

**лабораторная работа № 10**

**VISUAL BASIC FOR APPLICATIONS.  
ФОРМЫ**

# 1. Создание функций пользователя

Функция – это подпрограмма, которая действует в пределах своего блока и возвращает только одно значение.

Синтаксис функции имеет следующий вид:

```
[Private / Public] Function ИмяФункции (Список _  
    Параметров) As Тип Данных  
    «Тело функции»  
    Имя Функции = Возвращаемое Значение  
End Function
```

Например:

```
Function GetRandomNumber ()  
    GetRandomNumber = Int(10*Rnd())  
End Function
```

Рассмотрим порядок создания функции пользователя. Для создания функции пользователя нужно:

1. Если в проекте нет модуля, то создать его командой *Insert / Module* (*Вставка / Модуль*);
2. Выполнить команду меню редактора VB *Insert / Procedure* (*Вставка / Процедура*);
3. В окне *Вставка процедуры* установить переключатель *Function* (*Функция*);
4. В поле *Name* (*Имя*) ввести, например, имя функции (*Summa*);
5. Установить переключатель *Public* (*Общая*) в *Области Определения*;
6. Щелкнуть *OK*. После выполнения этих действий в окне модуля появится заготовка функции (Заголовок *Public Function Summa (Список параметров)* и Окончание *End Function*), между которыми нужно поместить код тела функции (рис. 1.1);

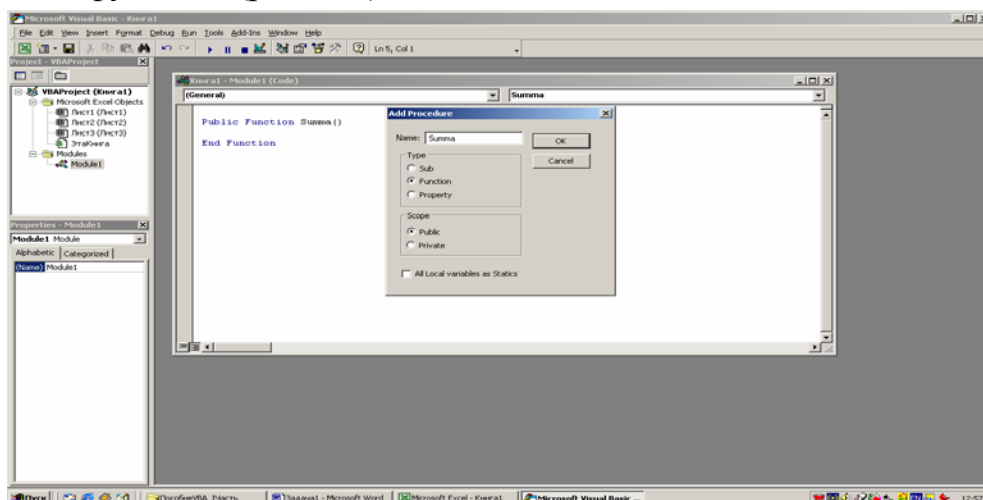


Рис. 1.1. Лист модуля с окном «Создание функции»

7. Ввести список параметров функции, их типов данных, а также указать тип возвращаемого функцией значения;
8. Используя команду меню *Вид / Просмотр объектов* или клавишу <F2>, вызвать окно *Просмотр объектов*;
9. Раскрыть список верхнего левого окна (*Список проектов*) и выбрать из него *VBA Project*. В окне *Классы* отобразятся элементы текущего проекта;
10. Выбрать в этом окне модуль, в котором создана функция – в области *Компоненты* отобразятся элементы, которые содержатся в этом модуле;
11. Выделить в области *Компоненты* элемент с именем созданной вами функции и включить контекстное меню;
12. Выполнить команду этого меню *Свойства (Properties)* – откроется окно *Параметры компонента*;
13. В поле *Описание* этого окна ввести текст краткого описания этой функции (рис. 1.2).

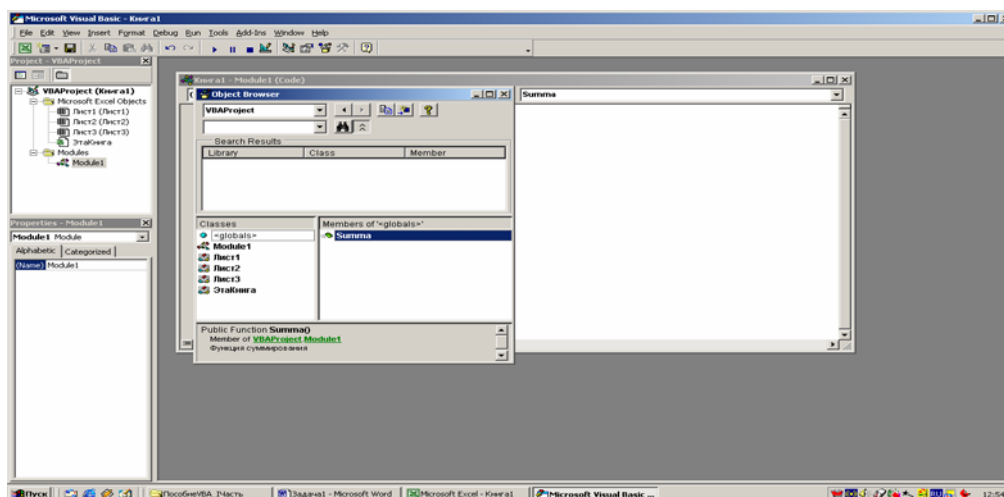


Рис. 1.2. Диалоговое окно «Просмотр объектов»

Существует и другой способ создания функции.

1. Находясь в Excel и не открывая окно редактора Visual Basic, выполнить команду *Сервис / Макрос / Макросы*;
2. В открывшемся окне *Макрос* ввести имя макроса, например *Summa*, и нажать на кнопку <Создать>;
3. В результате откроется приложение VBA, в котором будет открыт существующий модуль (или создан новый), например *Модуль1*;
4. На экране будут выведены строки:  

```
Sub Summa ()
```

```
End Sub
```
5. Заменить эти строки на листе модуля строками заголовка и конца функции следующего вида:

```
Public Function Summa()
```

```
End Function
```

Рассмотрим создание функции пользователя на примере функции «*Doxod*», вычисляющей разность между текущим объемом вклада и размером ссуды при постоянной годовой процентной ставке и неравномерных платежах, руководствуясь следующей последовательностью действий:

1. Находясь в Excel и не открывая окно редактора Visual Basic, выполнить команду *Сервис / Макрос / Макросы*;
2. В открывшемся окне *Макрос* ввести имя макроса, например *Doxod*, и нажать на кнопку *<Создать>*;
3. В результате откроется приложение VBA, в котором будет создан новый модуль, например *Модуль1*;
4. На экране будут выведены строки:

```
Sub Doxod()
```

```
End Sub
```

5. Заменить эти строки на листе модуля строками заголовка и конца функции следующего вида:

```
Public Function Doxod()
```

```
End Function
```

6. Набрать следующий программный код между этими строками

```
Option Explicit
```

```
Option Base 1
```

```
Function Doxod(procent As Double, platezh As _  
Variant, god As Variant) As Double
```

```
Dim i As Integer, N As Integer, Sum As Double
```

```
N = platezh.Rows.Count
```

```
Sum = 0
```

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

```
Sum = Sum + platezh(i) / (1+ procent) ^ ((god(i) - _  
god(1)) / 365)
```

```
Next i
```

```
Doxod = Sum
```

```
End Function
```

7. Вернуться в Excel и на рабочем листе создать табл. 1.1.

Таблица 1.1

*Решение задачи о неравномерных платежах*

	А	В	С
1	Наименования показателей	Неравномерные платежи	Даты
2	Ссуда	-10000	11.01.2008
3	Суммы возврата:	2000	20.05.2008
4		4000	18.10.2008
5		7000	12.12.2009
6			
7	Годовая процентная ставка	10%	
8	Разность между чистым текущим объемом вклада и размером ссуды	1480,83	
9	2 – ой вариант возврата ссуды		
10	Ссуда	-10000	11.01.2008
11	Суммы возврата:		
12			
13			
14			
15	Годовая процентная ставка		
16	Разность между чистым текущим объемом вклада и размером ссуды		
17	3 – ий вариант возврата ссуды		

8. В ячейку B8 ввести формулу вида:

$$=Doxod(B7; B2 : B5; C2 : C5)$$

9. В ячейке B8 будет получен результат, приведенный в табл. 1.1. Так как результат положителен, данная сделка выгодна.

10. Во втором варианте изменить суммы и даты возврата таким образом, чтобы сделка оказалась выгодной дающему ссуду.

11. В третьем варианте изменить суммы и даты возврата таким образом, чтобы сделка была выгодной берущему ссуду.

Замечание. В данном программном коде функции используются:

- метод Rows, который возвращает строки диапазона platezh;
- свойство Count, которое считает число элементов объекта;
- platezh.Rows.Count, который определяет число строк в диапазоне platezh.

В рассмотренном примере с функцией *Doxod* вызов функции происходит из ячейки рабочего листа *Excel*. Кроме этого способа обращаться к функции можно и из процедуры. Рассмотрим такой вариант вызова функции в примере организации модуля для вычисления площадей геометрических фигур.

```
Option Explicit
Option Base 1
Const PI As Double = 3,1416
' PI - глобальная постоянная
' Priz, A, B, R - глобальные переменные
Dim Priz As Integer
Dim A As Double, B As Double, R As Double

Function Figura (a1 As Double, b1 As Double) _
    As Double
    ' Figura Вычисляет площадь прямоугольника
    Figura = a1 * b1
    Priz = 1
End Function

Function Crug (r1 As Double) As Double
    ' Crug вычисляет площадь круга
    Crug = PI * r1 ^ 2
    Priz = 2
End Function

Sub Rezalt ()
    Dim PlFig As Double, PlCr As Double
    ' PlFig, PlCr - локальные переменные
    R = InputBox ("Введите значение радиуса круга")
    PlCr = Crug (R)
    A = InputBox ("Введите ширину прямоугольника")
    B = InputBox ("Введите высоту прямоугольника")
    PlFig = Figura (A, B)
    If Priz = 1 Then
        MsgBox "Площадь прямоугольника =" & CStr(PlFig)
    Else
        MsgBox "Площадь круга = " & CStr(PlCr)
    End If
End Sub
```

Часто при решении экономических задач возникает необходимость проведения расчетов, в основе которых лежат процессы преобразования матриц и векторов или их совместная обработка. В этом случае используются функции рабочего листа Excel для работы с матрицами, приведенные в табл. 1.2.

Таблица 1.2

*Функции рабочего листа для работы с матрицами*

Наименование		Описание
русское	английское	
Мумнож	Mmult	Произведение двух матриц
Мобр	Minverse	Обратная матрица
Трансп	Transpose	Транспонированная матрица
Мопред	Mdeterm	Определитель матрицы

Если в программном коде VBA используются стандартные функции Excel, то обращение к ним из программного кода начинается со служебного слова *Application*.

Рассмотрим использование этих функций в процессе разработки функции пользователя для решения системы линейных уравнений вида  $A * X = B$ , где  $A$  – матрица коэффициентов, расположенная в диапазоне  $A2 : C4$ ,  $B$  – столбец свободных членов, расположен в диапазоне  $D2 : D4$ ,  $X$  – столбец неизвестных. Решение системы имеет вид:

$$X = A^{-1} * B,$$

где  $A^{-1}$  – обратная матрица.

Ниже приведен программный код функции пользователя для решения системы линейных уравнений.

```
Function Solver (A As Variant, B As Variant) _
    As Variant
    Solver = Application .Mmult(Application _
        .Minverse(A), B)
End Function
```

Обратиться к этой функции можно из рабочего листа Excel, выполнив следующие действия:

- выделить диапазон ячеек  $E2 : E4$
- в строке формул ввести формулу
- $=\text{Solver}(A2 : C4; D2 : D4)$

## Задания для самостоятельной работы по созданию функций пользователя

**Задание 1.** Создайте функцию пользователя для генерации случайных чисел в диапазоне от 1 до 9 с именем *GetRndN*. Проверьте, что вновь созданная функция появилась в общем списке функций, вызвав, например, *Мастер функций* в MS Excel. Используйте эту функцию при написании процедуры для сообщения о выигрыше или проигрыше.

**Задание 2.** Создайте функцию пользователя с именем *GetPassword* для ввода пароля. При написании функции используйте окно ввода *Input-Box* - для ввода пароля и функцию *Lcase*, которая преобразует все буквы переданной в нее строки в строчные. Обратитесь к этой функции в процессе написания процедуры, проверяющей значение введенного пароля, для принятия решения о дальнейшем выполнении приложения.

**Задание 3.** Создайте функцию пользователя *Nadbavka* для начисления надбавок к окладам медицинских работников поликлиники № 10 по следующему алгоритму:

- Если стаж работы медицинского работника превышает 20 лет в данной организации, то надбавка к окладу должна составлять 20 % от оклада.
- Если стаж работы медицинского работника превышает 25 лет в данной организации, то надбавка к окладу должна составлять 25 % от оклада.
- Если стаж работы медицинского работника превышает 30 лет в данной организации, то надбавка к окладу должна составлять 30 % от оклада.

Замечание. В качестве параметров функции использовать переменные:

- *Stazh* – для стажа работы;
- *Salary* – для оклада сотрудника.

Записать обращение к функции из ячейки рабочего листа Excel.

**Задание 4.** Создайте функцию пользователя *Bonus* для начисления комиссионных менеджерам по продажам магазина «Эльдорадо» по следующему алгоритму:

- Если объем продаж не менее 500 000 рублей, то комиссионные равны P1 процентов от объема продаж.
- Если объем продаж от 100 000 до 500 000 рублей, то комиссионные равны P2 процентов от объема продаж.
- Если объем продаж от 50 000 до 100 000 рублей, то комиссионные равны P3 процентов от объема продаж.
- Если объем продаж менее 50 000 рублей, то с помощью функции *MsgBox* выдать сообщение о невозможности выплаты комиссионных.

Замечание. В качестве параметров функции использовать переменные:

- *Volume* – объем продаж;
- *P1*, *P2*, *P3* – проценты для начисления комиссионных.



С помощью функции «*Bonus*» вычислить комиссионные для двух сценариев: P1 = 5%, P2=3%, P3=1% и P1 = 7%, P2=5%, P3=3%.

Записать обращение к функции из ячеек рабочего листа Excel.

