

Использование VBA для подготовки данных

1. Первый элемент управления

Режим конструктора и элементы ActiveX. Программирование, заложенное в Microsoft Office, основывается на языке VBA – *Visual Basic for Application*. При ознакомлении с данным языком понадобятся элементы **ActiveX**, которые по-другому можно назвать «**Элементы управления**».

В качестве первого упражнения, необходимо в окне, показанном на рисунке 1.2:

1. Щелкнуть кнопкой «**Параметры Excel**». В результате откроется новое окно с набором закладок (рисунок 1);

2. На закладке «**Настроить ленту**» следует сделать установку для отображения на ленте вкладки «**Разработчик**», что реализуется с помощью флажка «**Показывать вкладку «Разработчик» на ленте**» (рисунок 2).



Рисунок 1

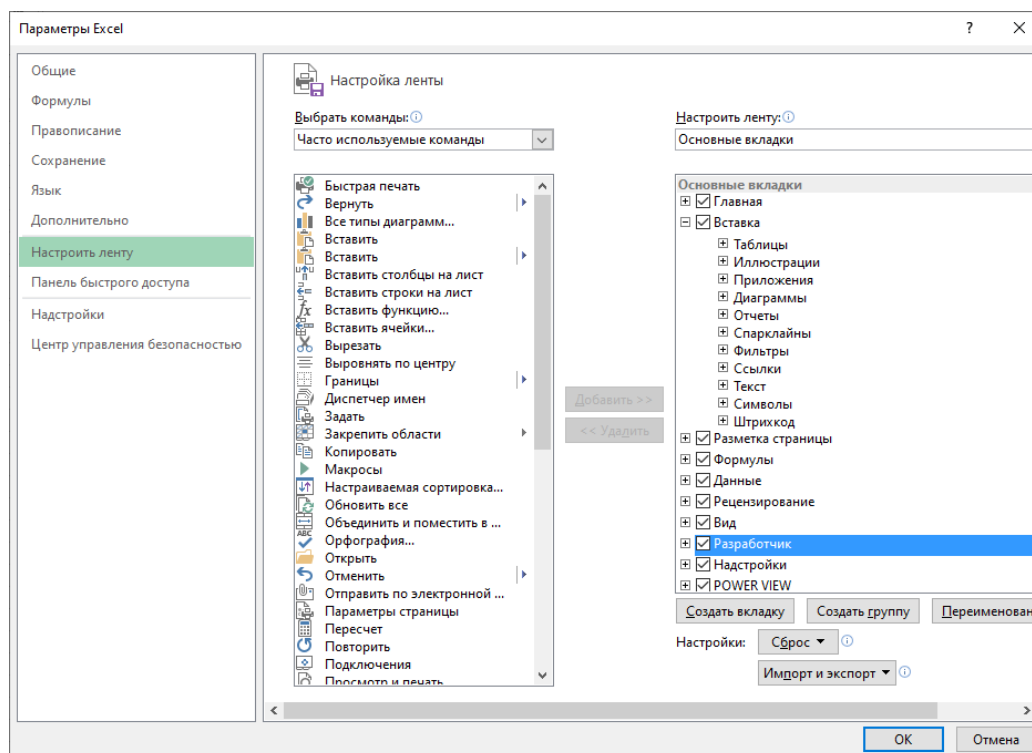


Рисунок 2

В результате окно приложения Microsoft Excel дополнится новыми пиктограммами, которые нам потребуются в дальнейшем (рисунок 3).

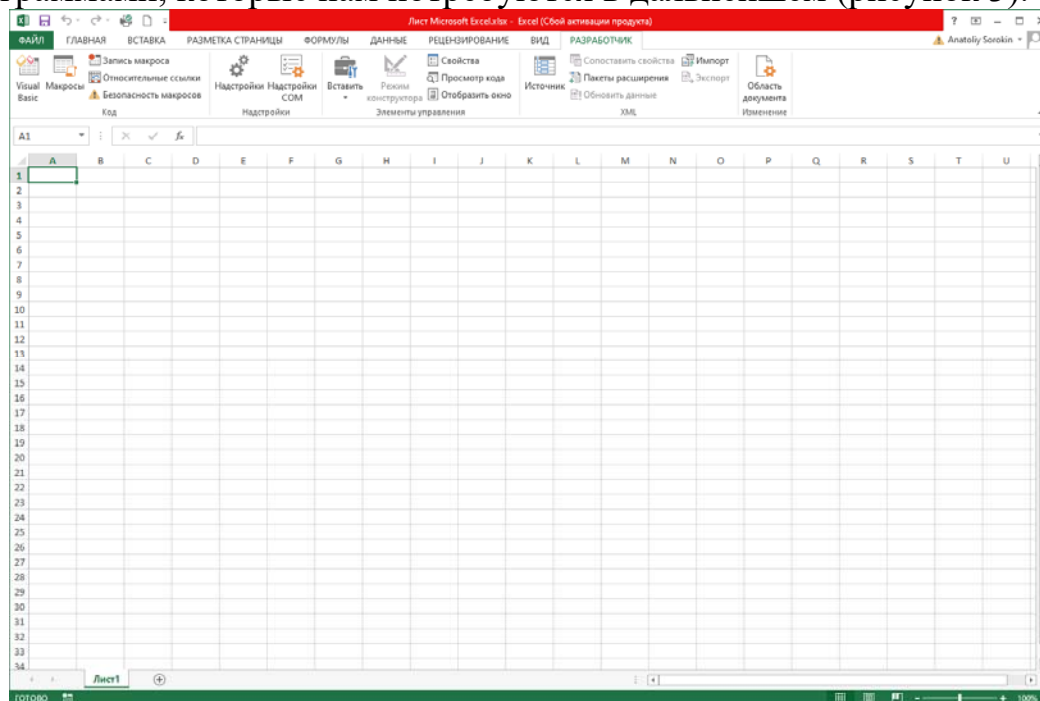


Рисунок 3

Наиболее часто используемой на протяжении данной лабораторной работы будет пиктограмма с надписью «**Вставить**». С ее помощью на рабочем листе можно размещать элементы ActiveX, которые существенно дополняют функциональность книг Microsoft Excel. Они типичны для различных приложений Windows (речь идет о кнопках, полосах прокрутки, текстовых окнах, переключателях, списках и т. д.). Кроме пиктограммы *Вставить*, необходимо обратить внимание на другие, присутствующие на ленте пиктограммы – *Режим конструктора*, *Свойства*, *Visual Basic* и *Макросы*.

Перед тем, как начать какие либо действия с тем или иным элементом ActiveX, его необходимо поместить на рабочий лист. Выполнить это легко, воспользовавшись пиктограммой *Вставить*. На рисунке 4 показано размещение на рабочем листе элемента ActiveX *Кнопка*.

Для того, чтобы перенести элемент ActiveX на рабочий лист:

1. Щелкните на панели инструментов на его пиктограмме левой кнопкой мыши;
2. Переместите курсор мыши на рабочий лист. Курсор мыши примет вид, аналогичный математическому знаку «плюс»;
3. Нажмите на левую кнопку мыши и, не отпуская ее, перемещайте мышью вправо и вниз, а затем отпустите ее левую кнопку.

В результате на рабочем листе мы увидим изображение появившегося элемента ActiveX, окруженного маркерами (рисунок 4).

