МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ

М.А. Борлакова
В.Ю. Кондратьев
А.М. Кумратова
Л.М. Эльканова

КОМПЬЮТЕРНЫЙ ПРАКТИКУМ

лабораторный практикум для обучающихся по направлению подготовки 09.03.03 Прикладная информатика

ЧЕРКЕССК
2025

УДК 004:34
ББК 32.81:67
Б 82

Рассмотрено на заседании кафедры «Общая информатика»
Протокол № 2 от «05» 09. 2024 г.
Рекомендовано к изданию редакционно-издательским советом СКГА
Протокол № 27 от «07» 11. 2024 г.

Рецензенты: Биджиева С.Х. – к.п.н., доцент кафедры «Общая информатика»

Б **Борлакова, М.А.** Компьютерный практикум: лабораторный практикум для обучающихся по направлению подготовки 09.03.03 Прикладная информатика / М.А. Борлакова, В.Ю. Кондратьев, А.М. Кумратова, Л.М. Эльканова. – Черкесск: БИЦ СКГА, 2025 – 48 с.

Данный практикум предназначен для проведения практических и лабораторных занятий по дисциплине «Компьютерный практикум» у обучающихся 1 курса направления подготовки 09.03.03 – Прикладная информатика. Содержит лабораторные работы, нацеленные на получение навыков использования программы Microsoft Word, Microsoft Excel входящей в состав пользовательского пакета Microsoft Office, в том числе в режиме разработчика. Материал изложен в соответствии с федеральным государственным образовательным стандартом.

УДК 004:34
ББК 32.81:67

© Кумратова А.М., Кондратьев В.Ю., Эльканова Л.М., Борлакова М.А, 2025
© ФГБОУ ВО СКГА, 2025

ОГЛАВЛЕНИЕ

Создание доверенности в Microsoft Word 4

Методические указания к созданию автоматизированного документа в Excel 16

Подготовительный этап 16

Работа в панели разработчика 18

Использование инструментов VBA 18

Создание функциональных форм 18

Создание макросов VBA 22

Основы синтаксиса VBA 23

Добавление кнопок 24

Создание справочников 25

Пример реализации технологии 27

Приложение 1. Листинг кода кнопок 34

Список использованной литературы 47

СОЗДАНИЕ ДОВЕРЕННОСТИ В MICROSOFT WORD

Первым шагом для создания формы документа в программе Microsoft Word будет создание пустой формы документа, с которым мы в дальнейшем будем работать. Форму можно либо написать вручную, либо найти форму в сети интернет и скопировать в программу Microsoft Word. После этого готовую форму нужно сохранить, чтобы можно было использовать пустую форму повторно при необходимости, сделать это можно при нажатии кнопки дискеты в верхнем левом углу, либо выбрав во вкладке «Файл» кнопку «Сохранить» или «Сохранить как».

После заполнения формы необходимо распечатать готовый документ. Это повлечёт за собой необходимость выполнения череды действий, которую мы можем избежать путём создания кнопки вывода документа на печать.

Для создания кнопки, отправляющей документ на печать необходим сам документ, который будет необходимо распечатать, и включенный режим разработчика, который поможет создать кнопку и настроить её. Для включения режима разработчика потребуется во вкладке «Файл» перейти в «Параметры», после чего, выбрав «Настроить ленту» поставить галочку напротив надписи «Разработчик» (Рис. 1). Стоит отметить, что в более ранних версиях Word настройка ленты происходит в пункте меню «Основные» вкладки «Параметры Word». Там также необходимо поставить галочку напротив соответствующей надписи.

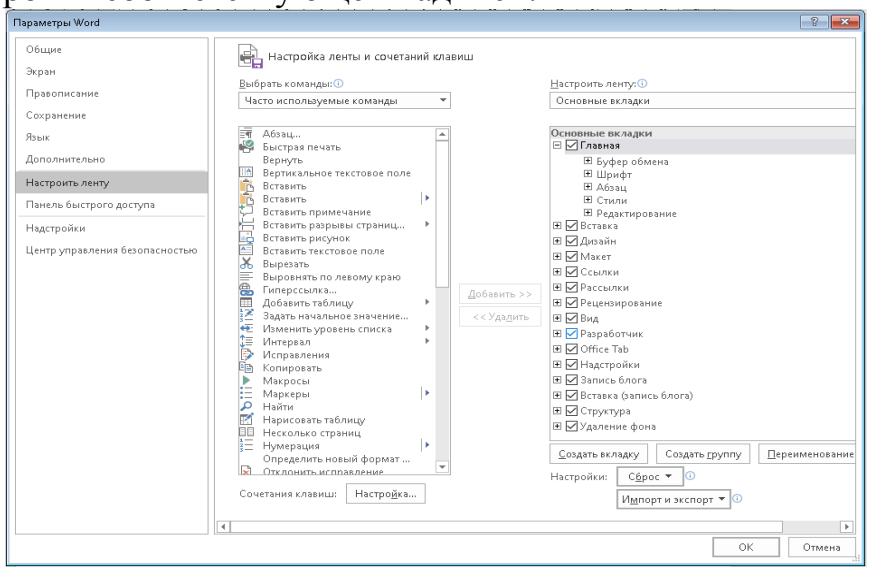

Рисунок 1 – Включение режима разработчика

После этих действий появится вкладка «Разработчик», благодаря которой появится возможность создавать и настраивать макросы. Макросы – это последовательность команд на языке Visual Basic которая будет выполняться при вызове макроса. Для начала создадим макрос, добавляющий кнопку, с помощью которой мы будем печатать документ. Во вкладе «Разработчик» необходимо нажать кнопку «Макросы», после чего задать имя и выбрать «Создать», для редактирования макроса в будущем можно, переходя по тому же пути до кнопки «Макросы», выбрать созданные макросы и редактировать их. В макросе нужно написать следующее:

```
Sub Test()
Dim shp As Word.InlineShape
Documents("Dov.docm").Activate
Set shp =
Word.ActiveDocument.Content.InlineShapes.AddOLEControl(ClassType:="Form
s.CommandButton.1")
shp.OLEFormat.Object.Caption = "Печать"
End Sub
```

Этот макрос создаёт кнопку с надписью «Печать». Осталось снова перейти в «Макросы» и нажать кнопку «Выполнить». Далее нужно настроить действия самой кнопки. Для этого во вкладке «Разработчик» нужно выбрать «Режим конструктора», и, после нажатия правой кнопкой мыши на кнопку, нажать на «Просмотреть код». В появившемся окне нужно записать следующие команды:

```
ActiveDocument.Range(Start:=0, End:=0).Select
Word.ActiveDocument.PrintPreview
```

Первая команда выделяет лист, на котором расположен документ. Это необходимо для работы второй команды, отвечающей непосредственно за вывод на печать.

Теперь, когда кнопка настроена, осталось выйти из «Режима конструктора», сохранить файл, и кнопка, при нажатии, выведет предпросмотр печати.

Типовая межотраслевая форма № М-2а
Утверждена постановлением Госкомстата России от 30.10.97 № 71а

	КОДЫ
Форма по ОКУД	
по ОКПО	

Организация _____
наименование организации

ДОВЕРЕННОСТЬ № _____

Дата выдачи: _____
Доверенность действительна по: _____

_____
наименование потребителя и его адрес

_____
наименование плательщика и его адрес

_____
наименование банка

Доверенность выдана _____ _____
должность фамилия, имя, отчество

Паспорт: серия _____ _____
Кем выдан _____
Дата выдачи _____

На получение от _____
наименование поставщика

материальных ценностей по _____
наименование, номер и дата документа

Перечень товарно-материальных ценностей, подлежащих получению

Номер по порядку	Материальные ценности	Единица измерения	Количество (прописью)
	2	3	4

Подпись лица, получившего доверенность _____ удостоверяем

Руководитель _____ _____
подпись расшифровка подписи

М.П.

Главный бухгалтер _____ _____
подпись расшифровка подписи

Следующим шагом будет создание самой формы, которая позволит пользователю с наибольшим удобством заполнять документ. Создадим пустую форму. Для этого необходимо во вкладке «Разработчик» перейти в Visual Basic, после чего нажать кнопку UserForm (Рис. 2).

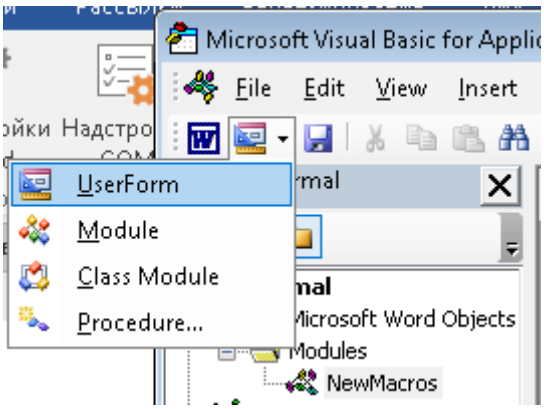

Рисунок 2 – Создание пустой формы

После этого необходимо открыть список инструментов. Для этого нужно нажать на кнопку Toolbox (Рис .3), после чего откроется окно с инструментами (Рис. 4), которые помогут добавлять на форму всё необходимое для конкретного документа.

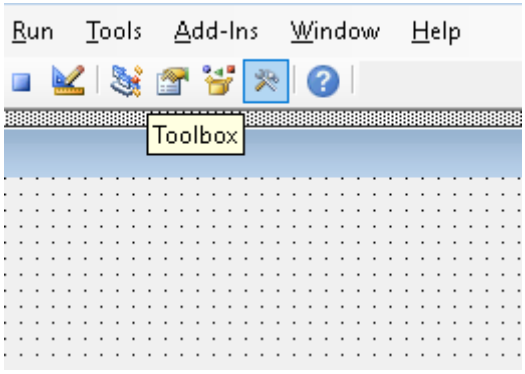

Рисунок 3 – Кнопка Toolbox

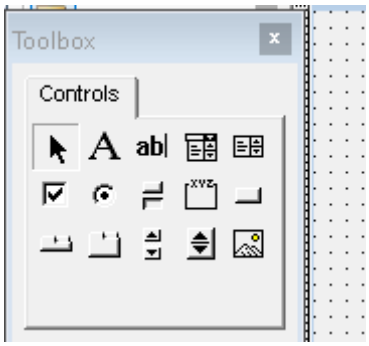

Рисунок 4 – Toolbox

Основные инструменты, которые используются на любой форме являются Label, на иконке которого видна буква «А», позволяющий разместить текст на форме, Textbox, иконка с буквами «аб», позволяющий считать текст, введённый пользователем документа, и CommandButton, четвёртый инструмент слева в первом столбце, позволяющий создать кнопку на самой форме. С помощью инструментов создадим макет формы, с которым будет взаимодействовать пользователь (Рис 5).

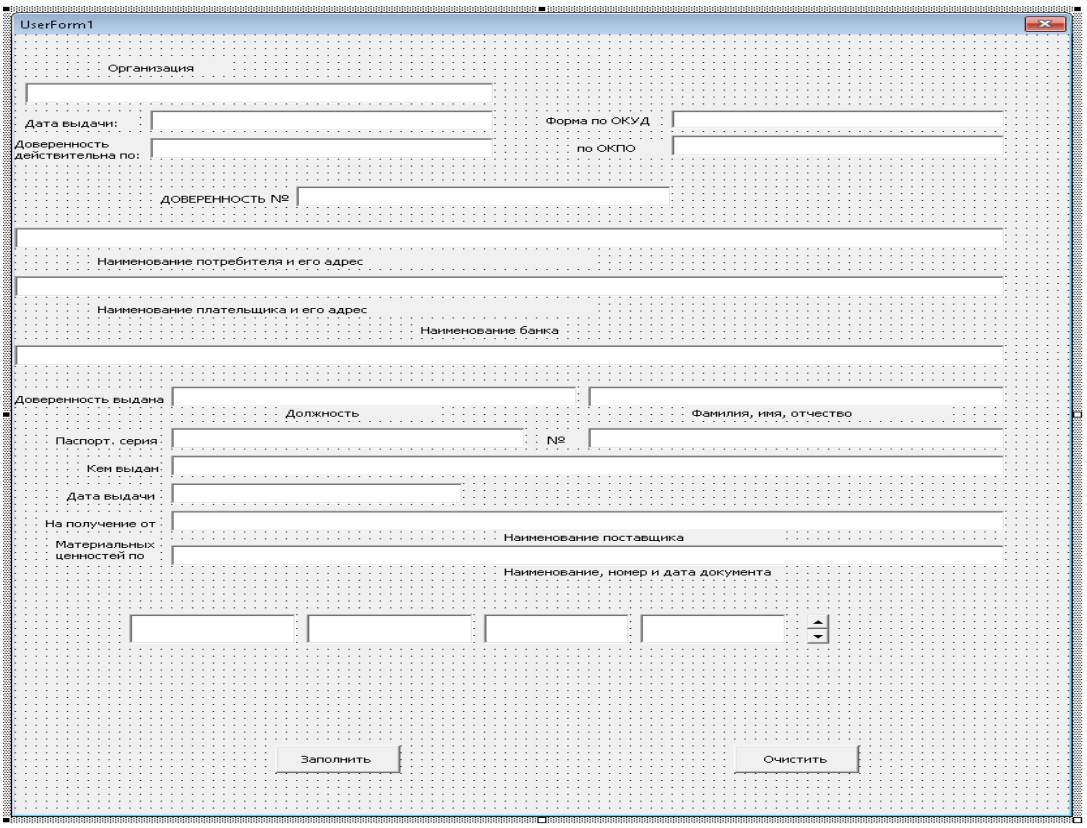

Рисунок 5 – Форма доверенности

Теперь следует настроить взаимодействие формы и самого документа. Сделать это можно с помощью полей. Чтобы создать поле необходимо выделить текст и, во вкладке Вставка, выбрать Экспресс-блоки, в открывшемся окне выбрать поле. В появившемся окне необходимо выбрать категорию «Автоматизация», затем поле «DocVariable», ввести имя поля, и нажать «ОК». Таким образом, мы создадим поле, с которым можно взаимодействовать посредством макросов, чтобы с помощью формы можно было обратиться к конкретному месту документа.

Теперь, всё готово для перехода к написанию макроса формы. Основной код в макросе будет сосредоточен в функциях кнопок «Заполнить» и «Очистить», т.к. большинство действий будет происходить именно при нажатии кнопок. Функции собой последовательность команд, выполняемых при определённых действиях, совершаемых над элементом интерфейса, заключенных между «Private sub <название элемента> _ <условие действия>» и «End sub». Чтобы перейти к функции любого элемента интерфейса нужно два раза нажать на элемент. Над окном написания макросов можно выбрать элемент и условие действия (Рис. 6).

