

ЛАБОРАТОРНАЯ РАБОТА.

Разработка VBA-приложений в Microsoft Excel (занятие 5).

ЦЕЛЬ ЗАНЯТИЯ: МОДИФИКАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ И КОДА МАКРОСОВ, ПОЛУЧЕННЫХ В ХОДЕ ВЫПОЛНЕНИЯ ЗАНЯТИЙ 1-4.

ОПИСАНИЕ ПЛАНИРУЕМЫХ ИЗМЕНЕНИЙ ПРИЛОЖЕНИЯ

- Цель этапа – ознакомление с планируемыми изменениями пользовательского интерфейса и кода приложения Microsoft Excel.

На рис.1 приведен общий вид рабочего листа *Список товаров*, который должен быть получен к концу текущего занятия. Необходимо выполнить следующие действия:

- 1) Выделить ранее написанный код сохранения диаграммы в графическом файле и связать его с кнопкой *Сохранить диаграмму*, добавленную на рабочий лист.
- 2) Выделить ранее написанный код загрузки графического файла и связать его с кнопкой *Загрузить рисунок*.
- 3) Модифицировать ранее созданный код макроса построения диаграммы.
- 4) Добавить возможность удаления строк таблицы.
- 5) Добавить код отображения полученной диаграммы на экранной форме.

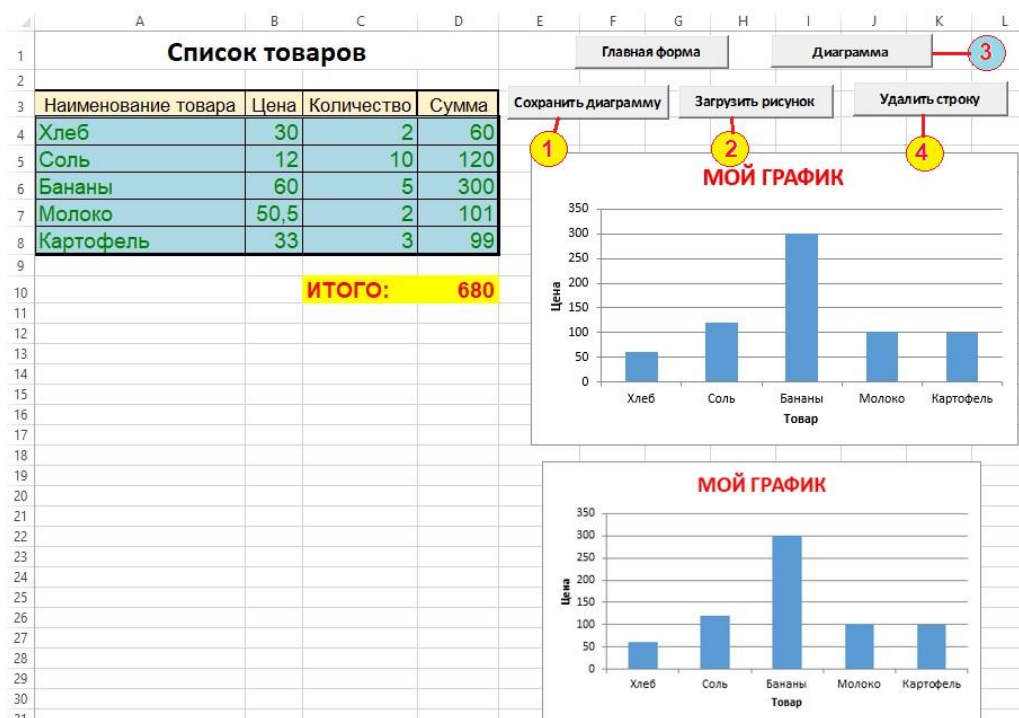


Рис. 1. Планируемые изменения рабочего листа *Список товаров*

Фрагменты кода макроса для построения диаграмм, которые будут использоваться при выполнении шагов 1-3 показаны на рис. 2.

```
Private Sub cmdDiag_Click()
    Charts.Add
    ActiveChart.ChartType = xlColumnClustered

    Dim NRow As Integer
    ' Поиск последней заполненной ячейки в таблице
    NRow = 4
    Do While Worksheets(2).Range("A" & NRow).Value <> ""
        NRow = NRow + 1
    Loop
    NRow = NRow - 1

    'Формирование строки-диапазона
    Dim SRng As String
    SRng = "A3:A" & NRow & ", D3:D" & NRow

    'Установка источника данных диаграммы
    ActiveChart.SetSourceData _
        Source:=Worksheets("Список товаров").Range(SRng), _
        PlotBy:=xlColumns

    ActiveChart.Location Where:=xlLocationAsObject, Name:="Список товаров"

    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "МОЙ ГРАФИК"
        .ChartTitle.Characters.Font.Color = xlRed

        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Товар"

        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Цена"

        .HasLegend = False
    End With

    ' Выгрузка диаграммы в графический файл
    Dim FullFileName As String
    FullFileName = CurDir + "\" + "График.gif"
    MsgBox Prompt:=FullFileName, Title:="Имя создаваемого файла"
    ActiveChart.Export Filename:=FullFileName, FilterName:="GIF"

    ' Загрузка графического файла на рабочий лист
    Range("A30").Select
    ActiveSheet.Pictures.Insert (FullFileName)
    MsgBox "Рисунок загружен!"
End Sub
```