**Задание 5.** Создайте функцию пользователя «*Tax*» для начисления налога с корпорации. Выручка от продаж корпорации за год равна 4 200 000 рублей, производственные расходы составляют 1 650 000 рублей. Корпорация сделала заем в банке в сумме 1 500 000 рублей, за который выплатила 15% годовых. Амортизация оборудования равна 360 000 рублей. С помощью функции «*Tax*» вычислить сумму налога, которую должна выплатить корпорация. Алгоритм решения задачи следующий:

- вычислить расходы на выплату процентов за пользование капиталом, полученным в долг от банка;
- вычислить налогооблагаемый доход, который равен выручке за вычетом производственных расходов, расходов на выплату процентов за пользование капиталом, полученным в долг и амортизации оборудования;
- вычислить сумму налога путем умножения налогооблагаемого дохода на ставку налога.

На рабочем листе Excel создайте табл. 1.3.

Таблица 1.3

*Решение задачи о начислении налога*

	А	В	С
1	Наименования показателей	Варианты решения	
2		1-й	2-й
3	Выручка от реализации		
4	Производственные расходы		
5	Величина займа в банке		
6	Годовая процентная ставка		
7	Амортизация		
8	Ставка налога на доход		
9	Сумма налога		

Внесите в табл. 1.2 в ячейки столбца В исходные данные из задачи, а в ячейки столбца С те же данные, но с другим значением годовой процентной ставки за пользование капиталом, полученным в долг от банка.

В ячейки В9 и С9 введите формулы с обращением к функции «*Tax*».

## 2. Создание процедур

**Задача 2.1.** Разработать процедуру *Substitute*, которая заменяет на рабочем листе с именем «Замена» все положительные числа диапазона ячеек *B2 : E5* знаком «\*», все отрицательные числа – знаком «%», а нули заменяет знаком «!».

Порядок решения задачи следующий:

1. Рабочему листу Лист1 присвоить имя «Замена».
2. Заполнить диапазон ячеек *B2 : E5* рабочего листа «Замена» положительными, отрицательными числами и нулями.
3. Перейти в режим редактора *Visual Basic*, выполнив команду *Сервис / Макрос / Редактор Visual Basic*.
4. Выполнить команды *Insert / Module* и *Insert / Procedure*.
5. Ввести имя процедуры *Substitute*.
6. Набрать программный код:

```
Sub Substitute()  
    Dim A As Object  
    For Each A In Worksheets("Замена").Range(B2:E5)  
        If IsNumeric(A.Value) Then  
            If A.Value > 0 Then  
                A.Value = "*"   
            ElseIf A.Value < 0 Then  
                A.Value = "%"   
            Else  
                A.Value = "!"   
            End If  
        Else  
            MsgBox "В диапазоне B2 : E5 должны быть числа"  
        End If  
    Next A  
End Sub
```

7. Вернуться на рабочий лист «Замена».
8. Запуск процедуры оформить кнопкой, расположенной на рабочем листе, там же, где расположен список. Для этого вывести на экран панель инструментов *Формы*. Четвертый элемент управления этой панели предназначен для создания кнопок, которые запускают процедуры - макросы. Для создания кнопки щелкните на кнопке панели инструментов и нарисуйте прямоугольную кнопку на листе «Замена». Вам будет предложено назначить макрос для данной кнопки. Это нужно будет сделать немедленно. Затем ввести название для этой кнопки «*Заменить*». Для запуска программы нужно просто щелкнуть на этой кнопке.

**Задача 2.2.** Разработать процедуру *Colour*, которая преобразует внешний вид ячеек выделенной области рабочего листа в зависимости от содержимого ячеек этой области. Если число в ячейке  $> 0$ , то изменяется цвет фона ячейки, размер, начертание и цвет шрифта содержимого этой ячейки. Если число в ячейке  $< 0$ , то изменяется цвет фона ячейки, начертание шрифта содержимого этой ячейки. Если число в ячейке  $= 0$ , то изменяется цвет фона ячейки, шрифт становится подчеркнутым.

Порядок решения задачи следующий:

1. Рабочему листу Лист2 присвоить имя «*Цвет*».
2. Заполнить произвольный диапазон ячеек рабочего листа «*Цвет*» положительными, отрицательными числами и нулями.
3. Перейти в режим редактора *Visual Basic*, выполнив команду *Сервис / Макрос / Редактор Visual Basic*.
4. Выполнить команды *Insert / Module* и *Insert / Procedure*.
5. Ввести имя процедуры *Colour*.
6. Набрать программный код:

```
Sub Colour()  
    Dim B As Object  
    For Each B In Selection  
        If IsNumeric(B.Value) Then  
            If B.Value > 0 Then  
                B.Interior.Colorindex = 6  
                B.Font.Size = 18  
                B.Font.Bold = True  
                B.Font.Colorindex = 7  
            ElseIf B.Value < 0 Then  
                B.Font.Italic = True  
                B.Interior.Colorindex = 3  
            Else  
                B.Underline = xlDouble  
                B.Interior.Colorindex = 4  
            End If  
        Else  
            MsgBox "В выделенном диапазоне должны быть _  
                числа", vbOKOnly + vbInformation, "Числа"  
        End If  
    Next A  
End Sub
```

Для запуска процедуры на рабочем листе «*Цвет*» создать кнопку с панели инструментов *Формы* и написать на ней название «Изменить цвет выделенного диапазона».

**Задача 2.3.** Написать процедуру поиска в списке, расположенном на рабочем листе с именем «*Цены*» активной рабочей книги, кода товара, введенного с клавиатуры, с выдачей на экран сообщения о цене введенного товара. Если код не найден, то пользователю сообщается об отсутствии товара в списке. Список состоит из двух полей: «*Код товара*», «*Цена единицы товара*». Количество записей заранее неизвестно. В ячейках *A3* и *B3* располагаются названия полей списка. Пример кода товара: *П1203*.

Порядок решения задачи.

1. Рабочему листу Лист3 присвоить имя «*Цены*».
2. В ячейках *A3* и *B3* разместить заголовки столбцов «*Код товара*» и «*Цена единицы товара*» соответственно.
3. Заполнить список записями об имеющейся в продаже номенклатуре (коде) товара и цене за единицу товара. Если конкретный товар отсутствует в продаже, то в списке его нет.
4. Количество записей в списке должно быть от 15 до 20.
5. Перейти в режим редактора *Visual Basic*, выполнив команду *Сервис / Макрос / Редактор Visual Basic*.
6. Оформить решение задачи приложением. В процедуре *SearchPrice* использовать массивы для хранения кодов товаров и цен за единицу товара каждого вида. Сначала подсчитать общее количество товаров в списке, затем определить размер массивов *CodeGood* и *Price*; в цикле *For* эти массивы заполняются данными списка из столбцов *A* и *B* соответственно. Далее у пользователя запрашивается код товара и еще в одном цикле *For* просматривается массив *CodeGood* для поиска введенного пользователем значения. Если код товара в массиве кодов товаров найден, то в переменной *FindPrice* сохраняется цена из массива *Price*, соответствующая найденному коду, а логической переменной *Flag* присваивается значение *True*. В зависимости от значения этой переменной выдается сообщение либо о найденной цене, либо, если код товара не найден, сообщение об отсутствии товара в списке товаров, имеющихся в продаже.
7. Ввести программный код.

```
Sub SearchPrice()  
    Dim CodeGood () As String, Price () As _  
        Currency, FindPrice As Currency, _  
        Nproducts As Integer, i As Integer, _  
        Flag As Boolean, FindCode As String  
    ' Определяет количество записей в списке, переопределяет  
    ' размер массивов и формирует массивы  
    With .ActiveWorkbook .Worksheets("Цены"). _  
        Range("A3")  
        Nproducts=Range(.Offset(1,0),.End(xlDown)). _  
            Rows.Count
```

```

    ReDim CodeGood(Nproducts)
    ReDim Price(Nproducts)
    For i = 1 To Nproducts
        CodeGood (i) = .Offset (i,0)
        Price (i) = .Offset (i,1)
    Next I
End With
` Ввод кода товара пользователем
    FindCode = InputBox ("Введите код товара (с
        большой буквы и четырьмя цифрами.")
` Поиск кода товара в списке. Сохранение найденной
цены товара.
    Flag = False
    For i = 1 To Nproducts
        If CodeGood (i) = FindCode Then
            Flag = True
            FindPrice = Price (i)
            Exit For
        End If
    Next i
` Вывод сообщения в окно сообщения в зависимости от
значения логической переменной Flag.
    If Flag Then
        MsgBox "Товар с кодом " & FindCode _
            & " стоит " & Format (FindPrice, _
            "0.00p."), vbInformation, "Товар найден"
    Else
        MsgBox "Товара с кодом " & FindCode _
            & " нет в списке", vbInformation, "Товар _
            не найден"
    End If
End Sub

```

9. Запуск процедуры оформить кнопкой, расположенной на рабочем листе, там же, где расположен список. Для этого вывести на экран панель инструментов *Формы*. Четвертый элемент управления этой панели предназначен для создания кнопок, которые запускают процедуры - макросы. Для создания кнопки щелкните на кнопке панели инструментов и нарисуйте прямоугольную кнопку на листе «*Цены*». Вам будет предложено назначить макрос для данной кнопки. Это нужно будет сделать немедленно. Затем ввести название для этой кнопки «*Поиск цены*». Для запуска программы нужно просто щелкнуть на этой кнопке.

**Задача 2.4.** Торговая фирма «Шанс» занимается продажей компьютерной техники и ведет учет всех торговых операций, связанных с ее продажами. С этой целью на рабочем листе Excel с именем «*Торговые операции*» активной рабочей книги в столбцах с A по C создается список – база данных в миниатюре. В ячейке A1 размещается заголовок исходного списка «*Информация о продажах*». Список начинается со второй строки. Вторая строка содержит названия полей списка: «*Код товара*», «*Дата продажи*», «*Объем продаж в рублях*». Заголовки полей списка выделены жирным шрифтом и расположены по центру, размер шрифта 16. Каждая последующая строка содержит трехзначный код проданного товара, дату в формате даты, например, 1.12.2009 и размер сделки в денежном формате. Данные о продажах отсортированы по дате продаж. На основе этого списка нужно определить количество проданного товара каждого наименования, общее количество наименований проданных товаров, суммарный объем продаж каждого наименования товара и общую сумму, вырученную от продаж компьютерной техники. На основе этих результатов формируется новый список. В ячейке E1 размещается заголовок нового списка «*Сводные данные о продажах*», а во второй строке столбцов с E по G располагаются названия полей списка: «*Код товара*», «*Количество продаж каждого наименования*», «*Суммарный объем продаж в рублях каждого наименования*» соответственно. Полученные результаты размещаются в столбцах с E по G нового списка, начиная с третьей строки, и сортируются по убыванию объемов продаж.

Оформить решение данной задачи процедурой *TradeOperations*, которая выполняет все описанные выше действия. Запуск процедуры оформить кнопкой на рабочем листе там же, где расположены списки. Для запуска процедуры щелкнуть на кнопке. В результате будет выдано сообщение о том, сколько было продано товаров из базы данных и общая сумма выручки от продаж компьютерной техники. Кроме того, будет создан новый список, о котором было сказано выше.

Алгоритм решения задачи.

Основной конструкцией программы является цикл, просматривающий коды товаров из столбца A, которые хранятся в массиве *CodesData*. Этот цикл используется для создания массива *CodesFound*. В итоге полученный массив будет содержать уникальные коды товаров, указанные в столбце A. В каждой итерации цикла код товара сравнивается со всеми уже найденными кодами товаров. Если код был найден ранее, то к количеству продаж этого товара добавляется 1, а стоимость проданного товара добавляется к общей стоимости товара с данным наименованием. В противном случае, если код товара ранее не был найден, то он добавляется к массиву *CodesFound*, а количество продаж этого товара устанавливается равным 1. Объем продаж товара принимает значение, равное стоимости текущей операции. Три других массива помогают в ведении учета. Массив *VolData* содержит данные из столбца C, а

массивы *Quantity* и *SumVol* хранят количество продаж и общую их стоимость по каждой номенклатуре проданного товара.

Как только просмотр кодов товаров в столбце *A* будет завершен, данные из массивов *CodesFound*, *Quantity* и *SumVol* заносятся в столбцы с *E – G* нового списка. Этот список сортируется по значениям столбца *G* в порядке убывания объемов продаж.

```
Option Explicit
Option Base 1
Sub TradeOperations()
    Dim Qsales As Integer, CodesData() As _
        Integer, CodesFound() As Integer, _
        Quantity() As Integer, VolData() As _
        Double, SumVol() As Double, i As _
        Integer, j As Integer, Col As Integer, _
        Volume As Double, Flag As Boolean _
        ColSales As Integer
    ' Очищает содержимое столбцов E – G.
    With ActiveWorkbook.Worksheets("Торговые _
        операции").Range("E2")
        Range(.Offset(1,0),.Offset(0,2).End (xlDown _
            )).ClearContents
    End With
    'Определяет количество записей в списке,
    'изменяет размерность массивов CodesData и 'VolData
    и заполняет их данными столбцов A и C.
    With ActiveWorkbook.Worksheets("Торговые _
        операции").Range("A2")
        Qsales=Range(.Offset(1,0),.End(xlDown)) _
            .Rows.Count
        ReDim CodesData (Qsales)
        ReDim VolData (Qsales)
        For i = 1 to Qsales
            CodesData (i) = .Offset (i,0)
            VolData (i) = .Offset (i,2)
        Next i
    End With
    'Начальная инициализация переменных Col и Volume.
    Col = 0 : Volume = 0
    For i = 1 To Qsales
    'Присваивание переменной Flag значения = True;
    'переменная делается равной False при нахождении
    'уже найденного ранее кода в массиве CodesFound().
        Flag = True
```

```

        If Col > 0 Then
'Просмотр всех найденных кодов товаров и сравнение
'их с текущим кодом товара
        For j = 1 To Col
            If CodesData(i) = CodesFound(j) Then
'Текущий код товара был найден ранее, поэтому
'значения Quantity(j) и SumVol(j) увеличиваются
                Flag = False
                Quantity (j) = Quantity (j) + 1
                SumVol (j) = SumVol (j) + VolData (i)
            Exit For
        End If
    Next j
End If
If Flag = True Then
'Текущий код товара новый, поэтому обновляется —
'список кодов, а элементы массивов Quantity() и —
'SumVol() инициализируются исходными значениями
    Col = Col + 1
    ReDim Preserve CodesFound (Col)
    ReDim Preserve Quantity (Col)
    ReDim Preserve SumVol (Col)
    CodesFound (Col) = CodesData(i)
    Quantity (Col) = 1
    SumVol (Col) = VolData (i)
End If
Next i
'Значения элементов сформированных массивов
'заносятся в столбцы E - G
    For j = 1 To Col
        Volume = Volume + SumVol (j)
        ColSales = ColSales + Quantity (j)
        With Range ("E2")
            .Offset (j, 0) = CodesFound (j)
            .Offset (j, 1) = Quantity (j)
            .Offset (j, 2) = SumVol (j)
        End With
    Next j
' Сортировка нового списка по объемам продаж
' (столбец G) в порядке убывания их объемов.
    Range("E3") .Sort Key1 :=Range("G3"), Order1 —
        :=xlDescending, Header :=xlYes
    With Range ("D2").offset (Col+2, 0).Value= —
        "Всего продано:"
        With Range ("E2")
            .Offset (Col+2, 1) = ColSales
            .Offset (Col+2, 2) = Volume
        End With
    End With

```



```

MsgBox "В базе данных проданных товаров _ найде-
но " & Col & " наименований.
Общий объем продаж равен " & Volume & " рублей."
End Sub

```

Пояснения к программе. Массивы *CodesData* и *VolData* меняют размер без указания ключевого слова *Preserve*, а массивы *CodesFound*, *Quantity*, *SumVol*, - с сохранением исходного содержимого. Причиной такого поведения является однократное изменение размера первых двух массивов, когда в массивах еще не сохранены никакие данные. Поэтому сохранение содержимого массивов просто не выполняется. Остальные три массива изменяют размер при каждом добавлении нового элемента (кода товара). В этих массивах приходится сохранять уже существующие элементы. При обнаружении нового кода товара значение переменной *Col* увеличивается на 1, кроме того, изменяются размеры массивов *CodesFound*, *Quantity* и *SumVol*. Затем в добавленные элементы массивов заносятся соответствующие значения.