Рисунок 6 – Окно написания макросов

Теперь необходимо понять, как именно связать элементы формы и документа. Для начала нужно научиться обращаться к элементам документа. Необходимо выделить необходимый документ с помощью команды:

Documents("название документа").Activate

К созданным ранее полям, с целью замены их содержания, можно обратиться с помощью команды:

ActiveDocument.Variables.Item("название поля") =«текст»

Так мы можем присвоить полю содержание, которое мы получим из полей Textbox:

ActiveDocument.Variables.Item("название поля") =<название TextBox>.Text

Название, как и всю информацию об элементах можно узнать в левой нижней панели. При работе с полями требуется описать способ их обновления, для отображения актуальных значений. Это реализуется благодаря команде ActiveDocument.Fields.Update, которую следует написать в конец функций кнопок «Заполнить» и «Очистить».

В некоторых документах могут потребоваться переменные и массивы, чтобы создать их нужно использовать команды:

Dim <название переменной> As Integer

Dim <название массива>(<количество элементов>, <количество элементов>,…) As String

Для их корректной работы лучше всего писать их в самом начале окна написания макросов. Могут помочь, в том числе и циклы, с помощью которых можно улучшить форму и повысить качество заполнения документа, такие как Do While и For:

Do While(<условие продолжения цикла>)
<тело цикла>
Loop

For <начальное значение> To <конечное значение>
<тело цикла>
Next

В макросе формы документа «Доверенность» основная часть кода также сосредоточена в функциях двух этих кнопок. На примере функции кнопки «Заполнить» можно понять, как работают операторы и в каких случаях их можно использовать.

Функция кнопки «Заполнить»:

```
Private Sub CommandButton1_Click()
Documents("Dov.docm").Activate
        itera = 0
        Do While (Not (mass(itera, 1) = ""))
        itera = itera + 1
        Loop
        ActiveDocument.Tables(8).Select

Do While (Not (ActiveDocument.Tables(8).Rows.Count = 3))
ActiveDocument.Tables(8).Rows.Last.Cells.Delete
Loop
ActiveDocument.Tables(8).Select
```

```
With ActiveDocument.Variables

.Item("Org") = " "
.Item("Nom") = " "
 .Item("DataV1") = " "
  .Item("Deystv") = " "
   .Item("OKU") = " "
    .Item("OKP") = " "
     .Item("NPot") = " "
      .Item("NPla") = " "
       .Item("Bank") = " "
        .Item("Vidana") = " "
         .Item("FIO") = " "
         .Item("Pass") = " "
         .Item("No2") = " "
          .Item("Kem") = " "
          .Item("DataV2") = " "
          .Item("NaP") = " "
          .Item("Mat") = " "

End With


  If (Not (TextBox1.Text = "")) Then
  ActiveDocument.Variables.Item("Org") = TextBox1.Text
  End If
  If (Not (TextBox17.Text = "")) Then
  ActiveDocument.Variables.Item("Nom") = TextBox17.Text
  End If
  If (Not (TextBox2.Text = "")) Then
  ActiveDocument.Variables.Item("DataV1") = TextBox2.Text
  End If
  If (Not (TextBox3.Text = "")) Then
  ActiveDocument.Variables.Item("Deystv") = TextBox3.Text
  End If
  If (Not (TextBox7.Text = "")) Then
  ActiveDocument.Variables.Item("OKU") = TextBox7.Text
  End If
  If (Not (TextBox6.Text = "")) Then
  ActiveDocument.Variables.Item("OKP") = TextBox6.Text
  End If
  If (Not (TextBox4.Text = "")) Then
  ActiveDocument.Variables.Item("NPot") = TextBox4.Text
  End If
```

```
    If (Not (TextBox5.Text = "")) Then
    ActiveDocument.Variables.Item("NPla") = TextBox5.Text
    End If
    If (Not (TextBox8.Text = "")) Then
      ActiveDocument.Variables.Item("Bank") = TextBox8.Text
      End If
      If (Not (TextBox9.Text = "")) Then
       ActiveDocument.Variables.Item("Vidana") = TextBox9.Text
       End If
       If (Not (TextBox10.Text = "")) Then
        ActiveDocument.Variables.Item("FIO") = TextBox10.Text
        End If
        If (Not (TextBox11.Text = "")) Then
        ActiveDocument.Variables.Item("Pass") = TextBox11.Text
        End If
        If (Not (TextBox12.Text = "")) Then
        ActiveDocument.Variables.Item("No2") = TextBox12.Text
        End If
        If (Not (TextBox13.Text = "")) Then
         ActiveDocument.Variables.Item("Kem") = TextBox13.Text
         End If
         If (Not (TextBox14.Text = "")) Then
         ActiveDocument.Variables.Item("DataV2") = TextBox14.Text
         End If
         If (Not (TextBox15.Text = "")) Then
         ActiveDocument.Variables.Item("NaP") = TextBox15.Text
         End If
         If (Not (TextBox16.Text = "")) Then
         ActiveDocument.Variables.Item("Mat") = TextBox16.Text
         End If

   For i = 0 To ListBox1.ListCount - 1
   Documents("Dov.docm").Activate
   If (Not (ListBox1.ListCount = 0)) Then
   ActiveDocument.Tables(8).Select
   If Not (i = 0) Then
   Selection.InsertRowsBelow 1
   ActiveDocument.Tables(8).Cell(4 + i, 2).Range = ListBox1.List(i, 0)
   ActiveDocument.Tables(8).Cell(4 + i, 3).Range = ListBox1.List(i, 1)
   ActiveDocument.Tables(8).Cell(4 + i, 4).Range = ListBox1.List(i, 2)
   ActiveDocument.Tables(8).Cell(4 + i, 5).Range = ListBox1.List(i, 3)
   End If
   End If
   Next
ActiveDocument.Fields.Update
```

```
Unload Me
End Sub
```

Стоит уточнить, что все начальные настройки формы, задающиеся при её запуске, происходят в функции формы, отвечающей за инициализацию, в данном случае, это функция userform_initialise():

```
Private Sub UserForm_Initialize()
itera = 0
For i = 0 To 10
mass(i, 1) = ""
mass(i, 2) = ""
mass(i, 3) = ""
mass(i, 4) = ""
Next
End Sub
```

Здесь происходит присвоение переменной itera, заданной в начале макроса, значения «0», а также присвоение пустого значения полям массива mass. В зависимости от количества задаваемых при запуске формы значений можно дополнять эту функцию по мере необходимости.

Функция кнопки «Заполнить» начинается с выделения документа с помощью команды :

```
Documents("Dov.docm").Activate
```

Это действие необходимо, поскольку при нажатии кнопки в документе выделение переходит на кнопку, и обращение к элементам документа может не работать.

Далее происходит заполнение переменной itera в соответствии с количеством элементов массива. Массив в данный момент времени заполнен, поскольку все действия происходят после заполнения формы. Следующий шаг – работа с таблицей. Есть несколько операторой, которые необходимо использовать в подобных случаях:

ActiveDocument.Tables(<номер>).Rows.Count – количество строк таблицы;

ActiveDocument.Tables(<номер>).Rows.Last.Cells.Delete – удаление последней строки;

ActiveDocument.Tables(<номер>).Select – выбор таблицы;

Selection.InsertRowsBelow 1 – добавление строки.

Цикл, описанный в начале функции, очищает таблицу перед заполнением, удаляя строки до тех пор, пока не останется только шапка таблицы, после чего добавляет строку.

После этого идет заполнение полей документа в соответствии с полями формы, а затем заполнение таблицы. Здесь пригодилась переменная itera которая была посчитана в начале функции. С помощью цикла For и переменной itera можно посчитать точное количество необходимых строк,

которые нужно добавить к таблице. Чтобы добавить текст в новую строку используется команда:

ActiveDocument.Tables(<номер>).Cell(<строка>, <ячейка>).Range = <текст>

Для кнопки очистки нам требуется только заменить все поля документа пробелом. Важно не оставлять пустые значения переменным, иначе вместо пустого поля появится сообщение об ошибке.

Функция кнопки «Очистить»:

```
Private Sub CommandButton2_Click()

With ActiveDocument.Variables

.Item("Org") = " "
.Item("Nom") = " "
 .Item("DataV1") = " "
  .Item("Deystv") = " "
   .Item("OKU") = " "
    .Item("OKP") = " "
     .Item("NPot") = " "
      .Item("NPla") = " "
       .Item("Bank") = " "
        .Item("Vidana") = " "
         .Item("FIO") = " "
         .Item("Pass") = " "
         .Item("No2") = " "
          .Item("Kem") = " "
          .Item("DataV2") = " "
          .Item("NaP") = " "
          .Item("Mat") = " "

          End With
          Do While (Not (ActiveDocument.Tables(8).Rows.Count = 3))
ActiveDocument.Tables(8).Rows.Last.Cells.Delete
Loop
   ActiveDocument.Tables(8).Select
      Selection.InsertRowsBelow 1
 ActiveDocument.Fields.Update
 Unload Me
End Sub
```

В конце функций кнопок, для удобной работы с ними, можно вставить команду Unload Me, которая разгрузит форму и закроет её.

Существует и иной способ заполнения списка в форме. Осуществляется он с помощью инструмента «ListBox» (Рис. 7), первый инструмент во второй строке toolbox, который позволяет записывать данные в табличном виде внутри самой формы, после чего проецировать их на наш документ.

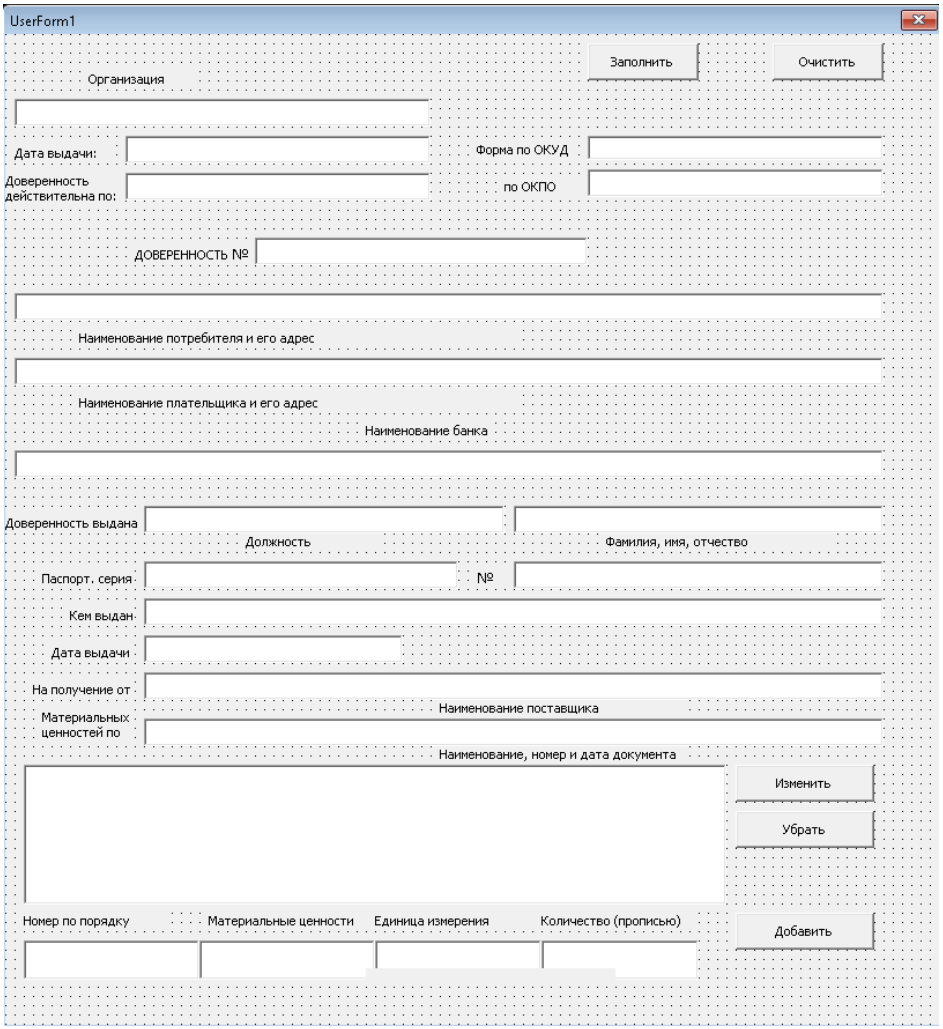

Рисунок 7 – Форма с использованием ListBox

Отличается такой вариант легкостью использования, поскольку позволяет просматривать большие списки с большей лёгкостью. Для создания такой формы необходимо разобраться, как использовать ListBox.

Для начала следует разметить ListBox на форме. Следующим шагом будет изменение функции UserForm_Initialize():

```
Private Sub UserForm_Initialize()
Documents("Dov.docm").Activate
itera = 0
ListBox1.AddItem "[eq"
ListBox1.List(ListBox1.ListCount - 1, 0) = "Номер по порядку"
ListBox1.List(ListBox1.ListCount - 1, 1) = "Материальные ценности"
ListBox1.List(ListBox1.ListCount - 1, 2) = "Единица измерения"
ListBox1.List(ListBox1.ListCount - 1, 3) =  "Количество (прописью)"
```

```
        Do While (Not itera = (ActiveDocument.Tables(8).Rows.Count - 3))
        ListBox1.AddItem ""
        ListBox1.List(ListBox1.ListCount - 1, 0) =
Left(ActiveDocument.Tables(8).Cell(4 + itera, 2).Range,
Len(ActiveDocument.Tables(8).Cell(4 + itera, 2).Range) - 2)
        ListBox1.List(ListBox1.ListCount - 1, 1) =
Left(ActiveDocument.Tables(8).Cell(4 + itera, 3).Range,
Len(ActiveDocument.Tables(8).Cell(4 + itera, 3).Range) - 2)
        ListBox1.List(ListBox1.ListCount - 1, 2) =
Left(ActiveDocument.Tables(8).Cell(4 + itera, 4).Range,
Len(ActiveDocument.Tables(8).Cell(4 + itera, 4).Range) - 2)
        ListBox1.List(ListBox1.ListCount - 1, 3) =
Left(ActiveDocument.Tables(8).Cell(4 + itera, 5).Range,
Len(ActiveDocument.Tables(8).Cell(4 + itera, 5).Range) - 2)
        itera = itera + 1
        Loop
        TextBox21.Text = ""
        TextBox22.Text = ""
        TextBox23.Text = ""
        Next
        End Sub
```

Эта функция позволит задать необходимые поля для ListBox, путём переноса их из документа. Следует разобрать команды, которые будут использованы для работы с ListBox.