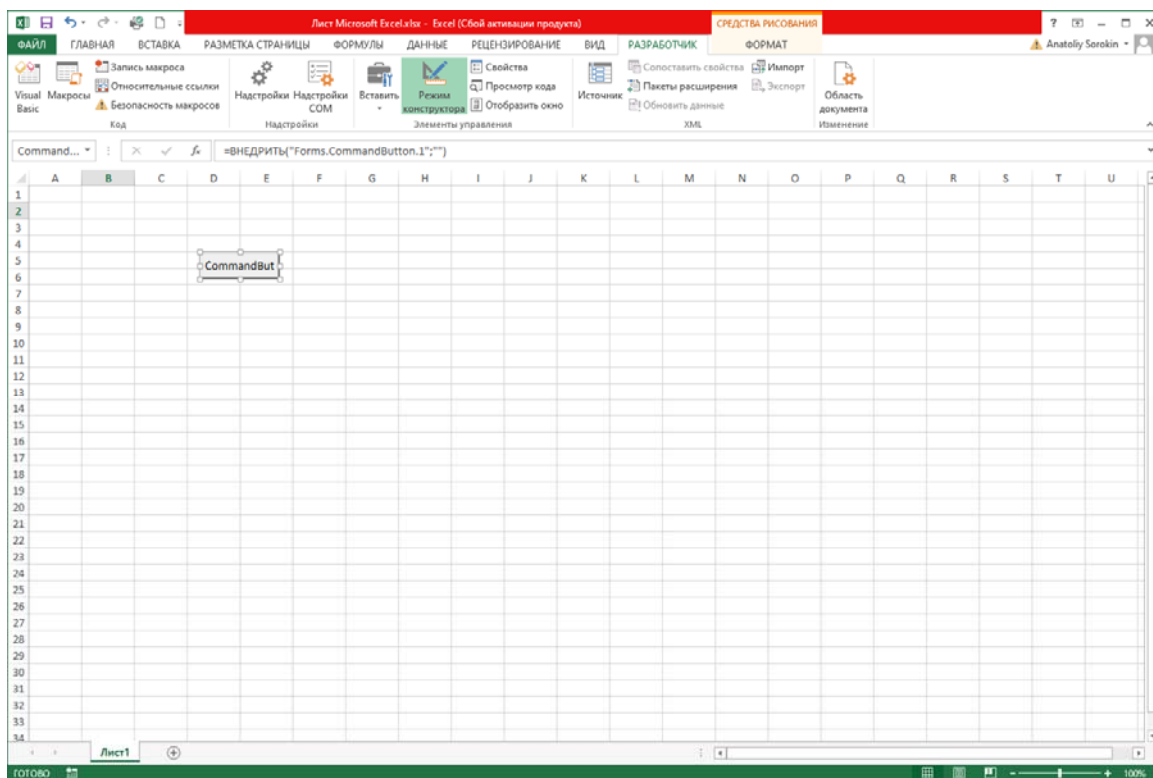


Рисунок 4

После размещения элемента ActiveX на рабочем листе активизируется кнопка *Режим конструктора*, которая позволяет пользователю далее работать в одном из двух режимов. Один из них так и называется – *режим конструктора*. В этом случае можно работать с элементами ActiveX для создания необходимого интерфейса на рабочем листе. Также в этом режиме пользователю предоставляется возможность создавать программные разработки.

Появление маркеров (рисунок 4), окружающих элемент ActiveX, как раз и является признаком того, что мы работаем в режиме конструктора (кнопка *Режим конструктора* в этом случае выглядит нажатой). На начальном этапе создания разработки работа в режиме конструктора сводится к размещению элементов управления на рабочем листе и изменению их свойств. Далее производится программирование различных событий, связанных с элементами управления, листами и книгой в целом.

Другой режим можно назвать *режимом выполнения* или *режимом работы* (какого-то однозначно принятого названия не существует). Чтобы перейти в этот режим (то есть выйти из режима конструктора), необходимо отжать кнопку *Режим конструктора* (рисунок 5). После этого будет возможность пощелкать кнопкой, которую мы создали на рабочем листе (функциональные действия, которые можно обеспечить в этом случае, будут рассмотрены в дальнейшем).

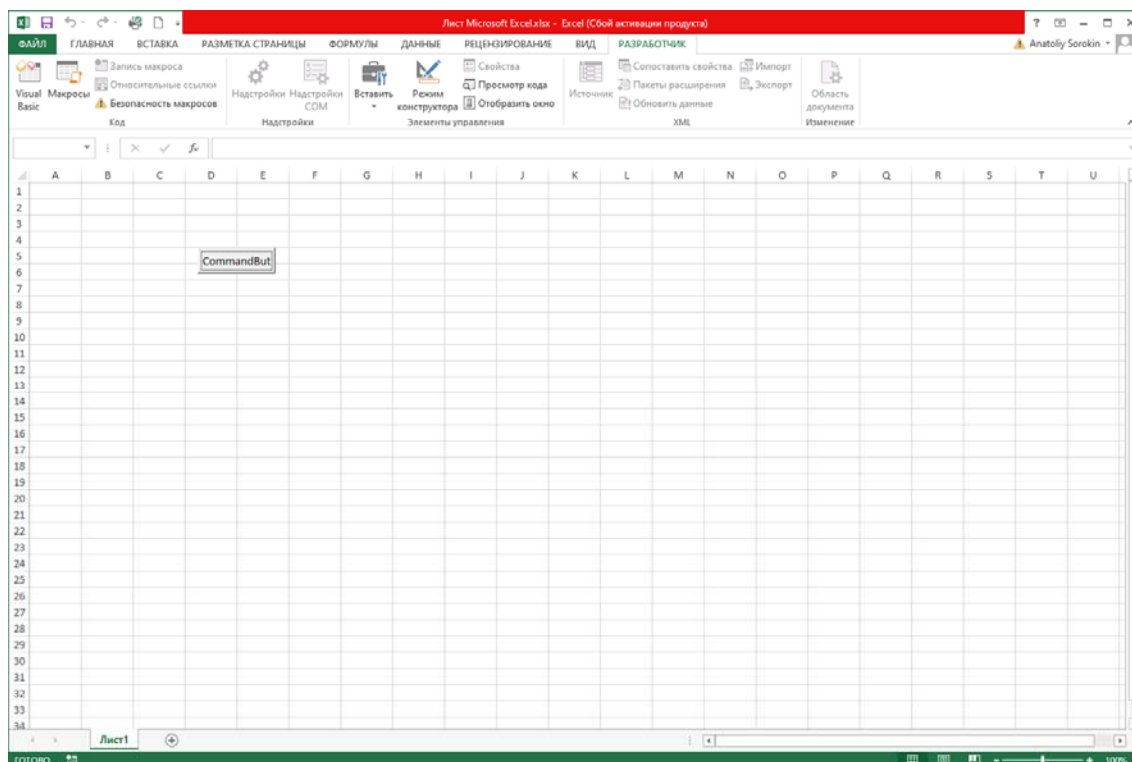


Рисунок 5

Далее опять вернитесь в режим конструктора, где с помощью маркеров, окружающих кнопку, измените ее размер и переместите по рабочему листу.

Свойства элемента управления «Кнопка». У программных объектов имеются различные свойства. Для просмотра свойств объекта его необходимо сначала выделить в режиме конструктора, щелкнув на нем мышью, а затем с помощью пиктограммы *Свойства* , которая расположена рядом с пиктограммой *Режим конструктора*, открыть окно свойств (рисунок 6).

В окне свойств отображается таблица – набор строк в два столбца. При этом левый столбец отводится для названий свойств, а в правом расположены значения этих свойств. Содержимое левого столбца изменять нельзя – свойства объекта уже определены его разработчиком, компанией Microsoft. А значения свойств мы поменять можем – для этого нужно щелкнуть в ячейке правого столбца и изменить ее содержание.

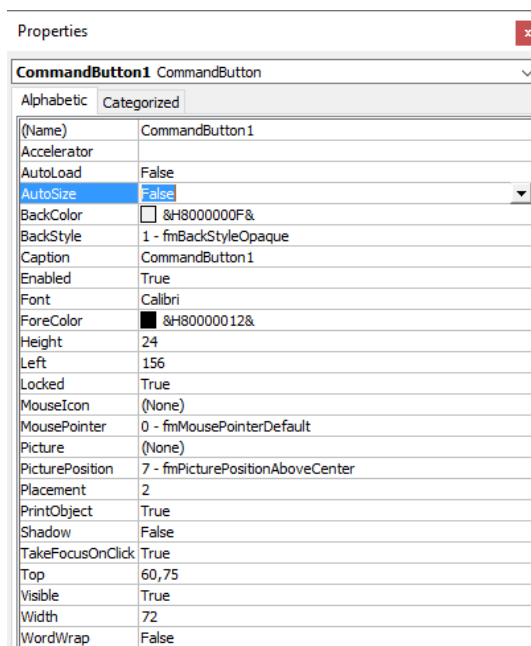


Рисунок 6

Если приглядеться к окну на рисунке 6, то мы увидим, что для ряда свойств возможны только два варианта значений. Например, для свойств *Visible* (определяет видимость объекта на экране) возможны только значения *False* и *True*. Такая ситуация наблюдается и для ряда других свойств. Среди них отметим *Enabled* – определяет доступ к объекту в режиме выполнения (если установить для этого свойства значение *False* и выйти из режима конструктора, то объект будет недоступен).