Чтобы указать диапазон, по которому выполняется сортировка, достаточно выбрать ячейку в его пределах. Аналогично, для задания столбца, по которому выполняется сортировка (аргумент *Key1*), достаточно выбрать ячейку в пределах этого столбца.

Порядок решения задачи.

1. На рабочем листе «*Торговые операции*» в ячейку *A1* ввести заголовок списка и заголовки полей списка. Сформировать исходный список, состоящий из 20 записей. Коды товаров - трехзначные числовые целые значения. Всего в списке 8 различных кодов товаров, которые повторяются в различных сочетаниях с разными датами продаж и различными объемами продаж в денежном выражении.
2. Сформировать заголовок нового списка в ячейке *E1* и заголовки полей нового списка, где будут размещены сводные данные. Заголовки полей списка отформатировать.
3. Перейти в режим редактора *Visual Basic*, выполнив команду *Сервис / Макрос / Редактор Visual Basic*.
4. Выполнить команды *Insert / Module* и *Insert / Procedure*.
5. Ввести имя процедуры *TradeOperations*
6. Ввести программный код процедуры *TradeOperations*.
7. Вернуться на рабочий лист «*Торговые операции*».
8. Запуск процедуры оформить кнопкой, размещенной на рабочем листе, там же, где расположены списки. Для создания кнопки щелкните на кнопке панели инструментов *Формы* и нарисуйте прямоугольную кнопку на листе «*Торговые операции*». Подтвердите создание макроса для данной кнопки. Затем измените название на этой кнопке на «*Подведение итогов по продажам*». Для запуска программы нужно просто щелкнуть на этой кнопке.

**Задача 2.5.** Отделу логистики корпорации «Славия» поручено разработать маршрут перемещения сотрудника отдела продаж корпорации, занимающегося сбытом мебели, для поиска новых рынков сбыта. Условия задачи следующие: агент по продажам должен выехать из города Воронеж, посетить все остальные города по одному разу, а затем возвратиться в город Воронеж. Определить маршрут перемещения сотрудника с минимальным общим расстоянием. Рассмотрим решение задачи на примере 10 городов. Для этого в ячейки диапазона с B2 по K2 ввести названия городов маршрута следования менеджера по продажам. В ячейки диапазона с A3 по A12 ввести те же названия городов и в том же порядке. Расстояния между городами будем формировать случайным образом, используя, оператор *Randomize* и функцию *Rnd*. Приложение будет состоять из двух процедур. Первая, *GetRandomTrack*, генерирует случайные расстояния между городами. Это сделано в целях экономии времени студента при выполнении лабораторной работы. Оператор *Randomize* позволяет при каждом запуске процедуры генерировать новые расстояния между городами. Другими словами каждая задача будет создаваться на основе собственного набора расстояний между городами. Программный код составлен в процедуре таким образом, что в этой матрице расстояния расположены симметрично относительно главной диагонали. Например, расстояние от города 5 до города 10 совпадает с расстоянием от города 10 до города 5. Диапазону ячеек *B3 : K12*, в которых генерируются случайным образом расстояния между городами, присвоить имя *MatrixTrack*.

Для запуска процедуры *GetRandomTrack* на рабочем листе рядом с диапазоном *MatrixTrack* расположить кнопку с панели инструментов *Формы* и дать ей название «*Формирование расстояний*».

```
Sub GetRandomTrack ()
    Dim i As Integer, j As Integer,
    Dim Quancity As Integer
    Randomize
    With Range ("MatrixTrack")
        Quancity = .Rows.Count
    ' Заполняет матрицу над главной диагональю _
    ' случайными числами
        For i = 1 To Quancity - 1
            For j = i + 1 To Quancity
                .Cells (i, j) = Int (Rnd * 100) + 1
            Next j
        Next i
    ' Создает зеркальное отражение уже введенных _
    ' значений над диагональю под диагональю
```

```

    For i = 2 TO Quancity
        For j = 1 To i - 1
            .Cells (i, j) = .Cells (j, I)
        Next j
    Next i
End With
End Sub

```

Для решения задачи составления маршрута, состоящего из посещений ближайших городов, разрабатывается процедура, содержащая массивы. Как только приложению станет известно расстояние между городами (диапазон ячеек будет заполнен расстояниями между городами), следует щелкнуть на кнопке, расположенной на рабочем листе, которая запустит процедуру поиска маршрута. Процедура разрабатывает маршрут к ближайшему соседу. После ее запуска на экране появится окно сообщения, в котором будет представлено общее расстояние, а на рабочем листе можно будет увидеть пройденный маршрут: названия и номера городов будут расположены в строках, начиная с ячеек *B15* и *B16* соответственно.

Пояснения к программе. В процедуре используется три массива, *Flag* типа *Integer*, целочисленный массив *Route* и массив текстовых данных *Cities*. Например, если элемент массива *Flag(6)* равен 1, значит, город 6 уже был посещен, и посещать его повторно не следует. В противном случае элемент *Flag(6)* приравнивается к 0, значит, город 6 является кандидатом на посещение. Массив *Route* определяет города, включенные в маршрут. Например, если элемент *Route(8)* равен 3, то город 3 является восьмым пунктом в маршруте.

Массивы *Flag*, *Route* и *Cities* в программном коде объявляются как динамические, чтобы сделать процедуру универсальной. Как только станет известно количество городов (определяется по счетчику строк в массиве *MatrixTrack*), приложение изменит размеры соответствующих массивов.

Перед выполнением процедуры необходимо инициализировать несколько переменных. В их число входят *Route(1)*, *Route(Quancity + 1)*, *NowAt*, *SumKil* и массивы *Flag*, *Cities*.

Алгоритм поиска маршрута наименьшей длины реализован с помощью двух вложенных циклов *For*. Внешний цикл задается для этапов маршрута. Назначением этого цикла является поиск второго, третьего и т. д. посещаемых городов. Внутренний цикл используется для поиска ближайшего соседнего города, предназначенного для посещения на данном этапе маршрута. Для этого производится поиск минимального расстояния от текущего города (который представлен переменной *NowAt*) до остальных городов, которые еще не посещались. Еще не посещавшийся город, расстояние до которого является минимальным, заносится в переменную *NextAt*. После завершения внутреннего цикла значение переменной *NextAt* приравни-

вается к переменной *NowAt*, используемой в следующей итерации внутреннего цикла.

```
Option Explicit
Option Base 1
Dim Quancity As Integer , Flag() As Integer, _
    Route() As Integer, SumKil As Integer
Dim Cities() As String

Sub MainProcedure()
    Call Initialize
    Call BuildRoute
    Call DrawRoute
End Sub

Sub Initialize()
    Dim i As Integer
    Quancity = Range("MatrixTrack").Rows.Count
    ReDim Flag(Quancity)
    ReDim Route(Quancity + 1)
    ReDim Cities(Quancity)
    Route(1) = 1
    Route(Quancity + 1) = 1
    Flag (1) = 1
    For i = 2 To Quancity
        Flag (i) = 0
    Next i
    SumKil = 0
    With Range ("A2")
        For i =1 To Quancity
            Cities(i) = .offset(0,i)
        Next i
    End With
End Sub

Sub BuildRoute ()
    Dim j As Integer, i As Integer, NowAt _
        As Integer, NextAt As Integer, _
        MinKil As Integer
    NowAt = 1
    For j = 2 To Quancity
        MinKil = 100000
        For i = 2 To Quancity
```

```

        If i <> NowAt And Flag (i) = 0 Then
            If Range("MatrixTrack").Cells(NowAt,i) _
                < MinKil Then
                NextAt = i
                MinKil = Range("MatrixTrack") _
                    .Cells(NowAt,NextAt)
            End If
        End If
    Next i
    Route(j) = NextAt
    Flag (NextAt) = 1
    SumKil = SumKil + MinKil
    NowAt = NextAt
Next j
SumKil = SumKil + Range("MatrixTrack") _
    .Cells(NowAt, 1)
End Sub

Sub DrawRout()
    Dim i As Integer
    Range("C14").Value = "МАРШРУТ СЛЕДОВАНИЯ"
    For i = 1 To Quancity + 1
        Range("A15").Offset(0, i) = Cities(Route(i))
        Range("A16").Offset(0, i) = Route(i)
    Next i
    MsgBox "Общее расстояние: " & SumKil & "км", _
        vbInformation, "Общее расстояние"
End Sub

```

**Задача 2.6.** Создать два списка, состоящих из фамилий клиентов фитнес – центра «Аврора», на листе «Списки»: один список расположить в столбце, начиная с ячейки *A4*, второй список расположить в столбце *B*, начиная с ячейки *B4*. Первый список должен содержать десять фамилий, а второй список – двадцать фамилий. Ячейки *A3* и *B3* должны содержать заголовки «**Клиенты 2007 года**» и «**Клиенты 2008 года**» соответственно. Отсортировать эти списки в алфавитном порядке. При создании списков обратить внимание на то, что некоторые клиенты должны быть указаны только в списке для 2007 года или только в списке для 2008 года, а некоторые клиенты должны упоминаться в обоих списках. Список в столбце *D*, полученный после слияния данных, должен включать неповторяющиеся имена клиентов из исходных списков.

Алгоритм слияния списков заключается в следующем.

- Начать просмотр фамилий списков с начала каждого списка и сравнивать имена клиентов следующим образом:
- Если фамилии клиентов в обоих списках совпадают, переместить эту фамилию в столбец D и сдвинуться на строку вниз в столбцах A и B для сравнения следующей пары фамилий.
- Если фамилия в столбце A расположена раньше фамилии в столбце B (в соответствии с алфавитным порядком), то следует переместить первое имя из столбца A в столбец D и перейти в следующую строку в столбце A (но не в столбце B) для сравнения следующей пары имен. Следует выполнить те же действия, если фамилия в столбце B расположена перед фамилией в столбце A (в соответствии с алфавитным порядком).
- Продолжать анализ, сравнивая фамилии в столбцах A и B, пока не будут скопированы все фамилии из столбца A или из столбца B. После этого оставшиеся во втором столбце или первом фамилии необходимо переместить в столбец D.

Пояснения к составлению программного кода процедуры.

- В начале выполнения программного кода процедуры предыдущий результат объединения списков удаляется из столбца D; для этого используется метод *ClearContents*. Затем вычисляется размер существующих списков, и массивы *List1* и *List2* заполняются содержимым исходных списков. Для заполнения массива *List3* используется процедура слияния, а результаты заносятся в столбец D.
- Процедура использует переменные *Index1* и *Index2* для указания позиции в соответствующем списке. Фамилии клиентов сохраняются в переменных *Name1* и *Name2*. Сравнение фамилий позволяет выяснить, какую из них заносить в результирующий список, а также какой счетчик (*Index1* или *Index2*)увеличивать на 1. Цикл *Do While* в нашем случае более удобен, поскольку он продолжает обработку списков до тех пор, пока хоть в одном из списков присутствуют фамилии.
- После завершения цикла *Do* оставшиеся фамилии (если такие есть) из одного списка заносятся в результирующий список. Затем содержимое массива *List3* заносится в столбец D.