<Название ListBox>.Additem - позволяет создать новое поле в ListBox

<Название ListBox>.ListCount – возвращает количество строк ListBox

<Название ListBox>.List(<Строка>,<Столбец>) – изменяет значение выбранной ячейки Listbox

<Название ListBox>.ListIndex – возвращает значение выделенной строки ListBox

Для настройки количества столбцов можно использовать метод <Название ListBox>.ColumnCount = <Значение>.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К СОЗДАНИЮ АВТОМАТИЗИРОВАННОГО ДОКУМЕНТА В EXCEL

ПОДГОТОВИТЕЛЬНЫЙ ЭТАП

Для того, чтобы автоматизировать документ, нужно проанализировать его структуру. Так, документ «Доверенность» мы можем условно разделить документ на две таблицы: таблица, состоящая из «ленточных» данных (шапка документа) и таблица, состоящая из основных данных (тело документа) (рис. 1).

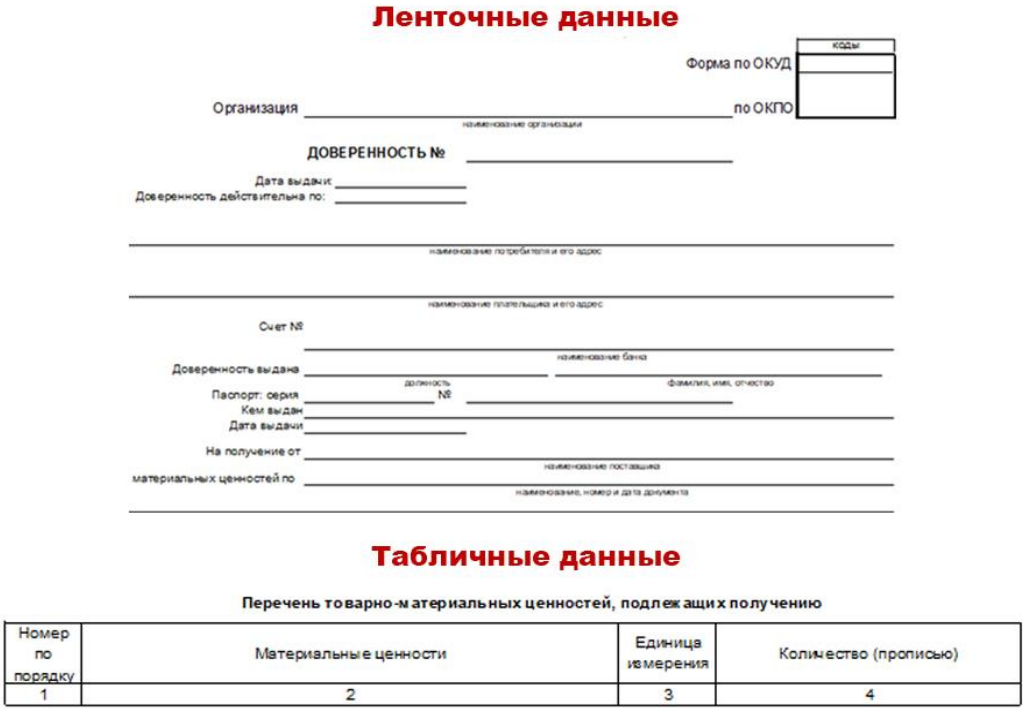

Рисунок 1. – Структура документа

Реализуем вышесказанное, создав листы в Excel для дальнейшей работы с ними. (рис. 2)

Рисунок 2. – Создание листов

Для того, чтобы начать работу с макросами и консолью разработчика, нужно добавить вкладку «Разработчик» в панель быстрого доступа: Файл>Параметры>Настроить ленту>Основные вкладки>Разработчик (поставить галочку) (рис. 3).

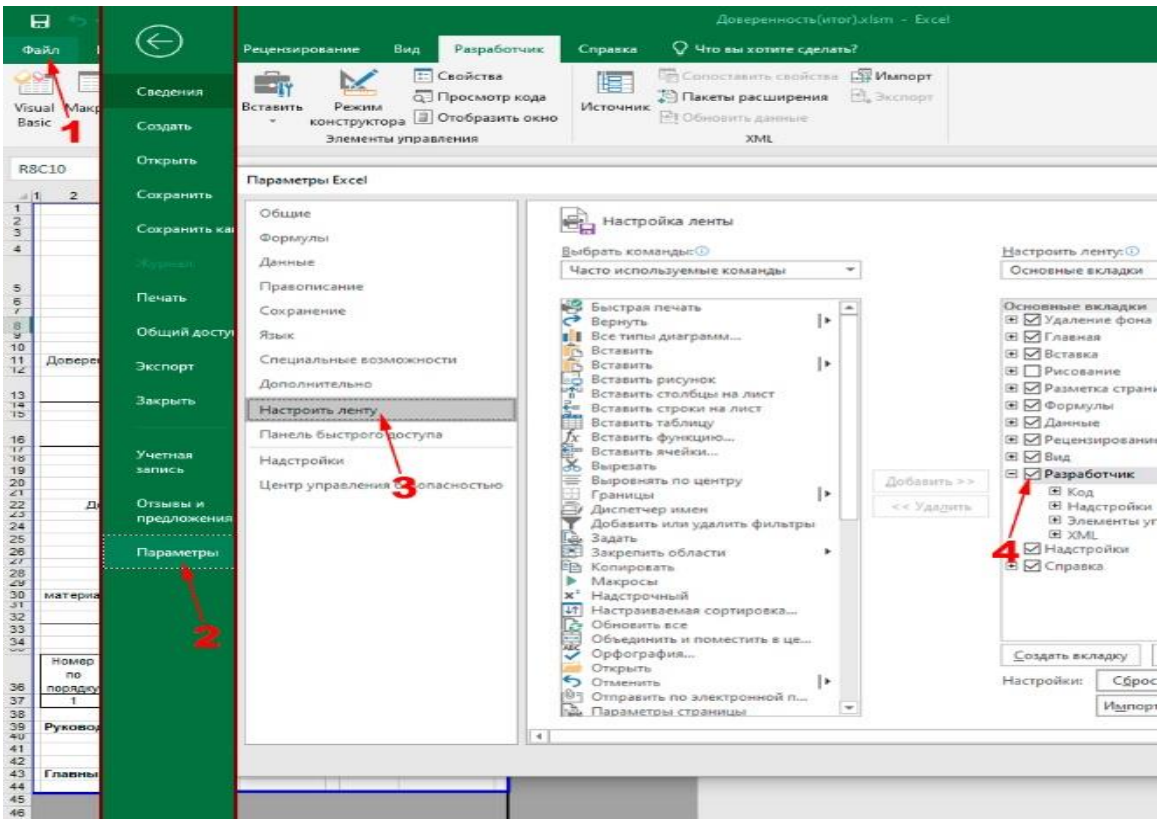

Рисунок 3. – Добавление вкладки "Разработчик"

Также, для удобства понимания адреса ссылки ячеек можно сменить стиль ссылок: Файл>Параметры>Формулы>«Стиль ссылок R1C1».

! Во избежание утери данных рекомендуется на данном этапе сохранить документ в формате «Книга с поддержкой макросов» (рис. 4).

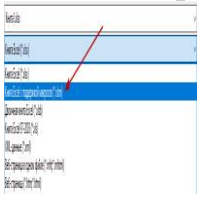

Рисунок 4. – Сохранение документа

РАБОТА В ПАНЕЛИ РАЗРАБОТЧИКА ИСПОЛЬЗОВАНИЕ ИНСТРУМЕНТОВ VBA

Для работы с VBA требуется открыть ранее подключенную вкладку «Разработчик» и перейти в режим просмотра кода (рис. 5).

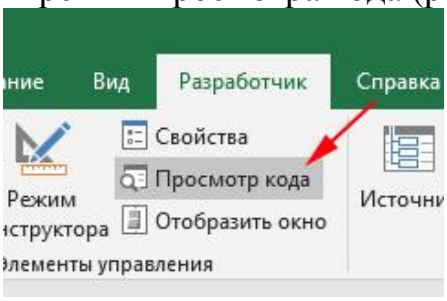

Рисунок 5. – Режим просмотра кода

СОЗДАНИЕ ФУНКЦИОНАЛЬНЫХ ФОРМ

Форма – элемент ООП, предоставляющий более удобный способ взаимодействия пользователя с документом посредством просмотра и правки его данных с помощью конкретных объектов. Ее создание происходит с помощью панели инструментов (рис. 6).

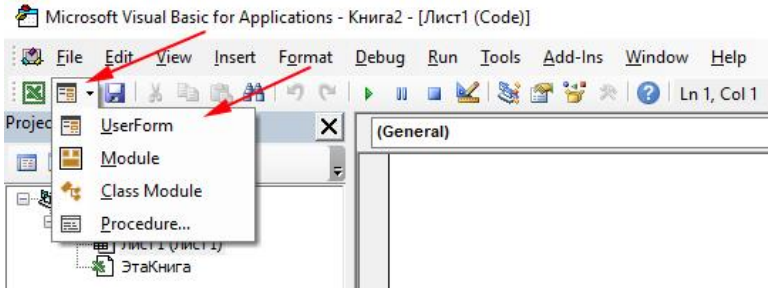

Рисунок 6. – Создание формы

О наличии созданной формы говорит появление раздела Forms в проекте и, непосредственно, появившаяся форма в редакторе (рис. 7).

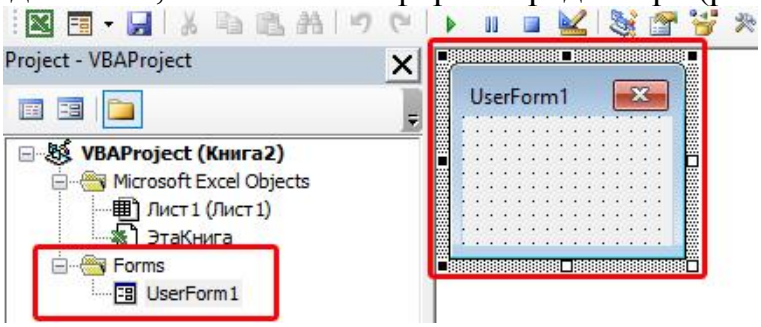

Рисунок 7. – Форма

Для редактирования формы используется «Ящик инструментов (ToolBox)» и окно свойств объекта:

«Ящик инструментов (ToolBox)» (рис. 8). – Позволяет добавить объекты на форму. Поскольку выбор невелик, можно скачать дополнительные объекты, однако их достаточно, чтобы разработать форму, требуемую в рамках задания.

Рисунок 8. – "Ящик инструментов (ToolBox)"

Ниже представлена таблица с описанием функций основных объектов и элементов управления (табл. 1):

Таблица 1. – Назначение объектов ToolBox

Объект	**Назначение**
Select Objects	Это единственный элемент в ящике инструментов, который не рисует элемент управления. При выборе этого инструмента можно только изменить размер или положение элемента управления, который уже нарисован на форме.
Label	Позволяет ввести текст, который не доступен для изменения пользователем, например подпись под рисунком.
TextBox	Содержит текст, который пользователь может ввести или изменить.
ComboBox	Позволяет рисовать комбинацию окна со списком и текстового поля. Пользователь может или выбрать элемент из списка, или ввести значение в текстовом поле
ListBox	Используется для отображения списка элементов, которые доступны для выбора пользователем. Если список содержит больше элементов, чем в нем отображается за раз, его можно прокрутить.
CheckBox	Создает поле, позволяющее пользователю указать, является ли чтонибудь истиной или ложью, либо отображающее несколько вариантов, когда у пользователя есть возможность выбора нескольких альтернатив.
OptionButton	Позволяет отобразить несколько

	вариантов, из которых для выбора пользователем доступен только один.
ToggleButton	Создает кнопку включения и выключения.
Frame	Позволяет создавать графическое или функциональное группирование для элементов управления. Чтобы сгруппировать элементы управления, нарисуйте сначала рамку, а затем нарисуйте внутри рамки элементы управления.
TabStrip	Позволяет определять несколько страниц для одной области окна или диалогового окна в приложении.
MultiPage	Представляет несколько экранов с информацией как единый набор.
ScrollBar	Предоставляет графический инструмент для быстрой навигации по длинному списку элементов или большому количеству информации, для индикации текущего положения на шкале или в качестве входного устройства либо индикатора скорости или количества.
SpinButton	Управление спиннером, которое можно использовать с помощью другого управления для приращения и расшифления номеров. Он также применяется для прокрутки назад и вперед по диапазону значений или списку элементов.
Image	Отображает графическое изображение из точечного рисунка, значка или метафайла на форме. Изображения, отображаемые в элементе управления Image, могут быть лишь декоративными и занимают меньше ресурсов, чем элемент PictureBox.

После добавления элемента на форму при двойном клике на него можно задать процедуру выполнения, а также условия его выполнения – событие. Здесь хранятся все процедуры, выполняемые в форме (рис. 9).

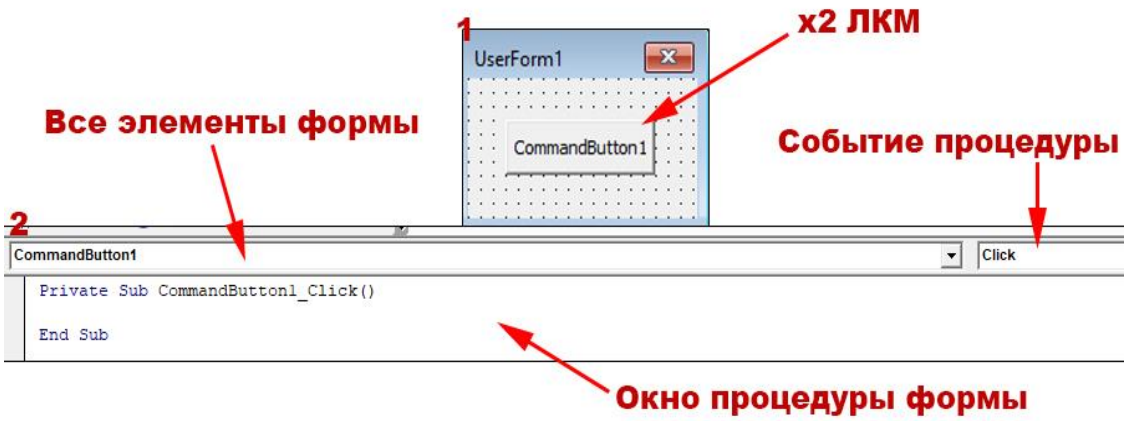

Рисунок 9. – Процедуры формы

– Окно свойств объекта (рис. 10). – Здесь можно управлять свойствами объектов. Например, использование свойств Top, Left позволяет сдвигать объект в двухмерной системе координат YoX, свойство Font позволяет настраивать шрифт объекта, а Visible в значении True – вовсе скрывает его от отображения для пользователя.

