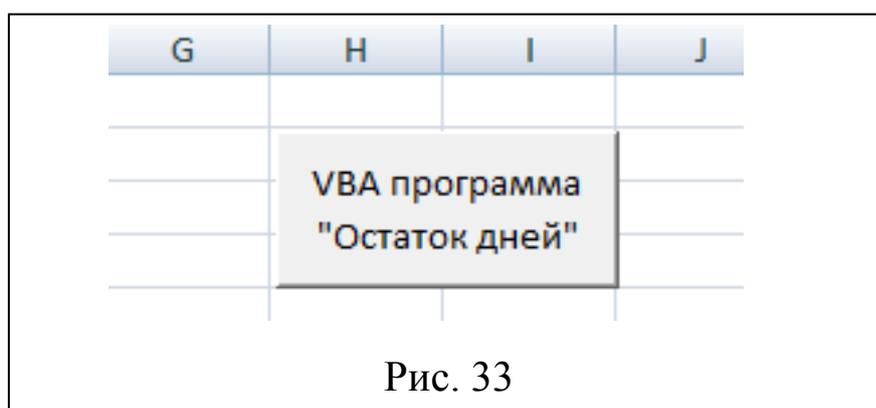


Рис. 33

Процедура для пользовательской формы, представленной на рис. 32 будет иметь следующий вид, см. рис. 34.

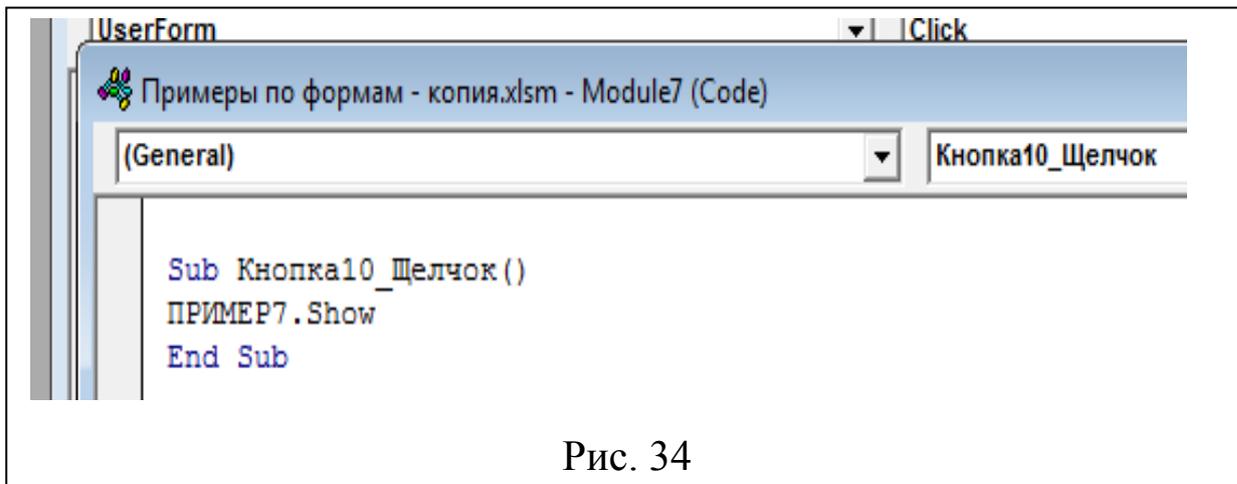


Рис. 34

### Замечания.

Функция DateDiff используется для вычисления разности двух дат. Возвращает значение типа Variant(Long), указывающее число временных интервалов между двумя датами.

### **Общая форма записи функции DateDiff**

DateDiff(Interval,Date1,Date2,[FirstDayOfWeek],[FirstWeekOfYear])

Функция содержит следующие именованные аргументы:

Interval - обязательный аргумент типа String, указывающий тип добавляемого временного интервала:

- уууу – год,
- q - квартал,
- m - месяц,
- y - день года,
- d - день месяца,
- w - день недели,
- ww - неделя,
- h - часы,
- n - минуты,
- s - Секунды;

Interval не чувствителен к регистру букв. Литерал должен быть заключен в кавычки. Для вычисления дней между двумя датами можно использовать значение интервала дня: "y" или "d";

Date1 - обязательный аргумент типа Variant(Date) или литерал даты - первая календарная дата при вычислении разности дат;

Date2 - обязательный аргумент типа Variant(Date) или литерал даты - вторая календарная дата при вычислении разности дат;

FirstDayOfWeek - необязательный аргумент типа Long, определяющего день недели, допускается использование констант;

vbUseSystem=0 - используется значение национальных системных установок:

vbSunday=1 - воскресенье(по умолчанию),

vbMonday=2 - понедельник,

vbTuesday=3 - вторник,

vbWednesday=4 - среда,

vbThursday=5 - четверг,

vbFriday=6 - пятница,

vbSaturday - суббота;

FirstWeekOfYear - необязательный аргумент - числовая константа, указывающая, какой день недели считать первым;

vbUseSystem=0 используется системная информация;

vbFirstJan1=1 неделя, содержащая 1 января;

vbFirstFourDays=2 первая неделя, содержащая как минимум 4 дня нового года;

vbFirstFullWeek=3 Первая полная неделя года.

### **Пример.**

*' Вычислить сколько дней осталось до 21 века*

Dim Today as Date *' сегодняшний день*

Dim Millennium As Date *' 1 день 21 века*

```
Dim retval      ' возвращаемое значение
Today=Date     ' узнаем текущую системную дату
Millenium="01.01.01"
retval=DateDiff ("d",Today,Millenium)
```

### Работа vba программы «Остаток дней»

При нажатии на клавишу с названием «VBA-программа «Остаток дней»» (см. рис. 33) на листе Excel появиться пользовательская форма. При вводе в неё интересующей Вас даты, например, 12.03.2012, с учётом того, что текущая дата 04.01.2012, форма примет следующий вид, см. рис.35.

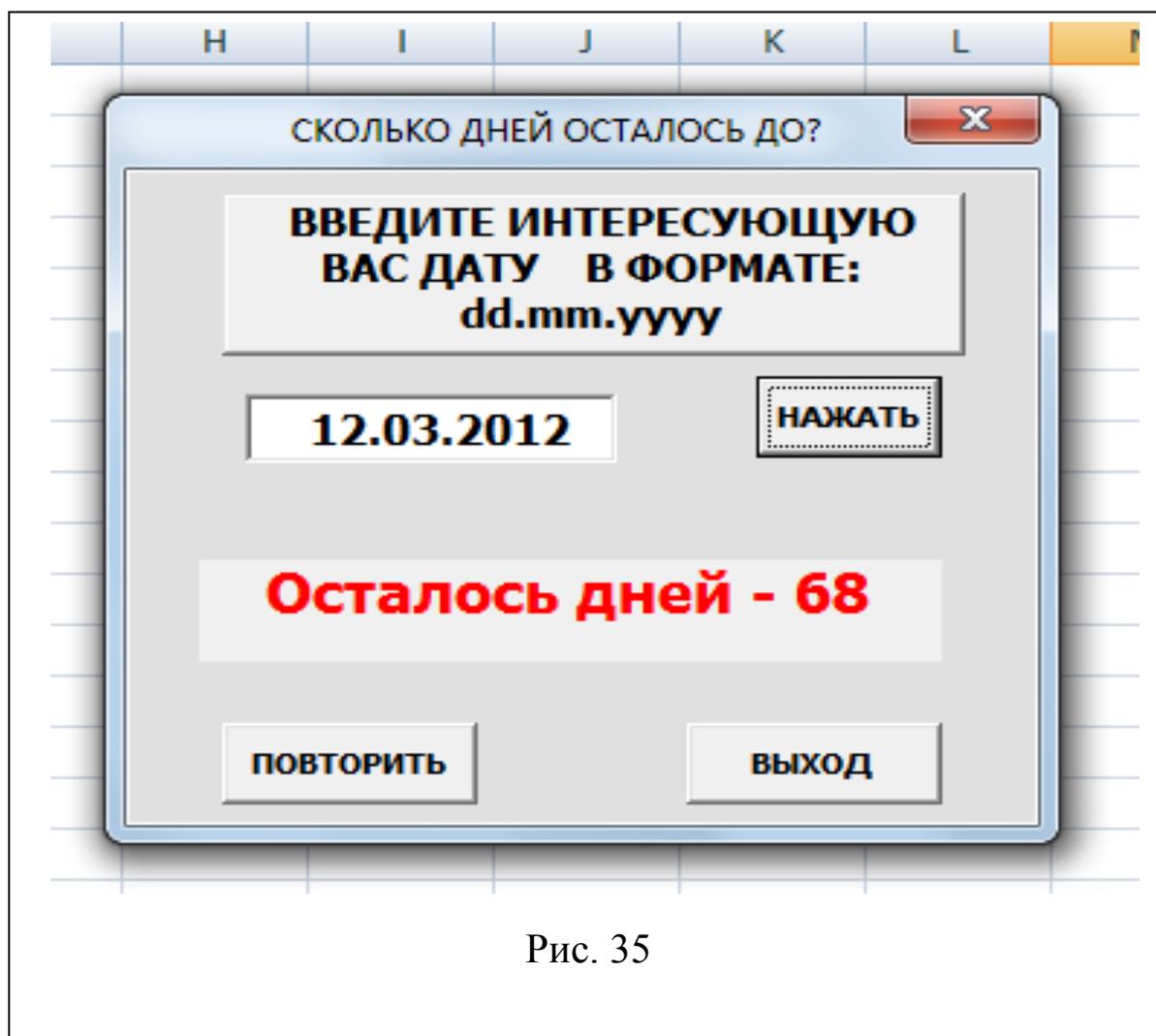


Рис. 35

### 3.4. Пример 4. VBA программа «Таймер»

**Задание.** Создать vba программу под названием «Таймер». В программу включить пользовательскую форму и процедуры к ней, в которых реализуется ввод временной задержки в секундах, сообщение об истечении заданного времени и организованы кнопки ввода данных, повтора и завершения работы программы. Предусмотреть запуск программы с листа Excel.

#### Создание пользовательской формы и процедур

Порядок создания пользовательской формы аналогичен примерам 1-3. Пользовательская форма для данного примера может быть выполнена, например, в следующем виде, см. рис. 36.