Sub MergeLists()

- ‘ Переменная LSx указывает размер списков
- ‘ (x изменяется от 1 до 3)
- ‘ Массивы Listx содержат элементы списков
- ‘ (x изменяется от 1 до 3)
- ‘ Списки индексируются от 1 до 3 следующим
- ‘ образом:
- ‘ List1– клиенты за 2007 год (существующие данные)
- ‘ List2– клиенты за 2008 год (существующие данные)

```

` List3 - клиенты, купавшие товар в обоих _ городах
(создается)
    Dim I As Integer, j As Integer
    Dim LS1 As Integer, LS2 As Integer, List1() _
        As String, List2() As String
    Dim LS3 As Integer, Index1 As Integer, Index2 _
        As Integer, Name1 As String, Name2 As _
        String, List3() As String
`Удаление старого объединенного списка в столбце D
    With Range("D3")
        Range(.Offset(1, 0), _
            .Offset(1, 0).End(xlDown)).ClearContents
    End With
` Получение размеров списков и их элементов в _
` столбцах A и B
    With Range("A3")
        LS1 = Range(.Offset(1, 0), .End(xlDown)) _
            .Rows .Count
        ReDim List1 (LS1)
        For i = 1 To LS1
            List1(i) = .Offset(i, 0)
        Next i
        LS2 = Range(.Offset(1, 1), .Offset(0, 1) _
            .End(xlDown)) .Rows .Count
        ReDim List2 (LS2)
        For i = 1 To LS2
            List2(i) = .Offset(i, 1)
        Next i
    End With
` Создание объединенного списка
` Размер объединенного списка = 0: LS3 = 0
` Одновременный просмотр списков List1 и List2
` Переменные Index1 и Index2 указывают текущее
` положение в списках. Переменные Name1 и
` Name2 содержат имена текущих клиентов
    Index1 = 1
    Index2 = 1
` Просмотр до конца одного из списков
    Do While Index1 <= LS1 And Index2 <= LS2
        Name1 = List1(Index1)
        Name2 = List2(Index2)
` Каждый шаг добавляет в объединенный список
` определенный элемент, поэтому приходится
` обновлять размер списка и массива List3

```

```

    LS3 = LS3 + 1
    ReDim Preserve List3(LS3)
` Анализ расположения сравниваемых имен в обоих
` списках.
` Расположенный первым в алфавитном порядке элемент
` перемещается в объединенный список.
` Значение переменной Index соответствующего списка
` обновляется. В случае совпадения обновляются
` индексы обоих списков
    If Name1 < Name2 Then
        List3(LS3) = Name1
        Index1 = Index1 + 1
    ElseIf Name1 > Name2 Then
        List3(LS3) = Name2
        Index2 = Index2 + 1
    ElseIf Name1 = Name2 Then
        List3(LS3) = Name2
        Index1 = Index1 + 1
        Index2 = Index2 + 1
    End If
    Loop
` Один из списков полностью просмотрен (List1 или
` List2).
` Оставшиеся элементы другого списка добавляются
` в конец объединенного списка.
    If Index1 > LS1 And Index2 <= LS2 Then
        For i = Index2 To LS2
            LS3 = LS3 + 1
            ReDim Preserve List3(LS3)
            List3(LS3) = List2(i)
        Next i
    ElseIf Index1 <= LS1 And Index2 > LS2 Then
        For i = Index1 To LS1
            LS3 = LS3 + 1
            ReDim Preserve List3(LS3)
            List3(LS3) = List1(i)
        Next i
    End If
` Занесение объединенного списка в столбец D
` рабочего листа
    With Range("D3")
        For i = 1 To LS3
            .Offset(i, 0) = List3(i)
        Next i
    End With

```



```

End With
' Переход к ячейке A2
Range("A2").Select
End Sub

```

**Задача 2.7.** Создать приложение, состоящее из процедур для оценки действующей финансово-экономической стратегии фирмы.

Методика оценки действующей финансово-экономической стратегии с использованием компьютерных технологий включает два этапа. На первом, с помощью экспертного или логического метода определяется состав следующих показателей: чистая прибыль, прибыль от продаж, выручка от реализации, производственные запасы, дебиторская задолженность, полная себестоимость реализованной продукции, фонд заработной платы.

На втором этапе методики оценки формируется нормативная структура (нормативный ряд) темпов роста (Т) показателей развития организации:

$T_{чп} > T_{пп} > T_{вр} > T_{пз} > T_{дз} > T_{сс} > T_{зп}$ ,

Реализация методики рассмотрена на примере деятельности одного из акционерных обществ г. Воронежа. Значения финансово-экономических показателей деятельности организации представлены в табл. 2.1 на рис. 2.1.

На основании сведений табл. 2.1 рассчитываются темпы изменения показателей, представленные в табл. 2.2 на рис. 2.1. Результаты заносятся в блок ячеек В30:Н34.

На основании темпов изменения финансово-экономических показателей формируются их фактические динамические ряды в табл. 2.3 на рис. 4. Приоритеты темпов роста показателей определяются следующим образом:

1 – наивысший приоритет, которым обладает показатель, имеющий в строке таблицы 2 максимальное значение;

7 – наименьший приоритет, которым обладает показатель, имеющий в строке таблицы 2 минимальное значение. Приоритеты располагаются в нормативном ряде показателей следующим образом:

1    2    3    4    5    6    7  
 $T_{чп} > T_{пп} > T_{вр} > T_{пз} > T_{дз} > T_{сс} > T_{зп}$

Результатом формирования табл. 2.3 являются фактические динамические ряды показателей, значения которых заносятся в блок ячеек В51-Н55 табл. 2.3, расположенной на рис. 2.2.

Степень соответствия нормативного ряда показателей (НРП) и фактического ряда показателей (ФРП) можно оценить с помощью коэффициента корреляции Спирмэна:

$$K = 1 - (6 \cdot \sum d^2) / (n^3 - n), \quad (1)$$

где  $\sum d^2$  – сумма квадратов разностей рангов показателей в ряду;

n – количество показателей в ряду, в нашем случае n = 7.

Этот коэффициент ранговой корреляции вычисляется по каждому фактическому динамическому ряду, в сравнении с нормативным динамическим рядом, приведенным в строке 50 табл. 2.3 на рис. 2.2.

Результаты расчета коэффициента Спирмэна по годам заносятся в блок ячеек I51 ÷ I55 табл. 2.3. По значениям коэффициента корреляции Спирмэна строится график, отражающий динамику изменения данного коэффициента, который приведен на рис. 2.2.

Порядок создания приложения следующий:

Создается форма с двумя вкладками, на каждой из которых расположены по две кнопки с названиями: «Очистить» и «Заполнить». Эта форма располагается на рабочем листе Excel и используется для запуска процедур с помощью этих кнопок.

Ниже приведен листинг программного кода приложения, автоматизирующего процедуру оценки действующей финансово-экономической стратегии и построения графика.

```
Private Sub CommandButton1_Click()  
    ' Заполнение таблицы 2.2  
    Dim i, j As Integer  
    For j = 2 To 8  
        For i = 8 To 12  
            Cells(i+22,j) = Round(Cells(i,j) / _  
                Cells(i-1,j) * 100, 2)  
        Next i  
    Next j  
End Sub  
  
Private Sub CommandButton2_Click()  
    ' Заполнение таблицы 2.3  
    Dim i, j, z As Integer  
    Dim Work, k1 As Variant  
    Dim Arr(1 To 8, 1 To 2)      ' Массив для сортировки  
даных строк  
    k1 = 7 ^ 3 - 7      ' 7 - количество рангов в ряду  
    For i = 30 To 34  
        ' Помещение i-й строки в массив Arr  
        For j = 1 To 7  
            Arr(j, 1) = Cells(i, j + 1)  
            Arr(j, 2) = j  
        Next j  
        ' Сортировка массива Arr  
        For j = 1 To 6  
            For z = j + 1 To 7  
                If Arr(j, 1) < Arr(z, 1) Then
```

```

        Arr(8, 1) = Arr(j, 1)
        Arr(8, 2) = Arr(j, 2)
        Arr(j, 1) = Arr(z, 1)
        Arr(j, 2) = Arr(z, 2)
        Arr(z, 1) = Arr(8, 1)
        Arr(z, 2) = Arr(8, 2)
    End If
Next z
Next j
' Помещение фактических динамических показателей в
очередную строку таблицы 2.3
For j = 2 To 8
    Cells(i + 21, Arr(j - 1, 2) + 1) = j - 1
Next j
' Вычисление коэффициента корреляции для текущей строки
Work = 0
For j = 2 To 8
    Work = Work + (Cells(50, j) - _
Cells(i + 21, j)) ^ 2
Next j
Cells(i + 21, 9) = 1 - 6 * Work / k1
Next i
End Sub

Private Sub CommandButton4_Click()
' Очистка таблицы 2.2
Range("B30:H34").Clear
End Sub

Private Sub CommandButton5_Click()
' Очистка таблицы 2.3
Range("B51:I55").Clear
End Sub

Private Sub UserForm_QueryClose(Cancel As _
Integer, CloseMode As Integer)
If CloseMode = 0 Then
    Cancel = True
    MsgBox "Так закрыть окно нельзя!"
End If
End Sub

```

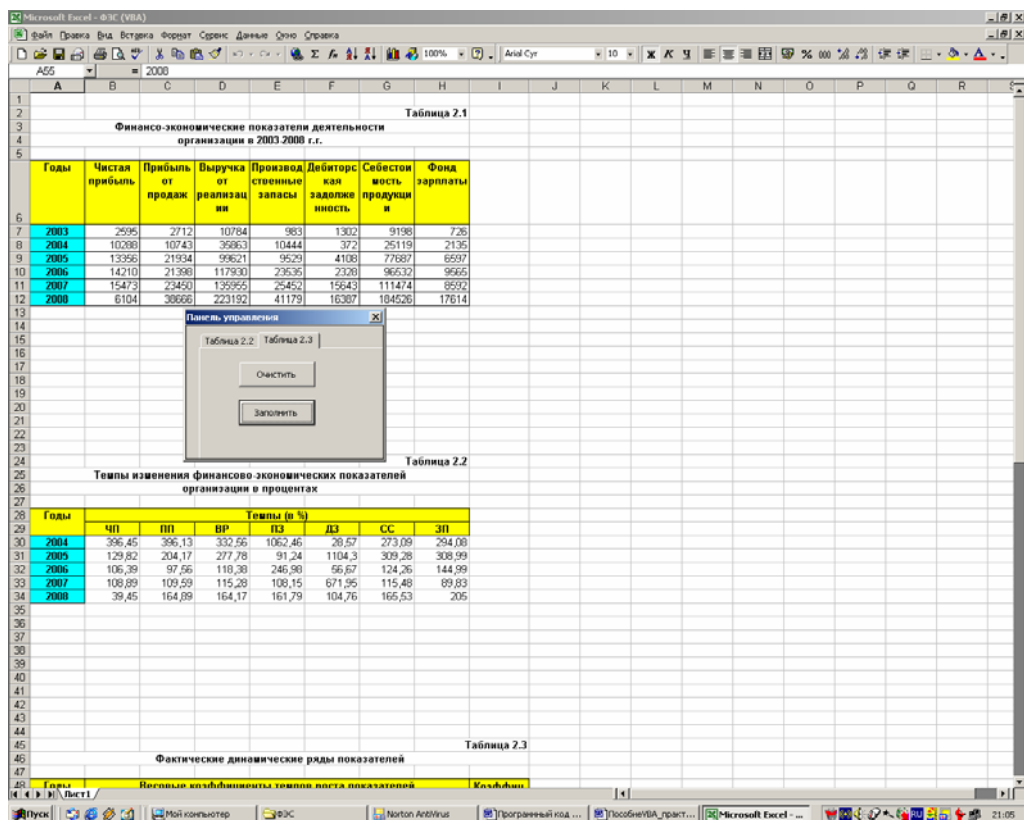


Рис. 2.1. Окно Excel с исходными и промежуточными вычислениями

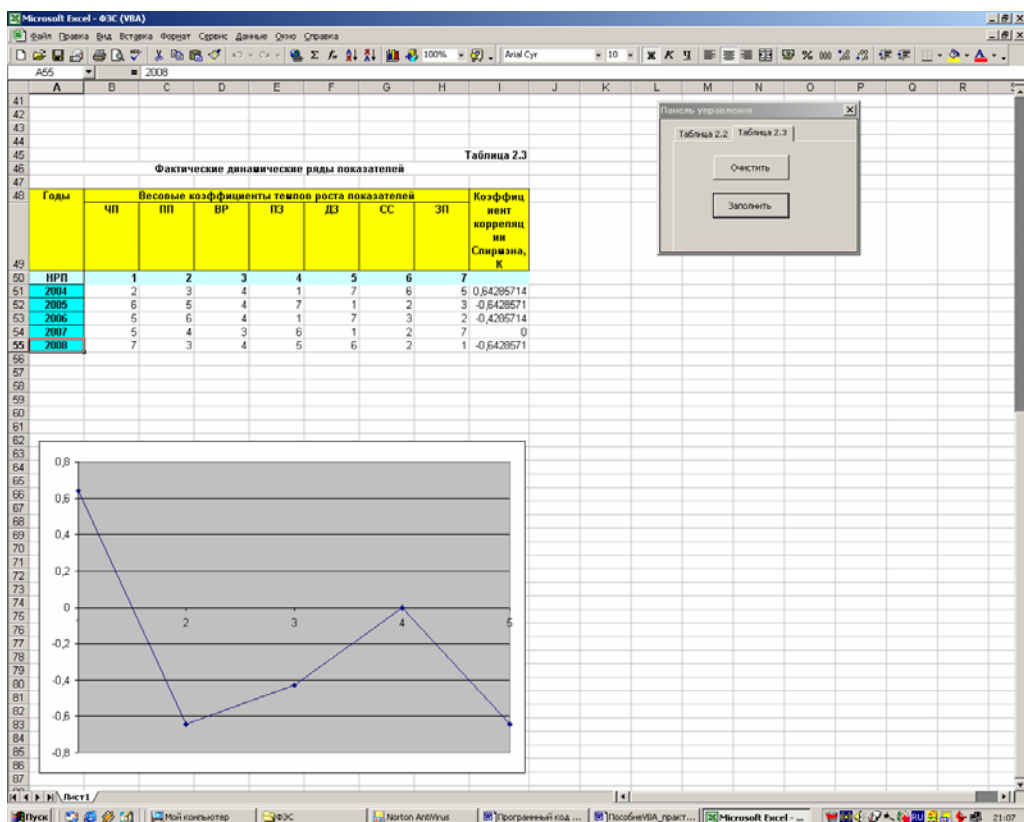


Рис. 2.2. Окно Excel с результатами и графиком

**Задача 2.8.** Создать процедуру для преобразования рабочего листа в форму-заставку для презентации фирмы.

Рассмотрим применение объекта *Shape* из семейства *Shapes* в процессе разработки формы-заставки для презентации возможностей РУС-ФИНАНС БАНКА. Объект *Shape* представляет любой графический элемент («фигуру»), размещенный «поверх» рабочего листа или диаграммы. Все объекты на данном листе входят в семейство *Shapes*, которое объединяет точечные и другие рисунки, прямоугольники, линии, текстовые поля, объекты WordArt и даже элементы управления. Некоторые методы и свойства присущи каждому объекту *Shape*, другие – только определенным типам фигур. Рассмотрим некоторые свойства и методы, которые будут использованы в приведенной ниже программе при размещении графического объекта типа *msoAutoShape* на форме.

#### **Свойства объекта**

**Type** – константа, определяющая тип объекта *Shape* (например, *msoLine*, *msoAutoShape*, *msoPicture*). Список констант представлен в справочной системе и в окне *Просмотр объектов (Object Browser)*.