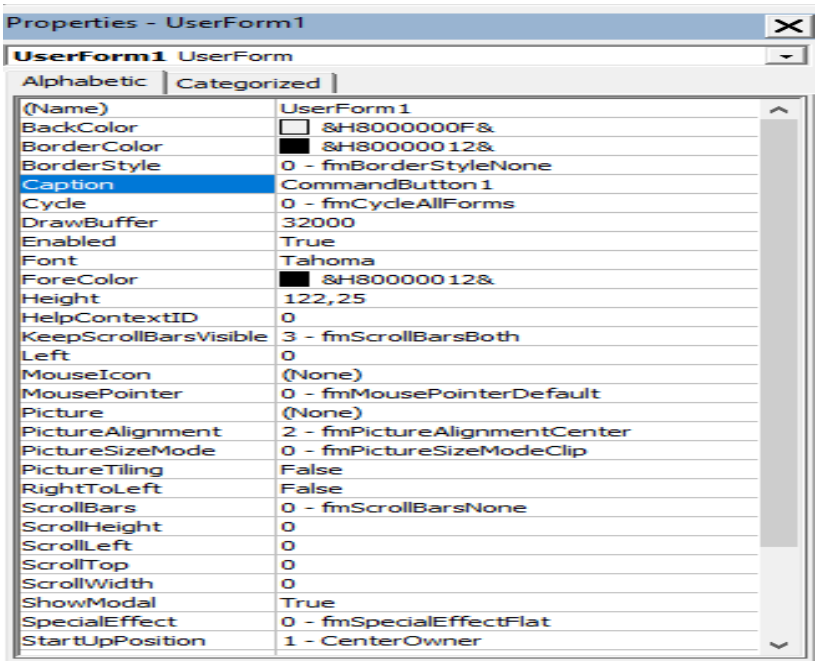

Рисунок 10. – Окно свойств объекта

Для того, чтобы запустить выполнение процедур формы, нужно нажать на кнопку Run(F5), остановить – Break, сбросить – Reset (рис. 11).

Рисунок 11. – Выполнение формы

Также, возможен запуск с отладкой с точками останова (Break Points), чтобы обнаруживать и локализовывать возможные ошибки в коде. Для этого потребуется поставить эти точки на поле, расположенном вертикально слева от редактора кода и нажать F8 (рис. 12).

```
Private Sub CommandButton3_Click() 'добавление строки'
Dim i, count As Integer
i = 38
count = 1
Do While Worksheets("Документ").Cells(i, 2) <> "" And i < 58
    i = i + 1
    count = count + 1
Loop
```

Рисунок 12. – Отладка кода по точкам останова

СОЗДАНИЕ МАКРОСОВ VBA

Макрос – это запрограммированная последовательность действий, записанная на языке программирования.

Для того, чтобы создать макрос, можно воспользоваться двумя способами:

1) Записать макрос (рис. 13). – Для этого потребуется во вкладке «Разработчик» на панели инструментов по нажатии клавиши «Запись макроса» выполнить действия, желаемые для многократного воспроизведения по вызову. Данный способ не желателен к использованию, поскольку не позволяет получить требуемого эффекта в полной его мере.

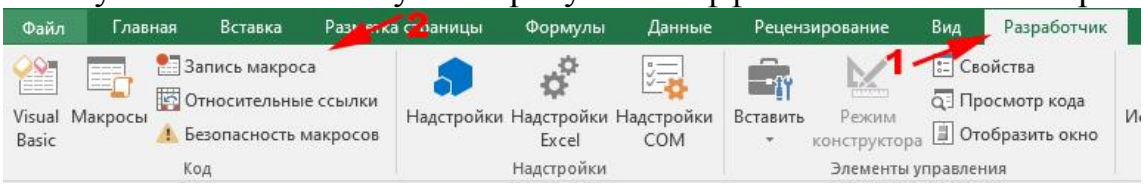

Рисунок 13. — Запись макроса

2) Написать макрос (рис. 14). – Более продвинутый способ создания макроса, заключающийся в полной подконтрольности алгоритма макроса, поскольку он описывается посредством языка VBA. Для этого требуется во вкладке «Разработчик» на панели инструментов нажать кнопку «Макросы» и создать макрос с уникальным именем (без использования пробелом), после чего Вы будете перенаправлены в редактор кода VBA.

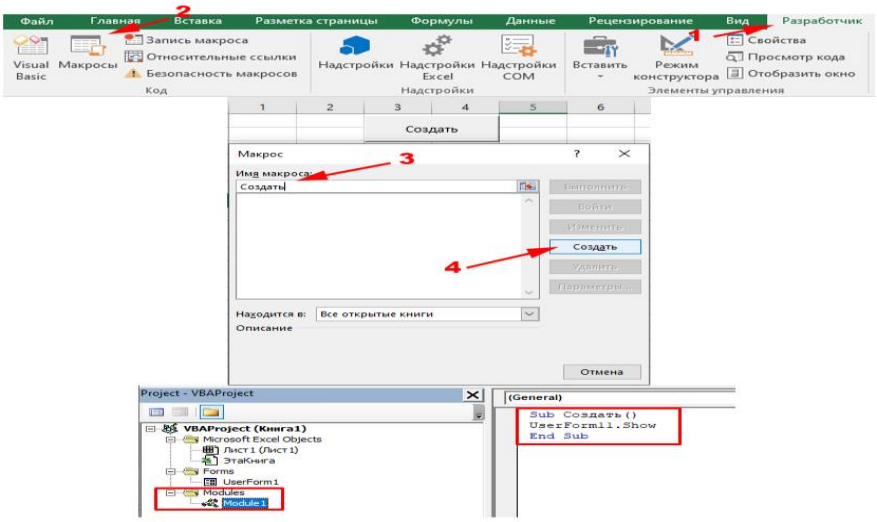

Рисунок 14. – Написание макроса

Макросы нам понадобятся для связывания кнопок документа с формами.

ОСНОВЫ СИНТАКСИСА VBA

Для того, чтобы делать формы и макросы функциональными, нужно знать базу, требуемую для написания кода, поэтому ниже представлена вспомогательная таблица, требуемая к ознакомлению (табл. 2):

Действие	Пример
Объявление переменной	Dim j As Integer j = 0
Цикл For	For i = 1 To N j=j+1 Next
Цикл While	Do While i<5 i=i+1 Loop
Обращение к ячейкам листа	WorkSheets(“ Лист1”).Cells(1,2) или WorkSheets(“ Лист1”).Cells(“A2”) – обращаемся ко второму столбцу первой строки
Действие	**Пример**
Обращение к объекту формы (например, к текстовому полю)	UserForm1.TextBox1.Value=2
Обращение к диапазону ячеек листа	WorkSheets(“ Лист1”).Range(Cells(i,1),Cells(i,5)) Или WorkSheets(“ Лист1”).Range(“A1:A5”), где i – любое число
Объединение ячеек листа	Worksheets("Документ").Range(Cells(1, 3), Cells(1, 12)).MergeCells = True
Добавление границ ячейке листа	Worksheets("Документ").Cells(1, 2).Borders.LineStyle = xlContinuous
Вставка строки в листе со сдвигом ячеек	Worksheets("Документ").Rows(i). Insert Shift:= xlShiftDown – сдвиг исходных ячеек вниз xlShiftToRight – сдвиг исходных ячеек вправо xlShiftToLeft – сдвиг исходных ячеек влево, где i – любое число
Отобразить форму	UserForm1.Show
Скрыть форму	UserForm2.Hide

Присвоить значение ячейки листа объекту формы	UserForm1.TextBox1.Value = Worksheets("Документ").Cells(24, 5).Value
Добавить в выпадающий список (ComboBox)	ComboBox1.AddItem Worksheets("Лист1).Cells(i, 1), где i – любое число
Отобразить всплывающее окно	Dim a AS Integer a = MsgBox("Успешно удалено", 64, "")
Преобразование типов	CInt – преобразует в int CStr – преобразует в string
Вывод окна печати документа	ActiveSheet.PrintPreview
Комментарий	‘’

ДОБАВЛЕНИЕ КНОПОК

Финальным этапом автоматизации документа является добавление кнопок работы с ним. Для этого нам потребуется воспользоваться вставкой элементов управления (рис. 15).

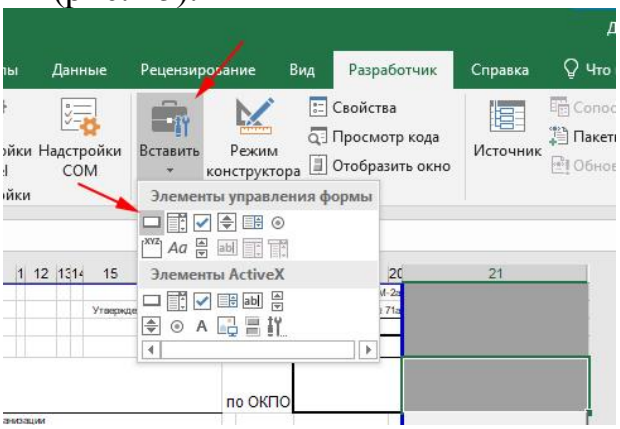

Рисунок 15. – Добавление кнопки

Чтобы кнопка выполняла определенную функцию, нужно назначить ей ранее созданный макрос. (рис. 16).

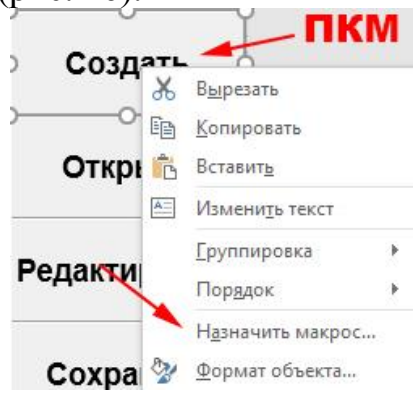

Рисунок 16. – Назначение макроса

Таким образом, получили следующий интерфейс (рис. 17):

Рисунок 17. – Кнопочный интерфейс

СОЗДАНИЕ СПРАВОЧНИКОВ

Создаем новые рабочие листы и называем их «Справочник(«Предмет справочника»)» (рис. 18).

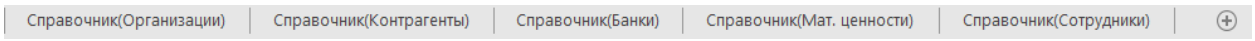

Рисунок 18. – Справочники

Выделяем нужное количество ячеек в соответствии с количеством столбцов в шапке справочника и во вкладке «Вставка» нажимаем «Таблица». Таким образом, мы создали «умную» таблицу, позволяющую автоматически расширять количество полей таблицы и сортировать их (рис. 19).

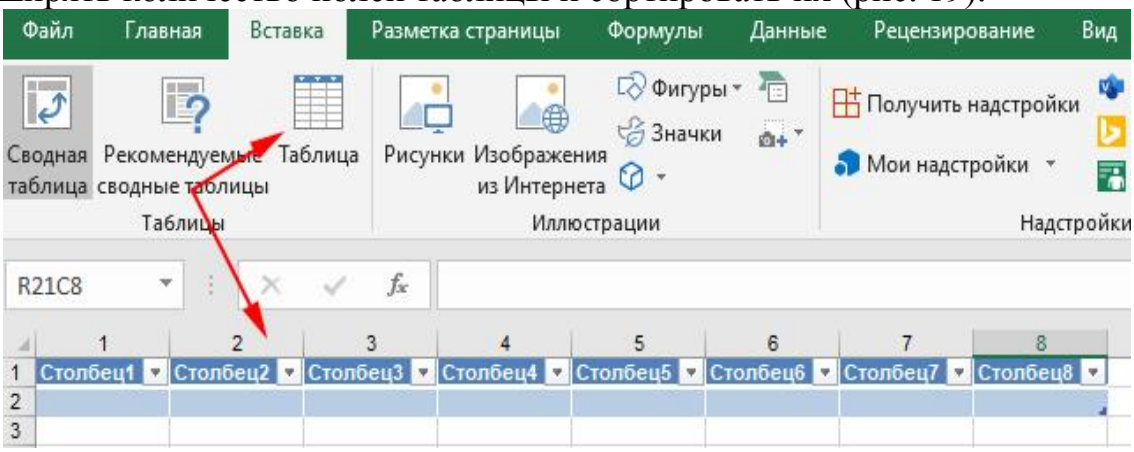

Рисунок 19. – Умная таблица

Нередко поля одних справочников могут содержать поля других справочников. Для того, чтобы их соединить, требуются кодовые поля таблицы и возможность их выбора посредством выпадающего списка подобно MS Acces. Однако, поскольку MS Excel не предусматривает функции создания связей «многие ко многим» (рис. 20), а также других полезных функций, как в MS Access, воспользуемся альтернативным способом.

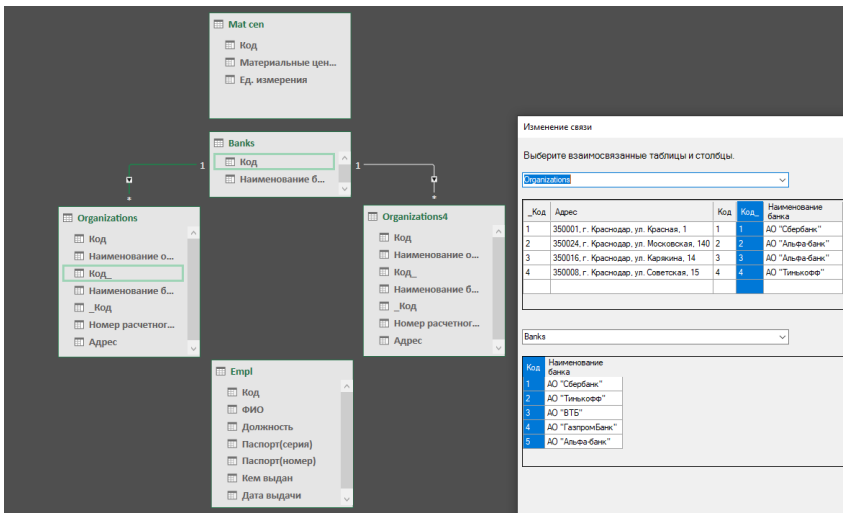

Рисунок 20. – Power Pivot. Связь 1 ко 1

Для того, чтобы реализовать выпадающий список нам потребуется выделить нужный столбец, перейти во вкладку «Данные», затем выбрать «Проверка данных». В диалоговом окне потребуется выбрать тип данных «Список», поставить галочку «Игнорировать пустые ячейки», «Список допустимых значений» и указать источник полей выпадающего списка (нужный столбец другого листа) (рис. 21).

