

**А.Н. Ефремова**

# Примеры использования языка программирования VBA в Microsoft Excel



Методические указания

Братск 2010

Министерство образования и науки РФ  
Государственное образовательное учреждение  
высшего профессионального образования  
«Братский государственный университет»

**А.Н. Ефремова**

# Примеры использования языка программирования VBA в Microsoft Excel

Методические указания

Братск 2010

УДК 681.3.06

Программирование в Microsoft Excel: метод. указания /  
А.Н. Ефремова. – Братск: ГОУ ВПО «БрГУ», 2010. – 58 с.

*Содержат основные сведения теоретического и практического характера по технологии создания пользовательских приложений в Microsoft Excel с использованием языка VBA (Visual Basic for Applications). Предназначено в помощь студентам всех специальностей и всех форм обучения, изучающих дисциплину «Информатика».*

© ГОУ ВПО «БрГУ», 2010

© Ефремова А.Н., 2010

<b>СОДЕРЖАНИЕ</b>	
<b>ВВЕДЕНИЕ</b> .....	5
1. СРЕДА ДЛЯ НАПИСАНИЯ ПРОГРАММЫ НА VBA. .....	7
2. ПРИМЕРЫ ПРОГРАММИРОВАНИЯ НА VBA .....	19
3. СВОЙСТВА И МЕТОДЫ ОБЪЕКТОВ EXCEL .....	29
4. ПРИМЕР - РАЗРАБОТКА БЛАНКА ЗАКАЗА .....	30
5. РАЗРАБОТКА ЗАЯВКИ НА ПОСТУПЛЕНИЕ ТОВАРОВ .....	40
6. РАЗРАБОТКА АВТОМАТИЗИРОВАННОГО БЛАНКА.....	47
7. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ. ..	57
<b>СПИСОК ЛИТЕАТУРЫ</b> .....	61

## ВВЕДЕНИЕ

На сегодняшний день **VBA** – один из самых простых в изучении и применении языков программирования для автоматизации приложений, входящих в пакет Microsoft Office.

В этом издании использование языка **VBA** рассматривается для **Microsoft Excel 2007**.

Следует заметить, что **VBA** относится к объектно-ориентированным языкам. И в целом объектно-ориентированное программирование – это технология написания приложений с помощью объектов. В свою очередь **объект** представляет программный компонент, который позволяет инкапсулировать (скрыть или расположить внутри себя) данные вместе с кодом (фрагментом текста программы), предназначенным для обработки объектов. При этом сам объект не представляет большого значения, а намного значительнее те действия, которые можно совершать над объектом. **Метод** как раз и представляет собой действие, выполняемое над объектом. Синтаксис применения того или иного метода выглядит следующим образом:

### **Объект1 . Метод.**

Здесь под названием **Объект1** подразумевается имя объекта (все объекты имеют имена и по этим именам к ним можно обращаться), а слово **Метод** как раз определяет конкретный метод для данного объекта. **Свойство** представляет атрибут объекта, который определяет его характеристики, такие как цвет, размер, положение на экране, состояние объекта (например, видимость). Чтобы изменить характеристики объекта, надо просто изменить значения тех или иных его свойств. Синтаксис установки значения свойства аналогичен обращению к методу:

## **Объект1 .Свойство = ЗначениеСвойства**

Например, чтобы установить красный цвет у объекта с именем **МоеЯблоко**, необходимо написать конструкцию следующего вида:

**МоеЯблоко . Цвет = Красный.**

Следующим понятием после объектов, методов и свойств, являются события.

**Событие** представляет собой действие, распознаваемое объектом (например, щелчок мышью или нажатие клавиши), для которого можно определить отклик (метод фактически определяет реакцию на событие). События возникают в результате действий пользователя или программы. Также они могут быть вызваны операционной системой.

Суть программирования на языке **Visual Basic** как раз и заключается в этих двух понятиях: событие и отклик на него. Если пользователь производит какое-либо воздействие (например, нажимает кнопку мыши), то в качестве отклика автоматически выполняется программный код созданной пользователем процедуры. Если такой отклик не создан, т.е. не написана соответствующая процедура, то и операционная система никак не реагирует на данное событие, и оно остается безответным.

## 1. СРЕДА ДЛЯ НАПИСАНИЯ ПРОГРАММЫ НА VBA

Для того, чтобы начать изучение возможностей **VBA** необходимо запустить Excel и вывести на экран панель инструментов – **Элементы управления** (для этого в пункте Главного меню **Разработчик** надо выбрать раздел **Вставить**). В результате на экране появится новая панель инструментов – **Элементы управления** (выглядеет это должно подобно изображению на рис.1).

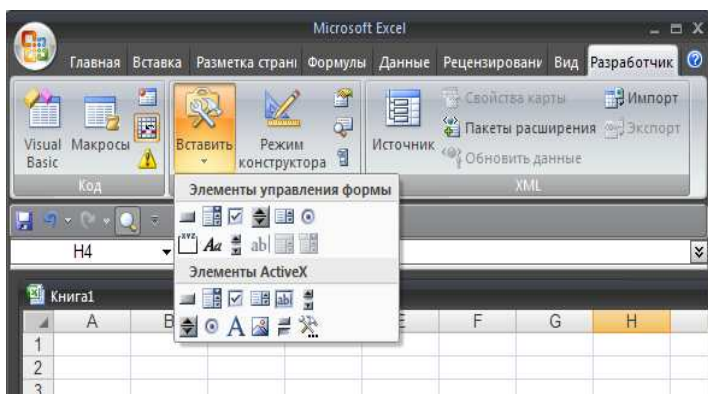


Рис. 1

Для размещения на рабочем листе какого-нибудь элемента управления (из имеющихся на указанной выше панели инструментов), необходимо перенести тот или иной элемент на рабочий лист, щелкнув на нем мышью (левой кнопкой), и далее разместить его на рабочем листе (при нажатой левой кнопке мыши). На рис.2 показано размещение командной кнопки (далее просто кнопка). Для перехода в режим конструктора необходимо выбрать пункт **Режим конструктора**.

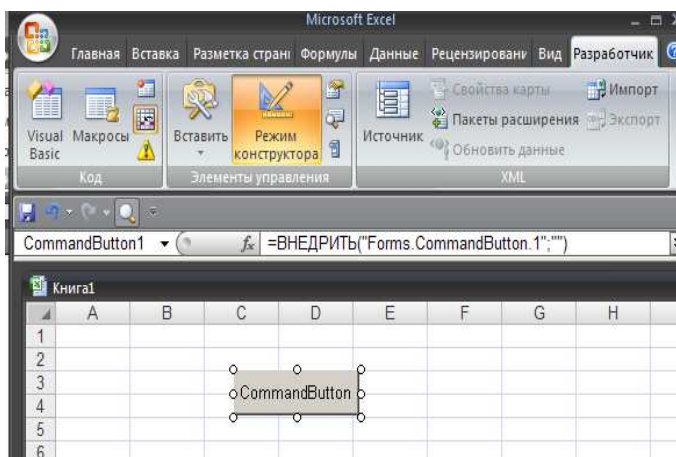


Рис.2

Важно отметить, что после размещения элемента управления на рабочем листе **Excel** этот элемент становится объектом. При этом первоначальное имя объекта назначается автоматически, но Вы сразу или в дальнейшем можете это имя изменить. Таким образом, мы создали элемент управления в режиме конструктора, и если Вы хотите воспользоваться этим элементом управления, необходимо выйти из режима конструктора (как это сделать указано выше). В результате выполненных действий на рабочем листе появилась кнопка, которая, правда, никакой полезной роли не играет. Мы можем просто по ней пощелкать без какого либо эффекта. Теперь вернемся в режим конструктора и рассмотрим некоторые свойства созданной кнопки.



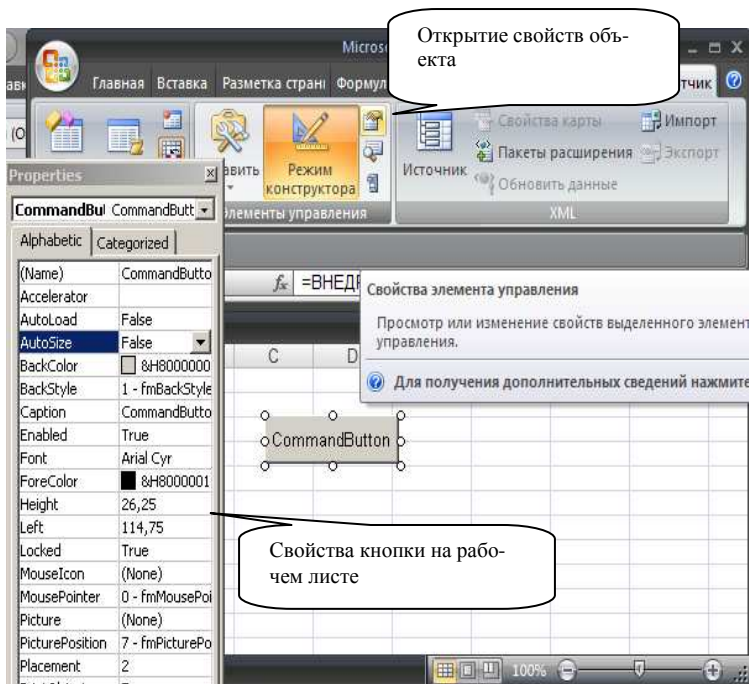


Рис. 3

Для просмотра свойств кнопки ее сначала необходимо выделить (щелкнуть мышкой), а затем нажать кнопку для открытия окна свойств объекта (рис.3). Принцип организации окна свойств следующий: левая колонка содержит название свойства, а правая значение этого свойства. Так первая строка окна свойств на рис.3 (**Name CommandButton1**) говорит о том, что свойству **Name** присваивается значение **CommandButton1**. Если мы хотим дать другое имя этой кнопке, то вместо **CommandButton1** мы должны набрать любое имя без пробелов. В качестве примера введем **Кнопка1**. Теперь выберем свойство **Caption**, которое означает надпись на кнопке (фактически это бирка или ярлык на кнопке).

Поменяйте значение этого свойства (например, на название **Кнопка на листе**), а далее поменяем цвет кнопки (сделаем кнопку не серой, а розовой). Для этого установим свойство **BackColor** с помощью мыши в розовый цвет. Можно также поменять шрифт букв – для это существует свойство **Font**. В результате кнопка на рабочем листе может принять примерно следующий вид (рис.4).

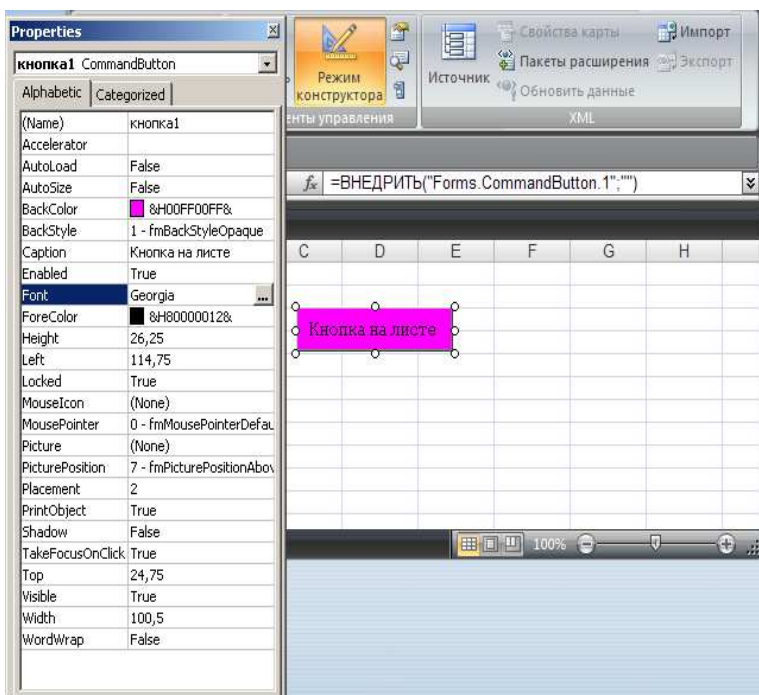


Рис. 4

Также можно с помощью свойства **Picture** вставить на кнопку картинку из графического файла.