**Fill** – возвращает объект *FillFormat*, свойства которого определяют параметры заливки объекта *Shape*.

**Shadow** – возвращает объект *ShadowFormat*, свойства которого определяют параметры тени объекта *Shape*.

**TextFrame** – возвращает объект *TextFrame*, подобъекты которого управляют текстом, отображаемым в объекте *Shape*, и его форматированием.

#### **Методы семейства Shapes**

**AddShape** – создает объект *AutoShape* (автофигуру).

Ниже предложен программный код, в котором объект *Worksheet* - рабочий лист *Excel* служит основой формы-заставки для презентации банка, которая занимает весь рабочий лист. Программный код выглядит следующим образом.

```
Public Sub Заставка( )
    Worksheets(1).Select
    With ActiveWindow
        'Определяет размер окна
        .WindowState = xlMaximized
        ' Не отображаются линии сетки
        .DisplayGridlines = False
        ' Не отображаются заголовки строк и столбцов
        .DisplayHeadings = False
        ' Нет названия объекта в строке заголовка
        .Caption = ""
        ' Не отображается полоса горизонтальной прокрутки
        .DisplayHorizontalScrollBar = False
    End With
End Sub
```

```

` Не отображается полоса вертикальной прокрутки
    .DisplayVerticalScrollBar = False
` Не отображаются ярлычки рабочих листов
    .DisplayWorkbookTabs = False
End With
`Задается фон диапазона: один из 56 цветов цветовой
  па\литры Excel, если 0 – стандартный цвет
  Range ("A1:Z5").Interior.ColorIndex = 16
` Изменяет свойства границы диапазона
  With Range ("A6:Z6").Borders (xlEdgeTop)
` Настройка стиля границы, принимает одно из 8 значе-
  ний
    .LineStyle = xlContinuous
    .Weight = xlThick ` Настройка толщины границы
    .ColorIndex = 1 ` Выбор цвета границы
  End With
` Задаёт фон диапазона
  Range ("A6:Z100"). Interior.ColorIndex = 5
` Осуществляет выбор диапазона
  Range ("A2:J2").Select
  With Selection ` С выбранным диапазоном
` Слияние выбранного диапазона в одну ячейку "A2"
    .Merge
` Горизонтальное выравнивание
    .HorizontalAlignment = xlCenter
` Ввод текста в ячейку A2
    .Value = "Добро пожаловать в РУСФИНАНС БАНК"
` Выбирает шрифт как объект работы с ним
    With.Font
      .Name = "Times New Roman" ` Задаёт имя
        шрифту
      .Size = 36 ` устанавливает размер шрифта
` Задаёт полужирное начертание
      .Bold = True
` Устанавливает цвет для шрифта
      .ColorIndex = 6
    End With
  End With
  With Worksheets (1)
` На рабочем листе "Лист 1" создаёт графический объ-
  ект и выбирает его
    .Shapes.AddShape(msoShapeUpRibbon, 170, 125,
      _ 300, 50).Select
    With Selection ` С выбранным объектом

```

```

` Устанавливает цвет
    .ShapeRange.Fill.ForeColor.SchemeColor = 54
` Вводит текст, располагая его на графическом объекте
    .Characters.Text = "Мы рады Вам помочь _
    с кредитом"
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    With.Font
        .Name = "Arial"
        .FontStyle = "Regular"
        .Size = 12
        .Bold = True
        .ColorIndex = 2
    End With
    End With
    End With
    Range ("A1").Select
End Sub

```

Самостоятельно модифицировать программный код процедуры, выполнив следующие действия:

1. Открыть диалоговое окно просмотра объектов *Object Browser* командой *View / Object Browser*. В открывшемся окне в списке библиотек выбрать элемент *Office*, а в следующем поле со списком выбрать или ввести элемент *msoShape* и нажать на кнопку поиска. После выбора в нижней части окна будет приведен список всех объектов из класса *msoShape*. В списке *Members of 'MsoAutoShapeType'* выбрать подходящий объект и расположить его на форме вместо объекта *msoShapeUpRibbon*.
2. Настроить размеры графического объекта таким образом, чтобы на нем полностью был размещен текст оказываемых услуг банком.
3. Расположить текст приглашения к сотрудничеству с банком по центру в верхней части заставки.
4. Изменить название банка в тексте приглашения в ячейке «A2».
5. Изменить текст предлагаемых услуг банком на графическом объекте формы.
6. Изменить цвет верхней и нижней частей заставки, выбрав его из 56 цветов цветовой палитры *Excel*.

## **Задания для самостоятельной работы по созданию процедур с использованием управляющих структур**

**Задание 1.** Написать процедуру Sub с именем Fuctorial, вычисляющую факториал целого положительного числа, по следующему алгоритму:

1. Выполнить описание переменных, используемых в программе, таких как strNumber, intNum, intFuctorial, intCount с помощью оператора Dim.
2. С помощью функции InputBox() ввести число для вычисления факториала, присвоив его значение переменной strNumber.
3. С помощью функции IsNumeric() проверить является ли введенное значение числом, если да, то, используя функцию Val(), преобразовать введенное текстовое значение в число и присвоить его значение переменной intNum.
4. Если значение  $\text{intNum} > 0$ , то вычислить факториал с помощью управляющей структуры For-Next, присвоив полученное значение переменной intFuctorial с выводом результата на экран функцией MsgBox.
5. При составлении программы выполнить проверки: если введено нечисловое значение или отрицательное число, то с помощью функции MsgBox() выдать соответствующие сообщения.

**Задание 2.** Написать процедуру Sub с именем Proc2, вычисляющую возведение значения целой переменной intBase в целочисленную положительную степень, значение которой содержит переменная intPower по следующему алгоритму:

1. Выполнить описание переменных, используемых в программе, таких как strBase, strPower, intBase, intPower, intResult, intCount с помощью оператора Dim.
2. С помощью функций InputBox() ввести числа для вычисления степени (числа для основания и показателя степени), присвоив введенные значения переменным strBase, strPower соответственно.
3. С помощью функции IsNumeric() проверить являются ли введенные значения числовыми, если да, то, используя функцию Val(), преобразовать введенные текстовые значения в числовые, присвоив их значения переменным intBase и intPower соответственно.
4. С помощью управляющей структуры For-Next возвести в степень intPower число intBase. Результат вычисления выдать в окне сообщения с соответствующими пояснениями.
5. При составлении программы выполнить проверки: если введено нечисловое значение или отрицательное число, то с помощью функции MsgBox() выдать соответствующие сообщения.



**Задание 3.** Написать процедуру Sub с именем Proc3\_WileWend для выделения определенного значения равного 7, указывающего на выигрыш из последовательности случайных чисел от 1 до 9 по следующему алгоритму:

1. Выполнить описание переменных, используемых в программе, таких как intLotteryEntry, intSum с помощью оператора Dim.
2. Выполнить начальную инициализацию этих переменных, присвоив каждой значение равное нулю.
3. С помощью управляющей структуры Wile-Wend генерировать случайные числа от 1 до 10. Цикл Wile-Wend должен выполняться до тех пор, пока значение переменной intLotteryEntry не станет равным 7. При каждом выполнении цикла этой переменной присваивается значение от 1 до 9 и происходит добавление этого значения к сумме intSum сгенерированных ранее чисел. При этом подается звуковой сигнал оператором Beep.
4. Как только значение intLotteryEntry станет равным 7 выдать на экран сообщение о выигрыше и значение накопленной суммы (intSum).

**Задание 4.** Написать процедуру Sub с именем Proc4\_DoWileLoop для выделения определенного значения равного 7, указывающего на выигрыш из последовательности случайных чисел от 1 до 9 по следующему алгоритму:

1. Выполнить описание переменных, используемых в программе, таких как intLotteryEntry, intSum с помощью оператора Dim.
2. Выполнить начальную инициализацию этих переменных, присвоив каждой значение равное нулю.
3. С помощью управляющей структуры DoWileLoop генерировать случайные числа от 1 до 10. Цикл DoWileLoop должен выполняться до тех пор, пока значение переменной intLotteryEntry не станет равным 7. При каждом выполнении цикла этой переменной присваивается значение от 1 до 9 и происходит добавление этого значения к сумме intSum сгенерированных ранее чисел. При этом подается звуковой сигнал оператором Beep.
4. Как только значение intLotteryEntry станет равным 7 выдать на экран сообщение о выигрыше и значение накопленной суммы (intSum).

**Задание 5.** Написать процедуру Sub с именем Proc5\_DoUntilLoop для выделения определенного значения равного 7, указывающего на выигрыш из последовательности случайных чисел от 1 до 9 по следующему алгоритму:

1. Выполнить описание переменных, используемых в программе, таких как intLotteryEntry, intSum с помощью оператора Dim.
2. Выполнить начальную инициализацию этих переменных, присвоив каждой значение равное нулю.

3. С помощью управляющей структуры DoUntilLoop генерировать случайные числа от 1 до 10. Цикл DoUntilLoop должен выполняться до тех пор, пока значение переменной intLotteryEntry не станет равным 7. При каждом выполнении цикла этой переменной присваивается значение от 1 до 9 и происходит добавление этого значения к сумме intSum сгенерированных ранее чисел. При этом подается звуковой сигнал оператором Beep.
4. Как только значение intLotteryEntry станет равным 7 выдать на экран сообщение о выигрыше и значение накопленной суммы (intSum).

**Задание 6.** Написать процедуру Sub с именем Proc6\_ForEachNext предназначенную для выполнения одной и той же группы действий над каждым элементом массива по следующему алгоритму:

1. Выполнить описание одномерного массива AArray(10), состоящего из десяти целых чисел и описание переменных, используемых в программе, таких как vntA, intSum, intProiz, dblSr с помощью оператора Dim.
2. Осуществить ввод элементов массива AArray() внутри цикла For-next с помощью окна ввода InputBox.
3. Выполнить начальную инициализацию переменной intSum для вычисления суммы элементов массива AArray(), переменной intProiz для вычисления произведения элементов этого массива и переменной dblSr для вычисления среднего арифметического элементов этого массива, присвоив значения 0, 1, 0 соответственно.
4. С использованием управляющей структуры For Each vntA In AArray ... Next вычислить сумму, произведение и среднее арифметическое элементов массива AArray().
5. Результаты вычислений выдать на экран с помощью функции MsgBox.

**Задание 7.** Написать процедуру Sub обработки события Click() для кнопки с именем КнопкаВычислить, выполняющую следующие действия:

1. Выполнить описание переменных, используемых в программе, таких как dblA, dblB, dblC с помощью оператора Dim.
2. С помощью управляющей структуры If-Then выполнить проверку содержимого текстовых полей Text1 и Text2. Если хотя бы одно поле окажется пустым выдать соответствующее сообщение с помощью функции MsgBox и осуществить досрочный выход из процедуры с помощью оператора Exit Sub.
3. Если в эти поля введены числа, то преобразовать их из текстового типа в числовой, применив функцию CDb1() к текстовым полям Text1 и Text2 и присвоив полученные значения переменным dblA, dblB.

4. Вычислить среднее арифметическое значений этих переменных и присвоить полученное значение переменной `dblC`.
5. Значение переменной `dblC` вывести на форму в текстовое поле `Text3`, присвоив свойству текст текстового поля `Text3` значение этой переменной в текстовом формате с помощью функции `CStr`.

**Задание 8.** Написать процедуру с именем `Proc8_IfThenElse`, выполняющую действия по следующему алгоритму:

1. Создать процедуру функцию `GetRandomNumber()`, с помощью которой осуществить генерацию случайных чисел в диапазоне от 1 до 9.
2. Выполнить описание переменной `intNum` с помощью оператора `Dim`.
3. Присвоить этой переменной случайное значение в диапазоне от 1 до 9 с помощью функции `GetRandomNumber`.
4. Если значение этой переменной будет равно 5, выдать сообщение с помощью функции `MsgBox` о выигрыше, в противном случае – сообщение о проигрыше.

**Задание 9.** Написать процедуру с именем `Proc9_IfThenElseIf`, выполняющую действия по следующему алгоритму:

1. Оформить в виде процедуры функции с именем `GetPassword` ввод пароля, используя для этого окно ввода `InputBox` и функцию `Lcase()`, которая преобразует все буквы переданной в нее строки в строчные.
2. После написания заголовка процедуры `Proc32_IfThenElseIf` выполнить описание переменных `strPassword` как текстовой и переменной `Elem` как объектной с помощью оператора `Dim`.
3. Выполнить вызов функции `GetPassword` и присвоить переданное функцией значение переменной `strPassword`.
4. С помощью управляющей структуры `If-Then-ElseIf` проверить введенное через окно ввода значение пароля. Если введенное значение пароля равно "P555", то каждому элементу управления формы с именем `Form1` с помощью управляющей структуры (`For Each Elem In Form1....Next Elem`) установить следующие атрибуты шрифта: имя шрифта – "Ms Sans Serif", размер шрифта – 14, начертание шрифта – курсив. Если значение пароля равно "P444", то каждому элементу управления формы установить следующие атрибуты шрифта: имя шрифта – "MT Extra", размер шрифта – 10, начертание шрифта – полужирное. Если значение пароля равно "P333", то каждому элементу управления формы установить следующие атрибуты шрифта: имя шрифта – "Times New Roman", размер шрифта – 18, начертание шрифта – не курсив.
5. Если пароль введен неправильный, то выдать сообщение о повторном вводе пароля с помощью функции `MsgBox`.

**Задание 10.** Написать процедуру с именем Proc10\_IfThenElseIf, выполняющую действия по следующему алгоритму:

1. Оформить в виде процедуры функции с именем GetPassword ввод пароля, используя для этого окно ввода InputBox и функцию Lcase(), которая преобразует все буквы переданной в нее строки в строчные.
2. После написания заголовка процедуры Proc10\_IfThenElseIf выполнить описание переменных strPassword как текстовой и переменной Sheet как объектной с помощью оператора Dim.
3. Выполнить вызов функции GetPassword и присвоить переданное функцией значение переменной strPassword.
4. С помощью управляющей структуры If-Then-ElseIf проверить введенное через окно ввода значение пароля. Если введенное значение пароля равно "P555", то перебрать все листы активной рабочей книги и с каждым из них с помощью управляющей структуры (For Each Sheet In ActiveWorkbook.WorkSheets....Next Sheet) выполнить два действия: снять защиту с листа паролем, сделать лист видимым в случае, если он скрыт. Если значение пароля равно "P444", то первый лист активной рабочей книги сделать видимым и снять защиту с него. Если значение пароля равно "P333", то первый лист рабочей книги сделать видимым.
5. Если пароль введен неправильный, то выдать сообщение о повторном вводе пароля с помощью функции MsgBox.

