

ЛАБОРАТОРНАЯ РАБОТА.

Разработка VBA-приложений в Microsoft Excel (часть 1).

1. Скопируйте в свою рабочую папку файл Excel Lab2Excel_Заготовка.xlsm (*Книга Excel с поддержкой макросов*) и сохраните его под именем Lab2Excel_<Фамилия>.xlsm.

ЧАСТЬ 1. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОЙ ФОРМЫ

- Цель этапа – создание главной формы приложения (см. рис. 2-3).
2. Откройте сохраненный документ и перейдите в редактор VBA.
 3. Вставьте в проект новую форму (команда меню **Insert->User Form**), рис.1.

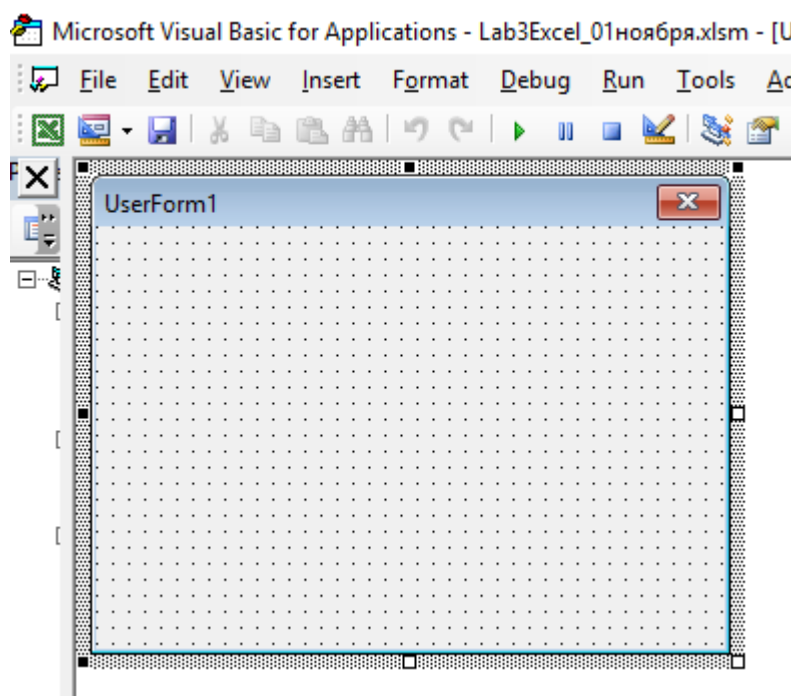






Рис.1. Новая форма в проекте VBA

4. В окне **Toolbox** (*Элементы управления*) выберите следующие элементы управления и поместите их на форму:
 - группирующую рамку (элемент управления Frame, );
 - 3 надписи-метки (элемент управления Label, );
 - 3 кнопки (элемент управления CommandButton, );
 - 3 поля редактирования (элемент управления TextBox, );

Внешний вид создаваемой формы см. на рис. 2-3.