Теперь разместим на листе другой элемент управления – **Текстовое окно** (рис.5). Действия для размещения элемента уже нам знакомы – необходимо щелкнуть на

пиктограмме элемента **Текстовое окно** (рис. 5) и далее при нажатой левой кнопке мыши разместить это элемент на рабочем листе. Установите значение свойства **Name** этого нового элемента – **ТекстовоеОкно**. Выберем цвет (свойство **BackColor** ) текстового окна голубым. Далее установим свойство **MultiLine** в значение **True**. В этом случае длинный текст в этом окне будет переноситься со строки на строку. Альтернативным вариантом является **False** – в этом случае текст будет только в одной строке.

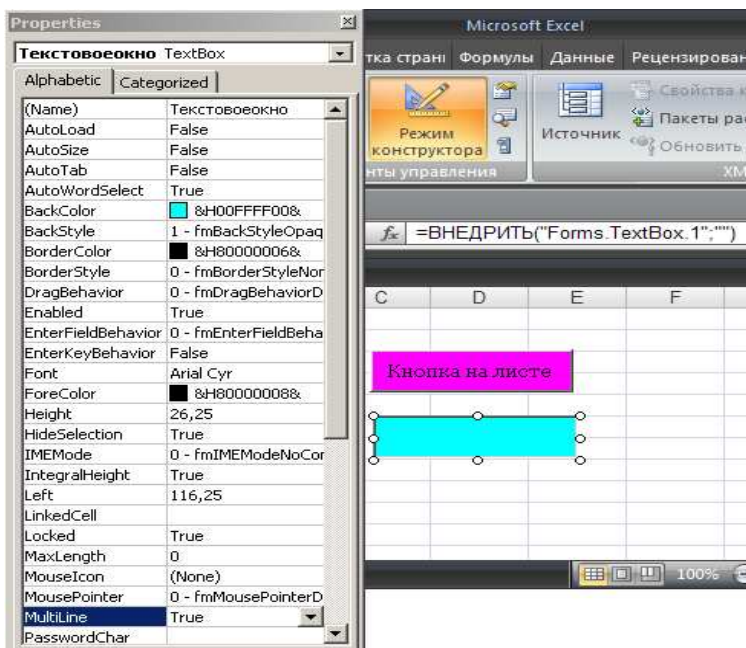


Рис. 5

Теперь перейдем к обработке событий. Наша задача сделать так, чтобы при щелчке по кнопке в текстовое окно выводился текст (например, **Пример вывода текста**).

Для начала перейдите в режим конструктора. После этого необходимо перейти в среду для написания кода (или текста) программы. Для этого дважды щелкните по созданной кнопке (**Кнопка1**).

Другой способ перехода в среду для написания кода – меню **Разработчик**, далее пункты **Макросы** и **Visual Basic**). В результате перед Вами появится новая среда редактора кода (рис.6). Здесь требуется сначала сделать выбор объекта, а затем для выбранного объекта можно написать содержимое процедуры, выполняемой при определенном событии.

Наша задача – обеспечить вывод в текстовое окно сообщения при щелчке по кнопке. Для этого в левом списке (рис.6) необходимо выбрать **Кнопка1** (объект который нас интересует), а затем в правом **Click** (другими словами щелчок по кнопке).

На рис.6 показан результат наших действий – на экране появляется окно процедуры, которая будет выполняться при щелчке по кнопке.

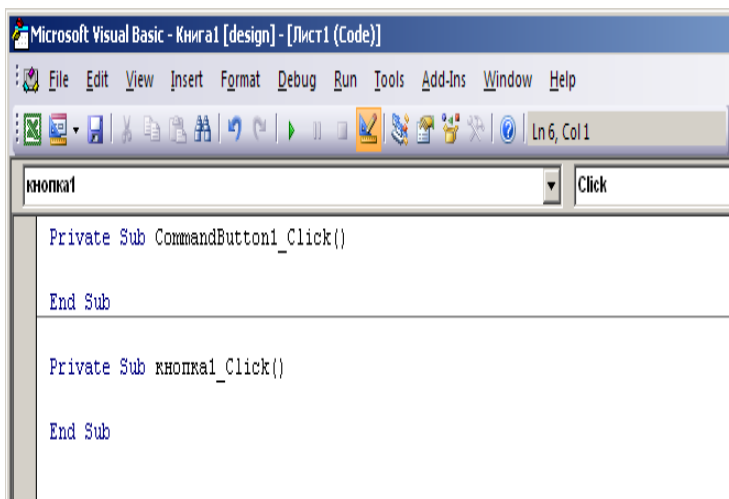


Рис. 6

Первое слово **Private** говорит о том, что это внутренняя процедура только для данного листа. Поэтому для других листов эта процедура недоступна. Пока данный параметр мы менять не будем, заметим только что альтернативным вариантом является **Public** (общая процедура – в этом случае доступна для других листов). Слово **Sub** говорит о том, что это процедура. Дальнейшая запись **Кнопка1\_Click( )** – имя процедуры. Это имя формируется автоматически из двух составляющих – **Кнопка1** (имя кнопки) и **Click** (действие – щелчок по кнопке). Это так называемая **предопределенная** процедура – она существует и выполняется всегда - когда мы щелкаем по кнопке.

Запись **End Sub** говорит о завершении рассматриваемой процедуры. Теперь, если мы между этими двумя строками напишем программный фрагмент, то этот фрагмент будет автоматически выполняться при щелчке по кнопке (рис.7).

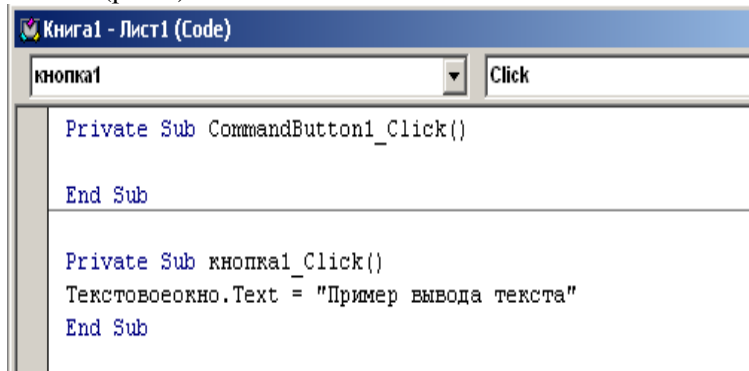


Рис. 7

Прокомментируем новые моменты, присутствующие на рис.7. Словосочетание **ТекстовоеОкно.Text** говорит о том, что выбирается объект с именем **ТекстовоеОкно** и рассматривается его свойство **Text** (фактически это –

текст, который располагается в этом текстовом окне). Знак равенства и последующая фраза говорят о том, что рассматриваемому свойству присваивается значение **Пример вывода текста**. Теперь перейдем из окна редактора **Visual Basic** в **Excel** (рис.8). В результате, если щелкнуть по кнопке, то эффект от этого действия будет аналогичен рис.9.

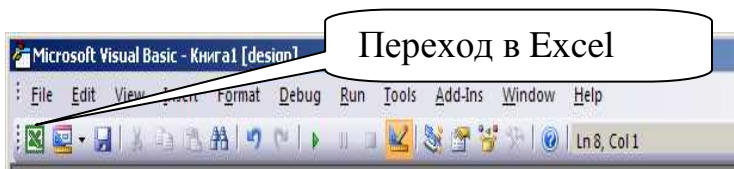


Рис.8

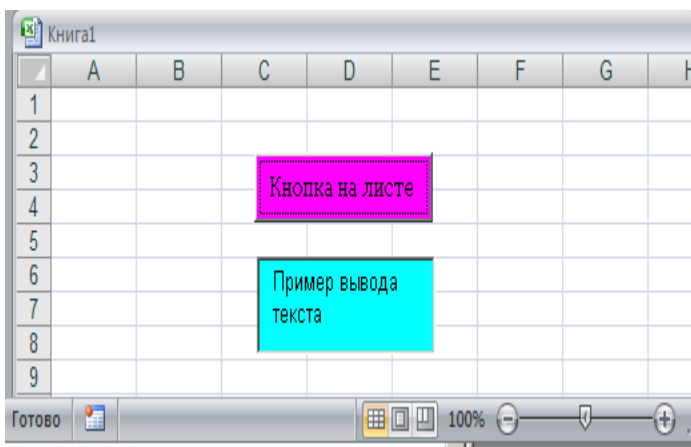


Рис. 9.

Рассмотрим теперь следующий элемент управления – **поле со списком** (рис.10). Этот элемент применяется для хранения списка значений ,последующего выбора элемента списка и передачи его в переменную, элемент интерфейса и т.д. В этом элементе до его активизации (щелчком

мышью) всегда отображается только одно значение списка. После активизации **поля со списком** список раскрывается и из него можно выбрать необходимый элемент. Кроме свойства **Name** у данного элемента есть несколько важных свойств. Рассмотрим их для дальнейшего использования.

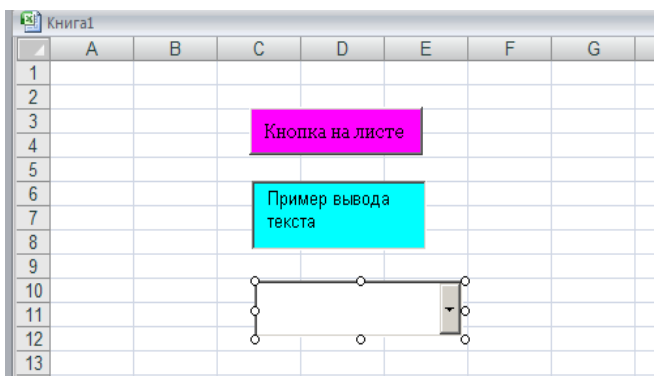


Рис. 10

Свойство **ListIndex** содержит номер текущего элемента списка (того, который подсвечен – на нем щелкнули мышкой). Нумерация элементов списка начинается с нуля. Фактически по значению этого свойства можно определить, какой элемент списка ему соответствует. Свойство **ListCount** содержит число элементов списка, а свойство **Text**, выбранный в списке элемент.

Теперь рассмотрим наиболее используемые методы работы со списком. Метод **Clear** удаляет все элементы из поля со списком. Синтаксис использования:

**ИмяОбъекта . Clear .**

Метод **AddItem** добавляет элемент в список. Синтаксис использования:

**ИмяОбъекта. AddItem элемент.**

Метод **RemoveItem** – удаление элемента из списка.  
Синтаксис использования:

**RemoveItem** номер удаляемого элемента .

Наша задача теперь разработать на рабочем листе следующий программный фрагмент (рис. 11).