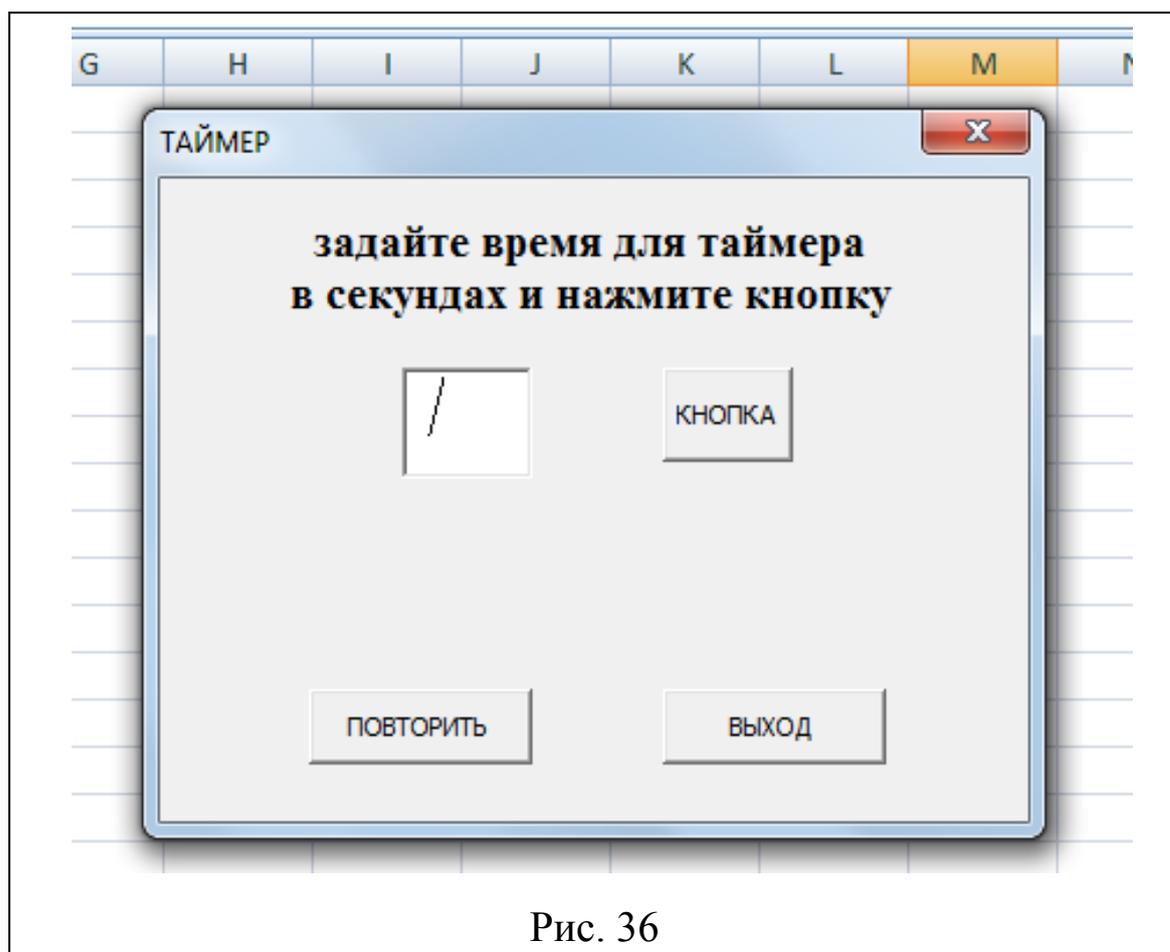


Рис. 36

Процедуры для данной пользовательской формы будут иметь вид, представленный на рис. 37.

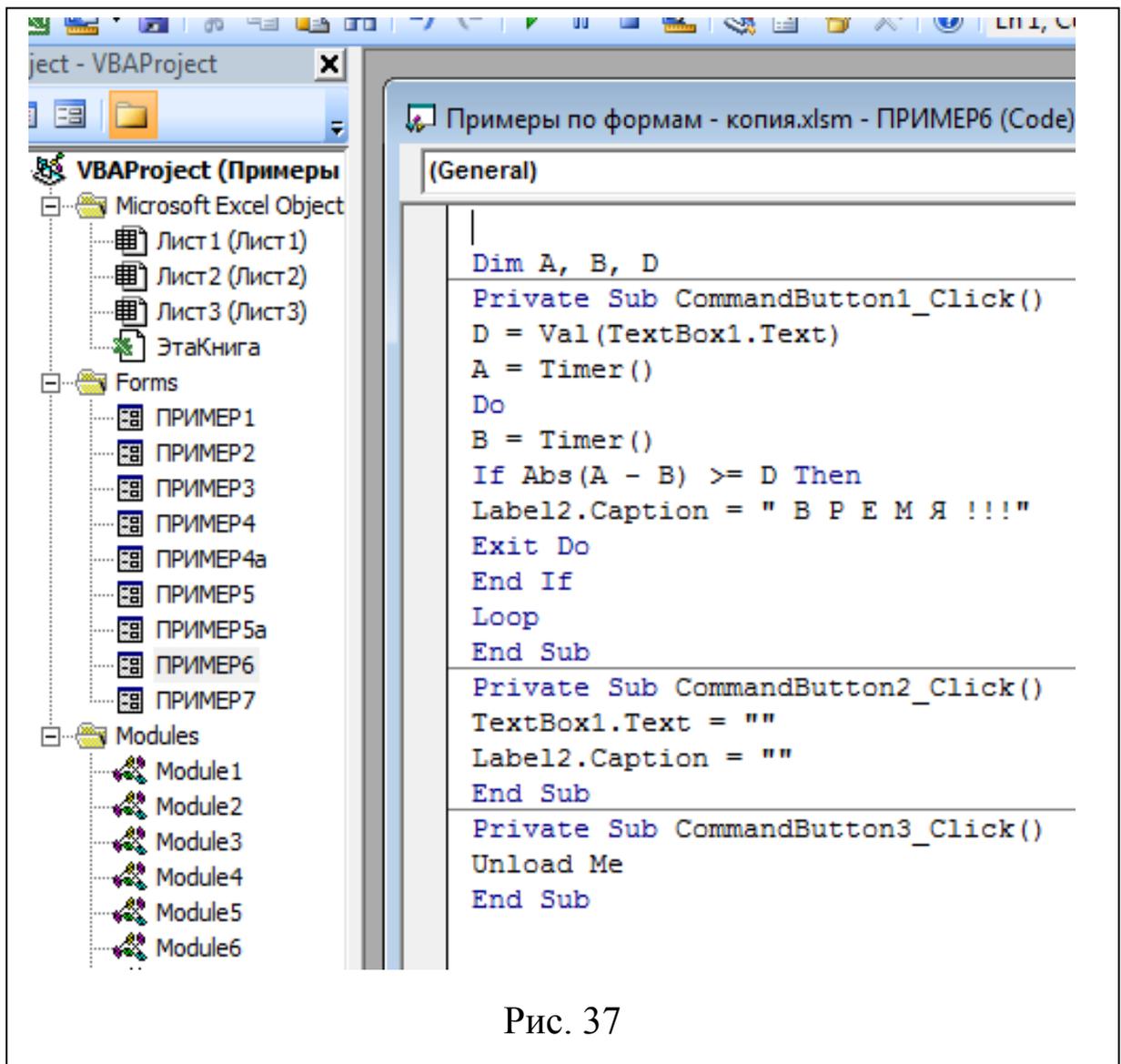
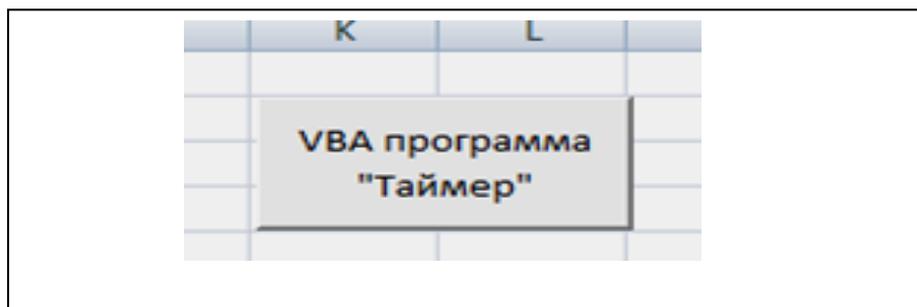


Рис. 37

Пользовательская форма для запуска VBA программы с листа Excel, может быть выполнена, например, как на рис.38. Здесь пользовательская форма представлена в виде кнопки (Кнопка9) с надписью «VBA программа « Таймер» ».



Процедура для пользовательской формы в виде кнопки, представленной на рис. 38 будет иметь следующий вид, см. рис. 39.

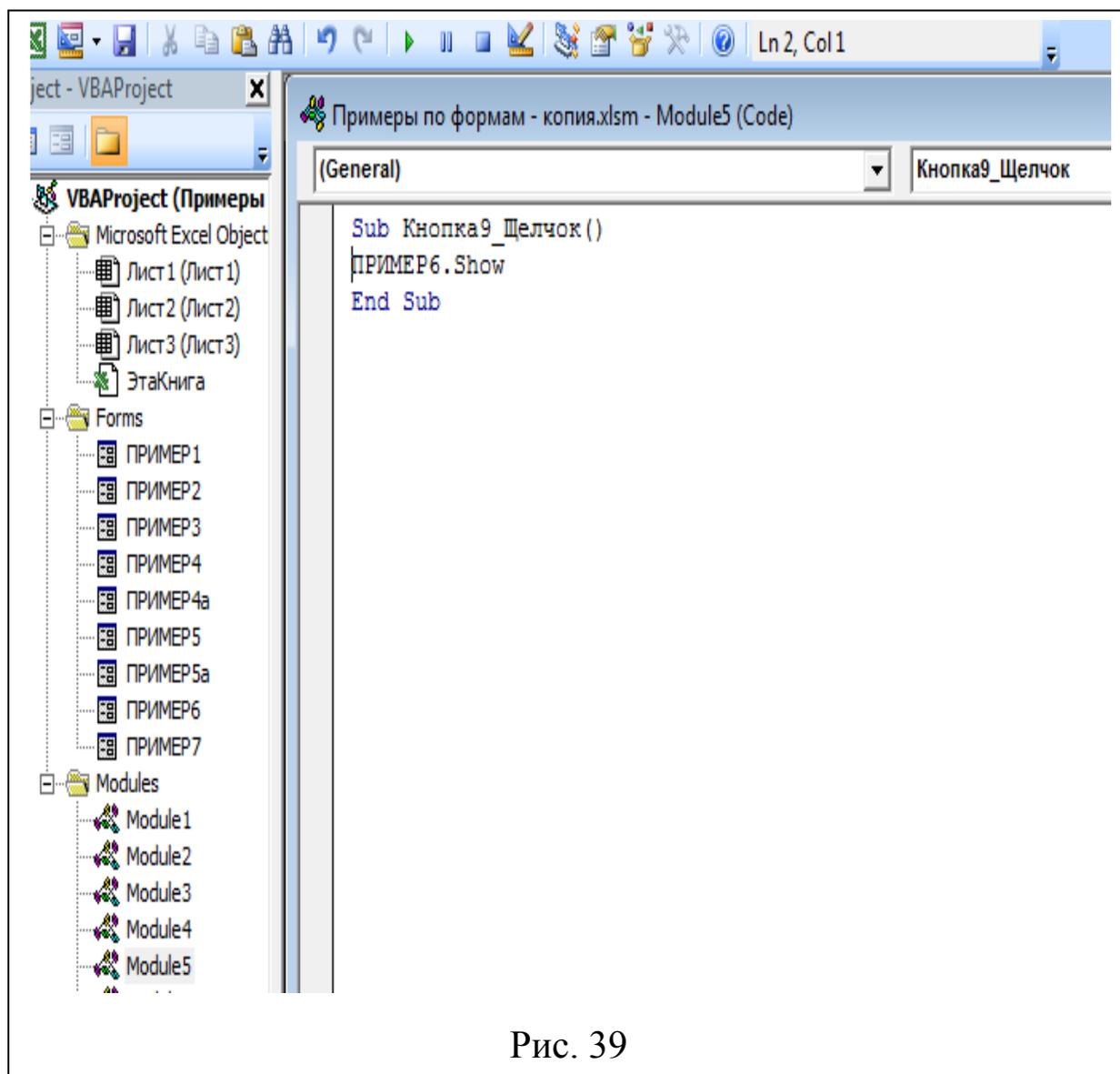


Рис. 39

### **Замечания.**

Функция Timer используется для получения числа секунд, прошедших с полуночи до текущего момента дня.