Оформить в виде отдельной функции

1

2

Рис.2. Код обработчика нажатия кнопки *Диаграмма*

ЧАСТЬ 1. СОХРАНЕНИЕ ДИАГРАММЫ В ГРАФИЧЕСКОМ ФАЙЛЕ

□ Цель этапа – добавление опции сохранения диаграммы в графическом файле.

1. Скопируйте в свою рабочую папку файл Lab2-4b_Excel_<Фамилия>.xlsm, полученный при выполнении предыдущей работы, и сохраните его под именем Lab2-5_Excel_<Фамилия>.xlsm.
2. Перейдите в режим конструктора (вкладка *Разработчик*, кнопка *Режим конструктора*).
3. Разместите на рабочем листе компонент-кнопку (вкладка *Разработчик*, кнопка *Вставить*, группа *Элементы ActiveX*) и задайте его следующие свойства:

Элемент управления VBA (имя по умолчанию)	Свойство	Новое значение
CommandButton1	Name	<i>cmdSavePict</i>
	Caption	<i>Сохранить диаграмму</i>

4. Дважды щелкните по кнопке – на экране появится заготовка процедуры обработки события нажатия кнопки. Перенесите в нее фрагмент кода сохранения диаграммы в графическом файле из макроса построения диаграммы (рис. 2, блок 1).
5. Внесите в код макроса изменение, указанное на рис. 3:

```
Private Sub cmdSavePict_Click()
' Выгрузка диаграммы в графический файл
  Dim FullFileName As String
  FullFileName = CurDir + "\" + "График.gif"
  MsgBox Prompt:=FullFileName, Title:="Имя создаваемого файла"
  ChartObjects(1).Chart.Export Filename:=FullFileName, FilterName:="GIF"
End Sub
```

Рис.3. Код обработчика нажатия кнопки *Сохранить диаграмму*

6. Перейдите на рабочий лист *Список товаров* и проверьте работоспособность внесенных изменений.

ЧАСТЬ 2. ЗАГРУЗКА ГРАФИЧЕСКОГО ФАЙЛА НА ЛИСТ EXCEL

□ Цель этапа – добавление опции загрузки графического файла на рабочий лист.

7. Перейдите в режим конструктора (вкладка *Разработчик*, кнопка *Режим конструктора*).
8. Разместите на рабочем листе компонент-кнопку и задайте его следующие свойства:

Элемент управления VBA (имя по умолчанию)	Свойство	Новое значение
CommandButton1	Name	<i>cmdLoadPicture</i>
	Caption	<i>Загрузить рисунок</i>

9. Дважды щелкните по кнопке – на экране появится заготовка процедуры обработки события нажатия кнопки. Перенесите в нее фрагмент кода загрузки графического файла на лист Excel (рис. 2, блок 2).

10. Внесите в код макроса изменения, указанные на рис. 4:

```
Private Sub cmdLoadPicture_Click()
' Загрузка графического файла на рабочий лист
Dim FullFileName As String
FullFileName = CurDir + "\" + "График.gif"
MsgBox Prompt:=FullFileName, Title:="Имя загружаемого файла"
Range(ActiveCell.Address).Select
ActiveSheet.Pictures.Insert (FullFileName)
MsgBox "Рисунок загружен!"
End Sub
```

Рис.4. Код обработчика события нажатия кнопки Загрузить рисунок

11. Перейдите на рабочий лист *Список товаров* и проверьте работоспособность внесенных изменений. Заметьте, что теперь рисунок загружается в позицию активной ячейки.

ЧАСТЬ 3. МОДИФИКАЦИЯ МАКРОСА ПОСТРОЕНИЯ ДИАГРАММЫ

- Цель этапа – внесение дополнительных изменений в код макроса построения диаграммы (см. рис. 2).