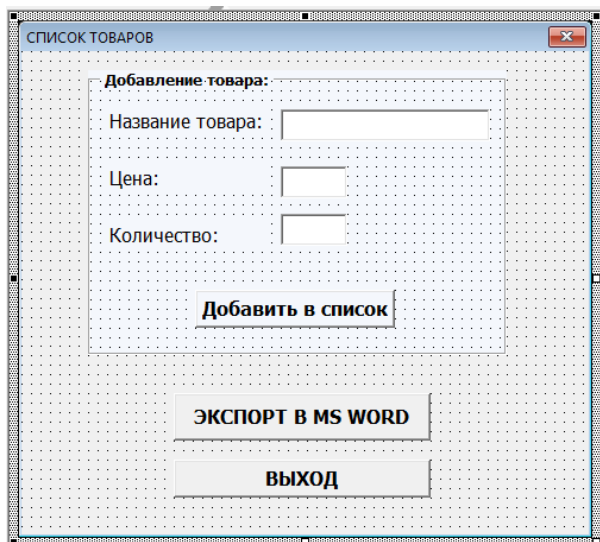


рис.2. Вид формы на этапе проектирования

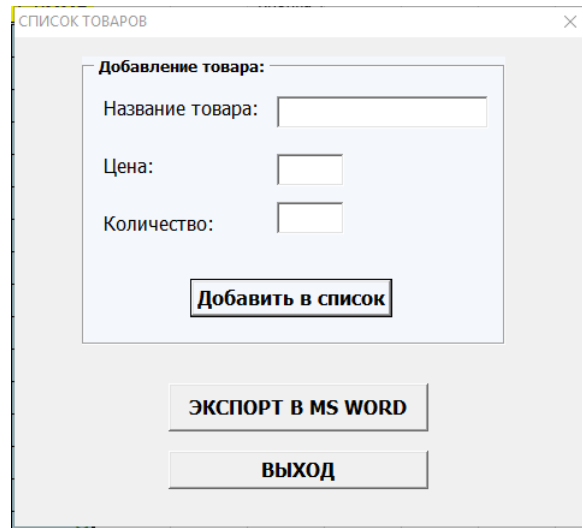


рис.3. Вид формы на этапе выполнения

5. В окне **Properties**, поочередно выделяя форму и размещенные на ней элементы управления, настройте их размер и относительное положение (свойства Left, Top, Width, Height), цвет заливки (свойство BackColor), шрифт для отображения надписей (свойство Font), а также измените значения следующих свойств компонентов:

Элемент управления VBA (имя по умолчанию)	Свойство	Новое значение
UserForm1	Name	<i>frmGoods</i>
	Caption	<i>Список товаров</i>
Frame1	Caption	<i>Добавление товара:</i>
CommandButton1	Name	<i>cmdAdd</i>
	Caption	<i>Добавить в список</i>
CommandButton2	Name	<i>cmdExport</i>
	Caption	<i>Экспорт в MS Word</i>
CommandButton3	Name	<i>cmdExit</i>
	Caption	<i>Выход</i>
Label1	Caption	<i>Название товара:</i>
Label2	Caption	<i>Цена:</i>
Label3	Caption	<i>Количество:</i>
TextBox1	Name	<i>txtName</i>
TextBox2	Name	<i>txtPrice</i>
TextBox3	Name	<i>txtAmount</i>

6. Проверьте работоспособность программы и вернитесь в редактор, закрыв выведенную форму. Внешний вид полученной формы см. на рис. 3.

ЧАСТЬ 2. ДОБАВЛЕНИЕ ТОВАРА В СПИСОК (ТАБЛИЦУ EXCEL)

- **Цель этапа** – добавление строки со сведениями о выбранном товаре в существующую таблицу Excel (рис. 4).

	A	B	C	D
1	Список товаров			
2				
3	Наименование товара	Цена	Количество	Сумма
4				

рис. 4. Таблица со списком товаров (исходный вид)

7. Дважды щелкните мышью по кнопке *Добавить* (cmdAdd). Откроется редактор кода, в котором автоматически будет создана заготовка процедуры обработчика **события** нажатия кнопки (**Click**), см. рис. 5.

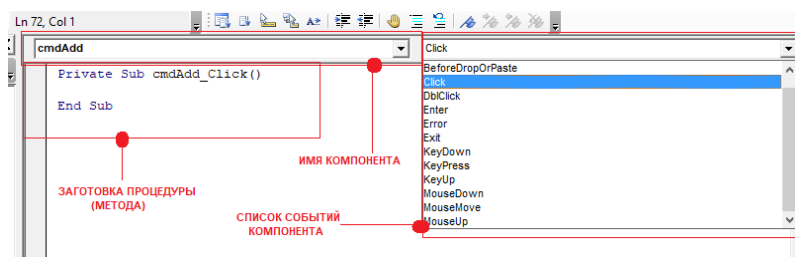


рис. 5. Заготовка процедуры-обработчика события нажатия кнопки

8. Введите в заготовку процедуры следующий код (полный листинг исходного кода см. в [Приложении 1](#)):

```
Private Sub cmdAdd_Click()
    Dim Name As String, Price As Double, Amount As Integer
    Dim varValue As Variant
    Dim NRow As Integer

    NRow = 2 + Range("A3").CurrentRegion.Rows.Count + 1

    ' Название товара
    Name = txtName.Text
    ' Цена товара
    varValue = txtPrice.Text
    If IsNumeric(varValue) Then
        Price = CDBl(varValue)
    Else
        MsgBox ("Некорректное значение!")
        txtPrice.SetFocus
        Exit Sub
    End If
    ' Количество товара
    varValue = txtAmount.Text
    If IsNumeric(varValue) Then
        Amount = CInt(varValue)
    Else
        MsgBox ("Некорректное значение!")
        txtAmount.SetFocus
        Exit Sub
    End If

    Rows(NRow).Insert
    Cells(NRow, 1) = Name
    Cells(NRow, 2) = Price
    Cells(NRow, 3) = Amount

End Sub
```

□ ПРИМЕЧАНИЯ К П. 8.