Возвращает значение типа Single, представляющее число секунд, прошедших после полуночи. Значение, возвращаемое таймером, ограничено числом секунд в сутках и обнуляется ровно в полночь. Эта функция не содержит аргументов.

## Работа vba программы «Таймер»

При нажатии на клавишу с названием «VBA программа «Таймер»» (см. рис. 38) на листе Excel появится пользовательская форма. При вводе требуемой временной задержки в секундах и нажатии на кнопку, после истечения заданной задержки, на форме появится сообщение «**ВРЕМЯ!!!**» и форма примет вид представленный на рис. 40.

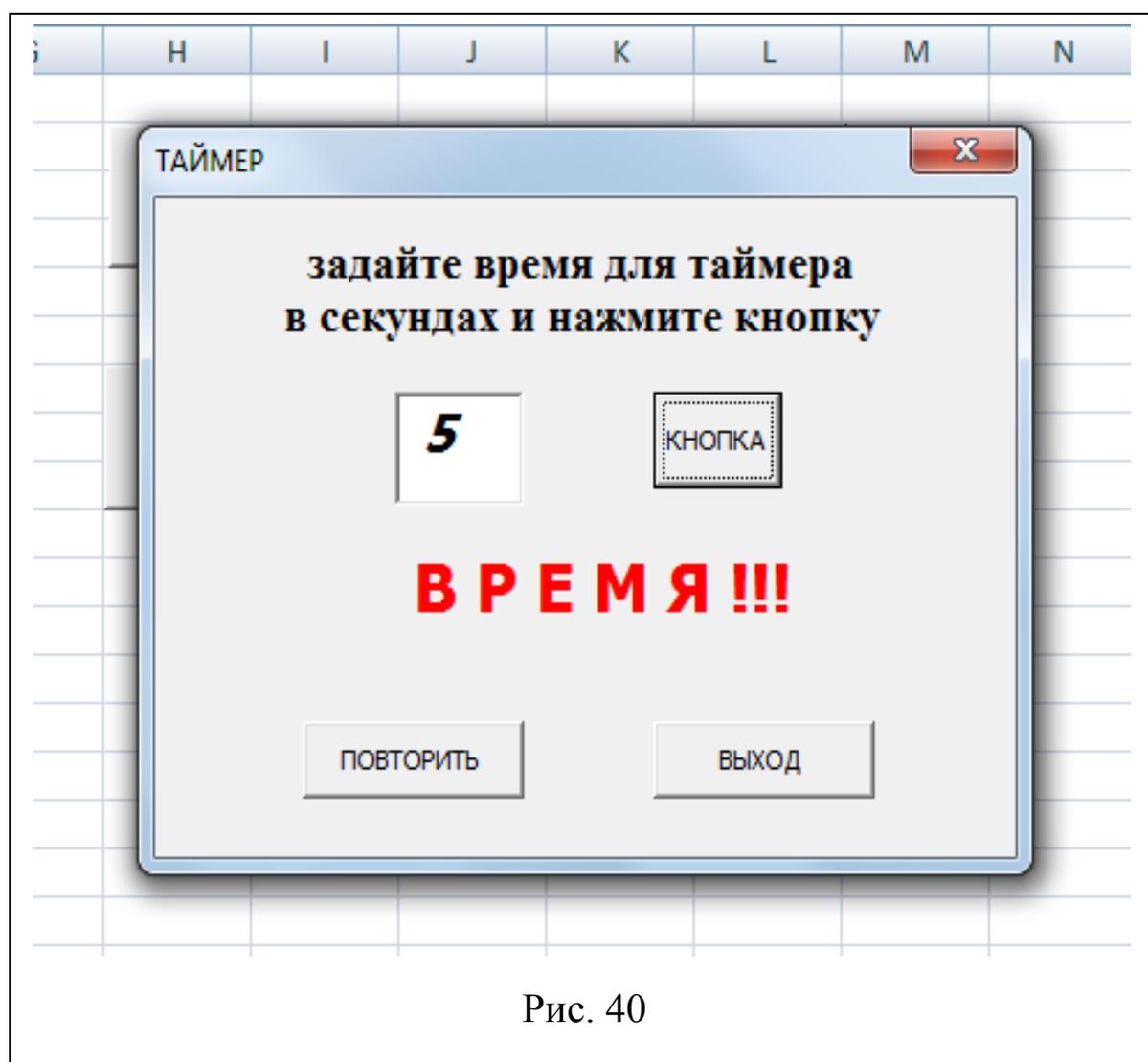


Рис. 40

### 3.5. Пример 5. VBA программа «Игра»

**Задание.** Создать vba программу под названием «Игра», в которой требуется угадать число, которое случайным образом выдаст программа. В пользовательской форме и процедурах к ней организовать ввод задуманного числа в пределах от 1 до 10-ти и его сравнение с числом, которое случайным образом, в тех же пределах, выдаёт программа. В случае если числа не равны, должно быть выдано сообщение «ВЫ ПРОИГРАЛИ» и значения чисел, а в случае если числа равны - сообщение «ВЫ ВЫИГРАЛИ!!!» и так же значения чисел. Предусмотреть организацию кнопок для повторения или завершения работы программы и запуска программы с листа Excel.

#### Создание пользовательской формы и процедур

Порядок создания пользовательской формы аналогичен примерам 1-4. Пользовательская форма для данного примера может быть выполнена, например, в следующем виде, см. рис. 41.

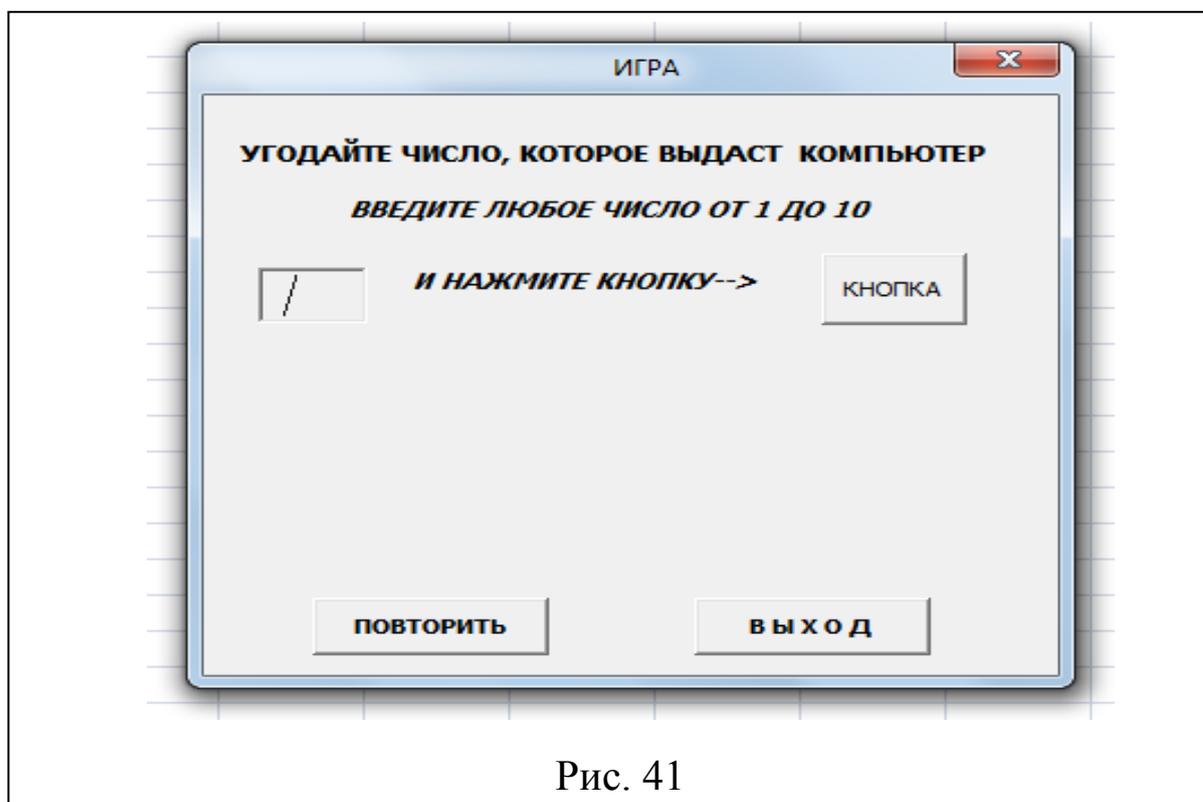


Рис. 41

Процедуры для данной пользовательской формы будут иметь вид, представленный на рис.42.