Удаление с рабочего листа ранее построенных диаграмм

12. Вставьте в начало макроса **cmdDiag_Click()** строки кода, удаляющие с текущего рабочего листа ранее построенные диаграммы (см. рис.5)

```
Private Sub cmdDiag_Click()
    For i = ChartObjects.Count To 1 Step -1
        ChartObjects(i).Delete
    Next
```

Рис. 5. Добавляемый фрагмент кода макроса cmdDiag_Click()

- **ПРИМЕЧАНИЕ.** Используемые методы и свойства объекта/коллекции ChartObjects

Метод/Свойство	Назначение
ChartObjects.Count	Количество диаграмм на рабочем листе (свойство)
ChartObjects(i).Delete	Удаляет элемент семейства (метод)

13. Перейдите на рабочий лист *Список товаров* и проверьте работоспособность внесенных изменений.

Описание функции, возвращающей номер последней строки таблицы

Оформите в виде отдельной функции фрагмент кода, определяющего номер последней заполненной строки таблицы (возвращает адрес последней строки таблицы на листе Excel – см. выделенный фрагмент кода на рис. 2). Для этого выполните следующие действия:

14. В редакторе VBA выберите команду Insert->Procedure и в появившемся окне укажите ее имя (Name, значение **RowsInTable**), тип (*Function*) и область видимости (*Private*), см. рис. 6. В результате в редакторе кода появится заготовка функции (рис. 7)

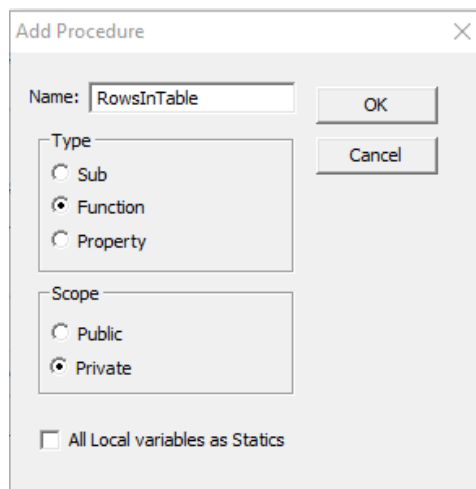


Рис.6. Диалоговое окно Add Procedure

```
Private Function RowsInTable()  
  
End Function
```

Рис.7. Заготовка пользовательской функции

15. Измените заголовок функции и перенесите в нее соответствующий фрагмент кода из макроса cmdDiag_Click() (см. рис. 2). Код, полученный в результате внесенных изменений, см. на рис. 8.

```
' Возвращает номер последней строки в таблице  
Public Function RowsInTable(wst As Worksheet) As Integer  
    Dim NRow As Integer  
    NRow = 4  
    Do While wst.Range("A" & NRow).Value <> ""  
        NRow = NRow + 1  
    Loop  
    NRow = NRow - 1 ' последняя строка в таблице  
    RowsInTable = NRow  
End Function
```

Рис.8. Код функции RowsInTable()

16. Вместо удаленного фрагмента кода вставьте в макрос cmdDiag_Click() строку с вызовом этой функции:

```
ActiveChart.ChartType = xlColumnClustered  
  
Dim NRow As Integer  
NRow = RowsInTable(Worksheets("Список товаров"))  
  
'Формирование строки-диапазона  
Dim SRng As String
```

17. Перейдите на рабочий лист *Список товаров* и проверьте работоспособность внесенных изменений.

ЧАСТЬ 4. УДАЛЕНИЕ СТРОКИ ИЗ ТАБЛИЦЫ *СПИСОК ТОВАРОВ*

- **Цель этапа** – программная реализация возможности удаления выбранной позиции из списка товаров.

Для удаления строки с выбранным товаром необходимо сделать текущей любую ячейку выбранной строки и нажать кнопку *Удалить строку*, размещенную рабочем листе. Текст макроса контролирует номер удаляемой строки (только для товарных позиций в таблице).

18. Перейдите в режим конструктора (вкладка *Разработчик*, кнопка *Режим конструктора*).

19. Разместите на рабочем листе компонент-кнопку и задайте его следующие свойства:

Элемент управления VBA (имя по умолчанию)	Свойство	Новое значение
CommandButton1	Name	<i>cmdDelete</i>
	Caption	<i>Удалить строку</i>

20. Дважды щелкните по кнопке – на экране появится заготовка процедуры обработки события нажатия кнопки. Наберите следующий код процедуры (рис. 9):

```
Private Sub cmdDelete_Click()
    Dim LastRow As Integer, NRow As Integer
    LastRow = RowsInTable(Worksheets("Список товаров"))
    NRow = ActiveCell.Row
    If NRow >= 4 And NRow <= LastRow Then
        Rows(NRow).Delete
        MsgBox Prompt:="Строка удалена", Title:="Удаление позиции списка"
    Else
        MsgBox Prompt:="Неверно выбрана строка!", Title:="Удаление позиции списка"
    End If
End Sub
```

Рис.9. Код макроса удаления строки

□ ПОЯСНЕНИЯ К КОДУ.