Свойство *AutoSize* позволяет обеспечить автоматическую подгонку размеров кнопки под размер текста, расположенного на ней. Такое будет происходить, если установить *True* в качестве значения данного свойства.

Для некоторых других свойств следует устанавливать числовые значения. Так, местоположение и размеры элемента управления определяются следующими свойствами:

- **Width** – ширина;
- **Height** – высота;
- **Top** – координата верхней границы элемента, начиная от верхней части листа;
- **Left** – координата левой границы элемента, начиная от левой части листа.

Теперь в режиме конструктора переместите созданную на листе кнопку по экрану и измените ее размеры, а соответствующие значения в окне свойств будут меняться. И наоборот, изменив значения в окне свойств, изменения отобразятся на экране. Для установки значений ряда свойств следует использовать пиктографические меню. Так, с помощью свойства *BackColor* измените цвет кнопки. Для этого требуется щелчком мыши выберите необходимый цвет (рисунок 7).

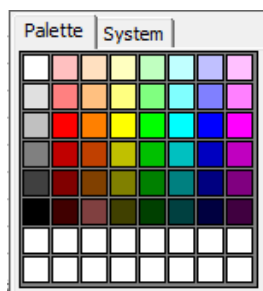


Рисунок 7

Аналогичное свойство *ForeColor* определяет цвет текста на кнопке. За надпись на кнопке отвечает другое свойство – *Caption*. Введем напротив названия этого свойства текст (рисунок 8) и мы увидим его и на кнопке.

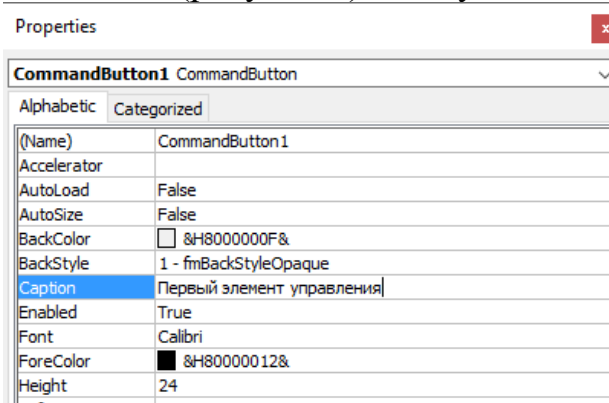


Рисунок 8

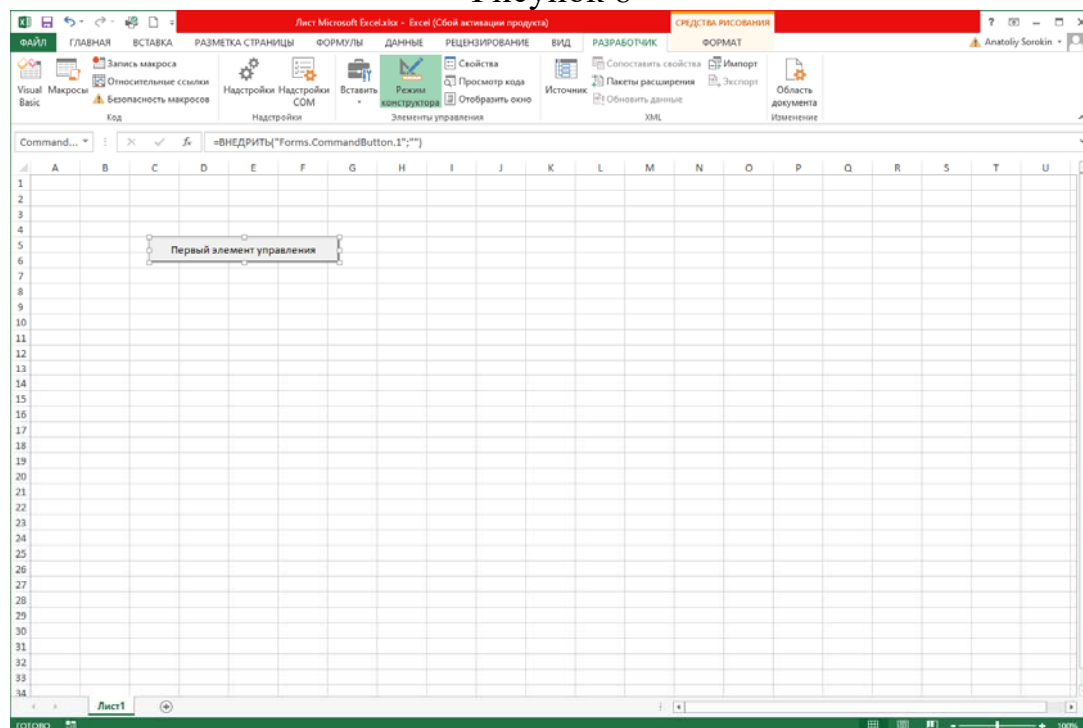


Рисунок 9

Для изменения шрифта текста на кнопке имеется свойство *Font*. Если в окне свойств попытаться изменить его значение, то перед нами откроется знакомое окно для выбора шрифта и размера букв. При этом, технические действия здесь выполняются с помощью щелчков мышью.

Еще одно интересное свойство, *Picture*, позволяет разместить на кнопке изображение (рисунок 2.5) из графического файла. Для этого в качестве значения свойства следует указать имя файла.

Это действие производится с помощью стандартного диалогового окна (рис.2.6), в котором необходимо выбрать один из графических файлов на компьютере, либо в сети. В случае, если необходимо убрать изображение, то в поле для значения данного свойства следует воспользоваться клавишей *Delete*.

Основным свойством объекта является *имя*. В окне свойств оно называется *Name*. Как уже было отображено на рисунке 2.1, на примере создания кнопки, Excel автоматически присвоил ей имя. Имя первой созданной кнопки на рабочем листе по умолчанию – *CommandButton1*, которое при желании можно изменить. Явной необходимости изменения имен, которые присваиваются по умолчанию, нет. Но предпочтительнее назначать объектам свои имена. Это связано с тем, что к любым объектам на листе можем получить доступ из текста программы, указав имя конкретного объекта. В связи с этим, с точки зрения организации программного кода часто удобно использовать свою систему назначения имен объектов.

Для изменения имени в окне свойств вместо *CommandButton1* следует набрать слово или словосочетание, которое вас устраивает. Но заметим, что оно должно быть без пробелов внутри! Изменим имя кнопки на *PrimerButton*.

После всего сделанного (также убрать изображение – очистить значение в свойстве *Picture*). Для свойства *AutoSize* нужно установить значение *True*, что позволит подогнать размеры кнопки под расположенную на ней надпись.

2. Написание кода

Для написания кода (текста) программы технически для этого в режиме конструктора следует сделать двойной щелчок (рисунок 1) мышкой на одном из размещенных в листе объектов (по кнопке, текстовой области или надписи).

В его правой части ленты (Visual Basic) расположен раздел, предназначенный для написания программного кода.