- 1) Переменные **Name**, **Price**, **Amount** используются для хранения значений *Наименование*, *Цена* и *Количество* товара соответственно
- 2) Переменная **varValue** используется для предварительного хранения значений *Цена* и *Количество* товара до проверки корректности их ввода.
- 3) Переменная **NRow** указывает позицию размещения очередной строки списка на рабочем листе.
- 4) При чтении сведений о цене и количестве товара выполняется дополнительный контроль корректности введенного значения: если введено число (функция **IsNumeric()** возвращает **True**), происходит соответствующее преобразование типа (функции **Cdbl()** и **Cint()**), в противном случае выводится сообщение и происходит выход из процедуры (**Exit Sub**). Фокус ввода (курсор) остается на компоненте с некорректным значением (метод **SetFocus**).

Сведения о других функциях проверки и преобразования типа выражения см. в [Приложениях 2 и 3](#).

9. Проверьте работоспособность программы (см. рис. 6) и вернитесь в редактор VBA.

	A	B	C	D
1	Список товаров			
2				
3	Наименование товара	Цена	Количество	Сумма
4	Молоко	50,5	2	

рис. 6. Таблица с введенными сведениями о товаре

10. Подсчет суммы по заданной позиции списка (для заданной строки – значение выражения *Цена*Количество*).

Введите в конце процедуры следующий оператор и проверьте работоспособность программы (рис. 7).

```
Cells(NRow, 4).Formula = "=" & Cells(NRow, 2).Address & "*" & Cells(NRow, 3).Address
```

	A	B	C	D
1	Список товаров			
2				
3	Наименование товара	Цена	Количество	Сумма
4	Молоко	50,5	2	101

рис. 7. Таблица с данными в столбце «Сумма»

□ ПРИМЕЧАНИЯ К П. 10.

- 1) В данном операторе формируется строковое значение формулы, которое помещается в соответствующую ячейку рабочего листа (столбец *Сумма*).
- 2) Свойство **Address** возвращает адрес заданного диапазона. Синтаксис:

```
Address (rowAbsolute, columnAbsolute, referenceStyle, external, relativeTo)
```

Аргументы:

RowAbsolute	По умолчанию True : абсолютная ссылка на строку, иначе - относительная
ColumnAbsolute	По умолчанию True : абсолютная ссылка на столбец, иначе - относительная
referenceStyle	Формат ссылки. Возможные значения: x1A1 (по умолчанию) - возвращается ссылка в формате A1 , x1R1C1 - в формате R1C1
external	False (по умолчанию) - возвращается относительная ссылка, True - абсолютная
relativeTo	Если rowAbsolute и ColumnAbsolute равны False , а referenceStyle - x1R1C1 , то аргумент определяет начальную ячейку диапазона, относительно которой производится адресация.

- Перейдите на рабочий лист Excel и проверьте значение формулы, внесенное в ячейку столбца *Сумма*.

ЧАСТЬ 3. ФОРМАТИРОВАНИЕ ТАБЛИЦЫ

- Цель этапа – программное оформление внешнего вида таблицы (см. рис. 8).

11. Дополните процедуру следующим кодом:

```
'Очистка формата таблицы
Dim Rng As Range
Set Rng = Range("A4", "D" & ActiveSheet.UsedRange.Rows.Count)
Rng.ClearFormats

'форматирование области с данными
Set Rng = Range("A4", "D" & NRow)
With Rng
    .Font.Name = "Arial"
    .Font.Size = 16
    .Font.Color = xlGreen
    .Interior.Color = xlLightBlue
    .Borders.LineStyle = True
    .Borders(xlEdgeTop).LineStyle = xlDouble
    .Borders(xlEdgeLeft).Weight = xlThick
    .Borders(xlEdgeRight).Weight = xlThick
    .Borders(xlEdgeBottom).Weight = xlThick
End With
```

□ ПРИМЕЧАНИЯ К П. 11.

- 1) Переменная **Rng** используется для хранения сведений о текущем диапазоне.
- 2) В данном фрагменте кода используются следующие свойства и методы объекта-диапазона:

Свойство/Метод	Описание
ClearFormats	Очистка формата указанного диапазона (метод). См. также методы Clear, ClearComments, ClearContents, ClearNotes
Font	Атрибуты шрифта диапазона (свойство)
• Font.Name	Имя шрифта
• Font.Size	Размер шрифта
• Font.Color	Цвет шрифта
Interior	Атрибуты заливки ячеек (диапазона)
• Interior.Color	Цвет заливки
Borders	Границы ячеек диапазона. Можно отдельно указать верхнюю (xlEdgeTop), нижнюю (xlEdgeBottom), левую (xlEdgeLeft) и правую (xlEdgeRight) границы.
• Borders.LineStyle	Стиль границы диапазона Возможные значения свойства: xlContinuous, xlDash, xlDashDot, xlDashDotDot, xlDot, xlDouble, xlSlantDashDot, xlLineStyleNone
• Borders.Weight	Стиль границы диапазона Возможные значения свойства: xlHairline, xlMedium, xlThick, xlThin

12. Проверьте работоспособность программы (см. рис. 8) и вернитесь в редактор VBA.