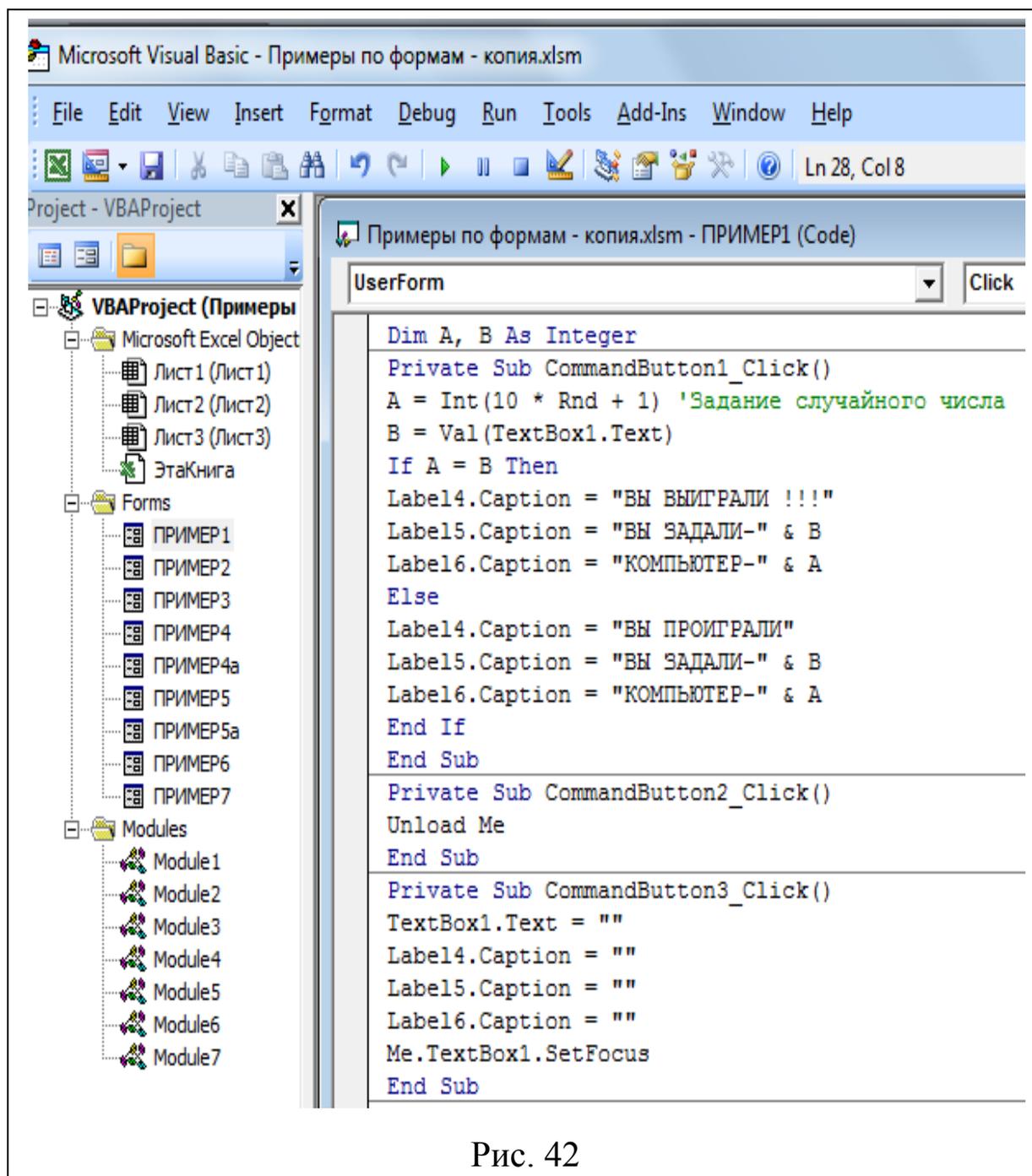


Рис. 42

Форма для запуска vba программы с листа Excel, может быть представлена, например, как на рис. 43. Здесь пользовательская форма представлена в виде кнопки (Кнопка 1) с надписью «VBA программа «Игра»».

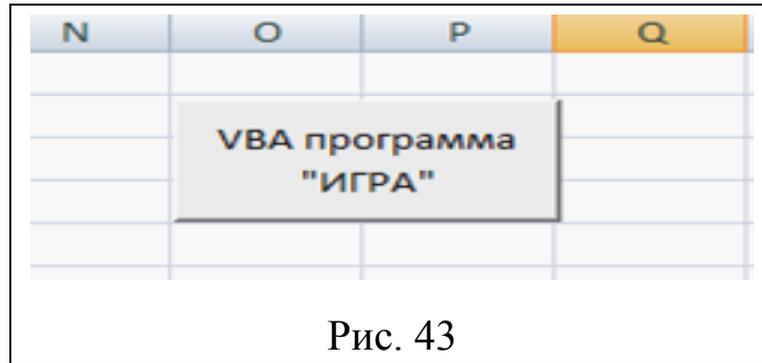


Рис. 43

Процедура для пользовательской формы, представленной на рис. 43 будет иметь следующий вид, см. рис. 44.

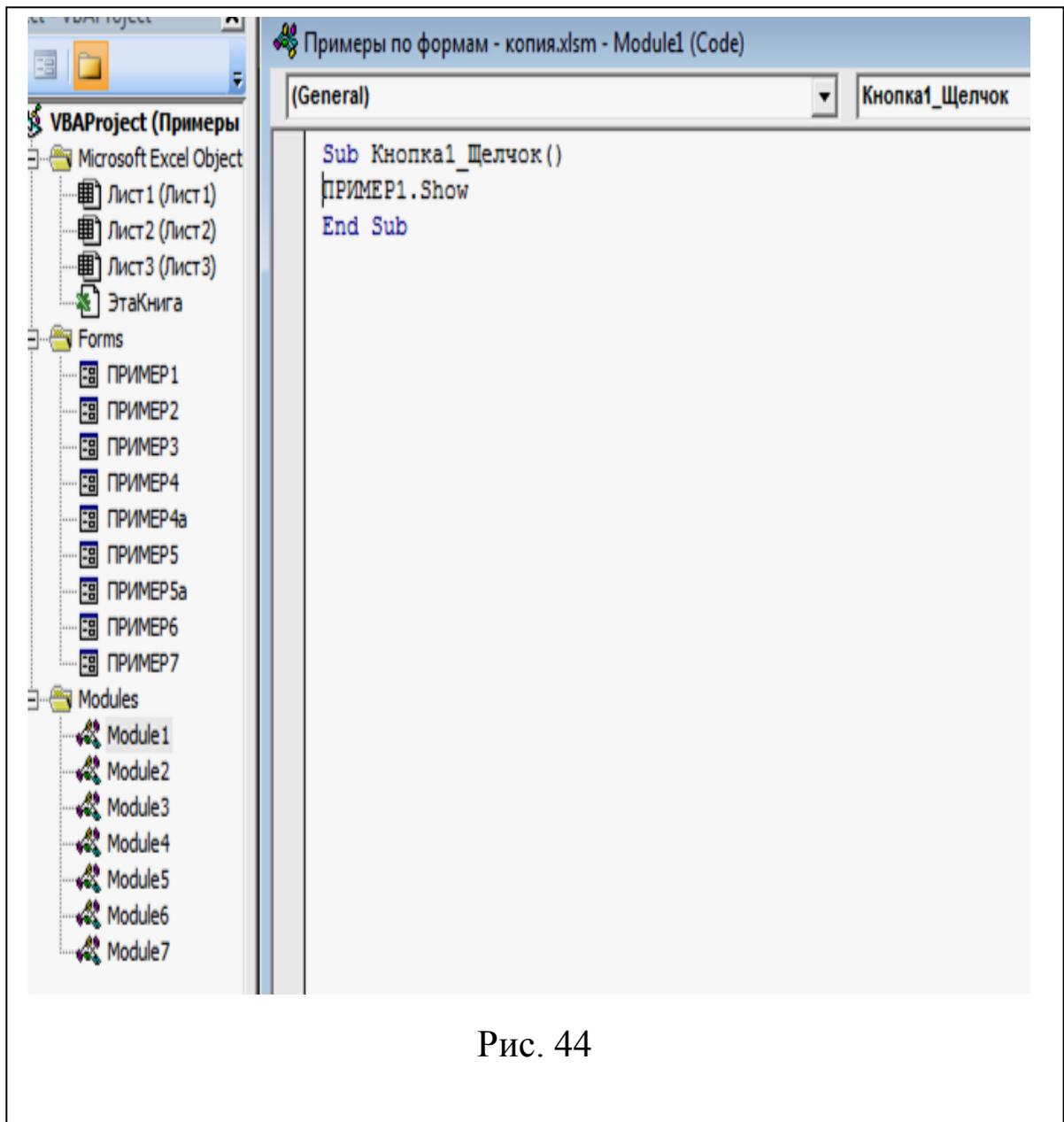


Рис. 44

### Замечания.

Функция **Rnd** [ (Number)], Rnd (Random) служит для генерации случайных чисел.

Функция Rnd возвращает значение в диапазоне от 0 до 1 типа Single, содержащее случайное число (причем 1 не входит в этот диапазон, а 0 входит). Строго говоря, функция возвращает псевдо-случайные числа. При каждом запуске программы, функция генерирует одну и ту же последовательность случайных чисел. Во избежание этого явления используйте инструкцию Randomize

Чтобы получить значения случайных чисел в интервале от min до max используется формула:

$$\text{Int} ( (\text{max} - \text{min} + 1) * \text{Rnd} + \text{min} )$$

где:

min и max-минимальное и максимальное число соответственно;  
Number - необязательный аргумент представляет число типа Single или любое допустимое числовое выражение.

Аргумент может принимать следующие значения: аргумент опущен или больше 0 - генерируется следующее случайное число в последовательности. При повторном запуске программы генерируется аналогичная последовательность случайных чисел.

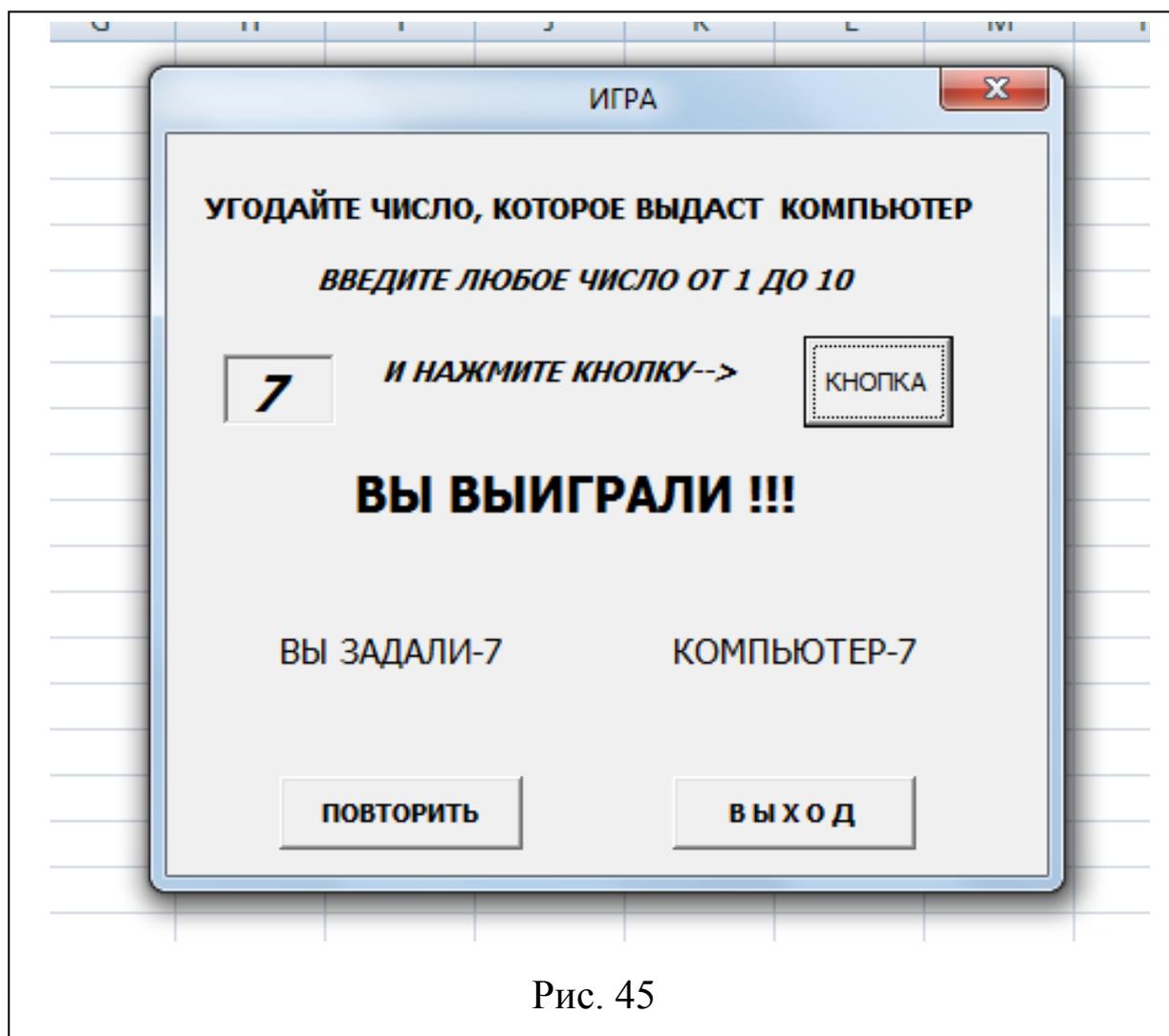
Аргумент отрицательный - генерируется одно и то же число, используя аргумент как опорное число.