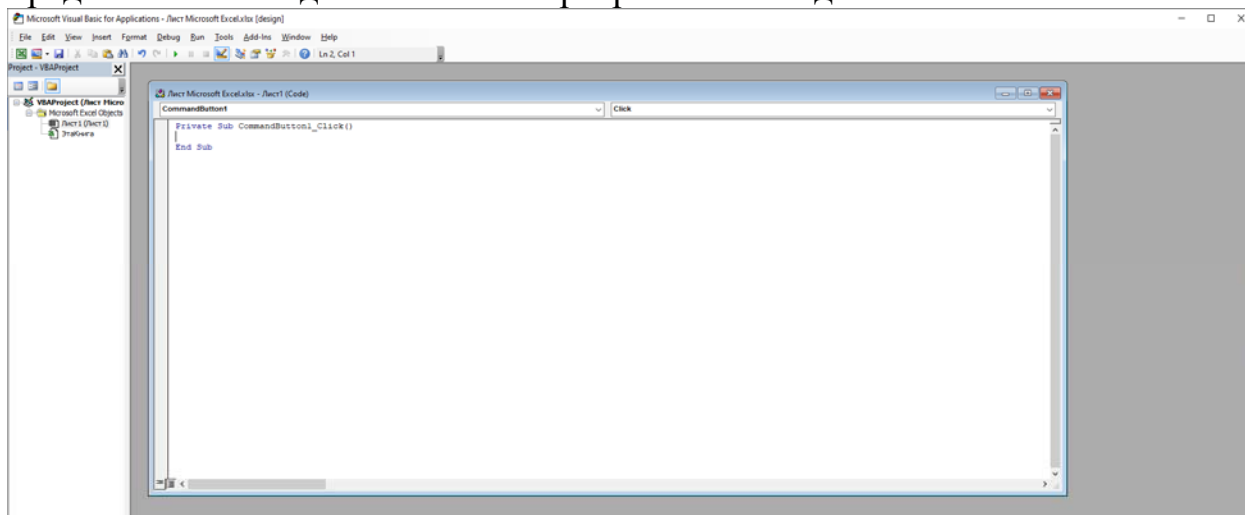


Рисунок 1

Напишем простой код, который будет выполняться при нажатии на кнопку. На листе 1 в диапазоне ячеек В3:С4 (рисунок 2) имеются значения и нам необходимо очистить данный диапазон.

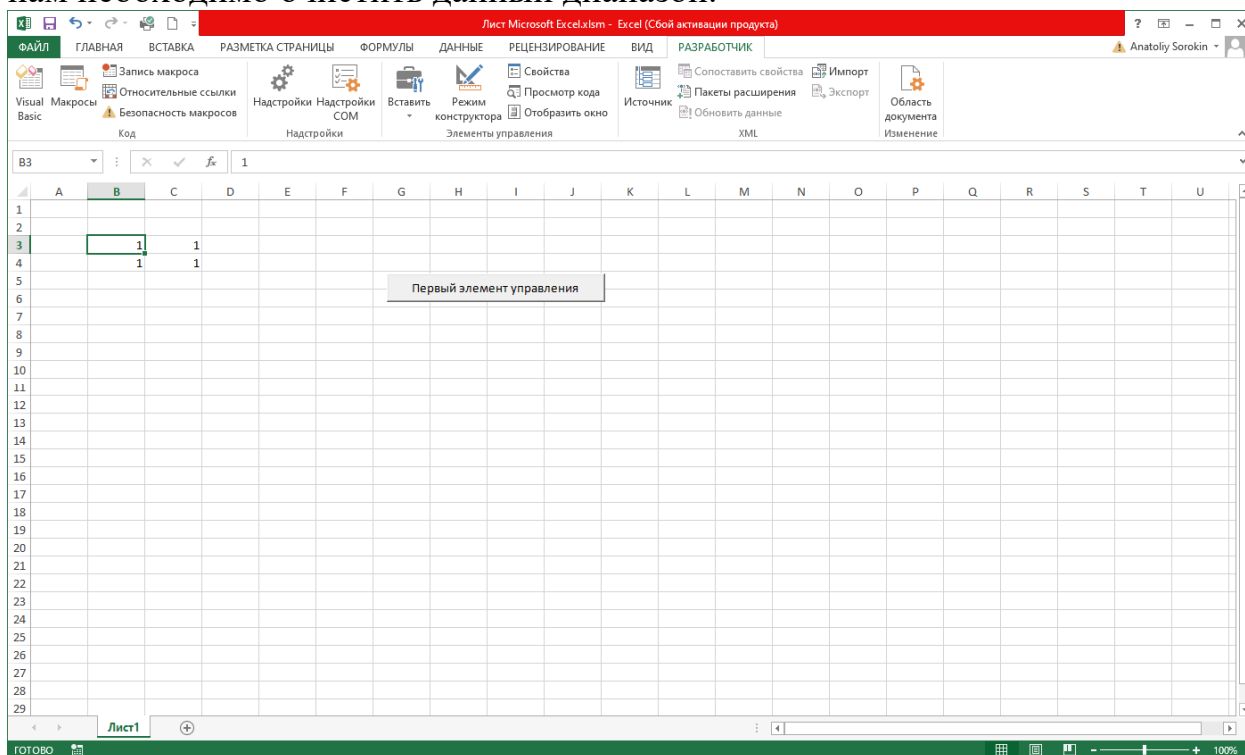


Рисунок 2

Переходим в редактор кода двойным щелчком по кнопке «Первый элемент управления» (рисунок 3).

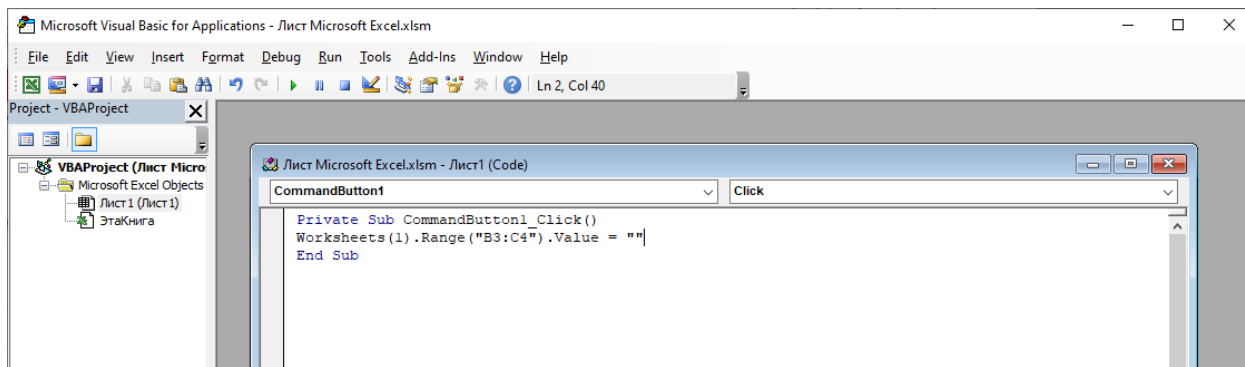


Рисунок 3

Добавляем следующий код:

```
Private Sub CommandButton1_Click()
Worksheets(1).Range("B3:C4").Value = ""
End Sub
```

Проверяем работоспособность (Рисунок 4).