	A	B	C	D
1	Список товаров			
2				
3	Наименование товара	Цена	Количество	Сумма
4	Молоко	50,5	2	101
5	Хлеб	30	1	30
6	Картофель	33	2	66
7				

рис. 8. Отформатированная таблица

ЧАСТЬ 4. ПОДСЧЕТ ИТОГОВОЙ СУММЫ

- **Цель этапа:** Подсчет итоговой суммы затрат - использование агрегирующих функций Excel, оформление полученного результата (рис. 9).

	A	B	C	D
1	Список товаров			
2				
3	Наименование товара	Цена	Количество	Сумма
4	Молоко	50,5	2	101
5	Хлеб	30	1	30
6	Картофель	33	2	66
7				
8			ИТОГО:	197

рис. 9. Таблица с подсчитанными суммарными затратами

13. Дополните процедуру следующим кодом:

```
'Подсчет и оформление итоговой суммы
Range("C" & (NRow + 2)).Value = "ИТОГО:"
Range("D" & (NRow + 2)).Value = WorksheetFunction.Sum(Range("D4", "D" & NRow))

Set Rng = Range("C" & (NRow + 2), "D" & (NRow + 2))
Rng.Font.Name = "Arial"
Rng.Font.Bold = True
Rng.Font.Size = 16
Rng.Font.Color = xlRed
Rng.Interior.Color = xlYellow
```

14. Проверьте работоспособность программы и вернитесь в редактор VBA.

□ ПРИМЕЧАНИЕ К П. 13.

- Объект **WorksheetFunction** используется для доступа к функциям рабочего листа Excel (worksheet) из кода VBA.
Полный список соответствий функций рабочего листа Excel и функций VBA см. по ссылке <https://msdn.microsoft.com/ru-ru/vba/excel-vba/articles/list-of-worksheet-functions-available-to-visual-basic>

ЧАСТЬ 5. КОПИРОВАНИЕ ТАБЛИЦЫ EXCEL В MICROSOFT WORD

- **Цель этапа:** Создание файла Microsoft Word, включающего в себя содержимое полученной таблицы Excel (результат нажатия кнопки *Экспорт в MS Word*, рис. 10).

Наименование-товара	Цена	Количество	Сумма
Молоко	50,5	2	101
Хлеб	30	1	30
Картофель	33	2	66
ИТОГО:			197

рис. 10. Результат экспорта в Microsoft Word

- Добавьте в проект ссылки на библиотеки *Microsoft Office ??? Object Library* и *Microsoft Word ??? Object Library* (пункт меню Tools>References, установите флажок у соответствующего элемента списка).
Значение поля **???** определяется используемой версией текстового процессора.
- Дважды щелкните мышью по кнопке *Экспорт в MS Word* (cmdExport). В результате откроется редактор кода, в котором автоматически будет создана заготовка процедуры обработчика события нажатия кнопки.
- Введите в заготовку процедуры следующий код:

```

cmdExport
Click

Private Sub cmdExport_Click()
    Dim oWord As Word.Application
    Dim oDoc As Word.Document

    Set oWord = CreateObject("Word.Application")
    oWord.Visible = True
    Set oDoc = oWord.Documents.Add()

    ActiveSheet.UsedRange.Select
    Selection.Copy
    oDoc.Range.Paste

    MsgBox "Экспорт данных завершен!"
End Sub

```

- Проверьте работоспособность программы и вернитесь в редактор VBA.

ЧАСТЬ 6. ВЫХОД ИЗ ПРИЛОЖЕНИЯ

- Цель этапа: Создание обработчика нажатия кнопки *Выход*, завершающего работу приложения.

19. Дважды щелкните мышью по кнопке *Выход* (cmdExit). В результате откроется редактор кода, в котором автоматически будет создана заготовка процедуры обработчика события нажатия кнопки.

20. Введите в заготовку процедуры следующую строку:

```
Unload frmGoods
```

□ ПРИМЕЧАНИЕ К П. 20.