**Задание 11.** Написать процедуру с именем Proc11\_SelectCase, выполняющую действия по следующему алгоритму:

1. Оформить в виде процедуры функции с именем GetPassword ввод пароля, используя для этого окно ввода InputBox и функцию Lcase(), которая преобразует все буквы переданной в нее строки в строчные.
2. После написания заголовка процедуры Proc33\_SelectCase выполнить описание переменных strPassword как текстовой и переменной Elem как объектной с помощью оператора Dim.
3. Выполнить вызов функции GetPassword и присвоить переданное функцией значение переменной strPassword.
4. С помощью управляющей структуры Select Case strPassword проверить введенное через окно ввода значение пароля (Case <значение пароля>). Если введенное значение пароля равно "P555", то каждому элементу управления формы с именем Form1 с помощью управляющей структуры (For Each Elem In Form1....Next Elem) установить следующие атрибуты шрифта: имя шрифта – "Ms Sans Serif", размер шрифта – 14, начертание шрифта – курсив. Если значение пароля равно "P444", то каждому элементу управления формы установить следующие атрибуты шрифта: имя шрифта – "MT Extra", размер шрифта – 10, начертание шрифта – полужирное. Если значение па-

роля равно “Р333”, то каждому элементу управления формы установить следующие атрибуты шрифта: имя шрифта – “Times New Roman”, размер шрифта – 18, начертание шрифта – не курсив.

5. Если пароль введен неправильный, то выдать сообщение о повторном вводе пароля с помощью функции MsgBox.

**Задание 12.** Написать процедуру Sub с именем Proc12\_ForEachNext предназначенную для выполнения одной и той же группы действий над каждым объектом семейства объектов по следующему алгоритму:

1. Выполнить описание переменной, используемой в программе, VrnSheet как объектной типа рабочего листа с помощью оператора Dim.
2. С использованием управляющей структуры For Each VrnSheet In ActiveWorkbook.WorkSheets ... Next vrnSheet выдать последовательно имена рабочих листов активной рабочей книги на экран в виде сообщений с помощью функции MsgBox.

**Задание 13.** Написать процедуру Sub с именем Proc13\_ForEachNext предназначенную для выполнения одной и той же группы действий над каждым объектом семейства объектов по следующему алгоритму:

1. Выполнить описание переменной, используемой в программе, Book как объектной типа рабочей книги и переменной x как целой с помощью оператора Dim.
2. С помощью цикла For – Next создать десять новых книг с использованием метода Add.
3. С помощью метода Arrange применительно к семейству объектов Windows, упорядочить расположение окон десяти созданных книг на экране и приостановить выполнение программы выводом на экран окна сообщения.
4. С использованием управляющей структуры For Each Book In Application.Workbooks ... Next Book сравнить имя каждой рабочей книги с именем книги ThisWorkbook. Если имена не совпадают, то для текущей книги вызвать метод Close.
5. Развернуть окно активной рабочей книги на всю рабочую область окна Excel.

**Задание 14.** Написать процедуру Sub с именем Proc14\_ForEachNext предназначенную для выполнения одной и той же группы действий над каждым объектом семейства объектов по следующему алгоритму:

1. Выполнить описание одномерного массива CountryArray(5), состоящего из названий пяти стран и описание переменной, используемой в программе, vrnCountry типа Variant с помощью оператора Dim.

2. Осуществить ввод элементов массива CountryArray(5) внутри цикла For-next с помощью окна ввода InputBox.
3. С использованием управляющей структуры For Each vrnCountry In CountryArray ... Next vrnCountry выдать названия стран на экран в виде сообщений с помощью функции MsgBox.

**Задание 15.** Написать процедуру Sub с именем Proc15\_ForEachNext предназначенную для выполнения одной и той же группы действий над каждым объектом семейства объектов по следующему алгоритму:

1. Выполнить описание переменной SheetVar как объектной типа рабочего листа и переменной Cell как объектной типа диапазонс помощью оператора Dim.
2. С использованием управляющей структуры For Each SheetVar In ActiveWorkbook.WorkSheets ... Next SheetVar перебрать все листы активной рабочей книги и вызвать для каждого из них метод Select. Внутренней управляющей структурой For Each Cell In Range ("A1:F20") ... Next Cell изменить цвет фона ячеек диапазона "A1:F20" на светло-серый (номер цвета 15).

**Задание 16.** Написать процедуру Sub с именем Proc16\_With() предназначенную для изменения содержимого и внешнего вида ячейки с адресом A1 третьего рабочего листа активной рабочей книги по следующему алгоритму:

1. С помощью управляющей структуры With – End With применить методы выделения и снятия защиты к третьему рабочему листу активной рабочей книги.
2. С помощью внутренней управляющей структуры With – End With вывести содержимое ячейки A1 выделенного листа, затем изменить значение этой ячейки на 200, высоту ячейки - на 60, ширину столбца - на 20 размер шрифта - на 20 цвет заливки этой ячейки изменить на красный.

## Задания для самостоятельной работы по созданию процедур для обработки массивов

Во всех заданиях исходные данные (значения элементов последовательностей или значения элементов матриц) располагайте в ячейках рабочих листов Excel активной рабочей книги. Вывод массивов, являющихся результатом вычислений, а также значений переменных, получаемых в результате выполнения программ, осуществляйте в ячейки рабочих листов, располагая их рядом с массивами исходных данных, сопровождая соответствующими пояснениями. При необходимости в процессе отладки программных кодов используйте отладчик приложения VBA.

При составлении программ вам могут потребоваться дополнительные пояснения:

1. Относительно заполнения рабочих массивов элементами последовательностей или матриц расположенных в ячейках рабочих листов.
2. Относительно размещения элементов сформированных массивов, а также вычисленных значений переменных в процессе выполнения программ в соответствующих ячейках активного рабочего листа активной рабочей книги Excel.

Рассмотрим фрагмент программного кода, в котором предусмотрено изменение размерности двумерного массива *Matrix* и заполнение этого массива значениями элементов исходной матрицы, расположенной на рабочем листе Excel, начиная с ячейки *A1*

```
With ActiveWorkbook.Worksheets(1).Range("A1")
    NRows = Range(.Offset(0,0), .End(xlDown)) _
        .Rows.Count
    NColumn = Range(.Offset(0,0), .End(xlToRight)) _
        .Columns.Count
    ReDim Matrix(NRows, NColumn)
    For i = 0 To NRows - 1
        For j = 0 To NColumn - 1
            Matrix(i, j) = .Offset(i, j)
        Next j
    Next i
End With
```

Рассмотрим фрагмент программного кода, в котором предусмотрено занесение преобразованной матрицы *Matrix1* в диапазон ячеек рабочего листа Excel, начиная с ячейки *H1*. Количество строк определяется переменной *NRows*, количество столбцов матрицы определяется переменной *NColumn*.

```

With Range("H1")
    For i = 0 To NRows - 1
        For j = 0 To NColumn - 1
            .Offset(i, j) = Matrix1(i, j)
        Next j
    Next i
End With

```

Запишем полный программный код процедуры *Matrica*, которая объединяет описанные выше два фрагмента. Кроме того, в этой процедуре представлено полное описание всех переменных и массивов, используемых в программе и фрагмент программного кода, осуществляющий преобразование исходного массива *Matrix* в массив *Matrix1*.

```

Sub Matrica()
    Dim i As Integer, j As Integer
    Dim Matrix() As Integer, Matrix1() As Integer
    Dim NRows As Integer, NColumn As Integer
    With ActiveWorkbook.Worksheets(1).Range("A1")
        NRows = Range(.Offset(0,0), .End(xlDown)) _
            .Rows.Count
        NColumn = Range(.Offset(0,0), .End(xlToRight)) _
            .Columns.Count
        ReDim Matrix(NRows, NColumn)
        For i = 0 To NRows - 1
            For j = 0 To NColumn - 1
                Matrix(i, j) = .Offset(i, j)
            Next j
        Next i
    End With
    For i = 0 To NRows - 1
        For j = 0 To NColumn - 1
            Matrix1(i, j) = Matrix(i, j) * 2
        Next j
    Next i
    With Range("H1")
        For i = 0 To NRows - 1
            For j = 0 To NColumn - 1
                .Offset(i, j) = Matrix1(i, j)
            Next j
        Next i
    End With
End Sub

```



**Задание 1.** Составить программу преобразования матрицы, разделив каждый элемент матрицы на ее максимальный элемент.

**Задание 2.** Составить программу преобразования матрицы, умножив каждый ее элемент на среднее арифметическое элементов матрицы.

**Задание 3.** Составить программу преобразования матрицы, разделив каждый ее элемент на произведение отрицательных элементов матрицы.

**Задание 4.** Составить программу преобразования матрицы, меняющую местами  $k$ -ю строку и  $r$ -й столбец квадратной матрицы. Номер строки и номер столбца ввести с клавиатуры с помощью окна ввода, воспользовавшись функцией `InputBox()`.

**Задание 5.** Составить программу преобразования матрицы, поменяв местами строки с минимальным и максимальным элементами матрицы.

**Задание 6.** Составить программу преобразования матрицы, разделив каждый ее элемент на среднее арифметическое отрицательных элементов матрицы.

**Задание 7.** Составить программу преобразования матрицы, поменяв местами столбцы с минимальным и максимальным элементами матрицы.

**Задание 8.** Составить программу преобразования матрицы, поменяв местами минимальный и максимальный элементы матрицы.

**Задание 9.** Матрица заполнена нулями и единицами. Считая, что строки матрицы представляют собой числа, записанные в двоичной системе, составить последовательность, элементы которой являются десятичными аналогами двоичных чисел матрицы. Элементы последовательности занести в ячейки рабочего листа Excel.

**Задание 10.** Дана матрица на рабочем листе Excel. Сформировать линейный массив  $B$ , элементами которого являются три минимальных элемента из каждой строки матрицы или 0, если сумма найденных элементов текущей строки равна 0. Элементы последовательности занести в ячейки рабочего листа Excel.

**Задание 11.** Дана матрица на рабочем листе Excel. Сформировать линейный массив  $C$ , элементами которого являются произведения положительных элементов каждого столбца (если такие есть) и нули, если положительные элементы отсутствуют. Элементы последовательности занести в ячейки рабочего листа Excel.

**Задание 12.** Дана матрица на рабочем листе Excel. Сформировать из элементов, стоящих выше главной диагонали линейный массив В, а из элементов, стоящих ниже главной диагонали линейный массив С. В последовательности С найти минимальный и максимальный элементы среди тех, которых нет в последовательности В. Элементы сформированных массивов, а также найденные элементы занести в ячейки рабочего листа Excel, расположив их рядом с исходной матрицей.

**Задание 13.** Дана матрица на рабочем листе Excel. Сформировать из элементов главной диагонали линейный массив В, а из элементов побочной диагонали линейный массив С. В последовательности С среди неповторяющихся элементов найти минимальный и максимальный элементы. Элементы сформированных массивов, а также найденные элементы занести в ячейки рабочего листа Excel, расположив их рядом с исходной матрицей.

**Задание 14.** Дана квадратная матрица на рабочем листе Excel. Составить программу определения строки матрицы, у которой среднее арифметическое элементов, предшествующих диагональному элементу больше среднего арифметического элементов, следующих за диагональным элементом. Элементы найденной строки поменять местами с элементами побочной диагонали.

**Задание 15.** На рабочем листе Excel расположены два линейных массива X и Y каждый размером N. Первый и второй расположены по вертикали, начиная с ячейки A1 и B1 соответственно. Составить программу преобразования массива X путем включения после его максимального элемента элементов, расположенных между максимальным и минимальным элементами массива Y. Элементы преобразованного массива X занести в ячейки рабочего листа Excel.

**Задание 16.** На рабочем листе Excel расположен по вертикали линейный массив, начиная с ячейки A1, и дана величина L, которую следует ввести с клавиатуры в процессе выполнения программы с помощью окна ввода InputBox(). Среди элементов этого массива найти три элемента, которые в сумме дают максимальное отклонение от введенной величины L. Преобразовать линейный массив, включив после каждого из найденных элементов значение индекса этого элемента. Элементы преобразованного массива занести в ячейки рабочего листа Excel.

### 3. Создание приложений с использованием пользовательских форм

**Задача 3.1.** Разработать приложение для расчета величины амортизации основных фондов с использованием формы пользователя для ввода исходных данных и выбора варианта расчета амортизации.

#### **Постановка задачи.**

Разработать программу для расчета амортизации с диалоговым окном.

Под амортизацией понимается уменьшение стоимости оборудования или другого имущества в процессе эксплуатации. Величину этого уменьшения оценивают за единицу времени.

Для расчета величины амортизации на период эксплуатации используется несколько функций VBA. В данном приложении используется две функции такие как: *SYD()* и *DDB()*.

Функция *SYD()* возвращает годовую амортизацию имущества для указанного периода и имеет следующий синтаксис:

*SYD (ПервичнаяСтоимость, ОстаточнаяСтоимость, ВремяЭксплуатации, ПериодРасчета)*

Функция *DDB()* возвращает величину амортизации имущества для указанного периода, используя метод двукратного (или к-кратного) учета амортизации и имеет следующий синтаксис:

*DDB (ПервичнаяСтоимость, ОстаточнаяСтоимость, ВремяЭксплуатации, ПериодРасчета, Кратность)*

где *ПервичнаяСтоимость* – начальная стоимость основных фондов;

*ОстаточнаяСтоимость* – стоимость в конце эксплуатации;

*ВремяАмортизации* – полная продолжительность эксплуатации (в периодах);

*ПериодРасчета* – номер периода, для которого производится расчет (измеряется в тех же единицах, что и *ВремяАмортизации*);

*Кратность* – норма снижения амортизации. Если коэффициент опущен, то предполагается, что он равен двум (метод двукратного учета амортизации).

Разработка приложения для расчета величины амортизации начинается с разработки формы в виде пользовательского диалогового окна.

Вид формы пользователя с элементами управления для ввода данных и выбора варианта расчета амортизации представлен на рис. 3.1.