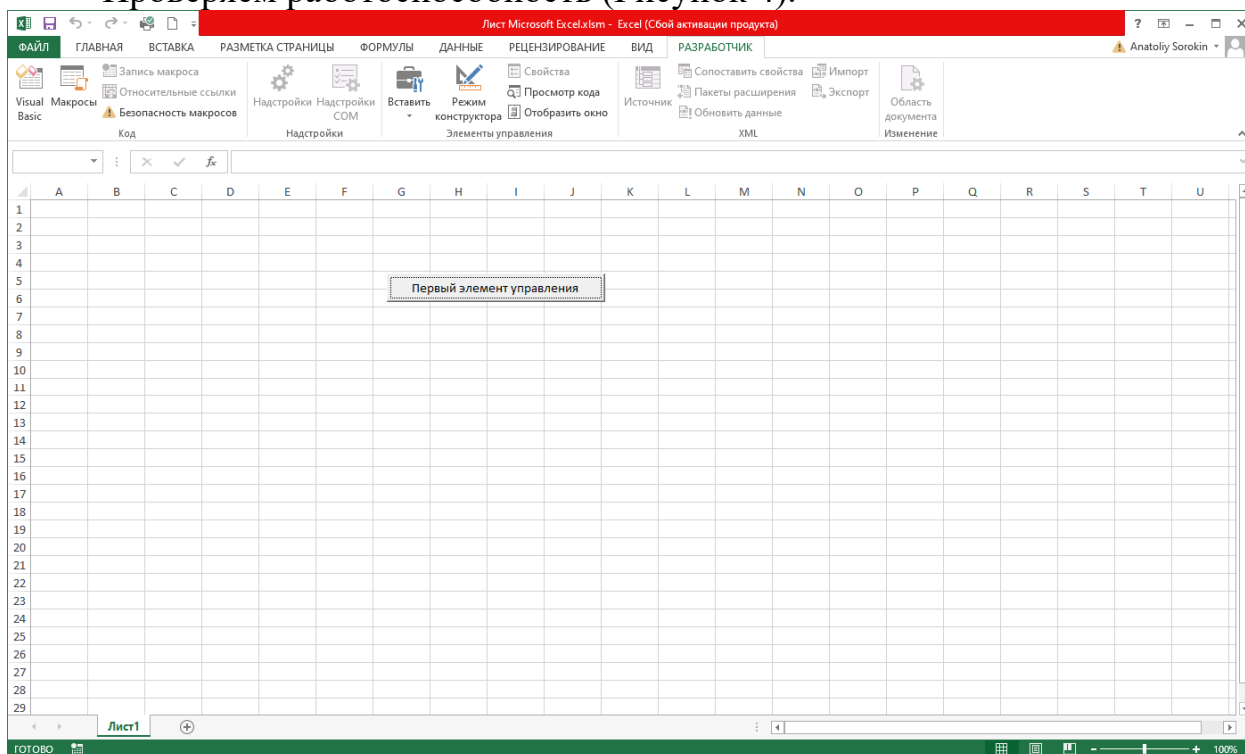


Рисунок 4

Подведем небольшой итог. Мы рассмотрели основные технические действия, которые следует выполнять при создании программных разработок в Microsoft Excel. Их можно сформулировать следующим образом:

- расположение на листе элементов ActiveX (элементов управления);
- присвоение созданным объектам имен и установка необходимых значений для других свойств;
- разработка процедур, связанных с событиями, относящимися к элементам управления.

Сохранение рабочих книг Microsoft Excel. Следующим важным моментом после создания книги Microsoft Excel с необходимой функциональностью является сохранение файла. Стандартная рабочая книга Microsoft Excel имеет одно из двух расширений:

- **xlsx** (если книга не содержит макросов);

– *xlsm* (если рабочая книга содержит макросы).

Макросом называется именованная последовательность команд (операторов языка VBA), которую пользователь может записать, а затем запускать на исполнение. Команды могут быть сформированы автоматически (как запись последовательных действий пользователя с помощью специального средства – *макрорекодера*) либо написаны непосредственно в редакторе VBA.

Сохраняя документ, содержащий макросы, необходимо обратить внимание, чтобы форматом служило расширение *xlsm*.

Для сохранения файла необходимо выбрать кнопку *Office* и затем раздел *Сохранить* (или *Сохранить как*). В окне *Сохранение документа* следует указать тип файла – *Книга Excel с поддержкой макросов*.

3. Создание формы для автоматизации заполнения листа

Изменим наш код так чтобы при нажатии на кнопку открывалась форма.
Для этого сначала создадим форму (рисунок 1).

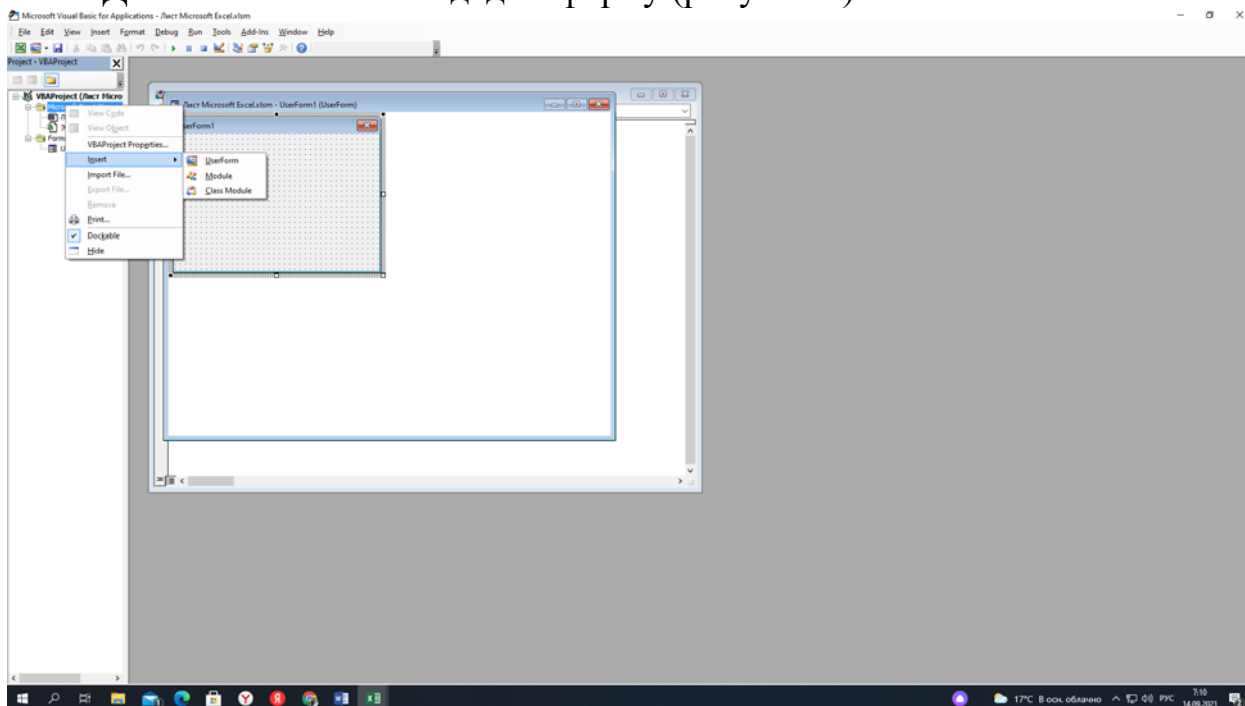


Рисунок 1

Следующим шагом добавим несколько элементов управления на нашу форму (рисунок 2):