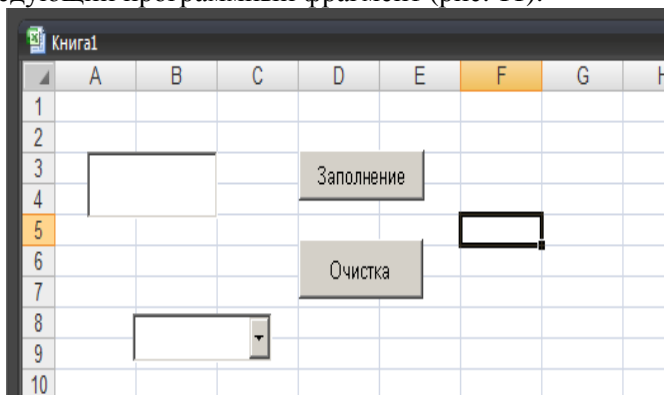


Рис. 11

Необходимо обеспечить выполнение следующих действий:

- по щелчку по кнопке **Заполнение** список заполняется строками – яблоко, груша, слива, дыня, арбуз;
- по щелчку по **полю со списком** выбранный элемент помещается в текстовое окно слева;
- по щелчку по кнопке **Очистка** текстовое окно и **поле со списком** очищаются – становятся пустыми.

Для заполнения поля со списком по щелчку по кнопке **Заполнение** необходимо процедуру **Заполнение\_Click( )** оформить так как показано на рис. 12.



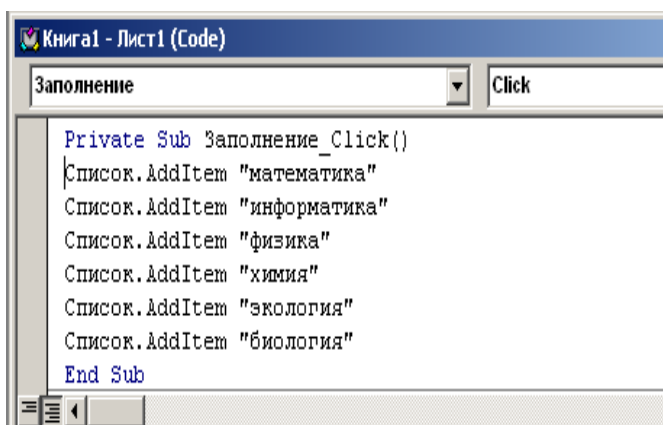


Рис. 12

Далее для заполнения текстового окна по щелчку по **полю со списком** (при выборе того или иного элемента списка) необходимо процедуру **Список\_Click ( )** оформить так как показано на рис.13.

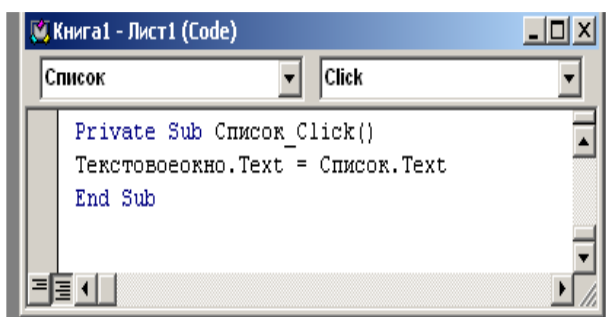


Рис. 13

Теперь для очистки списка и текстового окна необходимо написать обработку события **Щелчок** по кнопке **Очистка** (рис. 14).

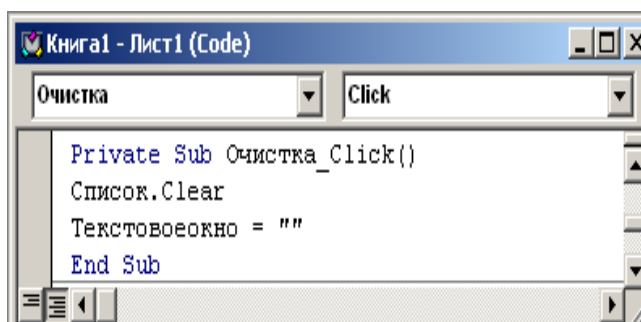


Рис. 14

Теперь разрабатываемый программный фрагмент на рабочем листе готов и Вы можете попробовать его в работе (выйдя из режима конструктора).

## 2. ПРИМЕРЫ ПРОГРАММИРОВАНИЯ НА VBA

В этом разделе рассматриваются примеры, поясняющие возможности VBA.

### Пример №1.

Необходимо разработать программный фрагмент: на рабочем листе располагаются две кнопки (с названиями **Старт** и **Стоп**) и текстовое окно. При щелчке по кнопке **Старт** в текстовом окне начинает постоянно отображаться текущее время. После щелчка по кнопке **Стоп** постоянное обновление времени останавливается. На рис. 15 показан вид рабочего листа с необходимыми элементами управления.

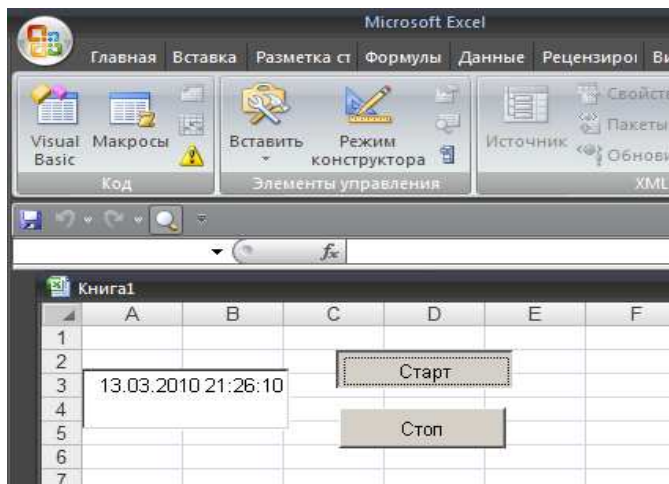


Рис. 15

Первый шаг – введем переменную (индикатор остановки обновления времени). Эта переменная должна быть известна как процедуре обработки щелчка по кнопке **Старт**, так и процедуре обработки щелчка по кнопке

**Стоп.** В среде **VBA** существует общая область, где можно описывать переменные (рис. 16).

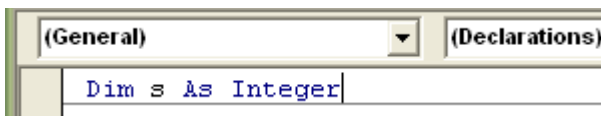


Рис. 16

Строка программы в окне на рис.16 означает следующее: **Dim** – определить (это ключевое слово **Visual Basic**), далее **s** – это имя переменной, **As Integer** - определение данной переменной как целой. Далее на рис.17 приведена процедура обработки события – щелчка по кнопке **Старт**.

**ИСПРАВИТЬ!!!!!!**

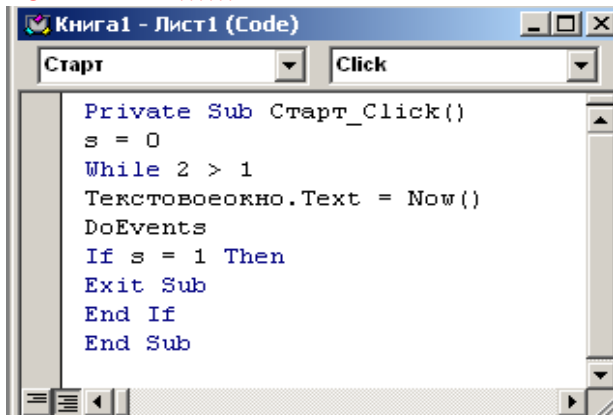


Рис. 17

Идея заключается в том, что переменной **s** присваивается изначально значение 0. Далее в цикле мы будем постоянно выводить текущее время (текущее время постоянно изменяется – поэтому время будет постоянно обновляться) в текстовое окно. При этом, если значение переменной **s** изменится (с 0 на 1), то вывод текущего вре-

мени прекращается. Это изменение происходит при щелчке по кнопке **Стоп**. Но сначала о процедуре на рис. 17.

Здесь выполняется цикл:

от строки **While 2 > 1**

до **Wend**.

А именно, пока выполняется условие (**While 2 > 1**) выполняются все строки программы до строки **Wend** (это ключевое слово помечает нижнюю границу цикла). Затем программа возвращается опять к строке **While 2 > 1**. Далее опять выполняются все строки программы до строки – **Wend**. И, так как условие всегда выполняется, то цикл может выполняться бесконечное число раз. В строке **ТекстовоеОкно.Text = Now( )** происходит обращение к внутренней функции Visual Basic – **Now()**. Эта функция возвращает текущую дату и текущее время, которые затем присваиваются свойству **Text** объекта **ТекстовоеОкно**. Далее по рис.17 следует – **DoEvents**. Это функция Visual Basic, которая позволяет переключаться на обработку других событий (в данном случае обработать щелчок по кнопке **Стоп**). Далее в программном фрагменте располагается условие:

**If s = 1 Then**

**Exit Sub**

**End If**

Эта конструкция означает, что если **s = 1**, то выполняются все строки до **End If**. В нашей ситуации здесь одна строка **Exit Sub**, которая означает выход из процедуры. Значение **s = 1** может установить только щелчок по кнопке **Стоп** (рис. 18).

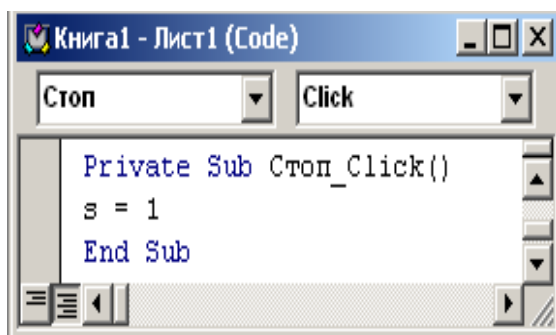


Рис. 18

После запуска (выхода из режима конструктора и щелчка по кнопке **Старт**) рабочий лист должен иметь примерно следующий вид (рис. 15).

### **Пример№2.**

Рассмотрим теперь программный пример с использованием элемента управления – **список**. На рис.19 показан вид рабочего листа программного фрагмента, который нам предстоит разработать.

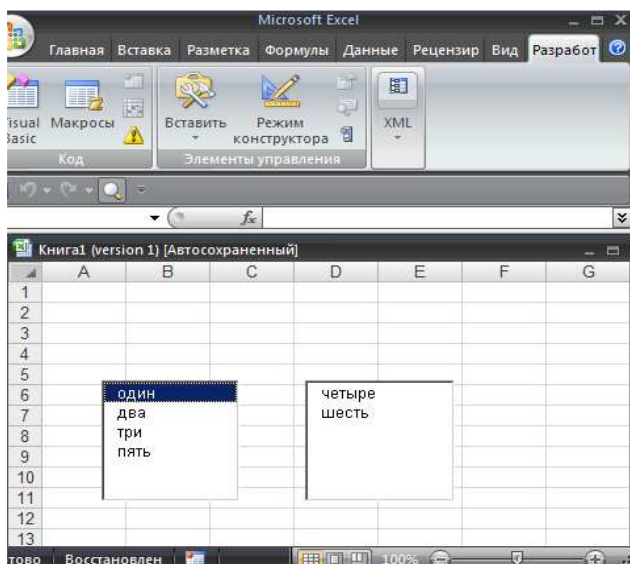


Рис. 19

Основные свойства и методы элемента **список** аналогичны свойствам и методам элемента **поле со списком**. Как показано на рис.19, необходимо разместить на листе два списка. Заполнить списки несколькими словами примерно поровну. Теперь при двойном щелчке мышкой на том или ином элементе этот элемент должен перемещаться в соседний список, а из текущего удаляться. Заполнение списков можно выполнить в предопределенной процедуре **Worksheet\_Activate()**.