Аргумент равен 0 - генерируется одно и то же число, используя число при предыдущем вызове функции.

### **Работа vba программы «Игра»**

При нажатии на клавишу с названием «VBA программа «Игра» » (см. рис. 43) на листе Excel появиться пользовательская форма. После ввода задуманного числа и нажатия кнопки ввода программа определяет случайное число и сравнивает его с введённым

числом. Если числа не равны, то на форме появляется сообщение «ВЫ ПРОИГРАЛИ» и значения чисел. Если числа равны, то на форме появится сообщение «**ВЫ ВЫИГРАЛИ!!!**» и значения самих чисел (см. рис. 45).



### 3.6. Пример 6. VBA программа «Компьютер»

**Задание.** Создать vba программу под названием «Компьютер» для вычисления суммы двух и более чисел. В пользовательской форме и процедурах к ней организовать ввод слагаемых чисел, вычисление и вывод результата, возможность ввода следующих слагаемых чисел вычисление их суммы и вывод результата. Предусмотреть организацию кнопок для повторения или завершения работы программы и запуска программы с листа Excel.

#### Создание пользовательской формы и процедур

Порядок создания пользовательской формы аналогичен примерам 1-5. Пользовательская форма для данного примера может быть выполнена, например, в следующем виде, см. рис. 46.

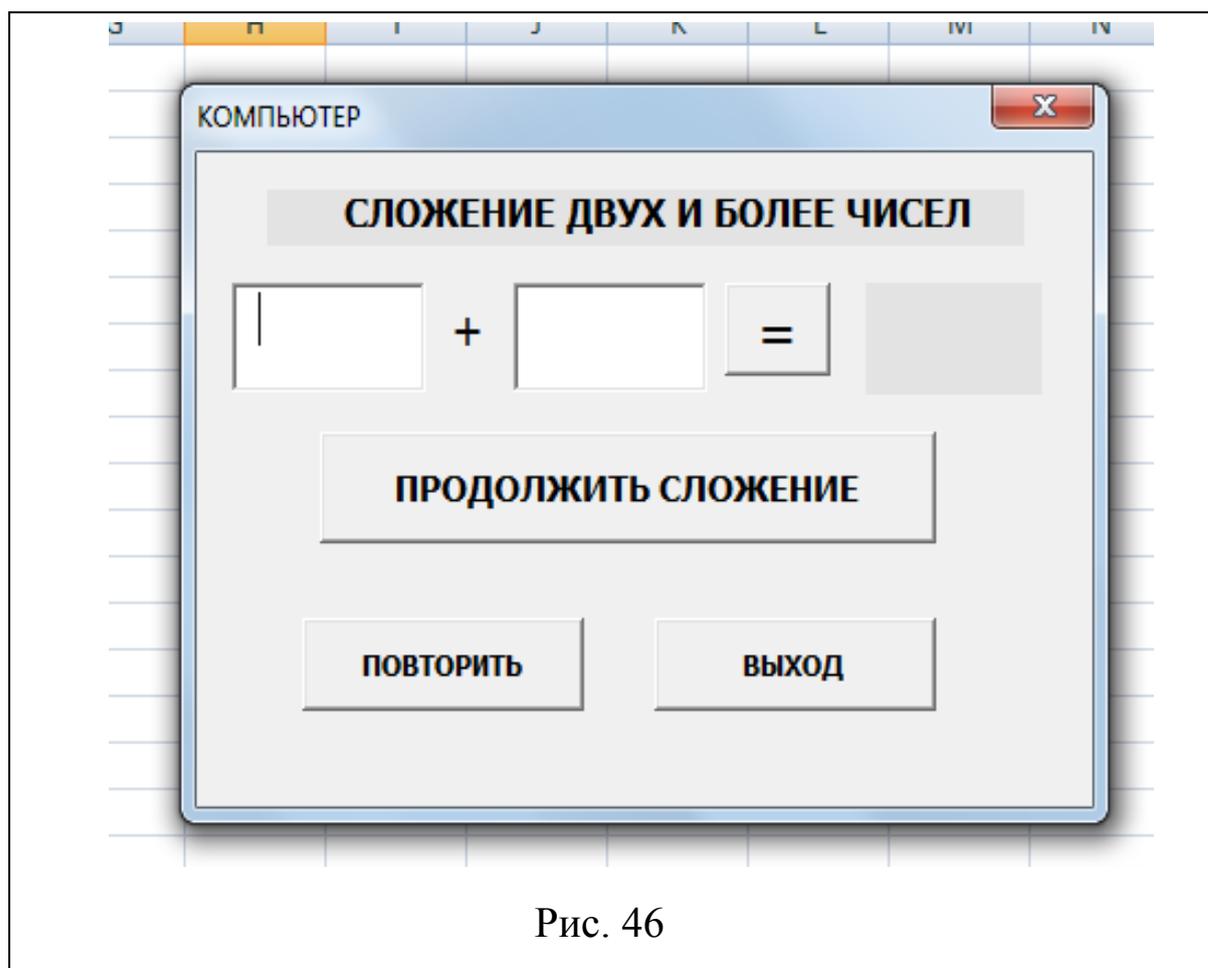


Рис. 46

Процедура для пользовательской формы, представленной на рис. 46 будет иметь следующий вид, см. рис. 47.

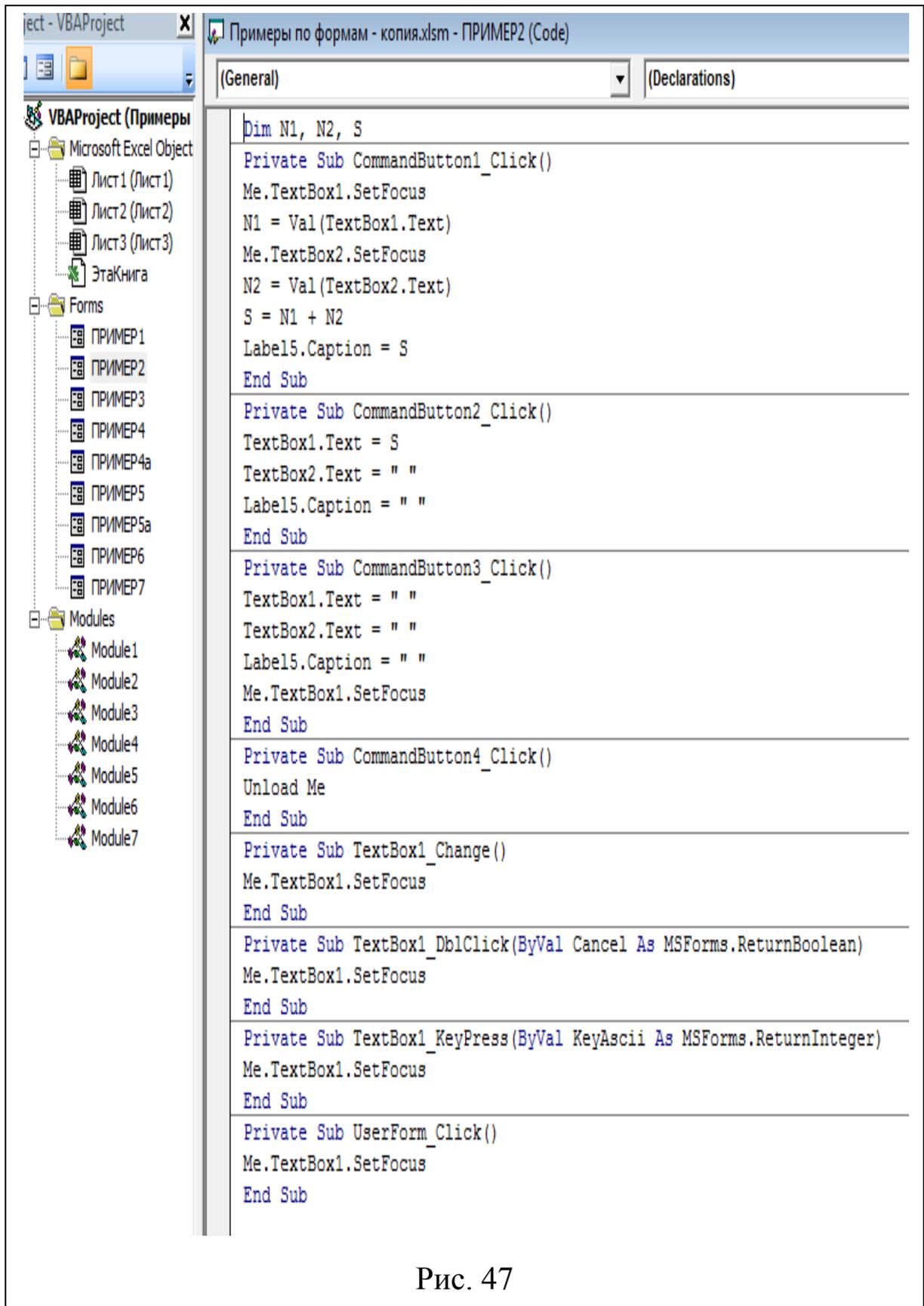


Рис. 47

Форма для запуска vba программы с листа Excel, может быть выполнена, например, как на рис. 48. Здесь пользовательская форма представлена в виде кнопки (Кнопка 1) с надписью «VBA программа «Компьютер»».

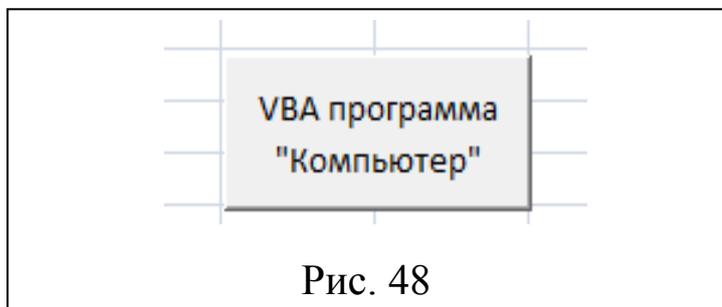


Рис. 48

Процедура для данной кнопки будет иметь следующий вид, см. рис. 49.