- Label – 3 штуки
- TextBox – 3 штуки
- CommandButton – 1 штука

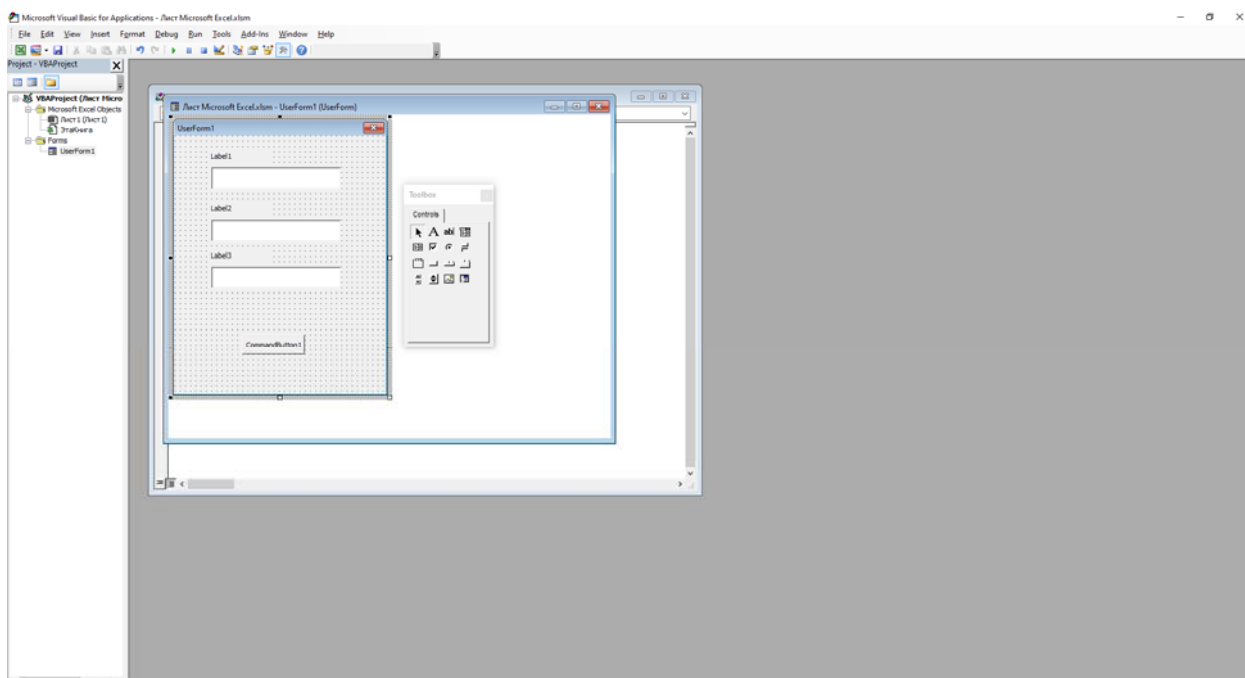


Рисунок 2

Теперь изменим свойства элементов Caption следующим образом (рисунок 3).

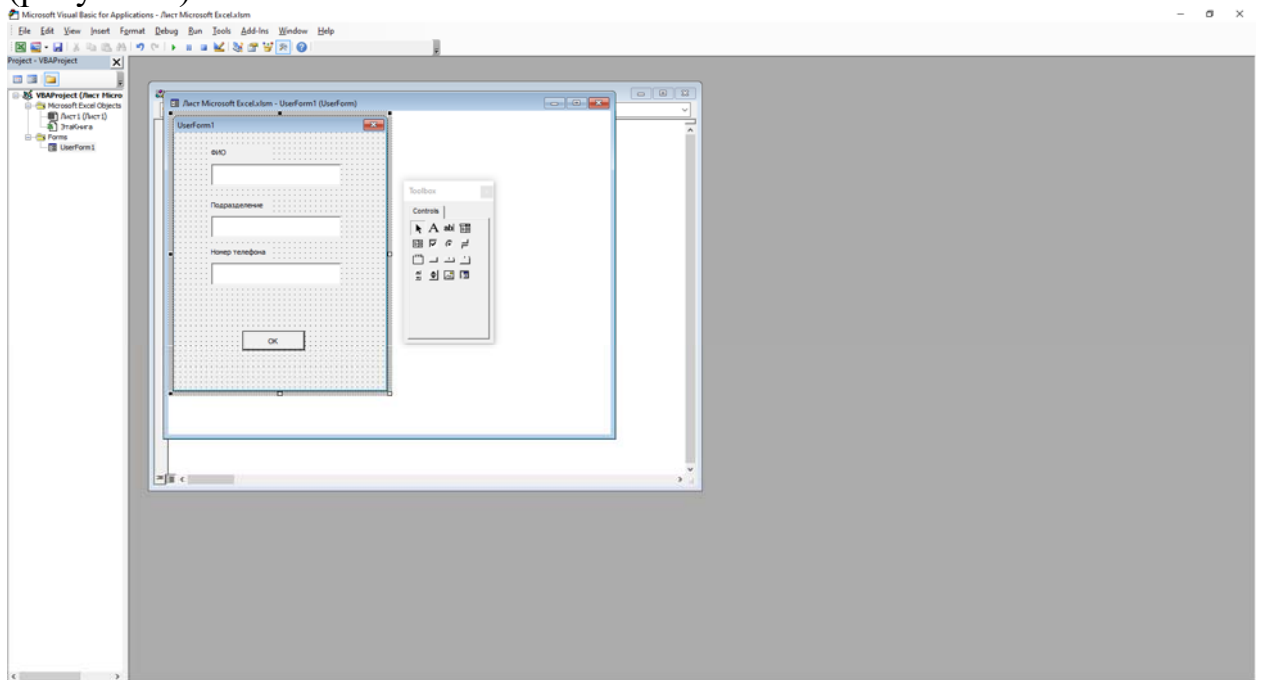


Рисунок 3

Теперь добавим код для кнопки, чтобы при нажатии открывалась форма (рисунок 4).

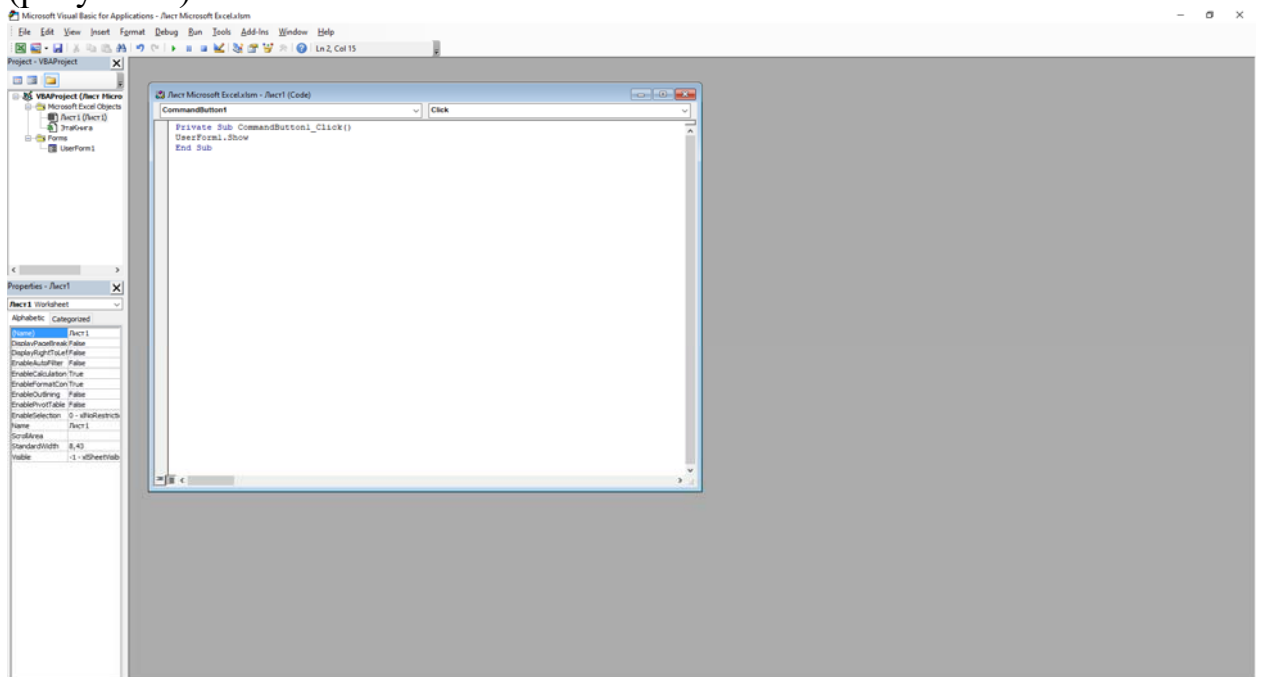


Рисунок 4

Перейдем на Лист1 и проверим работоспособность (рисунок 5).