а) Перечень товаров в таблице начинается с 4-й строки.

б) Для определения номера последней строки, содержащей описание товара, используется функция `RowsInTable()`, результат выполнения которой сохраняется в переменной `LastRow`

в) Номер удаляемой строки (`NRow`) равна номеру строки активной ячейки.

г) Условный оператор в макросе обеспечивает контроль корректности номера удаляемой строки.

□ ПРИМЕЧАНИЕ. Используемые методы и свойства

Метод/Свойство	Назначение
<code>ActiveCell.Row</code>	Номер строки активной ячейки (свойство)
<code>Rows(NRow).Delete</code>	Удаление строки с номером <code>NRow</code> (метод)




21. Перейдите на рабочий лист *Список товаров* и проверьте работоспособность внесенных изменений. Отметьте изменения, происходящие с диаграммой.

ЧАСТЬ 5. ОТОБРАЖЕНИЕ ДИАГРАММЫ НА ЭКРАННОЙ ФОРМЕ

- **Цель этапа** – создание дополнительной экранной формы для отображения на ней диаграммы.

22. Вставьте в проект новую форму (команда меню Insert->User Form).

23. В окне Toolbox (**Элементы управления**) выберите следующие элементы управления и поместите их на форму:

- 1 рисунок (элемент управления Image, );
- 1 кнопку (элемент управления CommandButton, );
- 3 переключателя (элемент управления OptionButton, .

24. 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>frmDiag</i>
	Caption	<i>Список товаров</i>
CommandButton1	Name	<i>cmdClose</i>
	Caption	<i>Заккрыть</i>
OptionButton1	Name	<i>obColumn</i>
	Caption	<i>Гистограмма</i>
OptionButton2	Name	<i>obLine</i>
	Caption	<i>График</i>
OptionButton3	Name	<i>obArea</i>
	Caption	<i>С областями</i>

Внешний вид создаваемой формы см. на рис. 10-11.

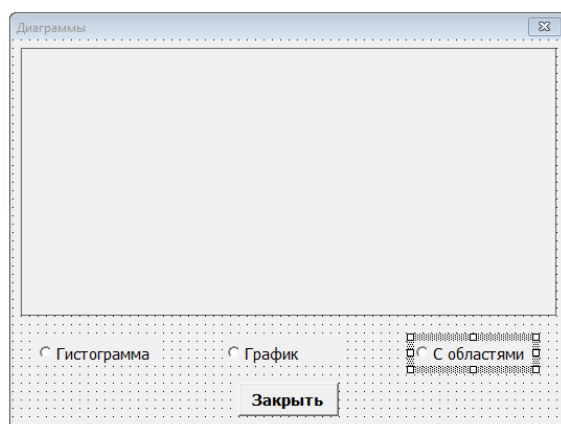


рис.10. Вид формы на этапе проектирования

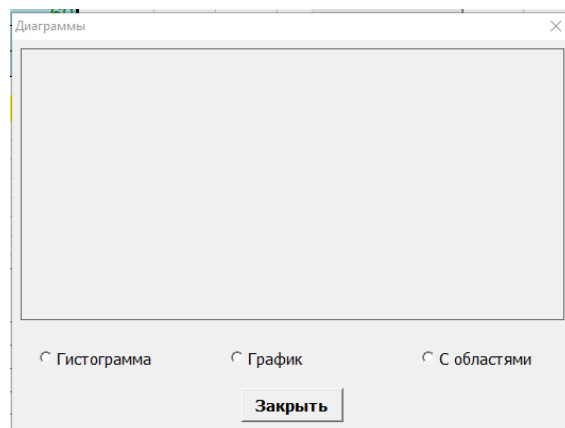


рис.11. Вид формы на этапе выполнения

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

26. Добавьте 1 кнопку на главную форму (**frmGoods**), задайте для свойства Name значение **cmdDiag**, а для свойства Caption - значение **Диаграммы**; настройте ее размер и размещение на форме, цвет заливки, формат шрифта в соответствии с оформлением формы. Примерный вид измененной формы см. на рис. 12.