Рис. 3.1. Форма «Расчет амортизации имущества»

### Программный код приложения.

#### 1. Код модуля запуска приложения.

На рабочем листе Excel создается элемент панели управления *Кнопка* панели инструментов *Visual Basic* с надписью на ней «*Вычисление амортизации*». С помощью этой кнопки будет запускаться приложение с выводом на рабочий лист *Excel* пользовательского диалогового окна. Для процедуры обработки события *Click* этой кнопки записывается следующий программный код:

```
Private Sub CommandButton1_Click()
    UserForm1.Show
End Sub
```

#### 2. Код модуля формы «Расчет амортизации имущества»

В общей области необходимо объявить вспомогательные переменные:

```
Option Explicit
Dim dblПервичнаяСтоимость As Double
Dim dblОстаточнаяСтоимость As Double
```

```

Dim intВремяАмортизации As Integer
Dim intПериодРасчета As Integer
Dim intКратность As Integer
Dim blnПризнак As Boolean
Dim dblВеличинаАмортизации As Double
Dim k As Integer

```

Код обработки события формы Initialize:

```

Private Sub UserForm_Initialize()
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    OptionButton1.Value = True
End Sub

```

Для активных элементов формы записывается соответствующий код.

Код процедуры обработки события Click кнопки «Вычислить» (в программе CommandButton1):

```

Private Sub CommandButton1_Click()
    If TextBox1.Text = "" Or TextBox2.Text = _
        "" Or TextBox3.Text = "" Or TextBox4.Text _
        = "" Then
        MsgBox "Нет данных для расчета", _
            vbExclamation, "Амортизация"
        Exit Sub
    End If
    dblПервичнаяСтоимость = CDb1(TextBox1.Text)
    dblОстаточнаяСтоимость = CDb1(TextBox2.Text)
    intВремяАмортизации = CInt(TextBox3.Text)
    intПериодРасчета = CInt(TextBox4.Text)
    If dblПервичнаяСтоимость < _
        dblОстаточнаяСтоимость Then
        MsgBox "Ошибка! Остаток больше начальной _
            амортизации", vbExclamation, "Амортизация"
        TextBox1.SetFocus
        Exit Sub
    End If
    If intВремяАмортизации < intПериодРасчета Then
        MsgBox "Ошибка в сроке амортизации", _

```

```

        vbExclamation, "Амортизация"
    TextBox3.SetFocus
    Exit Sub
End If
If OptionButton1.Value = True Then
    blnПризнак = True
Else
    blnПризнак = False
End If
If blnПризнак = True Then
    dblВеличинаАмортизации = _
    SYD(dblПервичнаяСтоимость, _
    dblОстаточнаяСтоимость, _
    intВремяАмортизации, intПериодРасчета)
Else
    TextBox6.SetFocus
    intКратность = CInt(TextBox6.Text)
    dblВеличинаАмортизации = _
    DDB(dblПервичнаяСтоимость, _
    dblОстаточнаяСтоимость, _
    intВремяАмортизации, intПериодРасчета)
End If
TextBox5.Text = CStr(dblВеличинаАмортизации)
ActiveSheet.Range("A1:F20").Clear
k = intКратность
ActiveSheet.Columns("A").Select
With Selection
    .ColumnWidth = 30
    .WrapText = True
End With
ActiveSheet.Range("B1").Select
With ActiveSheet
    .Range("A1").Value = "Начальная стоимость"
    .Range("A2").Value = "Остаточная _
        стоимость"
    .Range("A3").Value = "Время полной _
        амортизации"
    .Range("A4").Value = "Период, для _
        которого рассчитывается амортизация"
    .Range("A5").Value = "Расчет выполнен"
    .Range("A6").Value = "Величина _
        амортизации"
    .Range("B1").Value = dblПервичнаяСтоимость
    .Range("B2").Value = _

```

```

        dblОстаточнаяСтоимость
    .Range("B3").Value = intВремяАмортизации
    .Range("B4").Value = intПериодРасчета
    .Range("B5").WrapText = True
    .Range("B6").Value = _
        dblВеличинаАмортизации
If blnПризнак = True Then
    .Range("B5").Value = "Стандартным _
        Методом"
Else
    .Range("B5").Value = "Методом " & _
        CStr(k) & " кратного учета амортизации"
End If
End With
End Sub

```

Код процедуры обработки события Click кнопки «Завершить» (в программе CommandButton2):

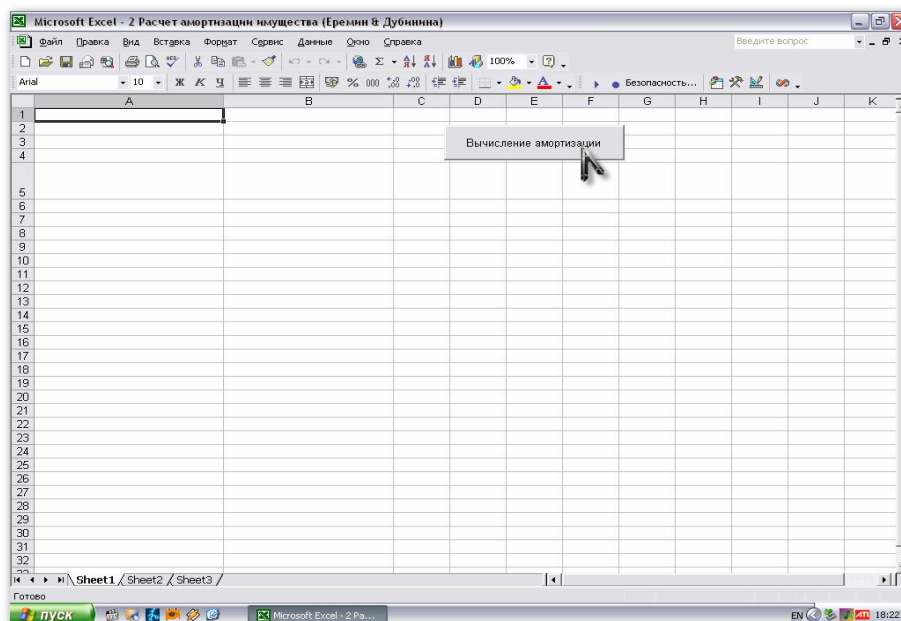
```

Private Sub CommandButton2_Click()
    End
End Sub

```

### **Порядок выполнения приложения «Расчет амортизации»**

1. Запускаем приложение нажатием на кнопку “Вычисление амортизации” на рабочем листе Excel (рис. 3.2).



*Рис. 3.2. Рабочий лист Excel с кнопкой запуска приложения*

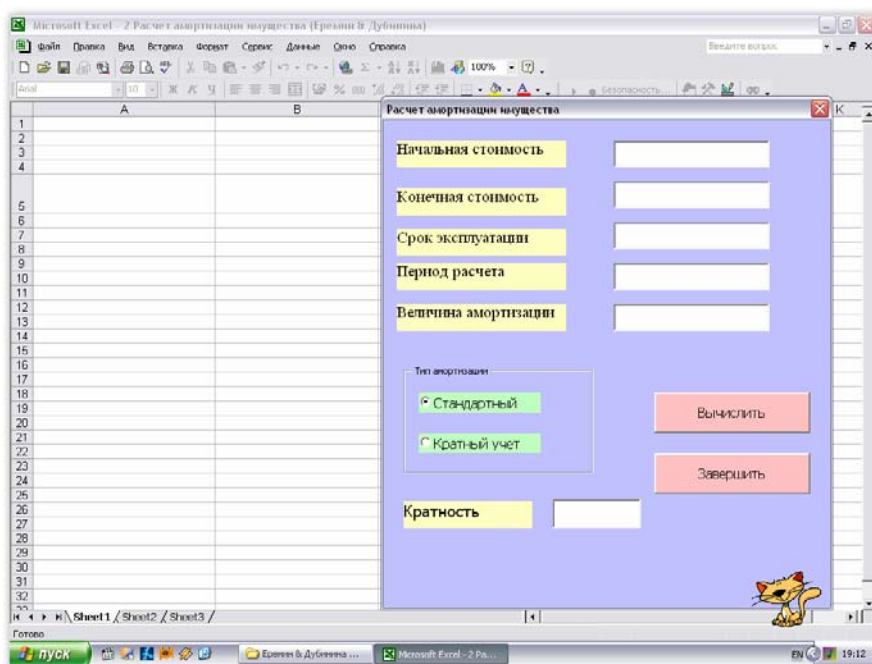


Рис. 3.3. Форма для ввода данных и выбора варианта расчета

2. На рабочем листе откроется форма. Заполните поля формы данными для проведения вычислений и нажмите на кнопку «Вычислить». Результат вычислений будет выведен на форму и в виде таблицы на рабочий лист Excel вместе с текстовыми пояснениями.

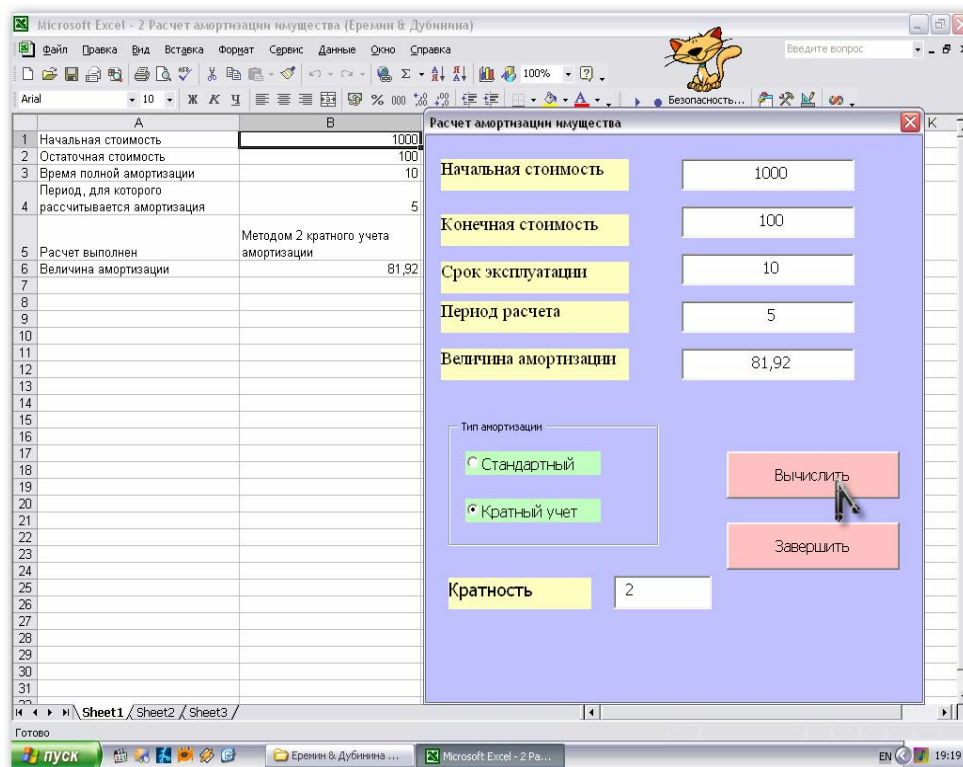


Рис. 3.4. Заполнение рабочего листа Excel результатами амортизации



**Задача 3.2.** Разработать приложение для проведения расчетов по ипотечному кредиту на недвижимость. Приложение создается с использованием формы пользователя для ввода исходных данных и выбора варианта расчета по ипотеке. Результаты вычисления выводятся в форму, а также на рабочий лист Excel с соответствующими текстовыми пояснениями.

### **Постановка задачи.**

Приложение предоставления кредита на покупку недвижимости реализует две основные функции.

1. Расчет регулярных периодических выплат (один раз в месяц или один раз в год), выплаченных процентов, сумму кредита и общую сумму выплат. Приложение принимает шесть параметров через оконный пользовательский интерфейс: стоимость недвижимости, начальный взнос в %, учетную ставку кредитования в %, время (в месяцах или годах) и период выплат (в начале или конце периода), а также срок погашения кредита (количество месяцев или лет, в течение которых должен выплачиваться кредит).
2. Отображение результатов выполнения приложения в соответствующих элементах управления формы и в табличной форме на рабочем листе Excel.

Для расчета величины регулярных периодических выплат существует специальная функция *VBA Pmt()*, ее синтаксис имеет следующий вид:

$$Pmt (Ставка, Kпер, -Hz [, Бз] [, Tин]),$$

где

*Ставка* – годовая процентная ставка;

*Kпер* – количество периодов выплат для погашения кредита;

*Hz* – сумма, которую нужно погасить;

*Бз* – баланс наличности, который нужно достичь после последней выплаты;

*Tин* – признак того, когда производится выплата (0 – в конце периода, 1 – в начале периода).

Разработка приложения для расчета по ипотечному кредиту начинается с разработки формы в виде пользовательского диалогового окна.

Вид формы пользователя с элементами управления для ввода данных и выбора варианта расчета по ипотечному кредиту представлен на рис. 3.5.

Рис. 3.5. Форма для расчета по ипотечному кредиту

### Программный код приложения.

#### 3. Код модуля запуска приложения.

На рабочем листе Excel создайте элемент панели управления *Кнопка* панели инструментов Visual Basic с надписью на ней “Расчет по ипотечному кредиту”. С помощью этой кнопки будет запускаться приложение с выводом на рабочий лист Excel пользовательского диалогового окна. Для процедуры обработки события *Click* этой кнопки запишите следующий код:

```
Private Sub CommandButton1_Click()
    UserForm1.Show
End Sub
```

#### 4. Код модуля формы “Расчет по ипотечному кредиту”.

Код обработки события формы *Initialize*:

```
Private Sub UserForm_Initialize()
    ListBox1.List = Array(3, 4, 5, 6, 7, 8, 9, _
                        10, 11, 12, 13, 14, 15)
    SpinButton1.Min = 0
    SpinButton1.Max = 1000
    OptionButton1.Value = True
    Label3.Caption = “Периодов”
```

```

ToggleButton1.Caption = "В начале периода"
Label7.Caption = ""
Label8.Caption = ""
Label9.Caption = ""
Label11.Caption = ""
End Sub

```

Для активных элементов формы записывается соответствующий код.