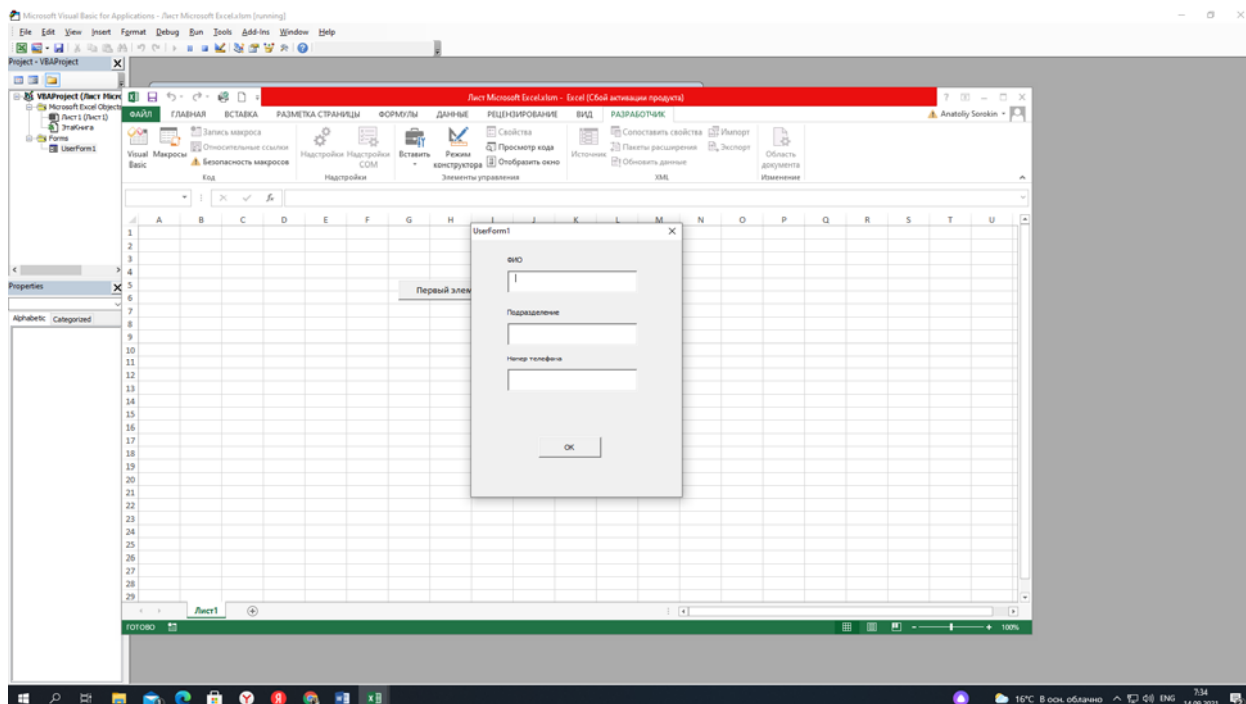


Рисунок 5

Следующим этапом добавим код на форму, чтобы после ее заполнения, данные заносились на Лист2 (Переименуем Лист2 в «Сотрудники») и добавим шапку таблицы (рисунок 6).

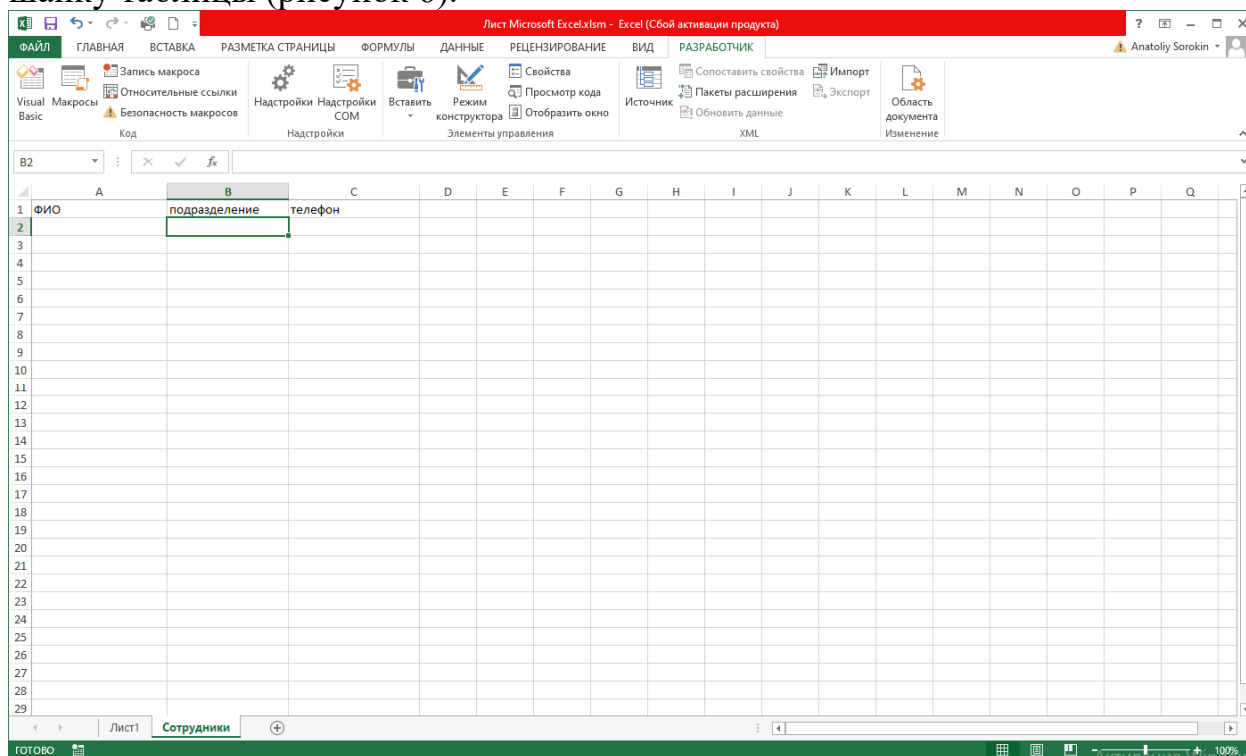


Рисунок 6

Переходим в конструктор формы и дважды нажимаем на кнопку ОК (рисунок 7).

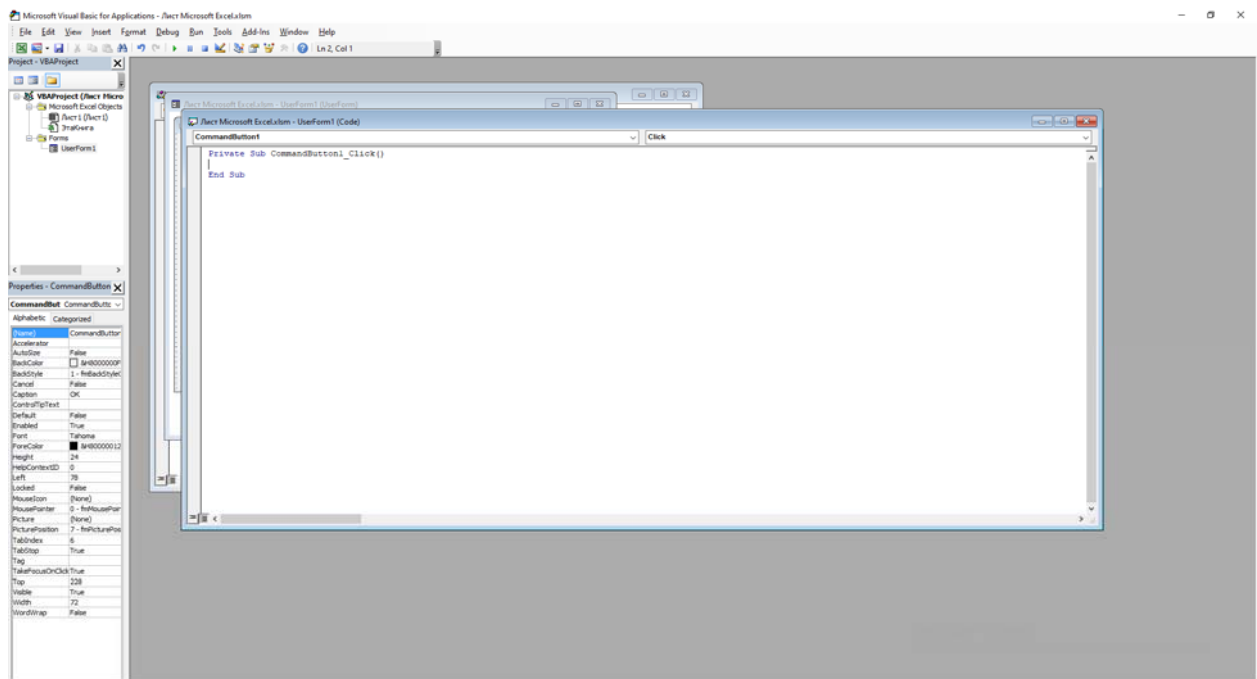


Рисунок 7

Создана процедура обработки щелчка на кнопке, которая расположена на созданной форме.