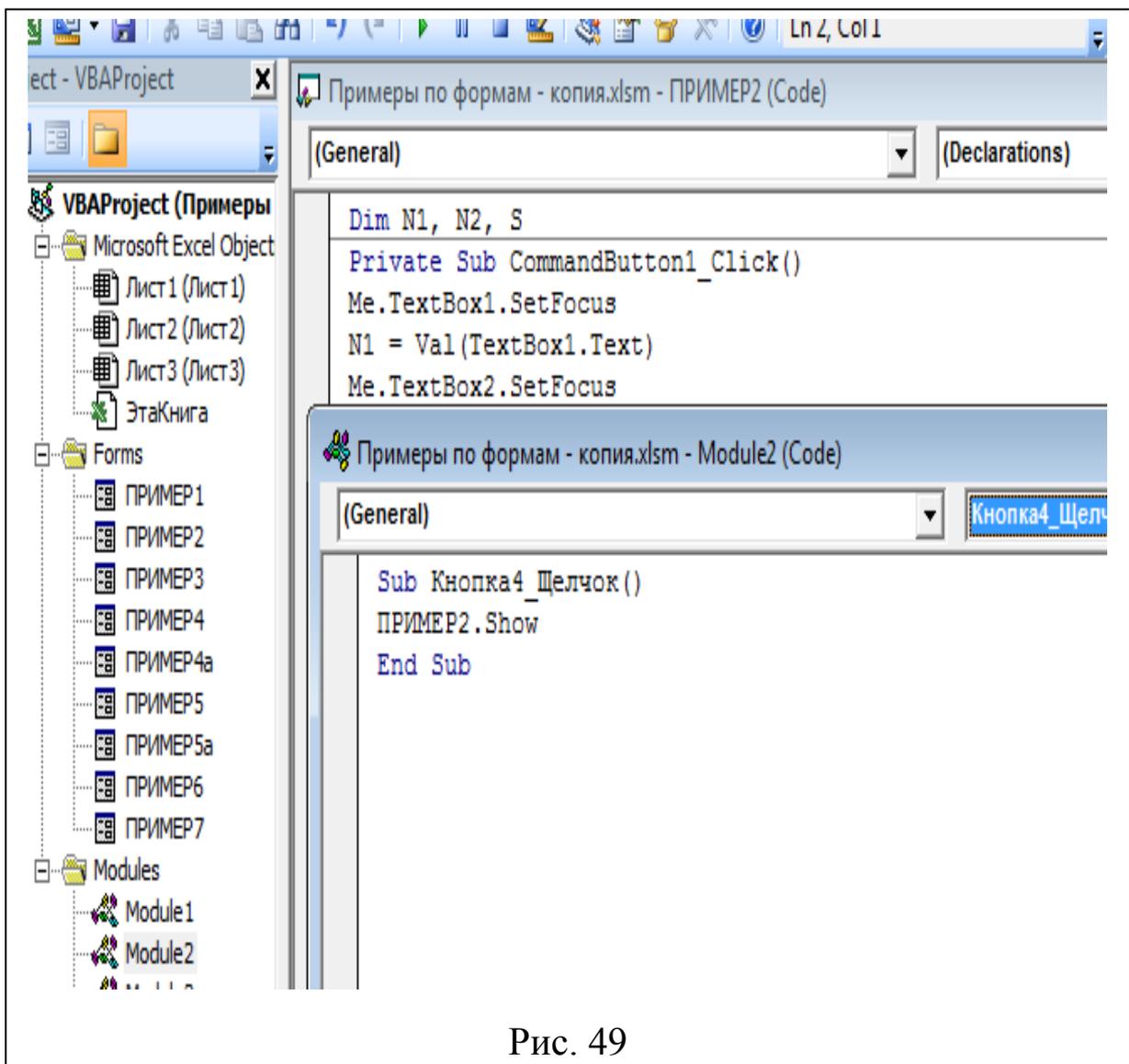


Рис. 49

## Работа vba программы «Компьютер»

При нажатии на клавишу с названием «VBA программа «Компьютер»», см. рис. 48, на листе Excel появится пользовательская форма. После ввода слагаемых чисел и нажатии на клавишу со знаком равенства появится результат сложения, см. рис. 50.

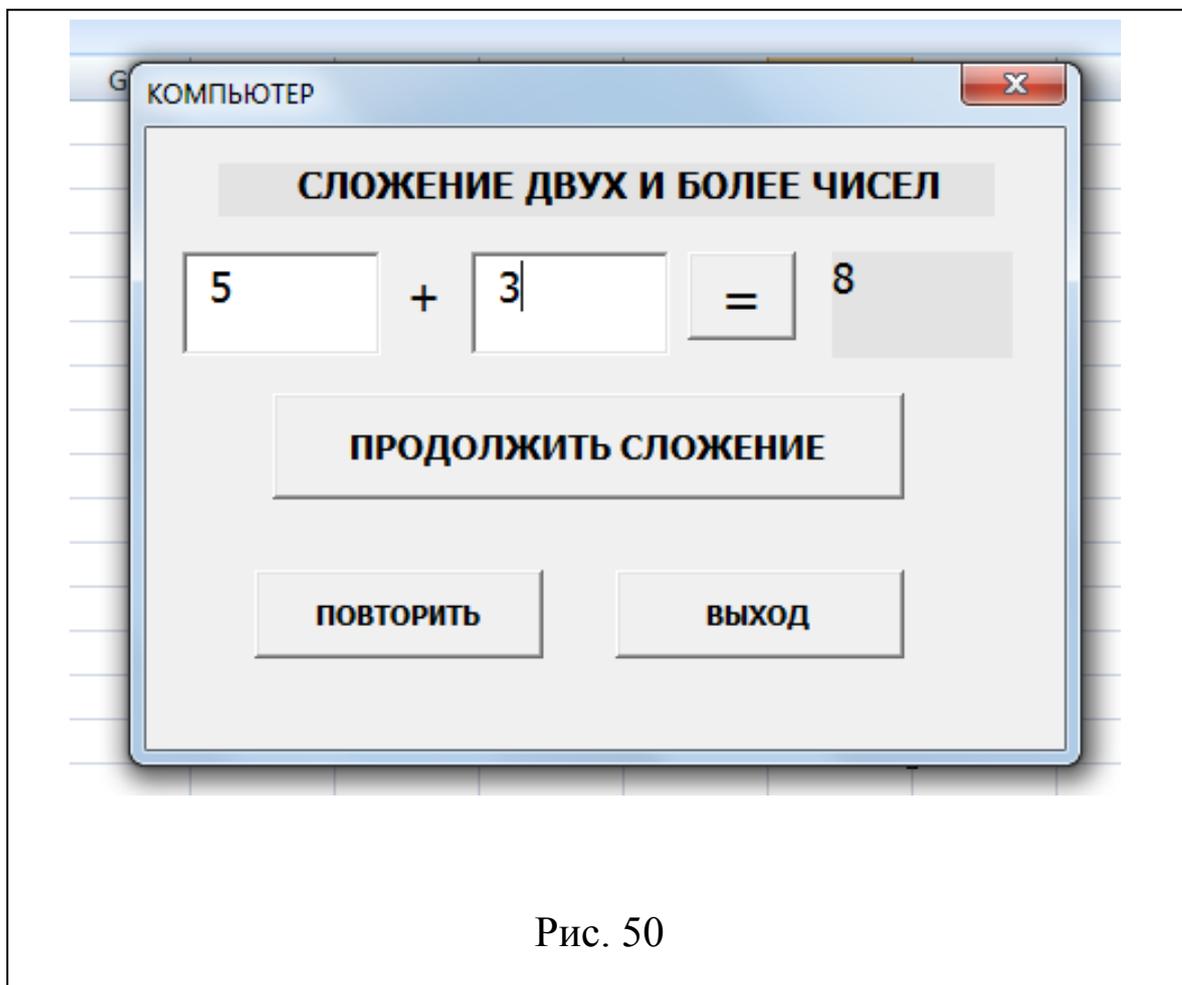


Рис. 50

Если требуется продолжить сложение, то надо нажать на соответствующую клавишу. Далее к полученному результату, в примере число восемь (см. рис. 50), добавить следующее слагаемое, например 15. Нажав, клавишу со знаком равенства получим результат - число 23, см. рис. 51.

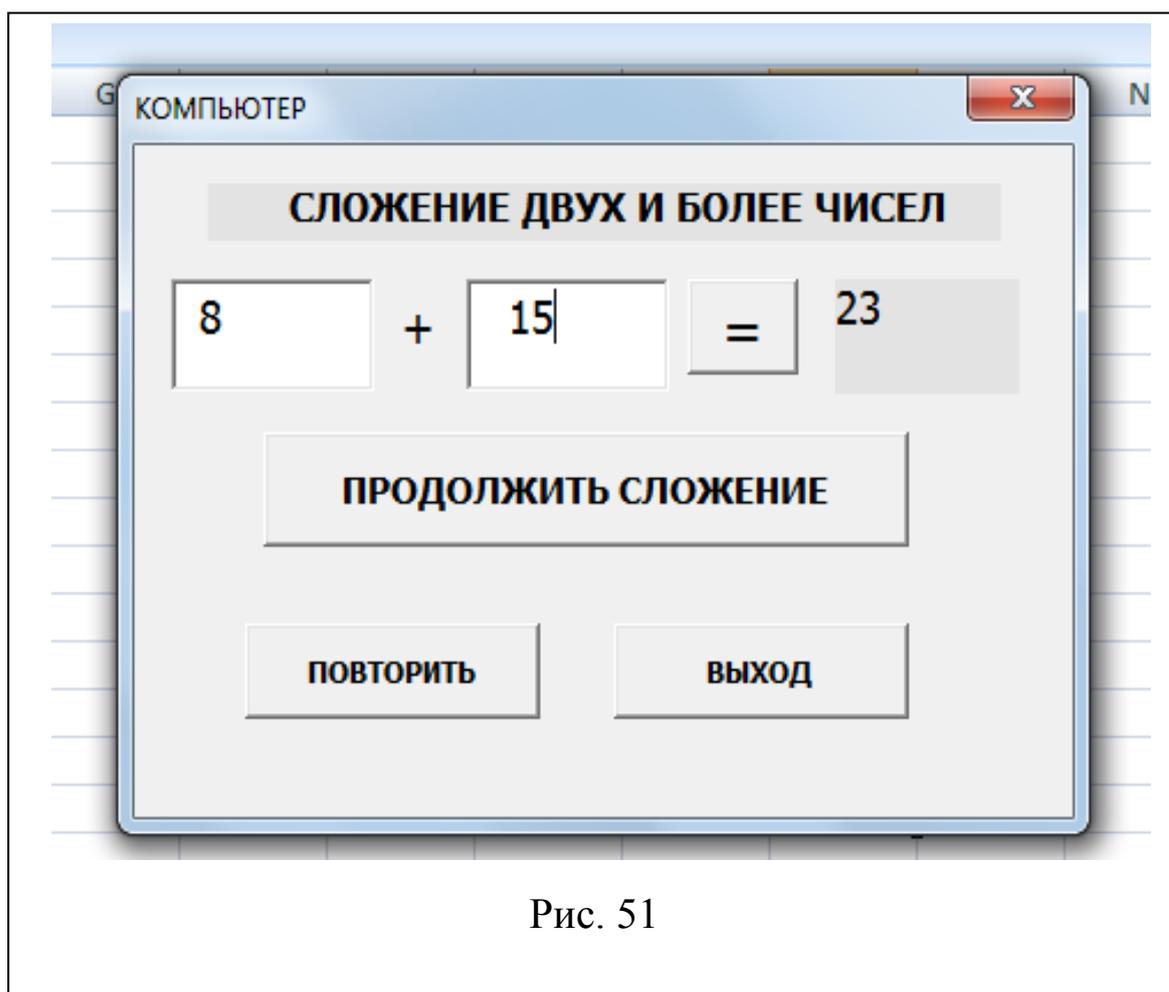


Рис. 51

### 3.7. Пример 7. VBA программа «Список»

В данном примере показано, каким образом можно организовать ввод данных (списка) через пользовательскую форму в одномерный массив и запись этого массива в файл, с последующим считыванием из него данных и их вывод в поле пользовательской формы при её запуске.