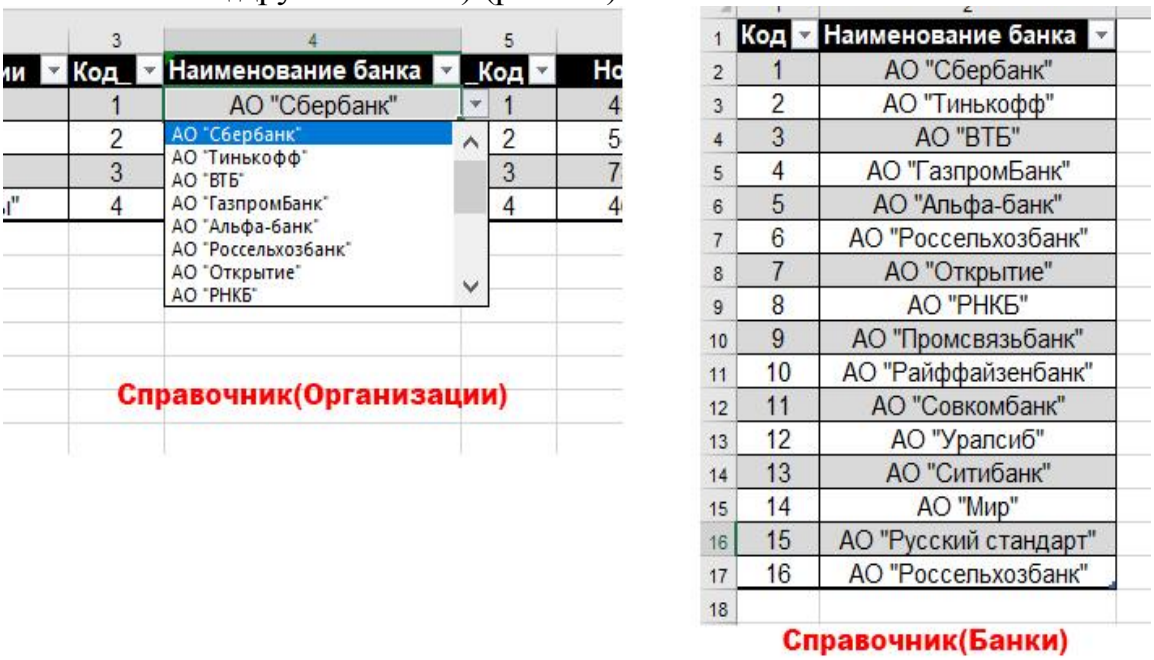

Рисунок 21. – Выпадающий список из справочника

ПРИМЕР РЕАЛИЗАЦИИ ТЕХНОЛОГИИ

Воспользуемся ранее вышесказанным и реализуем кнопку создания документа, рассмотрим ее функцию, а также функции других кнопок. Так, согласно первому разделу у нас бланк «Доверенность», два раздела документа (шапка – ленточные данные, тело – табличные данные) и 5 справочников (рис. 22).

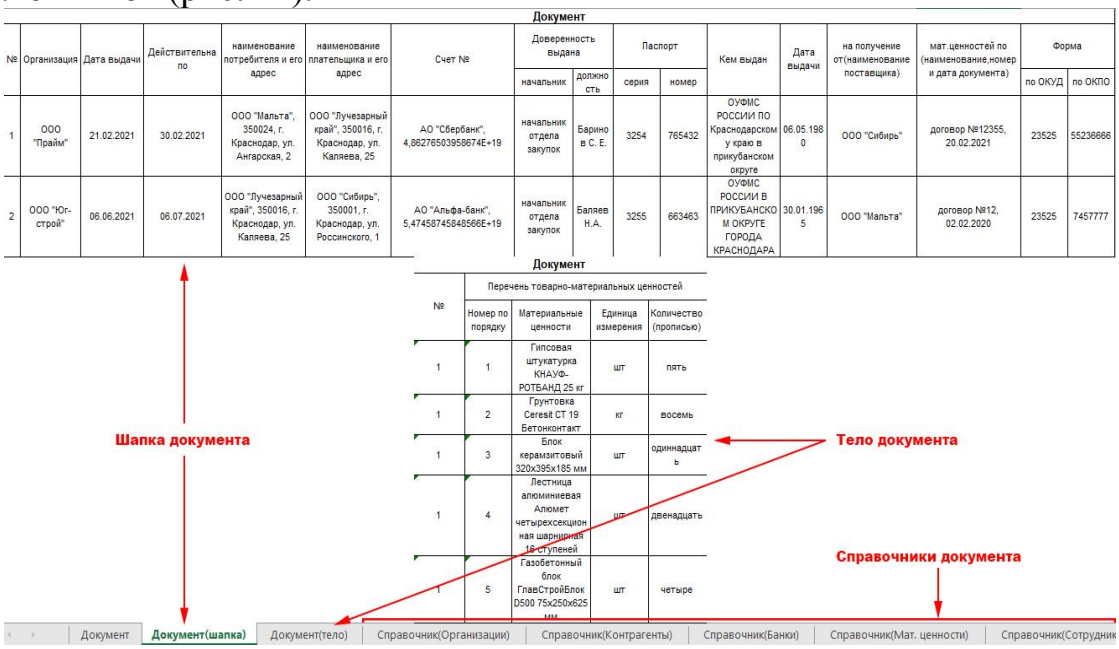

Рисунок 22. – Разделение документа

Далее в режиме просмотра кода VBA создадим форму UserForm2 и добавим на нее следующие объекты: Label1, TextBox1, Frame1, Button1 (рис. 23).

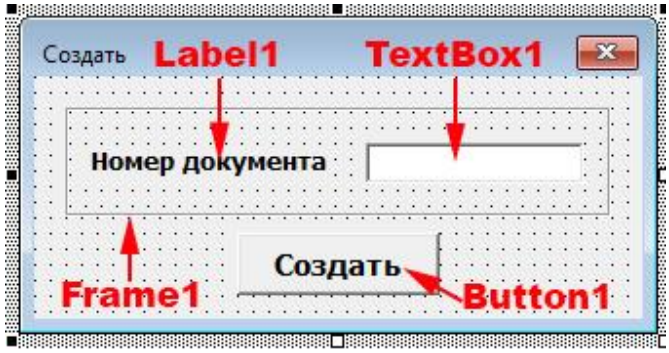

Рисунок 23. – UserForm2

По двойному клику на Button1 перейдем в режим редактирования кода процедуры выполнения кнопки и напишем для нее следующий код (рис. 24):

```
Private Sub CommandButton1_Click() - процедура с событием по клику кнопки «Создать»
Dim l, lm As Integer
Dim u As Boolean                    - объявление переменных
l = 4
u = False                           - инициализация переменных
Do While Worksheets("Документ(шапка)").Cells(l, 1) <> ""
    If CInt(Worksheets("Документ(шапка)").Cells(l, 1)) = CInt(UserForm2.TextBox1.Value) Then
        u = True
        lm = MsgBox("Такой документ уже существует", 64, "")
    End If                                                          - цикл с вложенным условием
    l = l + 1
Loop
If u = False Then
Worksheets("Документ").Cells(8, 10).Value = UserForm2.TextBox1
UserForm3.TextBox2.Value = UserForm2.TextBox1
UserForm2.Hide - закрытие формы «Создать»
Module3.Редактировать - обращение к макросу с открытием формы «Редактировать»
End If
End Sub - конец процедуры
```

Рисунок 24. – Описание функции UserForm2.Button1

UserForm3 – это следующая форма, которая открывается после нажатия кнопки создать в UserForm1 или при нажатии на кнопку «Редактировать» в документе.

Рассмотрим UserForm3 (рис. 25).

Рисунок 25. – UserForm2

Имеет аналогичные объекты, за исключением Image, ComboBox, ListBox.

Рассмотрим функцию кнопки «Сохранить» и «Добавить в таблицу» (рис.26).

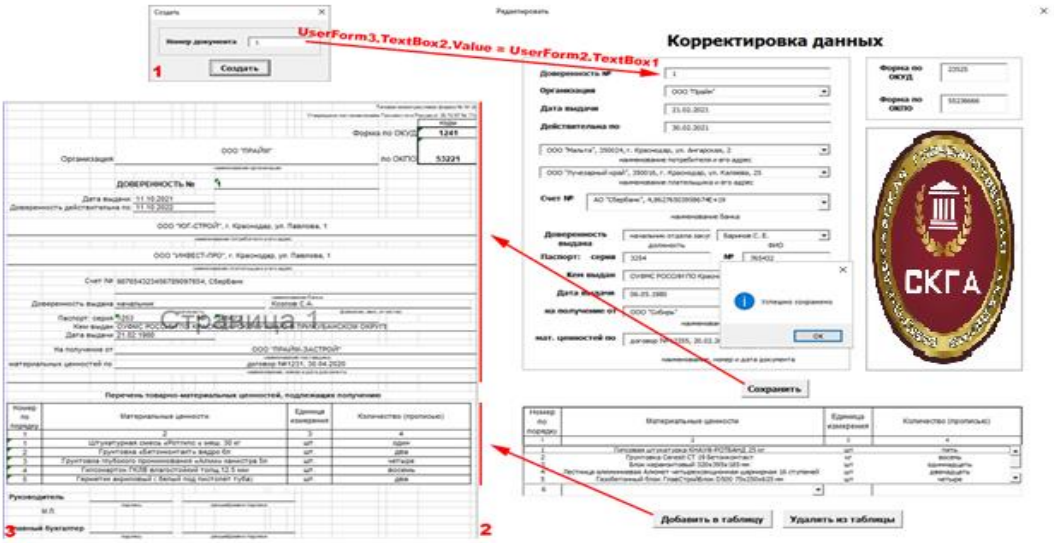

Рисунок 26. – Результат использования UserForm3

Ниже представлена обратная сторона кнопок «Сохранить» и «Добавить в таблицу» – выполняемый код (рис. 27 и рис. 28 соответственно):

```
Private Sub CommandButton2_Click() 'кнопка сохранить'
Worksheets("Документ").Cells(5, 5).Value = UserForm3.ComboBox1
Worksheets("Документ").Cells(8, 10).Value = UserForm3.TextBox2
Worksheets("Документ").Cells(4, 19).Value = UserForm3.TextBox3
Worksheets("Документ").Cells(5, 19).Value = UserForm3.TextBox4
Worksheets("Документ").Cells(10, 6).Value = UserForm3.TextBox5
Worksheets("Документ").Cells(11, 6).Value = UserForm3.TextBox6
Worksheets("Документ").Cells(13, 2).Value = UserForm3.ComboBox2
Worksheets("Документ").Cells(16, 2).Value = UserForm3.ComboBox3
Worksheets("Документ").Cells(19, 5).Value = UserForm3.ComboBox4
Worksheets("Документ").Cells(22, 5).Value = UserForm3.TextBox10
Worksheets("Документ").Cells(22, 12).Value = UserForm3.ComboBox7
Worksheets("Документ").Cells(24, 5).Value = UserForm3.TextBox12
Worksheets("Документ").Cells(24, 10).Value = UserForm3.TextBox13
Worksheets("Документ").Cells(25, 5).Value = UserForm3.TextBox14
Worksheets("Документ").Cells(26, 5).Value = UserForm3.TextBox15
Worksheets("Документ").Cells(28, 5).Value = UserForm3.ComboBox5
Worksheets("Документ").Cells(30, 5).Value = UserForm3.TextBox17

Dim a As Integer
a = MsgBox("Успешно сохранено", 64, "")
End Sub
```

Рисунок 27. – Форма "Редактировать". Кнопка "Сохранить"

```
Private Sub CommandButton3_Click() 'кнопка добавления строки'
Dim i, count, k As Integer
i = 38
k = 1
Dim a As Integer
Do While Worksheets("Документ").Cells(i, 2) <> "" And i < 58 And k = 1
    If (CStr(Worksheets("Документ").Cells(i, 2)) = CStr(UserForm3.TextBox18.Value)) Then
        Worksheets("Документ").Cells(i, 3) = UserForm3.ComboBox6.Value
        Worksheets("Документ").Cells(i, 13) = UserForm3.TextBox20.Value
        Worksheets("Документ").Cells(i, 16) = UserForm3.TextBox21.Value
        k = 0
    End If
    i = i + 1
Loop
If (k = 1) Then
i = 37
count = 1
Do While Worksheets("Документ").Cells(i, 2) <> "" And i < 58
    i = i + 1
    count = count + 1
    If (Worksheets("Документ").Cells(i, 3) = UserForm3.ComboBox6.Value) Then k = 3
Loop
'добавление строки+j+объединение+границы'
If (i < 58 And k = 1) Then
Worksheets("Документ").Rows(i).Insert Shift:=xlShiftTo добавление строки
Worksheets("Документ").Range(Cells(i, 3), Cells(i, 12)).MergeCells = True
Worksheets("Документ").Range(Cells(i, 13), Cells(i, 15)).MergeCells = True объединение ячеек
Worksheets("Документ").Range(Cells(i, 16), Cells(i, 19)).MergeCells = True
Worksheets("Документ").Cells(i, 2).Borders.LineStyle = xlContinuous
Worksheets("Документ").Range(Cells(i, 3), Cells(i, 12)).Borders.LineStyle = xlContinuous
Worksheets("Документ").Range(Cells(i, 13), Cells(i, 15)).Borders.LineStyle = xlContinuous добавление границ ячейке
Worksheets("Документ").Range(Cells(i, 16), Cells(i, 19)).Borders.LineStyle = xlContinuous
'заполнение'
UserForm3.TextBox18 = count
Worksheets("Документ").Cells(i, 2).Value = UserForm3.TextBox18
UserForm3.TextBox18 = count + 1
Worksheets("Документ").Range(Cells(i, 3), Cells(i, 12)).Value = UserForm3.ComboBox6
Worksheets("Документ").Range(Cells(i, 13), Cells(i, 15)).Value = UserForm3.TextBox20
Worksheets("Документ").Range(Cells(i, 16), Cells(i, 19)).Value = UserForm3.TextBox21
End If
End If

If (k = 3) Then a = MsgBox("Данная позиция уже существует", 64, "")

UserForm3.Hide
Module3.Редактировать
End Sub
```

Цикл замены содержимого строки при выборе ее из ListBox

Проверка на уже существующую строку

Рисунок 28. – Форма "Редактировать". Кнопка "Добавить в таблицу"

Наконец, назначаем кнопке «Создать» макрос вызова формы UserForm2. Для этого выполняем следующие действия (рис. 29):

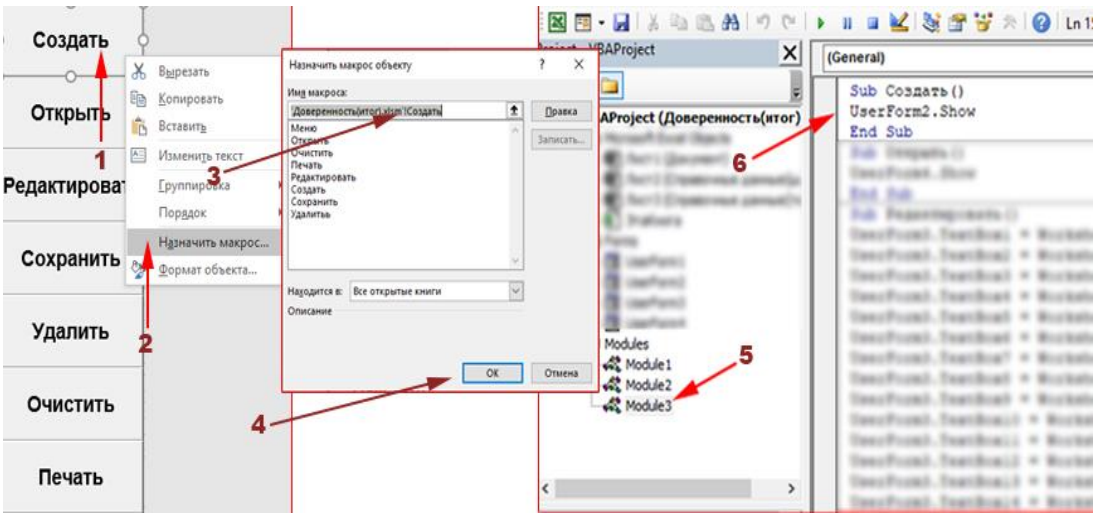

Рисунок 29. – Назначение макроса кнопке "Создать"

Подобным образом, используя выше писанную технологию, мы создаем остальные кнопки управления документом. Рассмотрим их функционал:

– Кнопка «Сохранить» (рис. 30). – Сохраняет в лист «Документ(шапка)» и лист «Документ(тело)» информацию с документа.