Теперь добавим следующий листинг (рисунок 8).

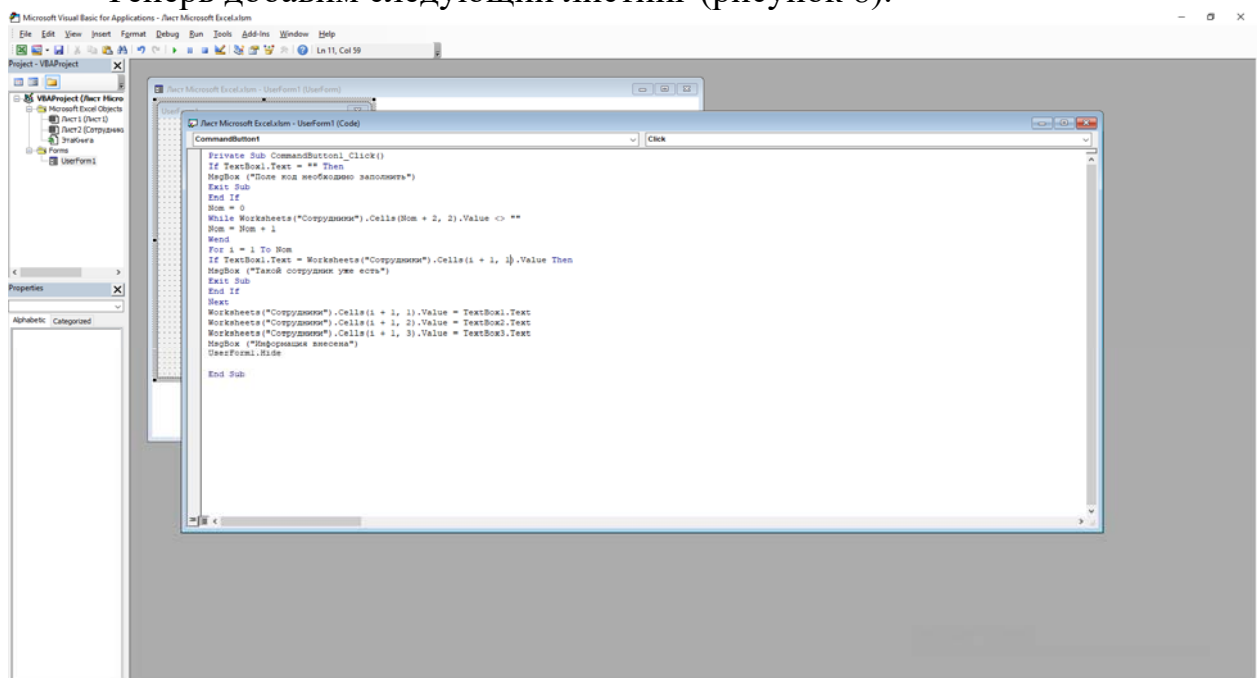


Рисунок 8

Листинг:

```
Private Sub CommandButton1_Click()
If TextBox1.Text = "" Then
MsgBox ("Поле код необходимо заполнить")
Exit Sub
End If
Nom = 0
While Worksheets("Сотрудники").Cells(Nom + 2, 2).Value <> ""
```

```

Nom = Nom + 1
Wend
For i = 1 To Nom
If TextBox1.Text = Worksheets("Сотрудники").Cells(i + 1, 1).Value Then
MsgBox ("Такой сотрудник уже есть")
Exit Sub
End If
Next
Worksheets("Сотрудники").Cells(i + 1, 1).Value = TextBox1.Text
Worksheets("Сотрудники").Cells(i + 1, 2).Value = TextBox2.Text
Worksheets("Сотрудники").Cells(i + 1, 3).Value = TextBox3.Text
MsgBox ("Информация внесена")
UserForm1.Hide
End Sub

```

После этого переходим на Лист1 для проверки работоспособности и заполняем форму (рисунок 9).

The image shows a standard Windows-style dialog box titled 'UserForm1'. It has a light gray background and a thin border. Inside, there are three vertically stacked text input fields. The first field is labeled 'ФИО' and contains the text 'Иванов Иван Иванович'. The second field is labeled 'Подразделение' and contains 'отдел 1'. The third field is labeled 'Номер телефона' and contains '89180001111'. At the bottom center of the dialog is a button labeled 'ОК'. The dialog box has a standard Windows title bar with a close button (X) in the top right corner.

Рисунок 9

После нажатия на кнопку данные будут занесены на лист «Сотрудники» (Рисунок 10).

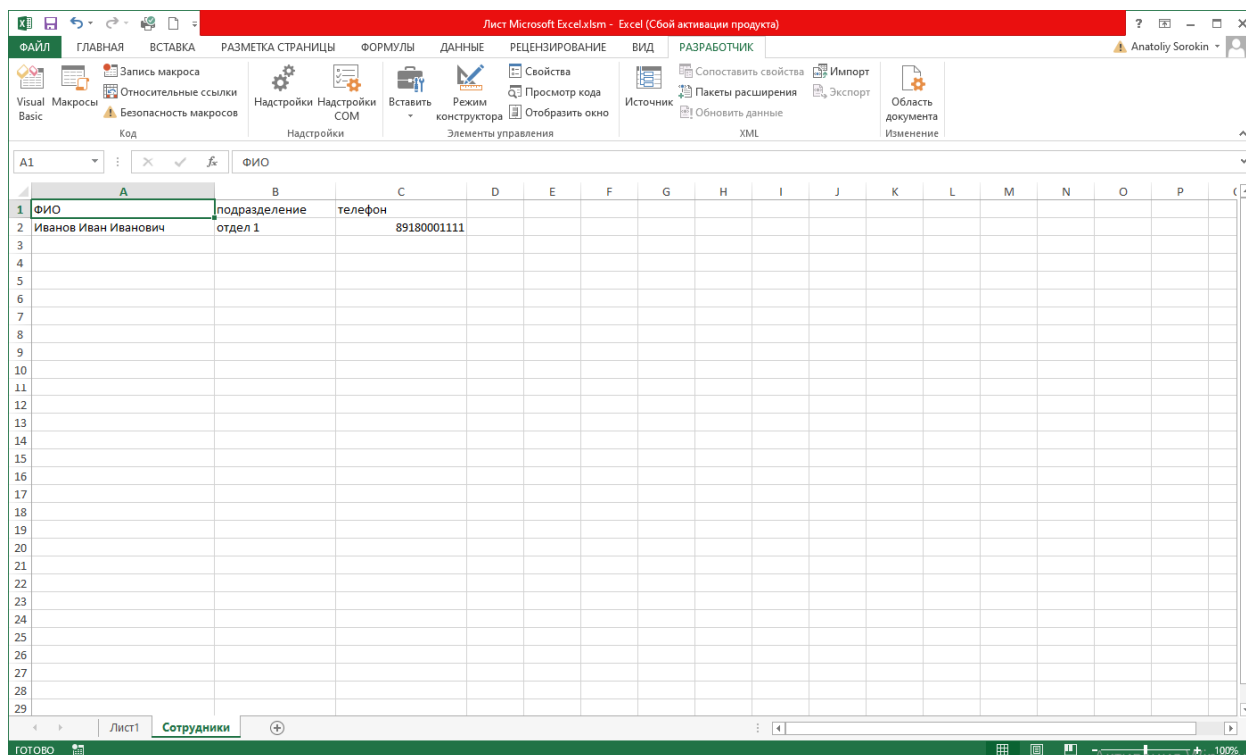


Рисунок 10

Также в нашем листинге реализована возможность проверки дублирующих записей. При повторной попытке внести информацию на лист появится сообщение "Такой сотрудник уже есть".

Самостоятельная работа:

Слияние данных MS Word и MS Excel

<https://office-guru.ru/excel/functions-tekstovye/sliyanie-dannyx-ms-word-i-ms-excel.html>