- Метод **Unload** закрывает форму и выгружает ее из оперативной памяти. См. также методы **Show** и **Hide**

21. Проверьте работоспособность программы и вернитесь в редактор VBA.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1. ИСХОДНЫЙ КОД ПРОЦЕДУРЫ ДОБАВЛЕНИЯ ТОВАРА

```

Private Sub cmdAdd_Click()
    Dim Name As String, Price As Double, Amount As Integer
    Dim varValue As Variant
    Dim NRow As Integer

    NRow = 2 + Range("A3").CurrentRegion.Rows.Count + 1

    ' Название товара
    Name = txtName.Text
    ' Цена товара
    varValue = txtPrice.Text
    If IsNumeric(varValue) Then
        Price = CDBl(varValue)
    Else
        MsgBox ("Некорректное значение!")
        txtPrice.SetFocus
        Exit Sub
    End If
    ' Количество товара
    varValue = txtAmount.Text
    If IsNumeric(varValue) Then
        Amount = CInt(varValue)
    Else
        MsgBox ("Некорректное значение!")
        txtAmount.SetFocus
        Exit Sub
    End If

    Rows(NRow).Insert
    Cells(NRow, 1) = Name
    Cells(NRow, 2) = Price
    Cells(NRow, 3) = Amount

    Cells(NRow, 4).Formula = "=" & Cells(NRow, 2).Address & "*" & Cells(NRow, 3).Address

    'Очистка формата таблицы
    Dim Rng As Range
    Set Rng = Range("A4", "D" & ActiveSheet.UsedRange.Rows.Count)
    Rng.ClearFormats

    'форматирование области с данными
    Set Rng = Range("A4", "D" & NRow)
    With Rng
        .Font.Name = "Arial"
        .Font.Size = 16
        .Font.Color = xlGreen
        .Interior.Color = xlLightBlue
        .Borders.LineStyle = True
        .Borders(xlEdgeTop).LineStyle = xlDouble
        .Borders(xlEdgeLeft).Weight = xlThick
        .Borders(xlEdgeRight).Weight = xlThick
        .Borders(xlEdgeBottom).Weight = xlThick
    End With

    'Подсчет и оформление итоговой суммы
    Range("C" & (NRow + 2)).Value = "ИТОГО:"
    Range("D" & (NRow + 2)).Value = WorksheetFunction.Sum(Range("D4", "D" & NRow))

    Set Rng = Range("C" & (NRow + 2), "D" & (NRow + 2))
    Rng.Font.Name = "Arial"
    Rng.Font.Bold = True
    Rng.Font.Size = 16
    Rng.Font.Color = xlRed
    Rng.Interior.Color = xlYellow

End Sub

```

ПРИЛОЖЕНИЕ 2. ФУНКЦИИ ПРОВЕРКИ ТИПОВ

Проверяют, является ли переменная выражением указанного типа.
Возвращают значение True, если переменная имеет заданный тип, и False - иначе.

Функция	Проверяет, является ли переменная
IsArray (переменная)	... массивом
IsDate (переменная)	... датой
IsError (переменная)	... кодом ошибки
IsNull (переменная)	... пустым значением (<i>Null</i>)
IsNumeric (переменная)	... числовым значением
IsObject (переменная)	... объектом

ПРИЛОЖЕНИЕ 3. ФУНКЦИИ ПРЕОБРАЗОВАНИЯ ТИПОВ

Таблица 1. Преобразование строки в число и обратно

Функция	Назначение
Val (строка)	Возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа
Str (число)	Возвращает значение типа Variant (String), являющееся строковым представлением числа. Допустимый десятичный разделитель - точка. При наличии другого десятичного разделителя (запятой) используйте функцию CStr.

Таблица 2. Функции преобразования типов данных в VBA

Функция	Возвращает тип	Действие
CBool (выражение)	Boolean	Преобразует значение в булевый тип
CByte (выражение)	Byte	Преобразует значение в тип Byte
CCur (выражение)	Currency	Преобразует значение в тип Currency
CDate	Date	Преобразует значение в Дату и Время
CDbl (выражение)	Double	Преобразует значение в тип Double
CDec (выражение)	Decimal	Преобразует значение в подтип Decimal типа Variant
CInt (выражение)	Integer	Преобразует значение в целый тип
CLng (выражение)	Long	Преобразует значение в длинное целое
CSng (выражение)	Single	Преобразует значение в тип Single
CStr (выражение)	String	Преобразует значение в строку
CVar (выражение)	Variant	Преобразует значение в тип Variant