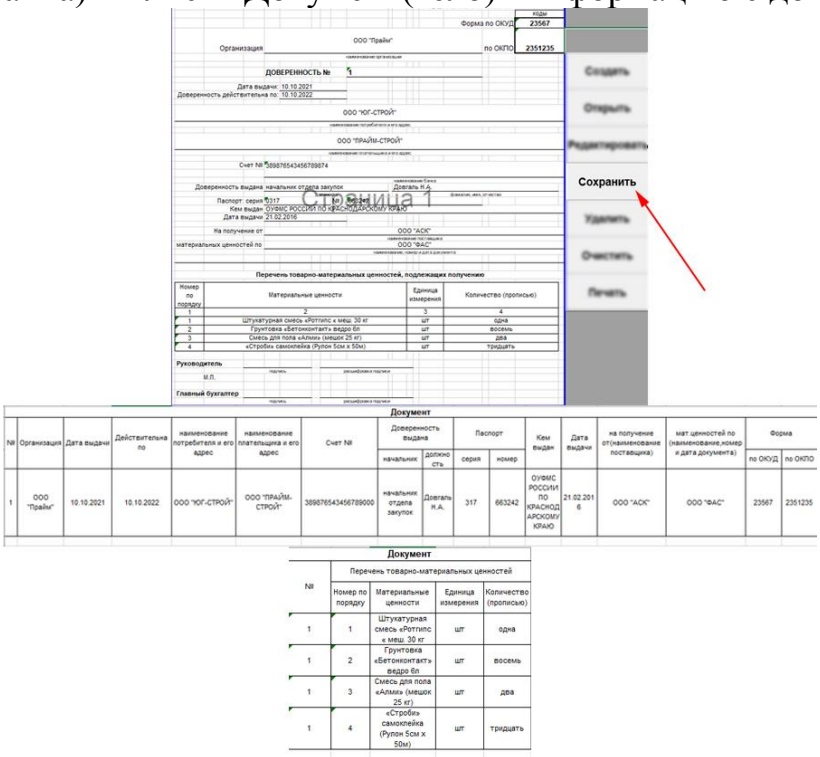

Рисунок 30. – Кнопка "Сохранить"

– Кнопка «Очистить» (рис. 31). – Очищает информацию с листа «Документ».

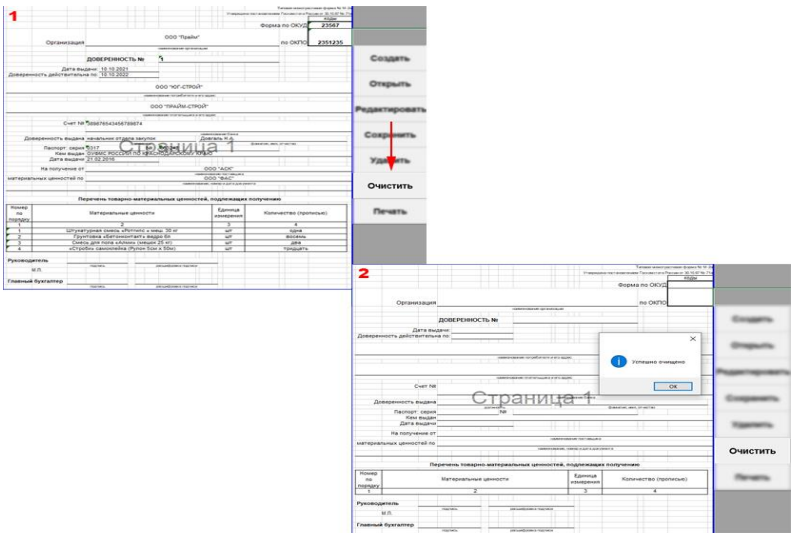

Рисунок 31. – Кнопка "Очистить"

– Кнопка «Открыть» (рис. 32). – Извлекает данные из справочников в документ по его номеру.

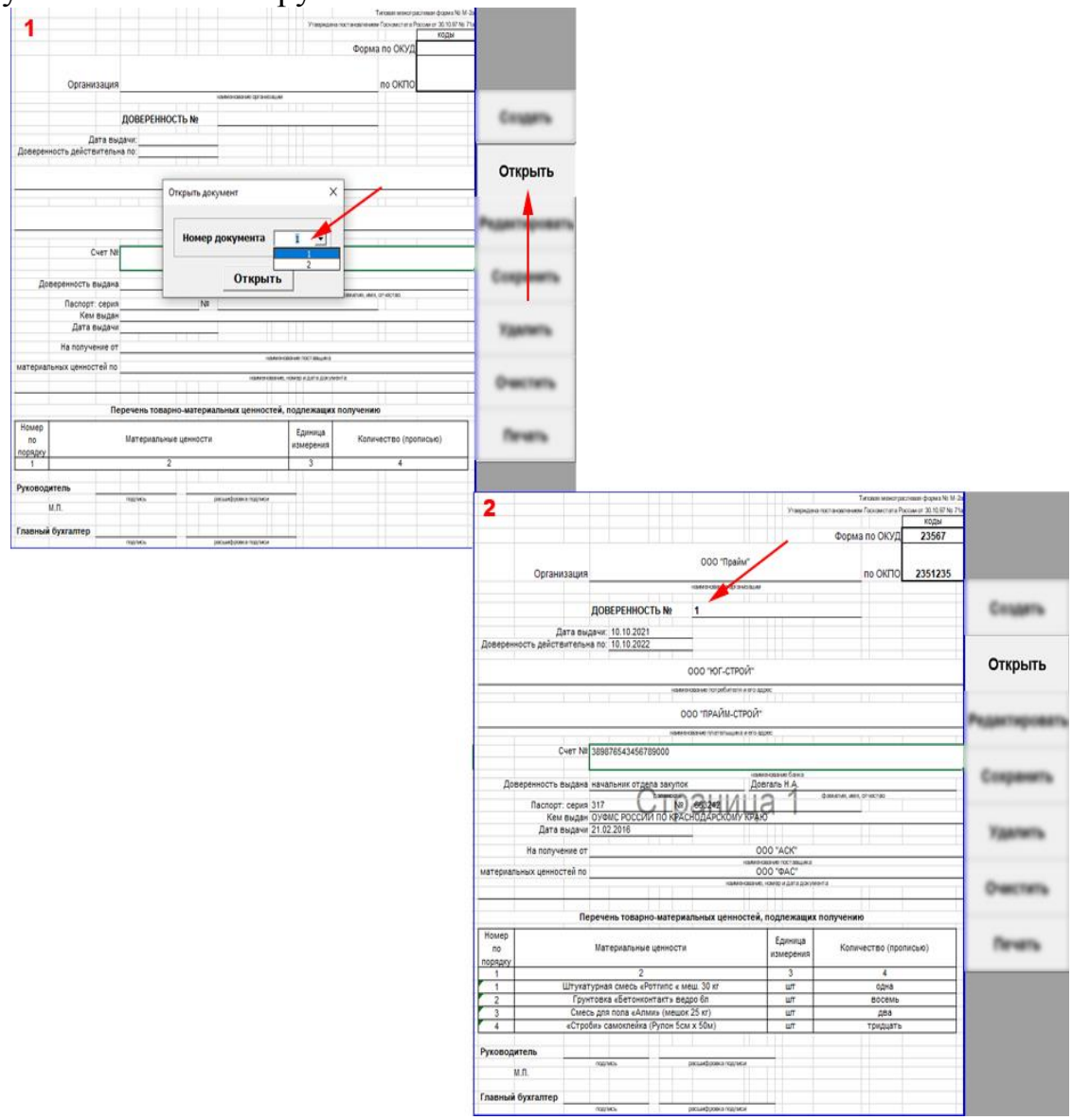

Рисунок 32. – Кнопка "Открыть"

– Кнопка «Редактировать» (рис. 33). – Открывает форму для редакции данных листа «Документ».

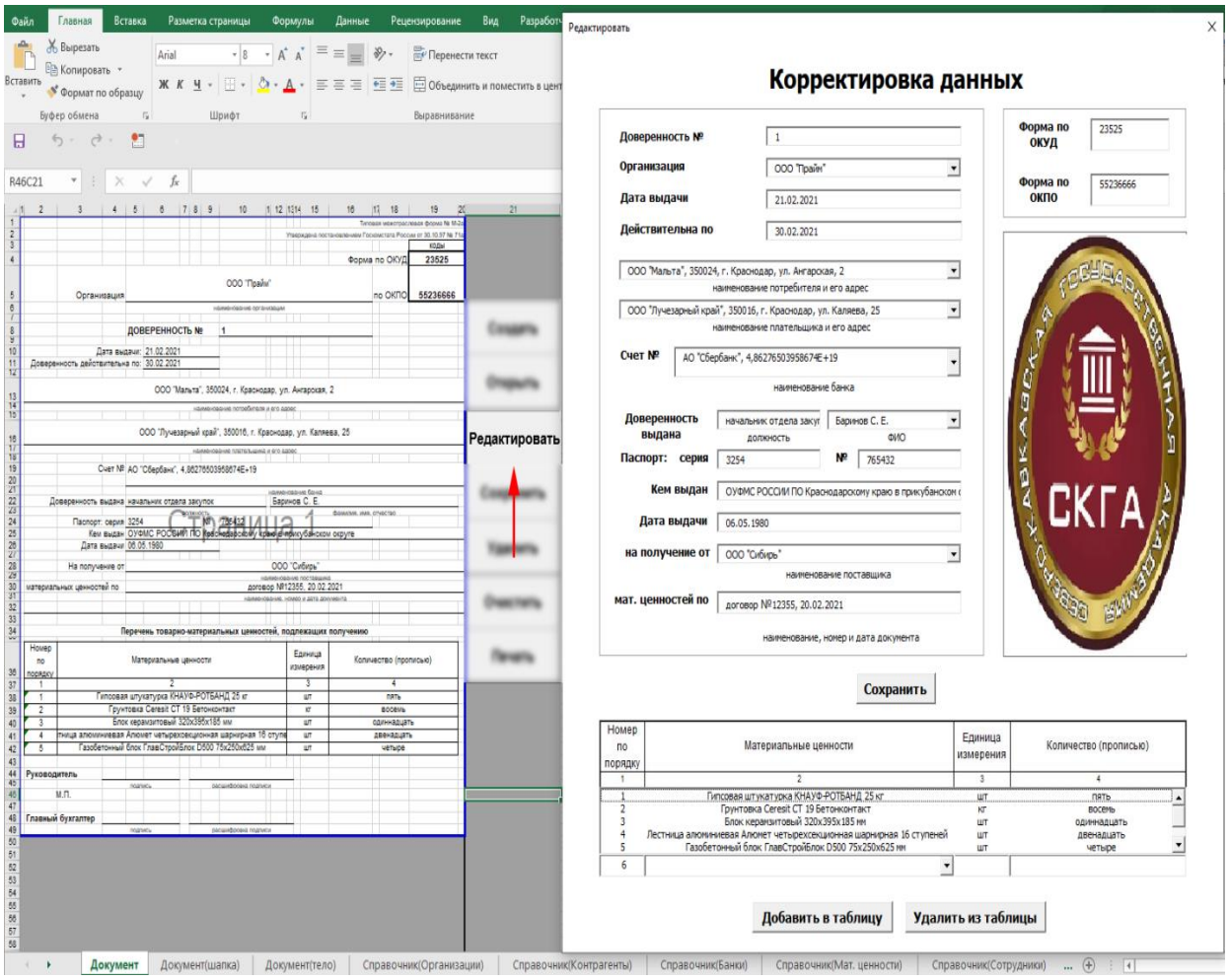

Рисунок 33. – Кнопка "Редактировать"

– Кнопка «Удалить» (рис. 34). – Удаляет данные по номеру документа из справочника.

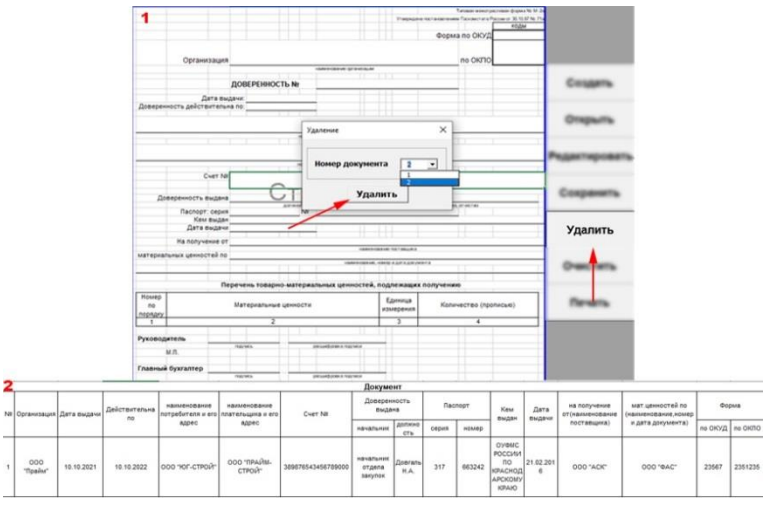

Рисунок 34. – Кнопка "Удалить"

– Кнопка «Печать» (рис. 35). – Выводит окно печати документа.