В примере показано, как можно организовать добавление и удаление через пользовательскую форму данных (записей) в созданном ранее массиве, с возможностью его сжатия при удалении из него данных.

**Задание.** Создать программу под названием «Список» для ввода через пользовательскую форму списка названий студенческих групп, добавления названий в список и удаление названий студенческих групп из списка. Список названий студенческих групп поместить в одномерный массив. При завершении программы массив со списком должен быть сохранён в файле, а при запуске программы считан из файла обратно в массив и выведен в список пользовательской формы. При удалении данных из массива организовать его сжатие, т.е. смещение данных с целью заполнения свободных мест в массиве, появившихся в результате удаления в нем отдельных записей (его элементов).

#### **Создание пользовательских форм и процедур, их работа**

Порядок создания пользовательских форм аналогичен предыдущим примерам.

В данном задании должно быть создано две формы. Первая пользовательская форма может быть выполнена в виде, представленном на рис. 52. При вводе или добавлении названия студенческой группы в поле с названием «Добавить группу» вводиться на-

звание группы, например 2АС-11 и нажимается клавиша «<- Ввод». В результате этого введённое название группы появляется в списке с заголовком «Выбрать группу» (см. рис. 53).



Рис.52

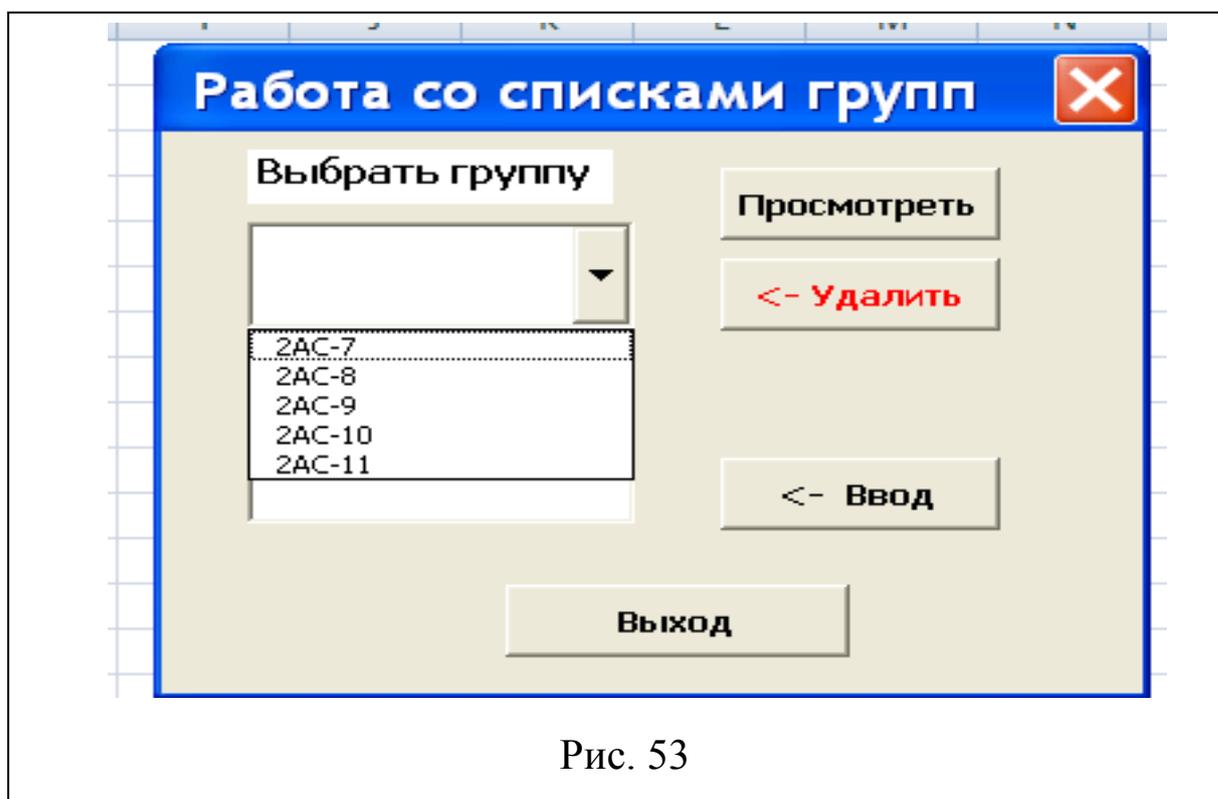


Рис. 53

Из списка студенческих групп выбирается нужная группа, например 2АС-11. Для этого выделяем мышью эту группу и нажимаем клавишу «Просмотреть» в случае необходимости её просмотра и – клавиша «Удалить» в случае её удаления. При удалении название группы из списка исчезает.

При просмотре группы осуществляется переход во вторую пользовательскую форму, в которой должен быть осуществлён ввод, добавление или удаление фамилий, имён и отчеств (ФИО) студентов этой группы. Данная пользовательская форма создаётся аналогично первой форме для ввода, добавлений и удалений наименований студенческих групп. Во второй пользовательской форме в отличие от первой вместо наименований групп будут фигурировать ФИО студентов той или иной группы. Данную пользовательскую форму (модуль) предлагается написать самостоятельно, в качестве тренировочного задания. Заготовка для него будет выглядеть следующим образом (см. рис.54)

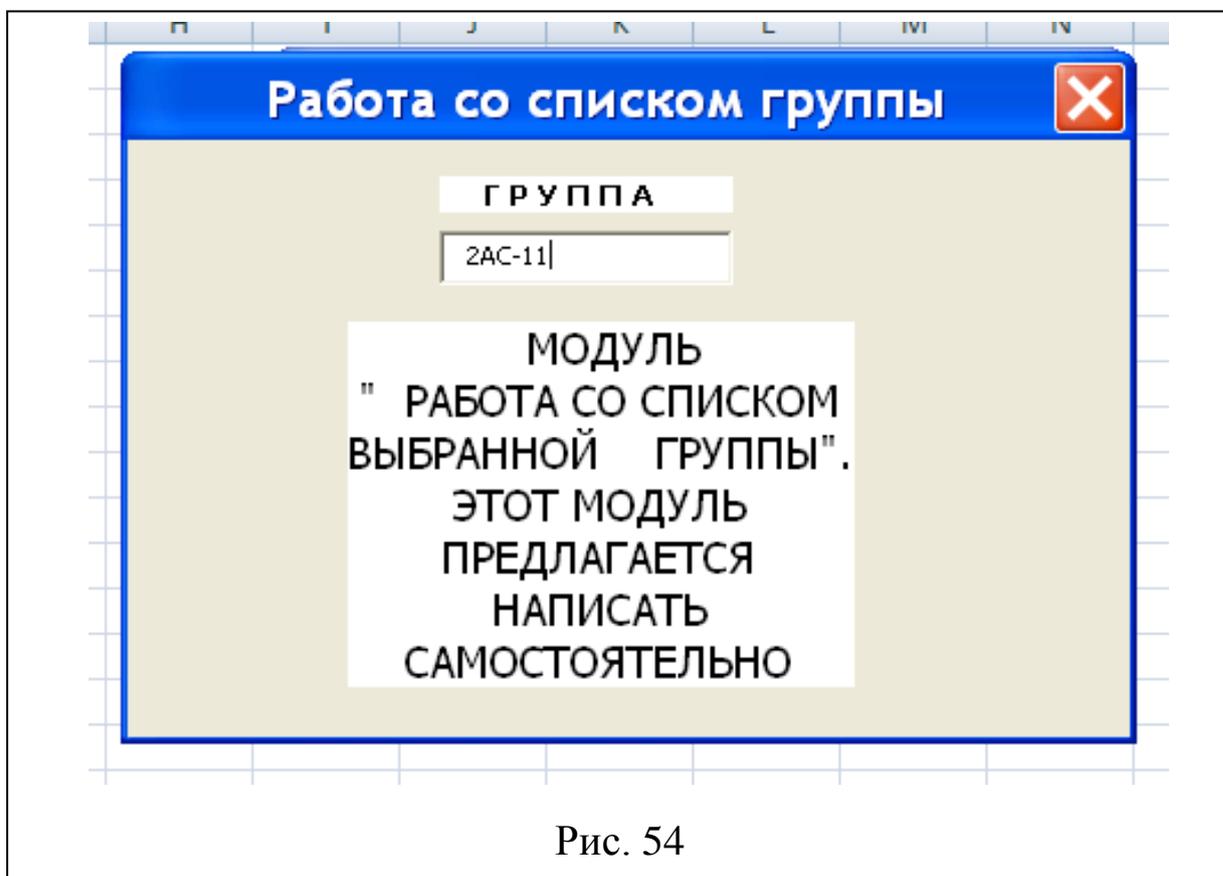


Рис. 54

Ниже представлен листинг процедур к пользовательским формам для рассмотренного задания.

```
Dim dgr As String, Gr(20) As String, _  
k As Integer, prm As String
```

```
Private Sub UserForm_Initialize()  
Open "D:\Grup3" For Input As #1 'Grup3-ФАЙЛ ХРАНЕНИЯ ДАННЫХ  
10 If EOF(1) Then GoTo 20  
Input #1, Gr(k) 'СЧИТЫВАНИЕ ДАННЫХ ИЗ ФАЙЛА  
ComboBox1.AddItem Gr(k)  
k = k + 1  
GoTo 10  
20 Close #1  
End Sub
```

```
Private Sub CommandButton1_Click() 'ДОБАВИТЬ В СПИСОК  
Dim i As Integer  
Open "D:\Grup3" For Append As #1  
dgr = TextBox1.Text  
Gr(k) = dgr 'МАССИВ ДЛЯ ХРАНЕНИЯ ДАННЫХ  
Print #1, dgr  
k = k + 1  
Close #1  
ComboBox1.AddItem dgr  
TextBox1.Text = ""  
End Sub
```