Эта процедура всегда выполняется, когда происходит переключение на текущий лист с другого листа. На рис.20 показана данная процедура.

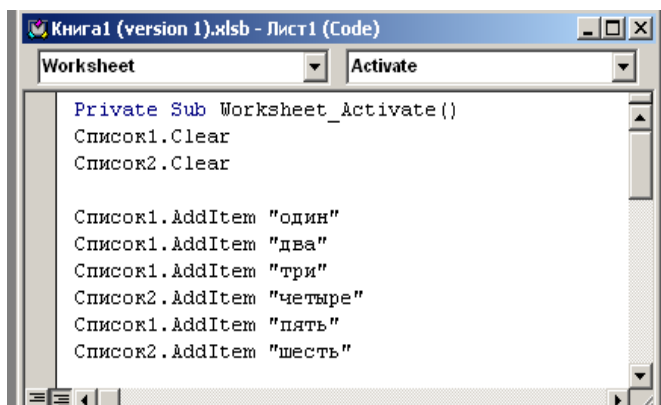


Рис. 20

Здесь мы сначала очищаем списки, а затем заполняем их данными. Далее нам необходимо написать процедуры, которые будут автоматически выполняться при двойном щелчке по левому и правому списку. Тексты этих процедур приведены на рис.21 и рис.22.

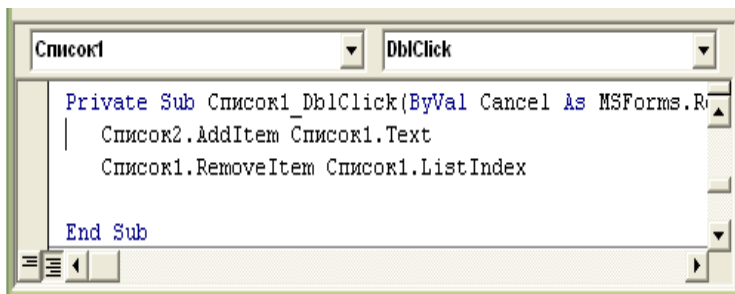


Рис. 21

Теперь программный фрагмент готов и Вы можете его протестировать. Заполнение списков происходит при событии – активизация листа. Это событие происходит тогда, когда производится переход на этот лист с другого листа. Поэтому для работы программы необходимо сначала



ла активизировать какой-нибудь другой лист, а затем вернуться на рассматриваемый.

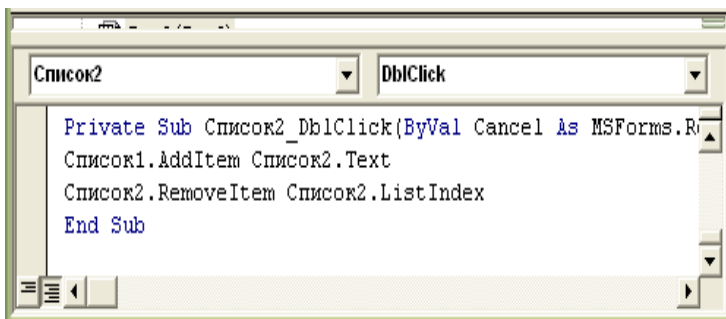


Рис. 22

### Пример №3.

Рассмотрим теперь свойство, которое позволяет разрешать и запрещать доступ к элементам управления. У всех элементов управления есть свойство **Enabled**. Когда это свойство принимает значение **True**, то доступ к данному элементу управления разрешен. В противном случае (свойство принимает значение **False**) доступ к данному элементу запрещен. Наша задача – разместить на листе две кнопки. Далее щелчок по одной из кнопок приводит к тому, что эта кнопка становится недоступной, а к другой кнопке наоборот доступ разрешается. Поэтому введите следующие функции обработки событий – щелчков по кнопкам:

```
Private Sub Кнопка1_Click()
    Кнопка1.Enabled = False
    Кнопка2.Enabled = True
End Sub
```

```
Private Sub Кнопка2_Click()
    Кнопка2.Enabled = False
```

Кнопка1.Enabled = True

End Sub

Выйдите из режима конструктора, запустите программу на выполнение и пощелкайте по кнопкам.

Теперь рассмотрим работу с линейкой прокрутки. Откройте новый лист, на котором расположите **линейку прокрутки** и **текстовое окно** (рис.23).

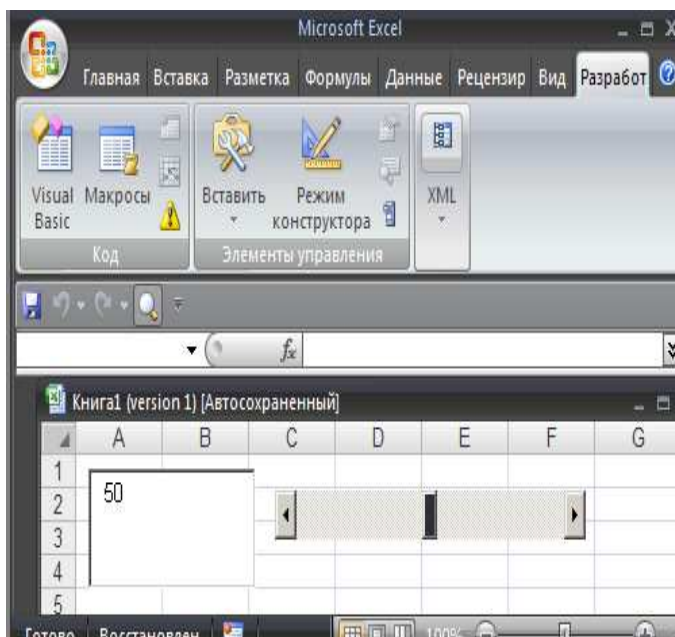


Рис. 23

У линейки прокрутки имеются следующие свойства:

- **Value** – значение, соответствующее положению ползунка на линейке прокрутки ;
- **Max** – значение, соответствующее правому крайнему положению ползунка на линейке прокрутки;
- **Min** – значение, соответствующее левому крайнему положению ползунка на линейке прокрутки;

- **SmallChange** – значение, соответствующее изменению **Value** при щелчках по стрелкам линейки прокрутки;
- **LargeChange** – значение, соответствующее изменению **Value** при щелчках по полосе линейки прокрутки.

Установим **Min** = 0, **Max** =100, **SmallChange** = 1, **LargeChange** = 5. Далее введите функцию, которая выполняется при передвижении движка линейки прокрутки:

```
Private Sub Линейка_Change()  
ТекстовоеОкно.Text = Линейка.Value  
End Sub
```

Выйдите из режима конструктора и посмотрите на результат работы программы.

Рассмотрим еще один пример. Задача заключается в том, чтобы разместить на листе кнопку, и эта кнопка при попытке навести на нее курсор должна автоматически перемещаться на другое место на рабочем листе. Для этого перейдем в режим конструктора и далее в процедуру:

### **Кнопка1\_MouseMove().**

Эта процедура (рис.24) автоматически выполняется при наведении курсора мыши на кнопку. В приведенном тексте используется внутренняя функция Visual Basic – **rnd**. Эта функция при обращении к ней выдает случайное число в интервале от 0 до 1. В результате при попадании курсора мыши на кнопку происходит вызов процедуры **Кнопка1\_MouseMove()**. Итог выполнения данной процедуры – -положение кнопки меняется (**Left** – левая граница кнопки, **Top** – смещение кнопки по вертикали).

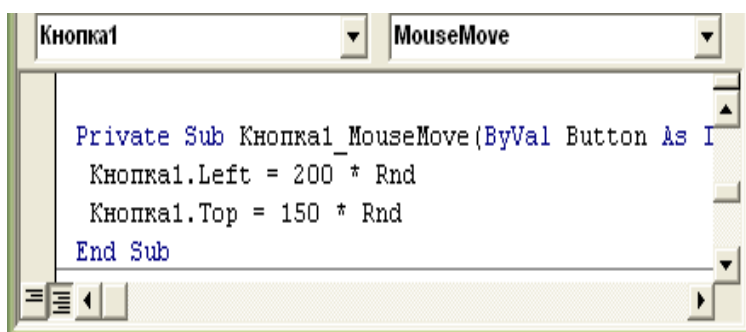


Рис. 24

### 3. СВОЙСТВА И МЕТОДЫ ОБЪЕКТОВ EXCEL

До настоящего времени мы рассматривали возможности Visual Basic без особой связи с информацией на рабочем листе. В этом разделе мы познакомимся с объектами VBA, которые позволяют работать с информацией в книгах Excel. Для работы с ячейками Excel в VBA существует объект **Range** – диапазон. Если установить в качестве параметра объекта **Range** значение **A1**, то тем самым мы выбрали объект – ячейка **A1**. Для изменения содержания этой ячейки у объекта **Range** есть свойство **Value**. Буквально оно означает значение ячейки (или группы ячеек). Запись

**Range("A1").Value = "Фамилия"**

позволяет программно внести в эту ячейку слово – Фамилия). Также, если в качестве параметра **Range** использовать обозначение другой ячейки, то можно обратиться и к ней.

Необходимо разработать на рабочем листе следующий программный фрагмент. При щелчке по кнопке на рабочем листе должны заполняться информацией следующие ячейки:

**A1** – Фамилия,

**A2** – Имя,

**A3** – Адрес.

Для этого в окне кода программы обработки щелчка по кнопке необходимо написать следующие строки:

**Range("A1").Value = "Фамилия"**

**Range("A2").Value = "Имя"**

**Range("A3").Value = "Отчество".**

Альтернативным способом работы с ячейками является объект **Cells**. Синтаксис использования этого объекта – **Cells ( номер строки, номер столбца)**.

Так ячейку **A2** можно выбрать двумя эквивалентными способами:

**Range(“A2”)      или      Cells(2,1) .**

Как известно, рабочая книга – это совокупность рабочих листов. Для работы с листами можно использовать коллекцию объектов – листов текущей книги следующим образом:

**Worksheets ( номер листа в книге).**

В качестве примера напомним фрагмент, который устанавливает значения двух ячеек на разных листах (может выполняться по щелчку по кнопке):

`Worksheets(3).Cells(5, 1).Value = 5`

`Worksheets(2).Range(“C5”).Value = “Строка”`

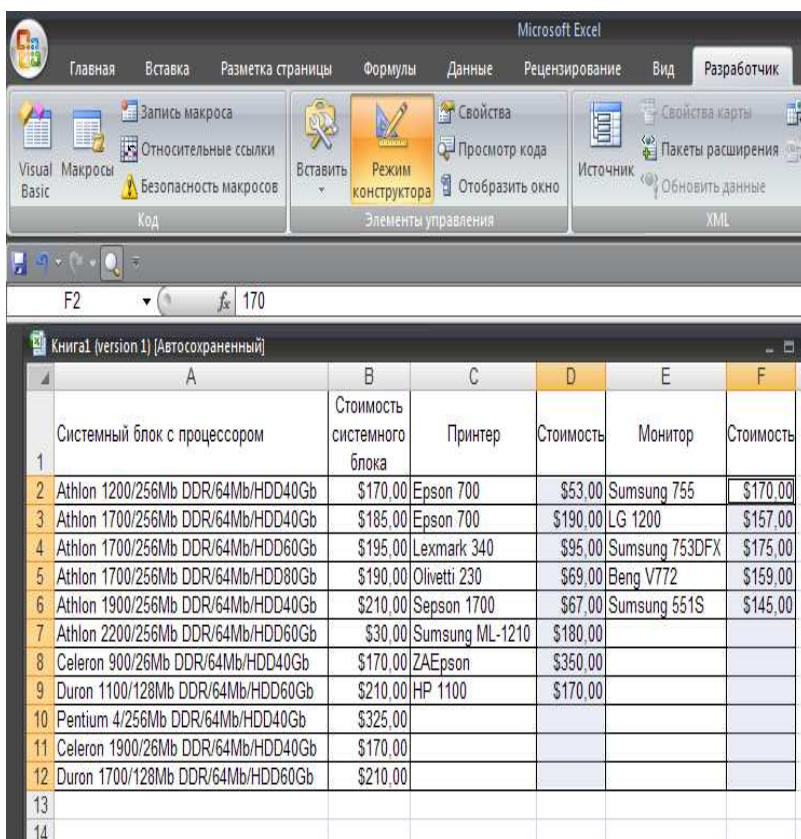
Теперь после того, как мы познакомились с простыми примерами использования VBA, перейдем к более сложной задаче.