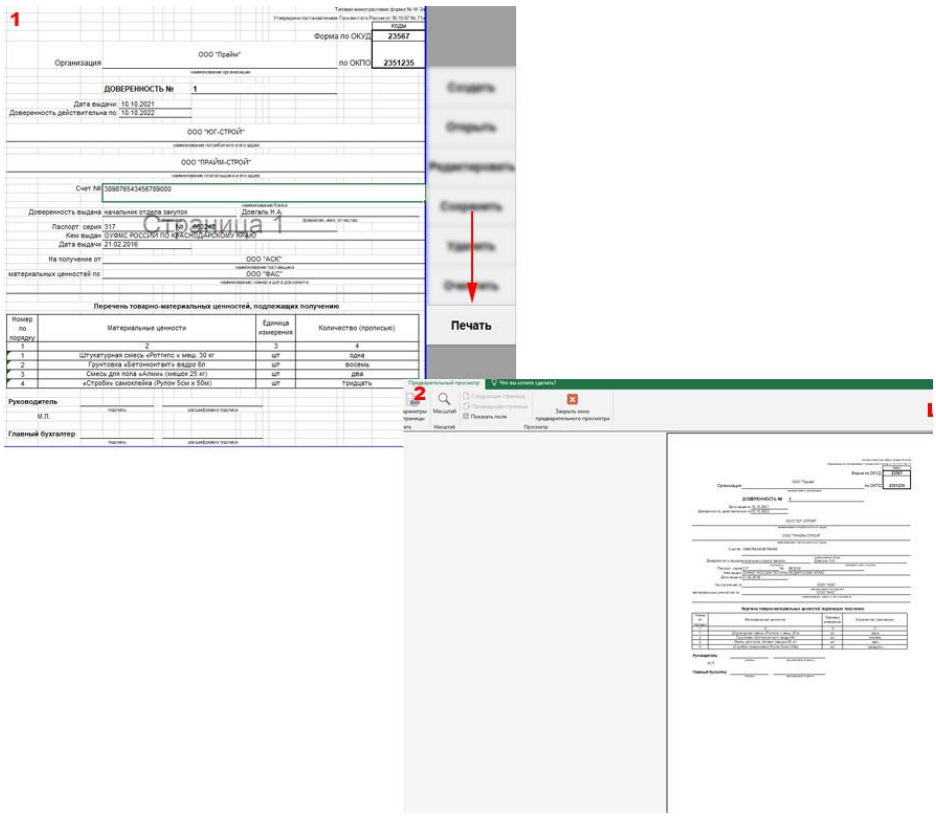

Рисунок 35. – Кнопка "Печать"

Для подробного ознакомления с кодом функций, выполняемых кнопках, вы можете ознакомиться в приложении 1.

ПРИЛОЖЕНИЕ 1. ЛИСТИНГ КОДА КНОПОК

1) **Кнопка «Создать»:**

– **Код, исполняемый в макросе:**

```
Sub Создать()
UserForm2.Show
End Sub
```

– **Код, исполняемый в форме:**

```
Private Sub CommandButton1_Click()
Dim l, lm As Integer
Dim u As Boolean
l = 4
u = False
Do While Worksheets("Документ(шапка)").Cells(l, 1) <> ""
   If CInt(Worksheets("Документ(шапка)").Cells(l, 1)) =
CInt(UserForm2.TextBox1.Value) Then
      u = True
      lm = MsgBox("Такой документ уже существует", 64, "")
   End If
   l = l + 1
Loop
If u = False Then
Worksheets("Документ").Cells(8, 10).Value = UserForm2.TextBox1
UserForm3.TextBox2.Value = UserForm2.TextBox1
UserForm2.Hide
Module3.Редактировать
End If
End Sub
Private Sub UserForm_Activate()
Dim i As Integer
i = 4
Do While Worksheets("Документ(шапка)").Cells(i, 1) <> ""
   i = i + 1
Loop
UserForm2.TextBox1.Value = Worksheets("Документ(шапка)").Cells(i  1,
1) + 1
End Sub
```

2) **Кнопка «Открыть»:**

– **Код, исполняемый в макросе:**

```
Sub Открыть()
UserForm4.Show
End Sub
```

– **Код, исполняемый в форме:**

```
Private Sub UserForm_Activate()
Dim i As Integer
```

```
        i = 4
        Do While Worksheets("Документ(шапка)").Cells(i, 1) <> ""
           ComboBox1.AddItem Worksheets("Документ(шапка)").Cells(i, 1)
        i = i + 1
        Loop
         ComboBox1.TabIndex = 0
        End Sub
        Private Sub CommandButton1_Click()
        Dim s As Integer
        s = 4
        Do While Worksheets("Документ(шапка)").Cells(s, 1) <> ""
           If CStr(Worksheets("Документ(шапка)").Cells(s, 1)) =
UserForm4.ComboBox1.Value Then
                Worksheets("Документ").Cells(8, 10).Value =
Worksheets("Документ(шапка)").Cells(s, 1)
                Worksheets("Документ").Cells(5, 5).Value =
Worksheets("Документ(шапка)").Cells(s, 2)
                Worksheets("Документ").Cells(10, 6).Value =
Worksheets("Документ(шапка)").Cells(s, 3)
                Worksheets("Документ").Cells(11, 6).Value =
Worksheets("Документ(шапка)").Cells(s, 4)
                Worksheets("Документ").Cells(13, 2).Value =
Worksheets("Документ(шапка)").Cells(s, 5)
                Worksheets("Документ").Cells(16, 2).Value =
Worksheets("Документ(шапка)").Cells(s, 6)
                Worksheets("Документ").Cells(19, 5).Value =
Worksheets("Документ(шапка)").Cells(s, 7)
                Worksheets("Документ").Cells(22, 5).Value =
Worksheets("Документ(шапка)").Cells(s, 8)
                Worksheets("Документ").Cells(22, 12).Value =
Worksheets("Документ(шапка)").Cells(s, 9)
                Worksheets("Документ").Cells(24, 5).Value =
Worksheets("Документ(шапка)").Cells(s, 10)
                Worksheets("Документ").Cells(24, 10).Value =
Worksheets("Документ(шапка)").Cells(s, 11)
                Worksheets("Документ").Cells(25, 5).Value =
Worksheets("Документ(шапка)").Cells(s, 12)
                Worksheets("Документ").Cells(26, 5).Value =
Worksheets("Документ(шапка)").Cells(s, 13)
                Worksheets("Документ").Cells(28, 5).Value =
Worksheets("Документ(шапка)").Cells(s, 14)
                Worksheets("Документ").Cells(30, 5).Value =
Worksheets("Документ(шапка)").Cells(s, 15)
                Worksheets("Документ").Cells(4, 19).Value =
Worksheets("Документ(шапка)").Cells(s, 16)
```

```
        Worksheets("Документ").Cells(5, 19).Value =
Worksheets("Документ(шапка)").Cells(s, 17)
        End If
        s = s + 1
        Loop

        'Таблица'
        Dim f, d As Integer
        f = 38
        d = f
        Do While Worksheets("Документ").Cells(f, 2) <> ""
           f = f + 1
        Loop

        For j = 1 To f  d
        Worksheets("Документ").Rows(38).Delete
        Next


        Dim i, c, count As Integer
        count = 0
        c = 4
        i = 38
        Do While Worksheets("Документ(тело)").Cells(c, 1) <> "" And i < 58
           If (CStr(Worksheets("Документ(тело)").Cells(c, 1)) =
CStr(UserForm4.ComboBox1.Value)) Then
              Worksheets("Документ").Rows(i).Insert Shift:=xlShiftTo
              Worksheets("Документ").Range(Cells(i, 3), Cells(i, 12)).MergeCells =
True
              Worksheets("Документ").Range(Cells(i, 13), Cells(i, 15)).MergeCells
= True
              Worksheets("Документ").Range(Cells(i, 16), Cells(i, 19)).MergeCells
= True
              Worksheets("Документ").Cells(i, 2).Borders.LineStyle = xlContinuous
              Worksheets("Документ").Range(Cells(i, 3), Cells(i,
12)).Borders.LineStyle = xlContinuous
              Worksheets("Документ").Range(Cells(i, 13), Cells(i,
15)).Borders.LineStyle = xlContinuous
              Worksheets("Документ").Range(Cells(i, 16), Cells(i,
19)).Borders.LineStyle = xlContinuous
              'заполнение'
              Worksheets("Документ").Cells(i, 2).Value =
Worksheets("Документ(тело)").Cells(c, 2)
              Worksheets("Документ").Range(Cells(i, 3), Cells(i, 12)).Value =
Worksheets("Документ(тело)").Cells(c, 3)
```

```
            Worksheets("Документ").Range(Cells(i, 13), Cells(i, 15)).Value =
Worksheets("Документ(тело)").Cells(c, 4)
            Worksheets("Документ").Range(Cells(i, 16), Cells(i, 19)).Value =
Worksheets("Документ(тело)").Cells(c, 5)
              i = i + 1
           End If
           c = c + 1
        Loop
        UserForm4.Hide
        End Sub
```

3) Кнопка «Редактировать»:

– **Код, исполняемый в макросе:**

```
        Sub Редактировать()
        UserForm3.ComboBox1 = Worksheets("Документ").Cells(5, 5).Value
        UserForm3.TextBox2 = Worksheets("Документ").Cells(8, 10).Value
        UserForm3.TextBox3 = Worksheets("Документ").Cells(4, 19).Value
        UserForm3.TextBox4 = Worksheets("Документ").Cells(5, 19).Value
        UserForm3.TextBox5 = Worksheets("Документ").Cells(10, 6).Value
        UserForm3.TextBox6 = Worksheets("Документ").Cells(11, 6).Value
        UserForm3.ComboBox2 = Worksheets("Документ").Cells(13, 2).Value
        UserForm3.ComboBox3 = Worksheets("Документ").Cells(16, 2).Value
        UserForm3.ComboBox4 = Worksheets("Документ").Cells(19, 5).Value
        UserForm3.TextBox10 = Worksheets("Документ").Cells(22, 5).Value
        UserForm3.ComboBox7 = Worksheets("Документ").Cells(22, 12).Value
        UserForm3.TextBox12 = Worksheets("Документ").Cells(24, 5).Value
        UserForm3.TextBox13 = Worksheets("Документ").Cells(24, 10).Value
        UserForm3.TextBox14 = Worksheets("Документ").Cells(25, 5).Value
        UserForm3.TextBox15 = Worksheets("Документ").Cells(26, 5).Value
        UserForm3.ComboBox5 = Worksheets("Документ").Cells(28, 5).Value
        UserForm3.TextBox17 = Worksheets("Документ").Cells(30, 5).Value

        Dim i As Integer "combobox1,2,3
        Dim s As String
        i = 2
        Do While Worksheets("Справочник(Организации)").Cells(i, 2) <> ""
           UserForm3.ComboBox1.AddItem
Worksheets("Справочник(Организации)").Cells(i, 2).Value "cb1
           s = Worksheets("Справочник(Организации)").Cells(i, 4).Value + ", " +
CStr(Worksheets("Справочник(Организации)").Cells(i, 6).Value) "c4
           UserForm3.ComboBox4.AddItem s
           s = Worksheets("Справочник(Организации)").Cells(i, 2).Value + ", " +
CStr(Worksheets("Справочник(Организации)").Cells(i, 7).Value) "c4
           UserForm3.ComboBox2.AddItem s "cb2
           UserForm3.ComboBox3.AddItem s "cb3
```

```
        i = i + 1
    Loop
    i = 2
    Do While Worksheets("Справочник(Контрагенты)").Cells(i, 2) <> ""
        s = Worksheets("Справочник(Контрагенты)").Cells(i, 2).Value + ", " +
Worksheets("Справочник(Контрагенты)").Cells(i, 7).Value "c2
        s = Worksheets("Справочник(Контрагенты)").Cells(i, 2).Value
        UserForm3.ComboBox5.AddItem s "cb5
        i = i + 1
    Loop
    i = 2 'cb6
    Do While Worksheets("Справочник(Мат. ценности)").Cells(i, 2) <> ""
        UserForm3.ComboBox6.AddItem Worksheets("Справочник(Мат.
ценности)").Cells(i, 2)
        i = i + 1
    Loop
    i = 2
    Do While Worksheets("Справочник(Сотрудники)").Cells(i, 2) <> "" 'cb7
        UserForm3.ComboBox7.AddItem
Worksheets("Справочник(Сотрудники)").Cells(i, 2)
        i = i + 1
    Loop
    'listbox
    Dim j As Integer
    Dim a As Range
    Dim k As Integer
    Dim count As Integer
    Dim stable() As String
    i = 37
    j = i
    k = 0
    count = 1
    Do While Worksheets("Документ").Cells(i, 2) <> ""
        i = i + 1
        count = count + 1
    Loop
    ReDim stable(i  j, 4) As String
    i = i  1
    If (i > 37) Then
    For h = 0 To i  j  1
        stable(h, k) = Worksheets("Документ").Cells(h + 38, 2)
        stable(h, k + 1) = Worksheets("Документ").Cells(h + 38, 3)
        stable(h, k + 2) = Worksheets("Документ").Cells(h + 38, 13)
        stable(h, k + 3) = Worksheets("Документ").Cells(h + 38, 16)
    Next
```

```
End If
UserForm3.ListBox1.List = stable
UserForm3.TextBox18 = count  1
UserForm3.Show
End Sub
```