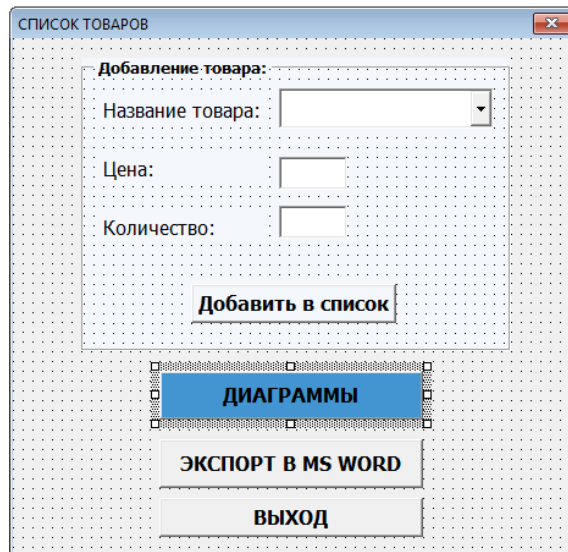


Рис. 12. Модифицированная главная форма (пример)

27. Дважды щелкните мышью по кнопке **Диаграммы** (cmdDiag). Откроется редактор кода, в котором автоматически будет создана заготовка процедуры обработчика события нажатия кнопки. Введите в заготовку процедуры следующую строку:

```
frmDiag.Show
```

28. Перейдите на рабочий лист *Список товаров* и проверьте работоспособность внесенных изменений.
29. Перейдите в редактор кода, **выберите в окне VBAProject форму *Диаграммы* (frmDiag)**, и создайте для нее макрос-обработчик события инициализации формы (рис. 13)

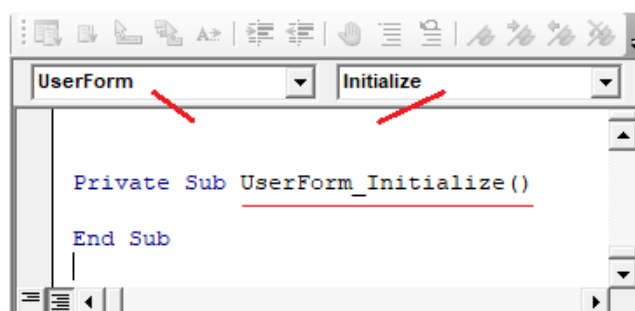


Рис. 13. Заготовка макроса инициализации формы

30. Введите в заготовку процедуры следующую строку:

```
obColumn.Value = True
```

31. Дважды щелкните мышью по кнопке **Закреть** (cmdClose) и в появившейся заготовке процедуры наберите строку:

```
Unload Me
```

32. Проверьте работоспособность программы и вернитесь в редактор VBA.

33. Для формы *Диаграммы* наберите код вспомогательной процедуры (рис. 14):

```
' Отображение диаграммы заданного типа
Private Sub UpdateChart(chtype)
    Dim CurrentChart As Chart
    Dim FullFileName As String
    ' Выбор диаграммы на рабочем листе
    Set CurrentChart = Sheets("Список товаров").ChartObjects(1).Chart
    CurrentChart.ChartType = chtype
    ' Сохранение диаграммы в виде GIF-файла
    FullFileName = ThisWorkbook.Path & "\" + "Temp.gif"
    CurrentChart.Export Filename:=FullFileName, FilterName:="GIF"
    ' Загрузка диаграммы на форму
    Image1.Picture = LoadPicture(FullFileName)
End Sub
```

Рис.14. Код вспомогательной процедуры

□ ПОЯСНЕНИЯ К КОДУ.

а) Параметр *chtype* (тип *Variant*) используется для указания типа отображаемой диаграммы

б) Перед отображением диаграммы на форме выполняется следующая последовательность действий: (1) на рабочем листе строится диаграмма выбранного типа, (2) построенная диаграмма сохраняется во временном графическом файле, (3) сохраненный на диске рисунок загружается на форму. Временный файл будет удален при закрытии формы (см. ниже по тексту).

34. Поочередно дважды щелкнув по переключателям *Гистограмма*, *График*, *С областями* внесите в заготовки макросов строку с кодом вызова созданной процедуры для отображения диаграммы соответствующего типа:

```
UpdateChart (xlColumnClustered) //для гистограммы
...
UpdateChart (xlLine) // для графика
...
UpdateChart (xlArea) // для диаграммы с областями
```

35. Проверьте работоспособность измененной программы и вернитесь в редактор VBA.

36. Дважды щелкните мышью по кнопке *Закрыть* (cmdClose) и допишите в начало макроса строку, уничтожающую временный графический файл, сохраненный на диске:

```
Private Sub cmdClose_Click()
    Kill ThisWorkbook.Path & "\" + "Temp.gif"
    Unload Me
End Sub
```