## **ПРИМЕР №1**

### **РАЗРАБОТКА БЛАНКА ЗАКАЗА**

Требуется разработать удобный бланк заказа для обслуживания покупателей в компьютерном салоне.

Необходимо подобрать необходимую конфигурацию компьютера из возможных вариантов по прайс-листу. Для упрощения ситуации будем считать, что комплектация компьютера состоит максимум из 3 компонентов: системного блока, монитора и принтера. При этом различных системных блоков, мониторов и принтеров много. Эта номенклатура отражается на одном из листов книги – лист2 с названием **Прайс-лист**. Фрагмент этого листа представлен на рис.25.

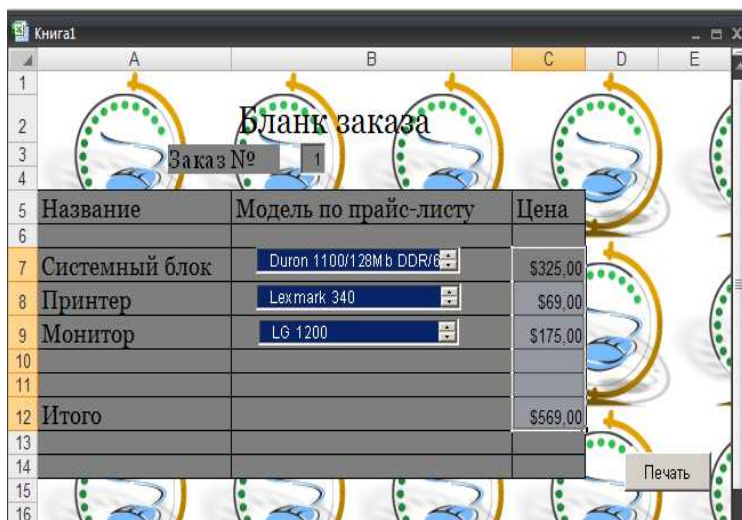


The screenshot shows the Microsoft Excel interface with the 'Разработчик' (Developer) ribbon selected. The active sheet is 'Книга1 (version 1) [Автосохраненный]'. The table below is displayed in the worksheet.

	A	B	C	D	E	F
	Системный блок с процессором	Стоимость системного блока	Принтер	Стоимость	Монитор	Стоимость
1						
2	Athlon 1200/256Mb DDR/64Mb/HDD40Gb	\$170,00	Epson 700	\$53,00	Sumsung 755	\$170,00
3	Athlon 1700/256Mb DDR/64Mb/HDD40Gb	\$185,00	Epson 700	\$190,00	LG 1200	\$157,00
4	Athlon 1700/256Mb DDR/64Mb/HDD60Gb	\$195,00	Lexmark 340	\$95,00	Sumsung 753DFX	\$175,00
5	Athlon 1700/256Mb DDR/64Mb/HDD80Gb	\$190,00	Olivetti 230	\$69,00	Beng V772	\$159,00
6	Athlon 1900/256Mb DDR/64Mb/HDD40Gb	\$210,00	Sepson 1700	\$67,00	Sumsung 551S	\$145,00
7	Athlon 2200/256Mb DDR/64Mb/HDD60Gb	\$30,00	Sumsung ML-1210	\$180,00		
8	Celeron 900/26Mb DDR/64Mb/HDD40Gb	\$170,00	ZA Epson	\$350,00		
9	Duron 1100/128Mb DDR/64Mb/HDD60Gb	\$210,00	HP 1100	\$170,00		
10	Pentium 4/256Mb DDR/64Mb/HDD40Gb	\$325,00				
11	Celeron 1900/26Mb DDR/64Mb/HDD40Gb	\$170,00				
12	Duron 1700/128Mb DDR/64Mb/HDD60Gb	\$210,00				
13						
14						

Рис.25

Здесь столбцы **A** и **B** содержат соответственно описание системного блока и его стоимость. Следующие два столбца – название принтера со стоимостью, и далее монитор с ценой. Необходимо на рабочем листе1 создать удобный электронный бланк заказа с использованием информации из прайс-листа. Вид этого бланка представлен на рис. 26.



Название	Модель по прайс-листу	Цена
Системный блок	Duron 1100/128Mb DDR/6	\$325.00
Принтер	Lexmark 340	\$69.00
Монитор	LG 1200	\$175.00
Итого		\$569.00

Рис. 26

Расширим столбцы **A** и **B**, а затем в ячейку **B2** введем текст – **Бланк заказа**. Обеспечим необходимое форматирование этого текста и далее разместим на рабочем листе элемент надпись. Мы ранее не рассматривали элемент управления **надпись**, но он достаточно прост и его роль заключается в создании поясняющего текста к какому-либо другому объекту (рис.27).





Рис. 27

Поменяйте значения свойств элемента **Надпись** следующим образом: свойство **Caption** установите – **Заказ №**, а свойство **BackColor** сделайте серым. Далее разместите на листе текстовое окно для отображения номера заказа (рис.27).

Имя (свойство **Name**) этого тестового окна можно оставить – **TextBox1**. Установите также серый фон текстового окна (с помощью свойства **BackColor**). Далее установите также свойство **AutoSize** этого окна в значение **True**. В этом случае ширина текстового поля будет автоматически увеличиваться при вводе символов.

Теперь заполните текстом (как показано на рис. 26) содержимое ячеек **A5, B5, C5, A7, A8, A9, A12**.

После этого уберите сетку с экрана (Кнопка «Office», раздел **Дополнительно** и далее следует убрать отметку с пункта **Показывать сетку**).

Для прямоугольного диапазона **A5 – C14** установите внешние и внутренние границы так, как показано на рис.26. Также установите серую заливку для рассматриваемого диапазона и выберите подложку для листа (Пункт меню **Разметка страницы – Подложка**).

Теперь разместите три элемента управления – **Поле со списком**. Имена этим спискам дайте **Список1, Список2 и Список3**.

Теперь мы подошли к написанию программного кода. Для этого перейдите в редактор **VBA**.

Необходимо, чтобы при открытии книги списки автоматически заполнялись содержимым листа – **Прайс-лист**.

Существует процедура **Workbook\_Open()**, которая автоматически выполняется при открытии книги. Эта процедура выполняется над объектом **Workbook** (текущая рабочая книга).

На рис. 28 показано окно редактора кода с заголовком данной процедуры и далее приводится подробное описание с комментариями.

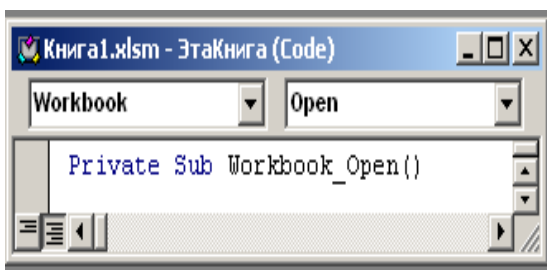


Рис. 28

```
Private Sub Workbook_Open( )
```

```
    'Worksheets(1) – первый лист книги,
```

```
    Список1 – первый список,
```

```
    'Clear – метод, заключающийся в очистке списка.
```

```
    Worksheets(1).Список1.Clear
```

```
    ' Далее производится подсчет количества видов  
    системных блоков.
```

```
    N = 0
```

```
    While Worksheets(2).Cells(N + 2, 1).Value <> ""
```

```
        N = N + 1
```

```
    Wend
```

‘ **N** – переменная, в которой подсчитывается количество записей в первом столбце **Прайс-листа**.

В качестве условия используется следующая конструкция

```
    ' Worksheets(2).Cells(N + 2, 1).Value <> "".
```

‘ Здесь Worksheets(2) – второй лист книги и этот лист – **Прайс-лист**.

‘ Cells (...) – функция выбора ячейки.

‘ В скобках указывается номер строки и номер столбца.

‘ Знак < > обозначает не равно.

‘ Далее "" – обозначает, что ячейка пустая.

Таким образом, эта запись обозначает выполнение цикла, пока значение в очередной ячейки в списке системных блоков не окажется пустым. Т.е. производится подсчет количества системных блоков в прайс-листе.

For i = 2 To N + 2

Worksheets(1).Список1.AddItem Worksheets(2).Cells(i, 1).Value  
Next

‘ В записи **Worksheets(1).Список1** используется список **Список1** на первом рабочем листе. Метод **AddItem** добавляет в список строку. Далее через пробел записывается строка, которая добавляется, а именно значение

‘ (свойства **Value**) ячейки ( **Cells( i, 1 )** ) из листа **Worksheets(2)**.

‘ В следующем фрагменте производится аналогичное заполнение второго списка.

Worksheets(1).Список2.Clear

N = 0

While Worksheets(2).Cells(N + 2, 3).Value <> ""

N = N + 1

Wend

For i = 2 To N + 2

Worksheets(1).Список2.AddItem Worksheets(2).Cells(i, 3).Value  
Next

‘В следующем фрагменте производится заполнение третьего списка.

```
Worksheets(1).Список3.Clear
N = 0
While Worksheets(2).Cells(N + 2, 5).Value <> ""
    N = N + 1
Wend
For i = 2 To N + 2
Worksheets(1).Список3.AddItem Worksheets(2).Cells(i, 5).Value
Next
End Sub
```

Таким образом, процедура **Workbook\_Open()** обеспечивает заполнение списков данными со второго листа (из прайс-листа).

Далее на рис.29 описывается процедура, выполняемая при щелчке по первому полю со списком (когда мы выбираем из списка ту или иную строку).

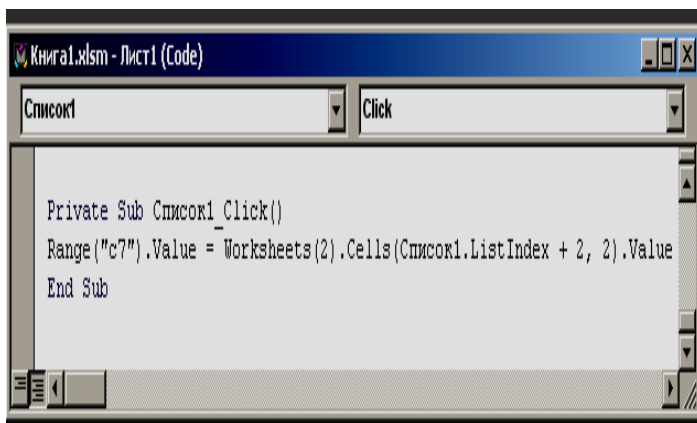


Рис. 29

На рис.29 строка `Range("c7").Value=Worksheets(2).Cells(Список1.ListIndex + 2, 2). Value` позволяет заполнить

ячейку **C7** ценой системного блока. А именно значение ячейки **C7** (в ней должна располагаться цена на первом листе) заполняется информацией из прайс-листа. При этом **Список1.ListIndex** – индекс элемента, который выделен щелчком мыши. Щелчки по двум другим спискам приводят к аналогичным действиям (они приводят к заполнению цен монитора и принтера). Эти процедуры приводятся далее.

```
Private Sub Список2_Click()  
Range("c8").Value = Work-  
sheets(2).Cells(Список2.ListIndex + 2, 4).Value  
End Sub
```

```
Private Sub Список3_Click()  
Range("c9").Value = Work-  
sheets(2).Cells(Список3.ListIndex + 2, 6).Value  
End Sub
```

Таким образом, мы получили автоматизированное заполнение бланка заказа. Осталось только в ячейку **C12** поставить формулу для вычисления суммы: **СУММ(C7:C9)**.