– **Код, исполняемый в форме:**

```
Dim stable() As String
Private Sub CommandButton1_Click()
UserForm3.Hide
UserForm1.Show
End Sub

Private Sub ComboBox1_Change() 'организация ~ счет'
Dim i As Integer
Dim s As String
i = 2
k = 1
If (UserForm3.ComboBox1.Value = "") Then k = 0
Do While Worksheets("Справочник(Организации)").Cells(i, 2).Value <>
"" And k = 1
    If CStr(Worksheets("Справочник(Организации)").Cells(i, 2).Value) =
CStr(UserForm3.ComboBox1.Value) Then
        s = Worksheets("Справочник(Организации)").Cells(i, 4).Value + ", "
+ CStr(Worksheets("Справочник(Организации)").Cells(i, 6).Value)
        UserForm3.ComboBox4.Value = s
        s = Worksheets("Справочник(Организации)").Cells(i, 2).Value + ", "
+ CStr(Worksheets("Справочник(Организации)").Cells(i, 7).Value)
        UserForm3.ComboBox2.Value = s
        UserForm3.ComboBox3.Value = s
        k = 0
    End If
    i = i + 1
Loop
End Sub

Private Sub ComboBox6_Change() 'Мат. ценности ~ ед.измерения'
Dim i, k As Integer
i = 2
k = 1
If (UserForm3.ComboBox6.Value = "") Then k = 0
Do While Worksheets("Справочник(Мат. ценности)").Cells(i, 2).Value <>
"" And k = 1
    If CStr(Worksheets("Справочник(Мат. ценности)").Cells(i, 2).Value) =
CStr(UserForm3.ComboBox6.Value) Then
```

```
        UserForm3.TextBox20.Value = Worksheets("Справочник(Мат.
ценности)").Cells(i, 3).Value
        k = 0
        End If
        i = i + 1
        Loop
        End Sub

        Private Sub ComboBox7_Change()
        Dim i As Integer
        i = 2
        k = 1
        If (UserForm3.ComboBox7.Value = "") Then k = 0
        Do While Worksheets("Справочник(Сотрудники)").Cells(i, 2).Value <> ""
And k = 1
        If CStr(Worksheets("Справочник(Сотрудники)").Cells(i, 2).Value) =
CStr(UserForm3.ComboBox7.Value) Then
        UserForm3.TextBox10.Value =
Worksheets("Справочник(Сотрудники)").Cells(i, 3).Value
        UserForm3.TextBox12.Value =
Worksheets("Справочник(Сотрудники)").Cells(i, 4).Value
        UserForm3.TextBox13.Value =
Worksheets("Справочник(Сотрудники)").Cells(i, 5).Value
        UserForm3.TextBox14.Value =
Worksheets("Справочник(Сотрудники)").Cells(i, 6).Value
            UserForm3.TextBox15.Value =
Worksheets("Справочник(Сотрудники)").Cells(i, 7).Value
            k = 0
          End If
          i = i + 1
        Loop
        End Sub
        Private Sub CommandButton2_Click() 'кнопка сохранить'
        Worksheets("Документ").Cells(5, 5).Value = UserForm3.ComboBox1
        Worksheets("Документ").Cells(8, 10).Value = UserForm3.TextBox2
        Worksheets("Документ").Cells(4, 19).Value = UserForm3.TextBox3
        Worksheets("Документ").Cells(5, 19).Value = UserForm3.TextBox4
        Worksheets("Документ").Cells(10, 6).Value = UserForm3.TextBox5
        Worksheets("Документ").Cells(11, 6).Value = UserForm3.TextBox6
        Worksheets("Документ").Cells(13, 2).Value = UserForm3.ComboBox2
        Worksheets("Документ").Cells(16, 2).Value = UserForm3.ComboBox3
        Worksheets("Документ").Cells(19, 5).Value = UserForm3.ComboBox4
        Worksheets("Документ").Cells(22, 5).Value = UserForm3.TextBox10
        Worksheets("Документ").Cells(22, 12).Value = UserForm3.ComboBox7
        Worksheets("Документ").Cells(24, 5).Value = UserForm3.TextBox12
```

```
        Worksheets("Документ").Cells(24, 10).Value = UserForm3.TextBox13
        Worksheets("Документ").Cells(25, 5).Value = UserForm3.TextBox14
        Worksheets("Документ").Cells(26, 5).Value = UserForm3.TextBox15
        Worksheets("Документ").Cells(28, 5).Value = UserForm3.ComboBox5
        Worksheets("Документ").Cells(30, 5).Value = UserForm3.TextBox17
        Dim a As Integer
        a = MsgBox("Успешно сохранено", 64, "")
        End Sub
        Private Sub CommandButton3_Click() 'кнопка добавления строки'
        Dim i, count, k As Integer
        i = 38
        k = 1
        Dim a As Integer
        Do While Worksheets("Документ").Cells(i, 2) <> "" And i < 58 And k = 1
           If (CStr(Worksheets("Документ").Cells(i, 2)) =
CStr(UserForm3.TextBox18.Value)) Then
              Worksheets("Документ").Cells(i, 3) = UserForm3.ComboBox6.Value
              Worksheets("Документ").Cells(i, 13) = UserForm3.TextBox20.Value
              Worksheets("Документ").Cells(i, 16) = UserForm3.TextBox21.Value
              k = 0
           End If
           i = i + 1
        Loop
        If (k = 1) Then
        i = 37
        count = 1
        Do While Worksheets("Документ").Cells(i, 2) <> "" And i < 58
           i = i + 1
           count = count + 1
           If (Worksheets("Документ").Cells(i, 3) = UserForm3.ComboBox6.Value)
Then k = 3
        Loop
        'добавление строки+j+объединение+границы'
        If (i < 58 And k = 1) Then
        Worksheets("Документ").Rows(i).Insert Shift:=xlShiftTo
        Worksheets("Документ").Range(Cells(i, 3), Cells(i, 12)).MergeCells = True
        Worksheets("Документ").Range(Cells(i, 13), Cells(i, 15)).MergeCells =
True
        Worksheets("Документ").Range(Cells(i, 16), Cells(i, 19)).MergeCells =
True
        Worksheets("Документ").Cells(i, 2).Borders.LineStyle = xlContinuous
        Worksheets("Документ").Range(Cells(i, 3), Cells(i, 12)).Borders.LineStyle
= xlContinuous
        Worksheets("Документ").Range(Cells(i, 13), Cells(i,
15)).Borders.LineStyle = xlContinuous
```

```
        Worksheets("Документ").Range(Cells(i, 16), Cells(i,
19)).Borders.LineStyle = xlContinuous
        'заполнение'
        UserForm3.TextBox18 = count
        Worksheets("Документ").Cells(i, 2).Value = UserForm3.TextBox18
        UserForm3.TextBox18 = count + 1
        Worksheets("Документ").Range(Cells(i, 3), Cells(i, 12)).Value =
UserForm3.ComboBox6
        Worksheets("Документ").Range(Cells(i, 13), Cells(i, 15)).Value =
UserForm3.TextBox20
        Worksheets("Документ").Range(Cells(i, 16), Cells(i, 19)).Value =
UserForm3.TextBox21
        End If
        End If
        If (k = 3) Then a = MsgBox("Данная позиция уже существует", 64, "")
        UserForm3.Hide
        Module3.Редактировать
        End Sub
        Private Sub CommandButton4_Click() 'кнопка удаления строки'
        Dim i As Integer
        i = 38
        Do While Worksheets("Документ").Cells(i, 2) <> "" And i > 37
           i = i + 1
           count = count + 1
        Loop
        If i <> 37 Then
           UserForm3.TextBox18 = count
           UserForm3.ComboBox6 = Worksheets("Документ").Cells(i  1, 3)
           UserForm3.TextBox20 = Worksheets("Документ").Cells(i  1, 13)
           UserForm3.TextBox21 = Worksheets("Документ").Cells(i  1, 16)
           i = i  1
        End If
        If i > 37 Then Worksheets("Документ").Rows(i).Delete
        UserForm3.Hide
        Module3.Редактировать
        End Sub
        Private Sub ListBox1_Click()
        UserForm3.TextBox18 =
UserForm3.ListBox1.List(UserForm3.ListBox1.ListIndex, 0)
        UserForm3.ComboBox6 =
UserForm3.ListBox1.List(UserForm3.ListBox1.ListIndex, 1)
        UserForm3.TextBox20 =
UserForm3.ListBox1.List(UserForm3.ListBox1.ListIndex, 2)
        UserForm3.TextBox21 =
UserForm3.ListBox1.List(UserForm3.ListBox1.ListIndex, 3)
```

```
End Sub
```

4) Кнопка «Сохранить»:

– **Код, исполняемый в макросе:**

```
'Документ(шапка)'
Dim l, lk As Integer
l = 4
lk = 0
Do While Worksheets("Документ(шапка)").Cells(l, 1) <> "" And lk = 0
   If CInt(Worksheets("Документ(шапка)").Cells(l, 1)) =
CInt(Worksheets("Документ").Cells(8, 10)) Then
      lk = 1
      l = l  1
   End If
l = l + 1
Loop
For g = 1 To 17
   Worksheets("Документ(шапка)").Cells(l, g).Borders.LineStyle =
xlContinuous
Next
Worksheets("Документ(шапка)").Cells(l, 1) =
Worksheets("Документ").Cells(8, 10).Value
Worksheets("Документ(шапка)").Cells(l, 2) =
Worksheets("Документ").Cells(5, 5).Value
Worksheets("Документ(шапка)").Cells(l, 3) =
Worksheets("Документ").Cells(10, 6).Value
Worksheets("Документ(шапка)").Cells(l, 4) =
Worksheets("Документ").Cells(11, 6).Value
Worksheets("Документ(шапка)").Cells(l, 5) =
Worksheets("Документ").Cells(13, 2).Value
Worksheets("Документ(шапка)").Cells(l, 6) =
Worksheets("Документ").Cells(16, 2).Value
Worksheets("Документ(шапка)").Cells(l, 7) =
Worksheets("Документ").Cells(19, 5).Value
Worksheets("Документ(шапка)").Cells(l, 8) =
Worksheets("Документ").Cells(22, 5).Value
Worksheets("Документ(шапка)").Cells(l, 9) =
Worksheets("Документ").Cells(22, 12).Value
Worksheets("Документ(шапка)").Cells(l, 10) =
Worksheets("Документ").Cells(24, 5).Value
Worksheets("Документ(шапка)").Cells(l, 11) =
Worksheets("Документ").Cells(24, 10).Value
Worksheets("Документ(шапка)").Cells(l, 12) =
Worksheets("Документ").Cells(25, 5).Value
```

```
        Worksheets("Документ(шапка)").Cells(l, 13) =
Worksheets("Документ").Cells(26, 5).Value
        Worksheets("Документ(шапка)").Cells(l, 14) =
Worksheets("Документ").Cells(28, 5).Value
        Worksheets("Документ(шапка)").Cells(l, 15) =
Worksheets("Документ").Cells(30, 5).Value
        Worksheets("Документ(шапка)").Cells(l, 16) =
Worksheets("Документ").Cells(4, 19).Value
        Worksheets("Документ(шапка)").Cells(l, 17) =
Worksheets("Документ").Cells(5, 19).Value
        'Документ(тело)'
        Dim d, k, f, pr As Integer
        k = 4
        d = 38
        f = d
        pr = 0
        Do While Worksheets("Документ").Cells(f, 2) <> ""
           f = f + 1
        Loop
        Do While Worksheets("Документ(тело)").Cells(k, 2) <> "" And pr = 0
           If CStr(Worksheets("Документ(тело)").Cells(k, 1)) =
CStr(Worksheets("Документ").Cells(8, 10)) Then
              pr = 1
              k = k  1
           End If
        k = k + 1
        Loop
        For n = d To f  1
           Worksheets("Документ(тело)").Cells(k, 1) =
Worksheets("Документ").Cells(8, 10)
           Worksheets("Документ(тело)").Cells(k, 2) =
Worksheets("Документ").Cells(n, 2)
           Worksheets("Документ(тело)").Cells(k, 3) =
Worksheets("Документ").Cells(n, 3)
           Worksheets("Документ(тело)").Cells(k, 4) =
Worksheets("Документ").Cells(n, 13)
           Worksheets("Документ(тело)").Cells(k, 5) =
Worksheets("Документ").Cells(n, 16)
           For h = 1 To 5
              Worksheets("Документ(тело)").Cells(k, h).Borders.LineStyle =
xlContinuous
           Next
           k = k + 1
        Next
        'Do While Worksheets("Документ(тело)").Cells(k, 1) <> ""
```

```
        '  If CStr(Worksheets("Документ(тело)").Cells(k, 1)) =
CStr(Worksheets("Документ").Cells(8, 10)) Then
Worksheets("Документ(тело)").Cells(k, 1).EntireRow.Delete
        '  k = k + 1
       'Loop
       Dim a As Integer
        a = MsgBox("Успешно сохранено", 64, "")
       End Sub
```

5) Кнопка «Удалить»:

- **Код, исполняемый в макросе:**

```
Sub Удалитьь()
UserForm1.Show
End Sub
```

- **Код, исполняемый в форме:**

```
       Private Sub UserForm_Activate()
       Dim i As Integer
       i = 4
       Do While Worksheets("Документ(шапка)").Cells(i, 1) <> ""
          ComboBox1.AddItem Worksheets("Документ(шапка)").Cells(i, 1)
       i = i + 1
       Loop
        ComboBox1.TabIndex = 0
       End Sub
       Private Sub CommandButton1_Click()
       Dim i, a As Integer
       i = 4
       Do While Worksheets("Документ(шапка)").Cells(i, 1) <> ""
          If CStr(Worksheets("Документ(шапка)").Cells(i, 1)) =
UserForm1.ComboBox1.Value Then
             Worksheets("Документ(шапка)").Rows(i).Delete
             i = i  1
          End If
       i = i + 1
       Loop
       i = 4
       Do While Worksheets("Документ(тело)").Cells(i, 1) <> ""
          If CStr(Worksheets("Документ(тело)").Cells(i, 1)) =
UserForm1.ComboBox1.Value Then
             Worksheets("Документ(тело)").Rows(i).Delete
             i = i  1
          End If
       i = i + 1
```

```
Loop
a = MsgBox("Успешно удалено", 64, "")
UserForm1.Hide
End Sub
"Документ(шапка)
```

6) Кнопка «Печать»:

– **Код, исполняемый в макросе:**

```
Sub Печать()
ActiveSheet.PrintPreview
End Sub
```

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Новиков Ф., Яценко А. Microsoft Office 2000 в целом. СПБ: БХВ – Санкт-Петербург, 2021, 728 с.

2. Симонович С.В.. Информатика для юристов и экономистов. С.-Петербург: Питер, 2022

3. Пасько В. MicroSoft Office 2000. – К.: Издательская группа BHV. 2000

4. Алексеев, А. П. Сборник лабораторных работ по дисциплине «Информатика». Часть 1: методические указания к проведению лабораторных занятий по дисциплине «Информатика»/ А. П. Алексеев. — Самара: Поволжский государственный университет телекоммуникаций и информатики, 2021. –305 с.

5. Ключников М. В. Применение MS Word и Excel в финансовых расчетах: Учеб. пособие. – М.: Market DS. 2022

6. Информатика. Учебник. – 3-е перераб. Изд. Под ред. Проф. Н.В. Макаровой. – М.: Финансы и статистика, 2021

БОРЛАКОВА Мариям Адильовна
КОНДРАТЬЕВ Валерий Юрьевич
КУМРАТОВА Альфира Менлигуловна
ЭЛЬКАНОВА Лиза Муратовна,

КОМПЬЮТЕРНЫЙ ПРАКТИКУМ

лабораторный практикум для обучающихся по направлению подготовки 09.03.03 Прикладная информатика

Корректор Чагова О.Х.
Редактор Чагова О.Х.

Сдано в набор 07.07.2025 г.
Формат 60х84/16
Бумага офсетная.
Печать офсетная.
Усл. печ. л. 2,79
Заказ № 5096
Тираж 100 экз.

Оригинал-макет подготовлен
в Библиотечно-издательском центре СКГА
369000, г. Черкесск, ул. Ставропольская, 36