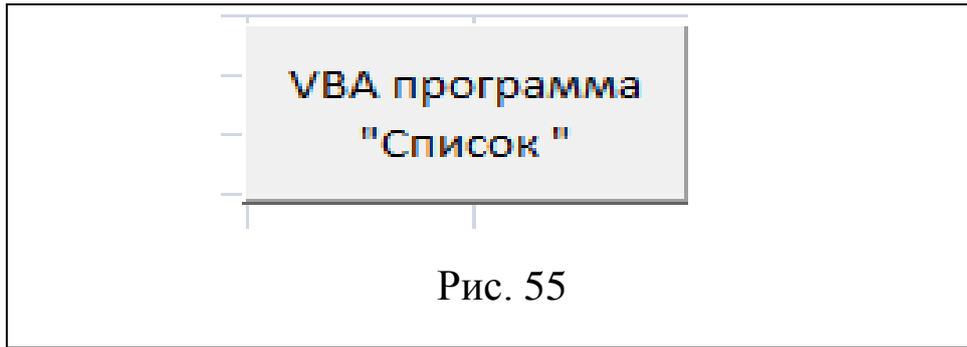
```
Private Sub CommandButton2_Click() 'ВЫХОД  
Unload Me  
End Sub  
Private Sub CommandButton3_Click() 'ПРОСМОТРЕТЬ СПИСОК  
prm = ComboBox1.Text  
MsgBox " Группа -" & prm & "." & _  
vbLf & " Работа со списком группы.", , _  
" Программа список "  
UserForm5.Show 'ПЕРЕХОД ВО ВТОРУЮ ФОРМУ  
End Sub
```

```

Private Sub CommandButton4_Click()  'УДАЛЕНИЕ ИЗ СПИСКА
Open "D:\Grup3" For Output As #1
dgr = ComboBox1.Text
MsgBox "      Группа -" & dgr & "." & _
vbLf & " Будет удалена из списка.", , _
" Программа: удалить      "
ku = k - 1
For i = 0 To ku
If dgr = Gr(i) Then
N = i
Gr(i) = ""
Exit For
End If
Next i
For i = N To ku - 1                               'СЖАТИЕ МАССИВА
Gr(i) = Gr(i + 1)
Next i
For i = 0 To ku - 1
Print #1, Gr(i)
Next i
Close #1
k = 0
ComboBox1.Clear
Call UserForm_Initialize
End Sub

```

Для удобства запуска программы можно организовать её запуск с рабочего листа Excel. Для этого надо на рабочем листе Excel создать кнопку, при нажатии которой происходило бы событие, связанное с запуском процедур первой пользовательской формы UserForm6. Процесс создание такой кнопки подробно рассмотрен в предыдущих примерах. В данном случае сама кнопка и процедура к ней будут выглядеть следующим образом, смотри соответственно рис. 55 и рис. 56. В процедуре сделано обращение к первой пользовательской форме, имя которой UserForm4.



```
Sub Кнопка2_Щелчок() 'ВЫЗОВ ПРОГРАММЫ  
UserForm4.Show  
End Sub
```

Рис. 56

## 4. Задания для самостоятельной работы

**Задание.** Для приведённых ниже вариантов заданий создать vba программу в виде пользовательских форм и процедур к ним, в которых реализуются указанные в вариантах действия. Варианты расположены в порядке возрастания сложности их заданий.

Предусмотреть организацию на форме кнопок для повторения и завершения работы программы, а также кнопки запуска программы с листа Excel.

При выполнении заданий использовать приведённые выше примеры создания аналогичных программ.

### Варианты заданий.

1. Организовать в программе определение количества дней оставшихся до дня вашего рождения или интересующего вас человека. Организовать ввод через форму имени пользователя, даты его рождения и вывод результата с приветствием пользователя.
2. Организовать в программе определение количества дней прошедших со дня вашего рождения. Организовать ввод через форму имени пользователя, даты его рождения и вывод результата с приветствием пользователя.
3. Организовать в программе определение количества дней оставшихся до студенческих каникул. Организовать ввод через форму имени пользователя, даты начала каникул и вывод результата с приветствием пользователя.
4. Организовать в программе определение количества дней прошедших со дня указанной вами даты. Организовать ввод через форму имени пользователя, указанной даты и вывод результата с приветствием пользователя.

5. Создать программу под названием «Кухонный таймер», в которой при достижении указанного в форме пользователем времени готовки блюда, в форме выдаётся соответствующее сообщение. В форме указать имя повара и название приготавливаемого блюда.
6. Создать программу под названием «Секундомер», в которой при нажатии соответствующей кнопки происходит отчет времени в секундах, а при нажатии другой кнопки в форму выдаётся сообщение с указанием количества прошедших секунд и имени пользователя программой.
7. Создать программу под названием «будильник», в которой при достижении указанного в форме пользователем времени в форме выдаётся надпись «подъём!».
8. Создать программу, которая через каждые 30 секунд выдаёт в пользовательскую форму сообщение с указанием количества минут оставшихся до конца занятий. Ввод времени до конца занятий должен быть осуществлён через форму.
9. В программе организовать ввод списка из десяти разных фамилий. Ввод фамилий организовать в процедуре, используя метод AddItem см. пример 2. В программе реализовать выбор случайным образом одной из введённых фамилий и вывод её в форму.
10. Создать программу под названием «Жребий», в которой пользователь должен угадать, что выдаст случайным образом программа «орёл» или «решка». Ввод стороны монеты и вывод результатов должны быть организованы с помощью пользовательской формы.
11. Создать программу под названием «Очередь», в которой пользователь должен организовать ввод через форму пяти фамилий и вывод фамилий в том порядке, который будет определён в программе с помощью функции случайных чисел, т. е. случайным образом.

12. Создать программу под названием «Спортлото», в которой для заданного через форму количества чисел от двух до шести определяются числа, лежащие в интервале от 1 до 99. Ввод данных и вывод результата должны быть организованы с помощью пользовательской формы.

13. Организовать в программе выполнение операции возведения в степень, как в целую, так и в дробную. Ввод данных и вывод результата производить через форму.

14. Организовать в программе вычисление тригонометрической функции косинуса для задаваемого через форму угла в градусах и вывод в форму полученного результата.

15. Организовать в программе вычисление логарифма заданного числа по заданному основанию логарифма. Ввод исходных данных и вывод полученного результата производить через форму.

16. Организовать в программе перевод одних единиц измерения в другие, по выбору пользователем соответственно для температуры - градусы по Цельсию в градусы по Фаренгейту, сантиметры в дюймы, угловые градусы в радианы.

17. Создать программу под названием «Вопрос», в которой на заданный через форму вопрос предлагается выбрать один из трёх предложенных ответов. В форме должен быть организован вывод сообщения о том правильный ли был ответ.

18. Создать программу под названием «Справочник по металлам» для пяти наименований металлов. Выбор нужного металла должен быть осуществлён из списка металлов представленного в пользовательской форме. После выбора из списка интересующего металла в форме должен быть выведен список с физико-механическими характеристиками металла, а именно: плотность, модуль упругости, предел прочности и предел текучести. Предусмотреть добавление наименований металлов в список справочника и их удаление.

19. Создать программу под названием «Телефонный справочник» для пяти фамилий. Выбор нужной фамилии должен осуществляться из списка фамилий представленного в пользовательской форме. После выбора из списка нужной фамилий - в форме должен быть выведен соответствующий номер телефона. Предусмотреть добавление фамилий в список справочника и их удаление.

20. Для программы, представленной в разделе 3.7 написать модуль « Работа со списком выбранной группы». В модуле предусмотреть ввод фамилий студентов, удаление фамилий из списка и добавление, аналогично тому, как это сделано в примере 7 с названиями студенческих групп.

## 5. Литература

1. Архипов В.Н., Калядин В.И., Любин А.Н., Макаров А.И. Программирование на фортране: методические указания к лабораторным работам. - М.;МГТУ «МАМИ», 2007, -144с.
2. Антомони В.И., Архипов В.Н., Любин А.Н., Тихомиров В.Н. Программирование на VBA в Microsoft Office: сборник лабораторных работ. - М.;МГТУ «МАМИ», 2011, -152с.
3. Антомони В.И., Архипов В.Н., Любин А.Н., Тихомиров В.Н. Основы программирование на VBA в Microsoft Office: учебное пособие. - М.;МГТУ «МАМИ», 2011, -142с.
4. Антомони В.И., Архипов В.Н., Любин А.Н., Тихомиров В.Н. Основы программирование на С#: сборник лабораторных работ - М.;МГТУ «МАМИ», 2011, -195с.
5. Гарнаев А. Самоучитель VBA технология создания пользовательских приложений. - М. ВHV, 2006 – 153с.
6. Калядин В.И., Макаров А.И. основы работы на персональном компьютере: сборник лабораторных работ. - М.;МГТУ «МАМИ», 2010, -85с.
7. Кузьменко В.Г. VBA эффективное использование М. БИНОМ 2009, -617с.
8. Лобанов А.С., Туманова М.Б. Решение задач на языке Visual Basic for Application: учебное пособие. - М.;МГТУ «МАМИ», 2009, -90с.
9. Слепцова Л.Д. Программирование на VBA в Microsoft Office 2010. - М. «Диалектика», 2010, -431с.
10. Уокенбах Дж. Профессиональное программирование на VBA в Excel 2003.- М. «Диалектика», 2005, -800с.

## СОДЕРЖАНИЕ

Введение.....	2
1.Объекты, методы, свойства и события в VBA.....	4
2. Пользовательские формы.....	6
2.1. Элементы управления.....	9
2.2. Свойства элементов управления.....	10
2.3. Связь пользовательской формы с процедурой.....	11
3. Примеры создания пользовательских форм и написания процедур к ним.....	12
3.1. Пример 1.VBA программа «Дата и время».....	112
3.2. Пример 2.VBA программа «Автомобиль года».....	20
3.3. Пример 3. VBA программа «Остаток дней».....	29
3.4. Пример 4. VBA программа «Таймер».....	34
3.5. Пример 5. VBA программа «Игра».....	38
3.6. Пример 6. VBA программа «Компьютер».....	43
3.7. Пример 7. VBA программа «Список».....	48
4. Задания для самостоятельной работы.....	54
5. Литература.....	54

*Учебное издание*

Любин Александр Николаевич

СОЗДАНИЕ  
ПОЛЬЗОВАТЕЛЬСКИХ ФОРМ  
В VISUAL BASIC FOR APPLICATIONS  
Учебное пособие

*Под редакцией автора*

*Оригинал-макет подготовлен редакционно-издательским отделом  
МГТУ «МАМИ»*

По тематическому плану внутривузовских изданий учебной литературы на 2012 г.

Подписано в печать. Формат 60x90 1/16. Бумага 80г/м<sup>2</sup>

Гарнитура «Таймс». Ризография. Усл. печ. л. 12.1

Тираж 100 экз. Заказ №.

МГМУ «МАМИ»

107023, г. Москва, Б. Семеновская ул., 38