Теперь рассмотрим действия при нажатии на кнопку **Печать**. Требуется, чтобы на 3-м листе создавалась печатная (автоматически заполняемая) форма. Вид этой формы может иметь примерно следующий вид (рис. 30).

	A	B	C	D	E
1					
2					
3		Заказ №			
4					
5					
6					
7					
8					
9	№	Название	Цена		
10					
11					
12					
13					
14					
15					
16					
17					

Рис. 30

Первая задача чисто техническая – обеспечить форматирование листа подобное рис.30. Далее необходимо написать процедуру, выполняемую по щелчку по кнопке **Печать**. Для этого перейдите в режим конструктора и дважды щелкните по кнопке **Печать**. В этом случае Вы автоматически попадаете в процедуру, обрабатывающую щелчок по кнопке, и далее показан текст данной процедуры. В этом тексте, как и ранее, название процедуры – это совокупность имени кнопки (**Печать**) и действия – щелчок **Click**.

```
Private Sub Печать_Click()
Worksheets(3).Cells(10, 2).Value = Work-
sheets(1).Cells(7, 1).Value + " "
+ Список1.Text
```

‘На третьем листе будет автоматически формироваться печатная форма бланка. Ячейка на пересечении 10-й строки и второго столбца будет содержать информацию

о системном блоке. В эту ячейку записывается информация с первого листа – о названии системного блока.

```
Worksheets(3).Cells(10,3).Value =  
Worksheets(2).Cells(Список1.ListIndex + 2,2).Value
```

‘Ячейка на пересечении 10-й строки и третьего столбца будет содержать информацию о стоимости системного блока. В эту ячейку записывается информация из второго листа с учетом выбора из списка.

```
Worksheets(3).Cells(11, 2).Value = Work-  
sheets(1).Cells(8, 1).Value + " " + Список2.Text
```

```
Worksheets(3).Cells(11, 3).Value =  
Worksheets(2).Cells(Список2.ListIndex + 2,4).Value
```

‘ Ячейка на пересечении 11-й строки и второго столбца будет содержать информацию о принтере. В эту ячейку записывается информация с первого листа – название принтера.

‘ Ячейка на пересечении 11-й строки и третьего столбца будет содержать информацию о стоимости принтера. В эту ячейку записывается информация из второго листа, соответствующая выбранному названию из списка.

```
Worksheets(3).Range("c3").Value = TextBox1.Text  
Worksheets(3).Activate  
End Sub
```

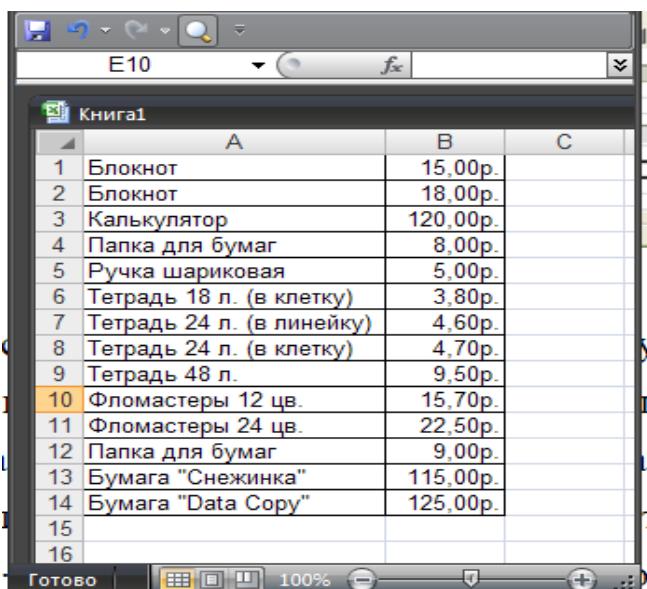
В приведенной процедуре не предусматривается заполнения параметров монитора и доработать данную возможность следует самостоятельно.

## ПРИМЕР №2

### ЗАЯВКА НА ПОСТУПЛЕНИЕ ТОВАРОВ

Требуется выполнить задание, близкое по содержанию предыдущему, а именно необходимо разработать удобное приложение для составления заявки на канцтовары.

Создайте новую книгу **Excel** и на втором рабочем листе создайте список некоторых товаров с их ценами (рис. 31).



The screenshot shows an Excel window with a single worksheet named 'Книга1'. The active cell is E10. A table is displayed with the following data:

	A	B	C
1	Блокнот	15,00р.	
2	Блокнот	18,00р.	
3	Калькулятор	120,00р.	
4	Папка для бумаг	8,00р.	
5	Ручка шариковая	5,00р.	
6	Тетрадь 18 л. (в клетку)	3,80р.	
7	Тетрадь 24 л. (в линейку)	4,60р.	
8	Тетрадь 24 л. (в клетку)	4,70р.	
9	Тетрадь 48 л.	9,50р.	
10	Фломастеры 12 цв.	15,70р.	
11	Фломастеры 24 цв.	22,50р.	
12	Папка для бумаг	9,00р.	
13	Бумага "Снежинка"	115,00р.	
14	Бумага "Data Copy"	125,00р.	
15			
16			

Рис. 31.

Далее на первом рабочем листе (рис. 32) должна быть расположена удобная форма для ввода информации о заказе.



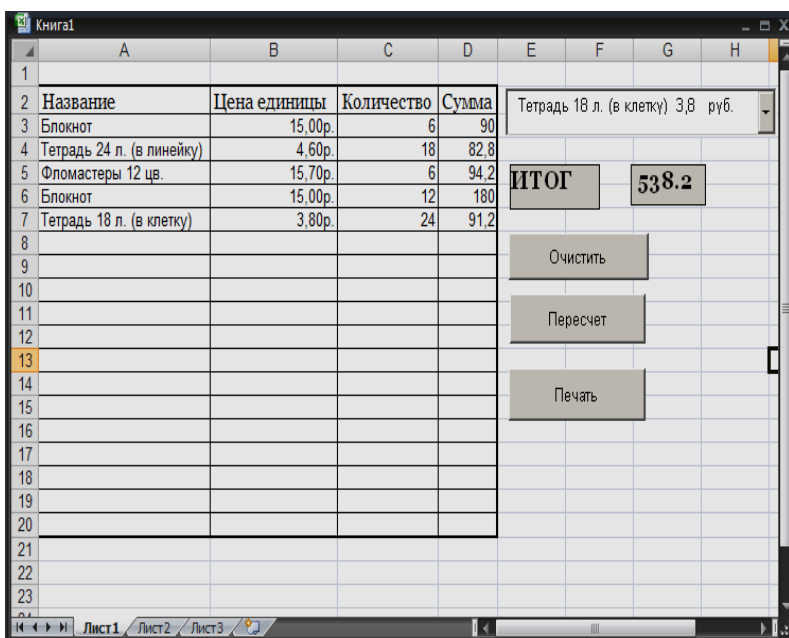


Рис. 32

Действия при выполнении этой задачи должны быть следующими. Во-первых уберем сетку с экрана ( меню **Сервис**, затем **Параметры**). Далее оформим столбцы **A**, **B**, **C** и **D** следующим образом: установим внешние границы, установим внутренние границы, введем названия в ячейки **A2**, **B2**, **C2** и **D2**. Далее расположим поле со списком ( **Name – Spisok1**), две метки – **Label1** и **Label2**, три кнопки **Очистить**, **Пересчет** и **Печать** (рис.33).

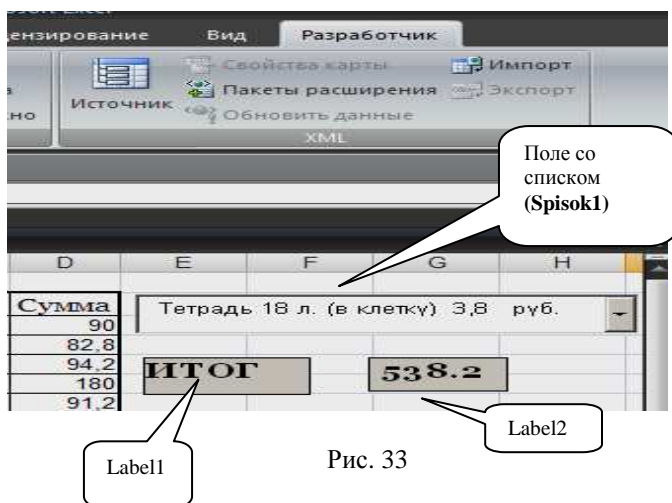


Рис. 33

Теперь наша задача – сделать так, чтобы **Spisok1** автоматически заполнялся при открытии книги. Для этого процедуру, автоматически, выполняемую при открытии книги, оформите следующим образом:

```
Private Sub Workbook_Open()
Worksheets(1).Spisok1.Clear
    ‘ Очистка списка. И далее подсчет
    ‘ количества записей на втором листе – в прайс-
    листе.
```

```
N = 0
```

```
While Worksheets(2).Cells(N + 1, 1).Value <> ""
```

```
    N = N + 1
```

```
Wend
```

```
‘Заполнение списка
```

```
For i = 1 To N
```

```
    a1 = Worksheets(2).Cells(i, 1).Value
```

```
    a2 = Worksheets(2).Cells(i, 2).Value
```

```
    a = a1 & " " & a2 & " " & " руб."
```

```
    Worksheets(1).Spisok1.AddItem a
```

Next  
End Sub

Теперь наша задача сделать так, чтобы при щелчке по строке списка эта информация фиксировалась в очередной строке первого рабочего листа. Мы таким способом включаем в заявку очередную строку. Поэтому оформим процедуру, обслуживающую щелчок по полю со списком:

```
Private Sub Spisok1_Click()  
    ' Подсчет в переменной N заполненных строк бланка  
заказа  
    N = 0  
    While Worksheets(1).Cells(N + 3, 1).Value <> ""  
        N = N + 1  
    Wend  
    Worksheets(1).Cells(N + 3, 1) = Work-  
sheets(2).Cells(Spisok1.ListIndex + 1, 1).Value  
    Worksheets(1).Cells(N + 3, 2) = Work-  
sheets(2).Cells(Spisok1.ListIndex + 1, 2).Value  
    ' Далее используется функция InputBox для ввода  
значения.  
    ' А именно то что мы вводим заносится в перемен-  
ную qw.  
    qw = InputBox("Введите количество", "Ввод числа",  
1)  
    Worksheets(1).Cells(N + 3, 3).Value = qw  
    ' Далее производится вычисление суммы по позиции  
товара.  
    Worksheets(1).Cells(N + 3, 4).Value = qw * Work-  
sheets(1).Cells(N + 3, 2).Value  
    Label2.Caption = Str(Val(Label2.Caption) + Work-  
sheets(1).Cells(N + 3, 4))  
End Sub
```

Здесь используются стандартные функции Visual Basic **Val** ( для перевода из тестового вида в числовой) и **Str** ( для перевода из числового вида в текстовый).

Мы пока не создали обработку событий по кнопкам **Пересчет** и **Очистка**. Это мы сделаем чуть позже, а пока оформим третий лист, так как показано на рис. 34.

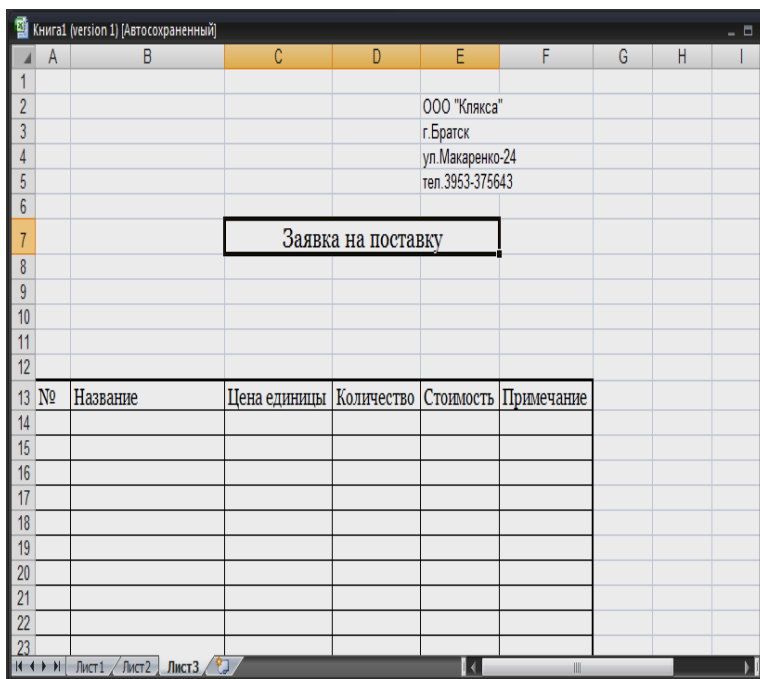


Рис. 34

Теперь наша задача сделать так, чтобы по кнопке **Печать** (на первом рабочем листе) заполнялся бланк (рис.34).

```
Private Sub Печать_Click()
```

```
‘ Подсчет строк в бланке заказа
```

```

N = 0
While Worksheets(1).Cells(N + 3, 1).Value <> ""
    N = N + 1
Wend
For i = 0 To N-1
    'Заполнение номера позиции
    Worksheets(3).Cells(14 + i, 1) = i + 1
    'Заполнение названия позиции заказа
    Worksheets(3).Cells(14 + i, 2) =
Worksheets(1).Cells(3 + i, 1)
    'Заполнение цены единицы товара
    Worksheets(3).Cells(14 + i, 3) = Work-
sheets(1).Cells(3 + i, 2)
    'Заполнение количества товара
    Worksheets(3).Cells(14 + i, 4) = Work-
sheets(1).Cells(3 + i, 3)
    'Заполнение цены с учетом количества
    Worksheets(3).Cells(14 + i, 5) =
Worksheets(1).Cells(3 + i, 4)
Next
'Активизация третьего листа
Worksheets(3).Activate
End Sub

```

Теперь вернемся к первому листу и рассмотрим про-  
цедуру, вызываемую по кнопке **Очистить**:

```

Private Sub Очистить_Click()
N = 0
While Worksheets(1).Cells(N + 3, 1).Value <> ""
    N = N + 1
Wend
For i = 3 To N + 2
    Worksheets(1).Cells(i, 1).Value = ""
    Worksheets(1).Cells(i, 2).Value = ""

```

```

Worksheets(1). Cells(i, 3).Value = ""
Worksheets(1). Cells(i, 4).Value = ""
Next
Label2.Caption = ""
End Sub

```

И теперь осталось только рассмотреть обработку нажатия на кнопку **Пересчет**. Смысл этой процедуры в том, что если мы в бланке заказа произведем изменения (удаляем строку или изменяем количество товара), то очевидно необходимо поменять и итоговые вычисления.

```

Private Sub Пересчет_Click()
N = 0
While Worksheets(1).Cells(N + 3, 1).Value <> ""
    N = N + 1
Wend
symma = 0
For i = 3 To N + 2
    a = Worksheets(1).Cells(i, 3).Value
    Worksheets(1).Cells(i,4).Value
=Worksheets(1).Cells(i, 2).Value * a
    symma = symma + Worksheets(1).Cells(i, 4).Value
Next
Label2.Caption = Str(symma)
End Sub

```

При открытии книги мы рассматривали ранее процедуру **Workbook\_Open()**. В процессе работы с книгой мы можем на втором листе вносить изменения в номенклатуру товаров. Чтобы эти изменения сразу же отражались в уже открытой книге, аналогичные действия выполним в процедуре **Worksheet\_Activate()**:

```

Private Sub Worksheet_Activate()

```

```

Worksheets(1).Spisok1.Clear
N = 0
While Worksheets(2).Cells(N + 1, 1).Value <> ""
    N = N + 1
Wend
For i = 1 To N
    a1 = Worksheets(2).Cells(i, 1).Value
    a2 = Worksheets(2).Cells(i, 2).Value
    a = a1 & " " & a2 & " " & " руб."
    Worksheets(1).Spisok1.AddItem a
Next
End Sub

```

### **ПРИМЕР №3** **РАЗРАБОТКА АВТОМАТИЗИРОВАННОГО БЛАН-** **КА**

В этом примере разберем решение еще одной практической задачи. Начнем разработку следующего рабочего листа (рис.35). Следует заметить, что здесь на представленном фрагменте используются **не элементы управления**, а просто ячейки, **отформатированные** определенным образом. При этом сами фамилии набирать не требуется (необходимо лишь подобрать заливку и размер шрифта для вводимого в последующем текста). В качестве первого действия следует убрать сетку с рабочего листа и далее отформатировать ячейки (рис. 35).

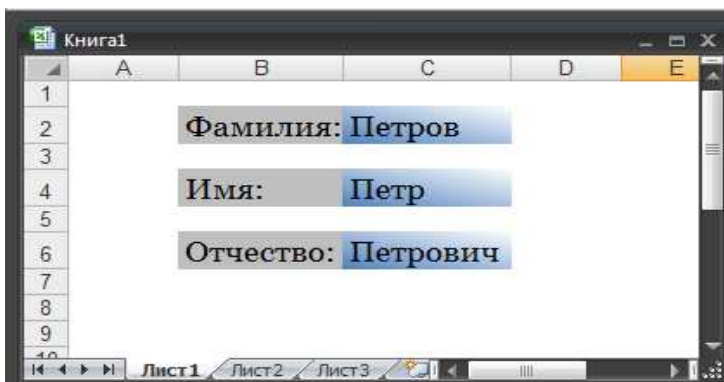


Рис.35

Теперь необходимо расположить на листе следующие элементы управления (рис.36).

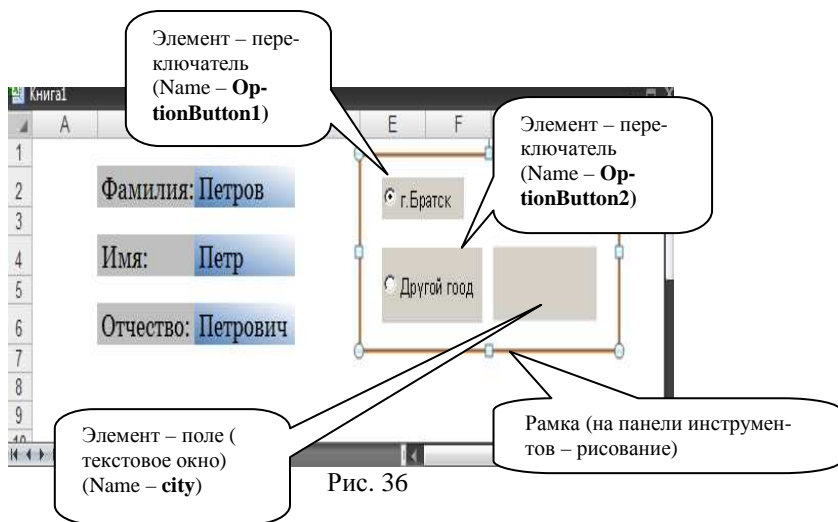


Рис. 36

На рис.36 новый элемент – переключатель. Этот элемент позволяет обеспечить два состояния – **включено** и **выключено**. Идея следующая – человек, заполняющий



бланк, указывает (щелчком по одному из переключателей) откуда он - из Братска или из другого города. При этом в случае выбора другого города человеку следует указать его в текстовом поле. На листе мы реализуем оформление следующим образом. В начальном виде установлен вариант – Братск, а текстовое окно **city** для выбора города невидимо (у свойства **Visible** установлено значение **False**). При щелчке на переключателе **Другой город** текстовое окно **city** для выбора города становится видимым (у свойство **Visible** устанавливается значение **True**). Поясним технические действия для реализации этого эффекта. В качестве начального значения свойства **Visible** элемента **city** установим **False**. Далее напомним процедуры обработки щелчков по переключателям следующим образом (рис.38):

```
Private Sub OptionButton1_Click()  
    OptionButton2.Value = False  
    City.Visible = False  
End Sub  
Private Sub OptionButton2_Click()  
    OptionButton1.Value = False  
    City.Visible = True  
End Sub
```

Теперь выйдите из режима конструктора и проверьте написанные действия.

На рис.37, 38 и 39 показаны фрагменты, которые нам необходимо далее создать на рабочем листе.

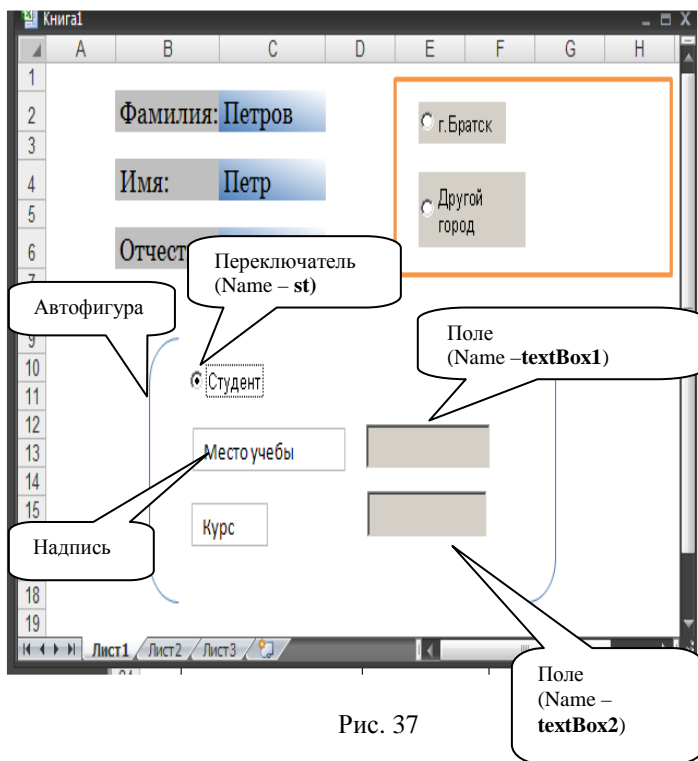


Рис. 37

После всех указанных действий лист примет вид аналогичный рис. 40.

Теперь установим в конструкторе начальные свойства элементов. Для установки города (Братск) следует в конструкторе задать начальные значения свойств:

```

OptionButton1.Enabled = True
OptionButton2.Enabled = False
City.Visible = False.

```

Для установки статуса по умолчанию (допустим – студент) выполним в конструкторе установку следующих свойств:

St.Enabled = True  
 TextBox1. Visible = True  
 TextBox1.Text = " "  
 TextBox2. Visible = True  
 TextBox2.Text = "1 "  
 sp. Enabled = False  
 TextBox3. Visible = False  
 TextBox3.Text = " "  
 TextBox4. Visible = False  
 TextBox4.Text = " "  
 flag1.Value= False  
 flag2.Value= False  
 flag3.Value= False.

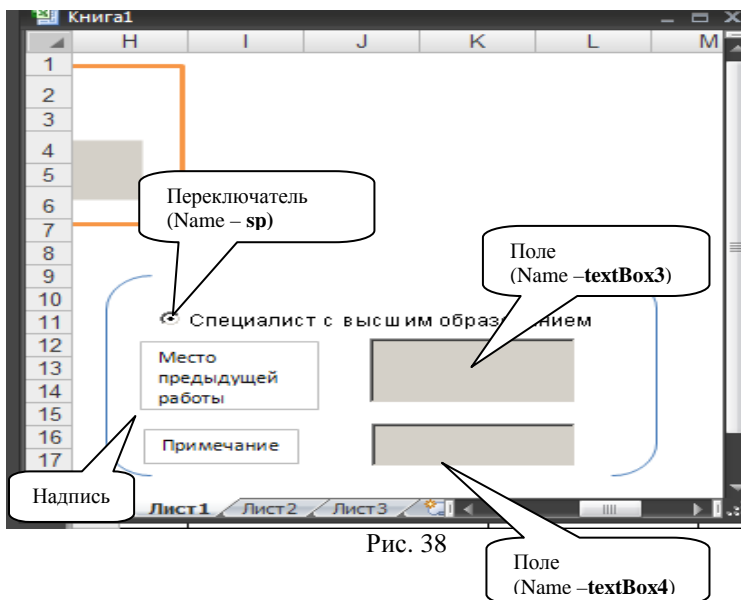


Рис. 38

Место учебы

БрГУ

Курс

Флажок (Name – flag1)

Флажок (Name – flag2)

Флажок (Name – flag3)

☒ Хорошее знание английского языка

☐ Наличие водительских прав

☒ Знание Microsoft Office

Рис.39

Исправить!!!

Фамилия: Петров

Имя: Петр

Отчество: Петрович

☐ г.Братск

☐ Другой город

☐ Студент

Место учебы

БрГУ

Курс

☐ Специалист с высшим образованием

Место предыдущей работы

Примечание

☒ Хорошее знание английского языка

☐ Наличие водительских прав

☒ Знание Microsoft Office

Рис.40

Теперь приведем процедуры (выполняемые по щелчкам на объектах) для рабочего листа:

```
Private Sub St_Click()  
    TextBox1.Visible = True
```

```

TextBox2. Visible = True
TextBox3. Visible = False
TextBox4. Visible = False
End Sub

```

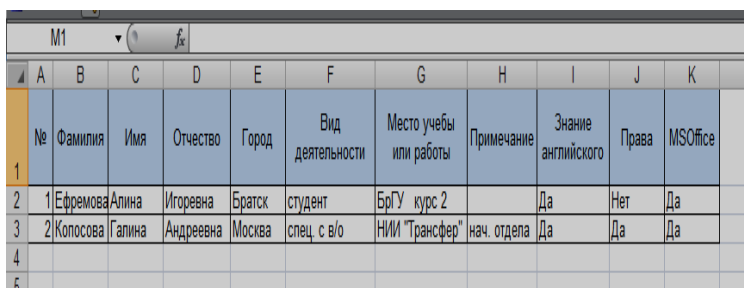
```

Private Sub Sp_Click()
TextBox1. Visible = False
TextBox2. Visible = False
TextBox3. Visible = True
TextBox4. Visible = True
End Sub

```

Следующая задача – это запись на второй лист введенных данных.

На рис.41 показана таблица данных, которая будет заполняться автоматически по щелчку по кнопке. Описание следует далее.



	A	B	C	D	E	F	G	H	I	J	K	L
	№	Фамилия	Имя	Отчество	Город	Вид деятельности	Место учебы или работы	Примечание	Знание английского	Права	MSOffice	
1												
2	1	Ефремова	Алина	Игоревна	Братск	студент	БрГУ курс 2		Да	Нет	Да	
3	2	Колосова	Галина	Андреевна	Москва	спец. с в/о	НИИ "Трансфер"	нач. отдела	Да	Да	Да	
4												
5												

Рис.41

При этом для отображения полной картины выполненных действий на рис.42 показан первый лист в рабочем состоянии с введенными данными.

Рис.42.

Напишем обработку события щелчка по кнопке с названием - **Записать на лист1**. Текст процедуры, обрабатывающей данное событие, приводится далее.

```
Private Sub CommandButton1_Click()
```

```
‘ Подсчет количества записей на втором листе.
```

```
N = 0
```

```
While Worksheets(2).Cells(N + 2, 1).Value <> ""
```

```
    N = N + 1
```

```
Wend
```

‘Заполнение очередной строки. Номер записывается в первую колонку.

```
Worksheets(2).Cells(N + 2, 1).Value = N + 1
```

‘Фамилия записывается во вторую колонку.

```
Worksheets(2).Cells(N + 2, 2).Value = Range("c2")
```

‘Имя записывается в третью колонку.

```
Worksheets(2).Cells(N + 2, 3).Value = Range("c4")
```

```
‘Отчество попадает в четвертую колонку.
```

```

Worksheets(2).Cells(N + 2, 4).Value = Range("c6")
'Название города располагается в пятой колонке.
If OptionButton1.Value = True Then
    Worksheets(2).Cells(N + 2, 5).Value = "Братск"
Else
    Worksheets(2).Cells(N + 2, 5).Value = city.Text
End If
'Вид деятельности расположим в шестой колонке,
место работы или
' учебы в седьмой ,а примечание в восьмой колонке.
If st.Value = True Then
    Worksheets(2).Cells(N + 2, 6).Value = "студент"
    Worksheets(2).Cells(N + 2, 7).Value = TextBox1.Text
+ " курс " +
    TextBox2.Text
Else
    Worksheets(2).Cells(N + 2, 6).Value = "спец. с в/о"
    Worksheets(2).Cells(N + 2, 7).Value = Text-
Box3.Text
    Worksheets(2).Cells(N + 2, 8).Value = Text-
Box4.Text
End If
'Характеристики, указанные во флагах.
If flag1.Value = True Then
    Worksheets(2).Cells(N + 2, 9).Value = "Да"
Else
    Worksheets(2).Cells(N + 2, 9).Value = "Нет"
End If

If flag2.Value = True Then
    Worksheets(2).Cells(N + 2, 10).Value = "Да"
Else
    Worksheets(2).Cells(N + 2, 10).Value = "Нет"
End If

```

```
If flag3.Value = True Then
    Worksheets(2).Cells(N + 2, 11).Value = "Да"
Else
    Worksheets(2).Cells(N + 2, 11).Value = "Нет"
End If
End Sub
```



## 4. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

### Задание 1

На рабочем листе расположить элемент управления - **кнопку**. Необходимо сделать так, чтобы при наведении курсора мыши на кнопку она могла принимать один из четырех цветов. Во-первых, красный, при наведении курсора мыши на левый верхний угол кнопки. Во-вторых, синий, при наведении курсора мыши на левый нижний угол кнопки. Далее зеленый, если курсор мыши располагается в правом верхнем углу кнопки, и белый, если курсор мыши располагается в правом нижнем углу кнопки.

При выполнении задания необходимо воспользоваться функцией **rgb** для задания цвета. Например, строка программы в виде

Кнопка1.BackColor = rgb(a,b,c)

приводит к установке цвета кнопки. Здесь a – интенсивность красного цвета (от 0 до 255), b – интенсивность зеленого цвета (от 0 до 255), c – интенсивность синего цвета.

Также следует воспользоваться методом **MouseMove**, который автоматически выполняется при наведении курсора мыши на тот или иной объект.

### Задание 2

Разместить на листе окно для рисунка (один из элементов управления). Сделать так, чтобы при наведении курсора мыши на левую половину окна рисунка в этом окне отображалась одна картинка, а при попадании курсора мыши на правую половину окна, другая.

### Задание 3

На втором рабочем листе создайте список сотрудников в следующем виде:

	А	В	С
1	Петров	Петр	Петрович
2	Иванов	Иван	Иванович
3	Сидоров	Николай	Николаевич
4			

Далее на первом рабочем листе разместите элементы управления:

Необходимо сделать так, чтобы при открытии книги элемент управления для отображения фамилии (**Поле со**

Фамилия

Имя

Отчество

Командная кнопка

Поле со списком

Текстовое окно (поле)

Текстовое окно (поле)

**списком**) заполнялся фамилиями со второго листа. А по щелчку по этому элементу управления поля **Имя** и **Отчество** автоматически заполнялись данными, соответствующими лицу с выбранной фамилией. При щелчке по командной кнопке информация о выбранном лице должна переписываться на третий лист в очередную свободную строку.

#### Задание 4

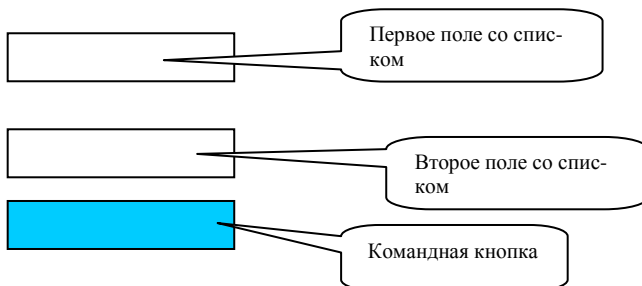
Заполните второй лист данными о сотрудниках:

	А	В	С
1	Петров	Петр	Петрович
2	Иванов	Иван	Иванович
3	Сидоров	Николай	Николаевич
4			

После этого заполните третий лист данными о подразделениях:

	А	В	С
1	Администрация		
2	Основное производство		
3	Транспортный отдел		
4			

На первом рабочем листе создайте следующие элементы управления:



Необходимо сделать так, чтобы при открытии книги первое поле со списком заполнялось данными о сотрудниках со второго листа, а второе поле со списком должно заполняться данными о подразделениях с третьего листа. По щелчку по командной кнопке данные должны перепи-

сываться на четвертый лист в очередную свободную строку.

### **Задание 5**

Необходимо разработать следующую игру. На рабочем листе разместите 3 окна для рисунков, а также кнопку и текстовое окно. При щелчке по кнопке в каждое окно для рисунка должна попадать (случайным образом) она из трех возможных картинок. Если все три картинки совпадают, то в текстовое окно добавляется 9 баллов, а если нет, то вычитается 1 балл.

## **СПИСОК ЛИТЕАТУРЫ**

1. Джон Уокенбах, Профессиональное программирование на VBA в Microsoft Office Excel 2003, Вильямс, 2006 г., с.800
2. С.Н. Белоусова, И.А. Бессонова, Основные принципы и концепции программирования на языке VBA в Excel, Интернет-университет информационных технологий, Бином Лаборатория знаний, 2010, с.200
3. Г.З. Гарбер, Основы программирования на Visual Basic и VBA в Excel 2007, Салон-Пресс, 2008, с.192