В модуле формы введите программный код процедур по обработке событий формы и элементов управления, расположенных на форме.

```

Private Sub CommandButton1_Click()
    Dim Ставка As Double
    Dim Кпер As Byte
    Dim Тип As Double
    Dim Нз As Double
    Нз = TextBox2.Value - TextBox2.Value * _
        TextBox3.Value / 100
    Label9.Caption = Нз
    Ставка = ListBox1.Value
    Ставка = Ставка / 100
    If OptionButton1.Value = True Then Ставка = _
        Ставка / 12
    Кпер = TextBox1.Value
    If ToggleButton1.Value = True Then Тип = 1 _
        Else Тип = 0
    Label7.Caption = Int(Pmt(Ставка, Кпер, -Нз, _
        , Тип))
    Label8.Caption = Label7.Caption * Кпер
    Label11.Caption = Label8.Caption - _
        Label9.Caption
    ActiveSheet.Range("A1:B10").Clear
    ActiveSheet.Columns("A").Select
    With Selection
        .ColumnWidth = 40
        .WrapText = True
    End With
    ActiveSheet.Range("B1").Select
    With ActiveSheet
        .Range("A1").Value = "Стоимость имущества"
        .Range("A2").Value = "Начальный взнос, %"
        .Range("A3").Value = "Периодов"
        .Range("A4").Value = "Ставка"
        .Range("A5").Value = "Порядок выплат"
    End With

```

```

.Range("A6").Value = "Результаты вычислений"
.Range("A7").Value = "Сумма кредита"
.Range("A8").Value = "Величина платежа"
.Range("A9").Value = "Полная сумма выплат"
.Range("A10").Value = "Комиссионные"
.Range("B1").Value = TextBox2.Text
.Range("B2").Value = TextBox3.Text
If OptionButton1.Value = True Then
    .Range("B3").Value = TextBox1.Text + " _
        Месяцев"
Else
    .Range("B3").Value = TextBox1.Text + " _
        Лет"
End If
.Range("B4").Value = ListBox1.Value
If ToggleButton1.Value = True Then
    .Range("B5").Value = "В начале периода"
Else
    .Range("B5").Value = "В конце периода"
End If
.Range("B6").Value = ""
.Range("B7").Value = Label9.Caption
.Range("B8").Value = Label7.Caption
.Range("B9").Value = Label8.Caption
.Range("B10").Value = Label11.Caption
End With
End Sub

```

Код процедуры обработки события Click кнопки Заккрыть (в программе CommandButton2):

```

Private Sub CommandButton2_Click()
    End
End Sub

```

Код процедуры обработки события Click элемента ToggleButton1 (на форме это кнопка с надписью "В начале периода"):

```

Private Sub ToggleButton1_Click()
    ToggleButton1.Value = True
End Sub

```

Код процедуры обработки события Change элемента SpinButton1:

```

Private Sub SpinButton1_Change()

```

```

    TextBox1.Text = SpinButton1.Value
End Sub

```

Код процедуры обработки события Change элемента TextBox1 (на форме это текстовое поле рядом с надписью “Периодов”):

```

Private Sub TextBox1_Change()
    If TextBox1.Text = "" Then Exit Sub
    SpinButton1.Value = CInt(TextBox1.Text)
End Sub

```

**Порядок выполнения приложения «Расчет по ипотечному кредиту» следующий.**

1. Запускаем приложение нажатием на кнопку “Расчет по ипотечному кредиту”, расположенную на рабочем листе Excel (рис. 3.6).

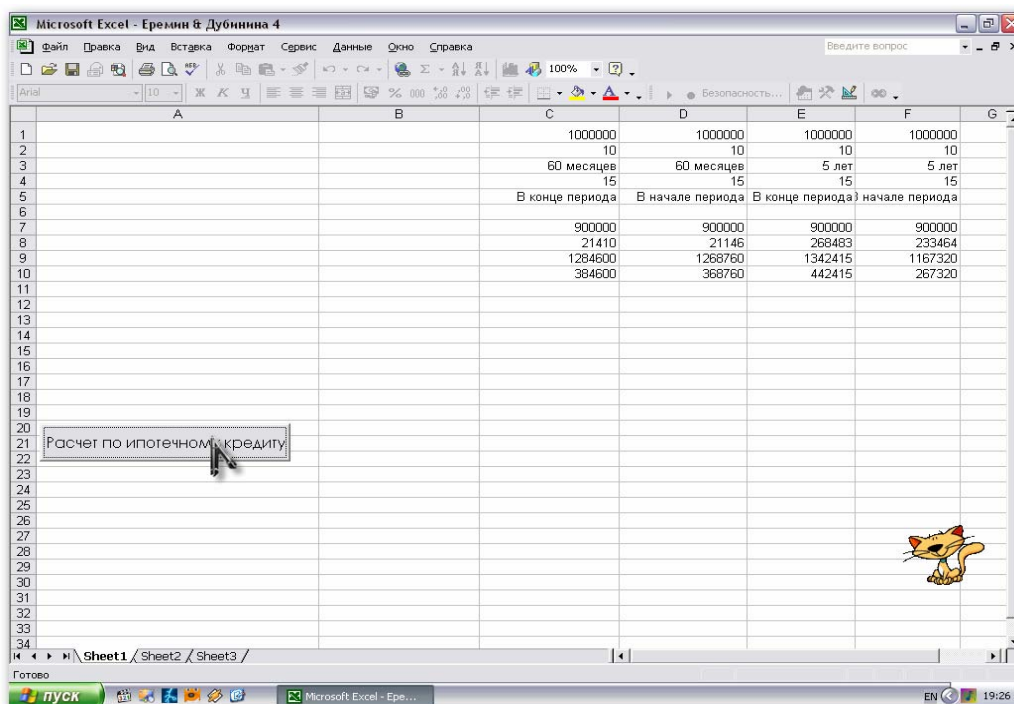


Рис. 3.6. Рабочий лист Excel с кнопкой запуска приложения

2. На рабочем листе откроется форма (рис. 3.7). Заполните поля формы данными для проведения вычислений. Образец заполнения элементов управления формы данными приведен на рис. 3.8. После проверки правильности ввода данных нажмите на кнопку «Вычислить». Результаты вычислений будут выведены на форму в элементы управления «Label» или «Надпись» и на рабочий лист Excel в виде таблицы с соответствующими текстовыми пояснениями.

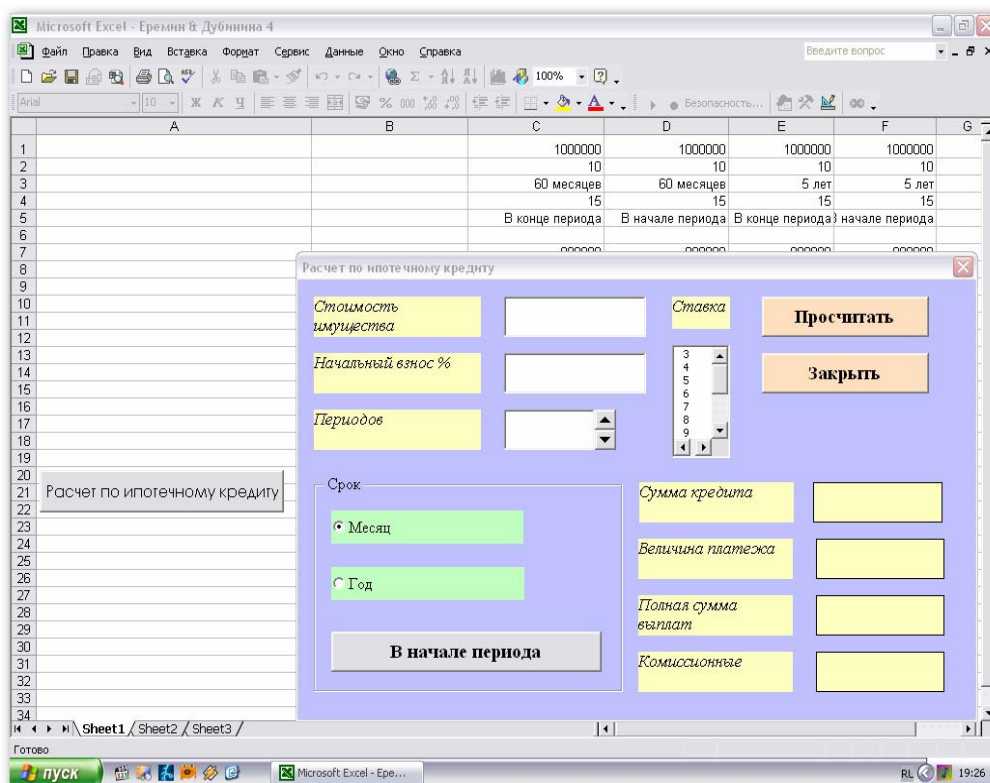


Рис. 3.7. Форма для ввода данных и выбора варианта расчета

3. Образец различных вариантов расчетов по ипотечному кредиту приведен на рис. 3.8 в виде таблицы.

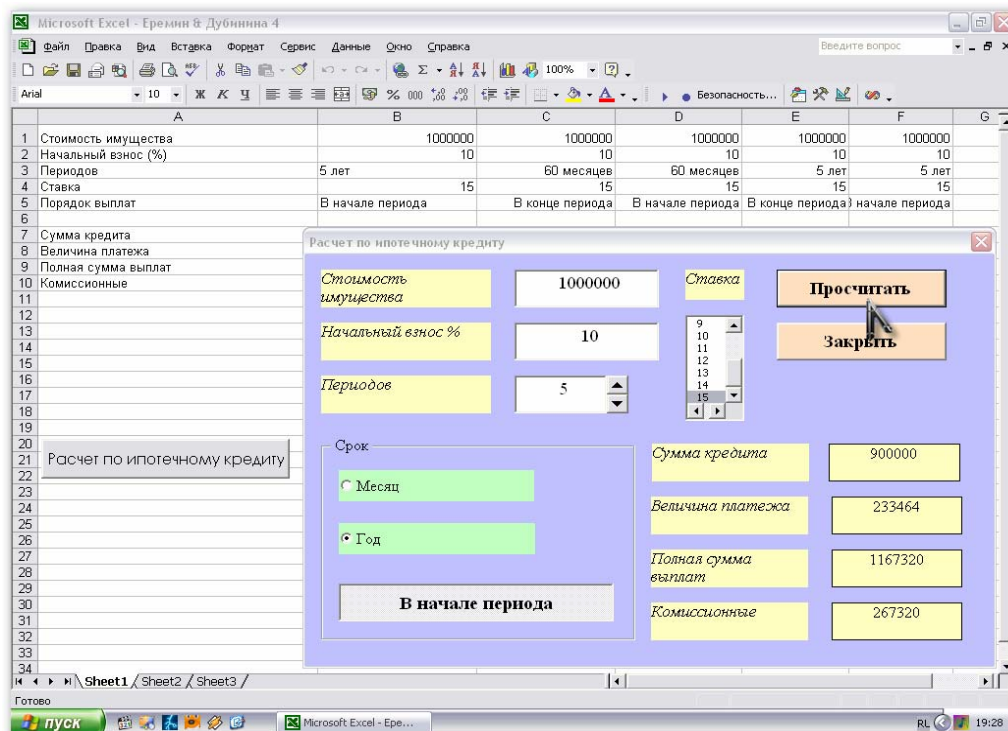


Рис 3.8. Заполнение рабочего листа Excel результатами расчетов

## 4. Создание приложений для формирования и заполнения баз данных (списков) в Excel

**Задача 4.1.** Разработать приложение с пользовательским диалоговым окном для создания на рабочем листе Excel базы данных по учету выданных кредитов. Приложение реализует четыре основных функции.

1. Принимает десять параметров: фамилия, имя, пол, адрес проживания, предмет и сумма залога, срок кредита, сумма кредита, ставка, тип выплат (в конце/начале месяца).
2. На основе полученных данных с помощью встроенной функции VBA Pmt() определяет сумму периодических выплат по кредиту.
3. Исходные данные и расчетные значения выводит в виде записи в базу данных на рабочий лист «Кредиты». При этом данные о залоге и дате оформления кредита (получаемой с помощью встроенной функции Now) выводит в примечание.
4. Вывод формы на рабочий лист Excel осуществляет с помощью созданной в процедуре строки меню.

Разработка приложения начинается с разработки формы в виде пользовательского диалогового окна.

Вид формы пользователя с элементами управления для ввода данных в поля очередной записи базы данных представлен на рис. 4.1.

Рис. 4.1. Форма для оформления кредита клиенту

Для удобства ввода данных, кроме элементов *Textbox*, в форме предусмотрены следующие элементы управления: *Option Button* – для выбора пола, *ComboBox* – для установки ставки кредита и вида залога, выбора города, *ToggleButton* – для выбора типа выплат (табл. 4.1).

Таблица 4.1

*Объекты, свойства и события элементов управления*

Элемент	Имя	Свойства	События
UserForm	UserForm1	По умолчанию	Initialize: 1. Формирует названия колонок таблицы. 2. Закрывает строку формул. 3. Устанавливает по умолчанию переключатель «муж». 4. Присваивает значения трем раскрывающимся спискам: Город, Вид залога, Ставка 5. Устанавливает по умолчанию переключатель «в конце месяца» 6. Назначает кнопкам ОК, Отмена, Завершить работу всплывающую подсказку
TextBox	TextLastname	По умолчанию	Нет
TextBox	TextName	По умолчанию	Нет
TextBox	TextStr	По умолчанию	Нет
TextBox	TextBld	По умолчанию	Нет
TextBox	TextFlat	По умолчанию	Нет
TextBox	TextTerm	По умолчанию	Нет
TextBox	TextAmount	По умолчанию	Нет
TextBox	TextPawn	По умолчанию	Нет
ComboBox	ComCity	По умолчанию	Нет
ComboBox	ComRate	По умолчанию	Нет
ComboBox	ComPawn	По умолчанию	Нет
ToggleButton	TogType	Caption=В конце месяца	Click: Изменяет свойство Caption и тип платежей
CommandButton	OKButton	Caption=ОК	Click: 1. Вычисляет номер первой пустой строки 2. Проверяет введенные данные на наличие ошибок 3. Заносит данные в таблицу БД
CommandButton	CancelButton	Caption=Отмена	Click: Отменяет надпись в строке заголовка приложения
CommandButton	QuitButton	Caption=Завершить	Click: Активирует лист



### Программный код приложения.

1. Рабочему листу Excel присвойте имя «Кредиты».
2. Создайте процедуру «Заголовки», которая проверяет, заполнена ли первая строка – строка заголовков полей базы данных. Если не заполнена, то процедура формирует названия полей и задает их ширину. Если первая строка заполнена, то продолжает ввод данных:

```
Private Sub Заголовки()  
    Application.Worksheets("Кредиты").Activate  
    With ActiveSheet  
        If .Range("A1").Value = "Фамилия" Then  
            Range("A2").Select  
        Else  
            ActiveSheet.Cells.Clear  
            Range("A1:H1").Value = Array("Фамилия", _  
                "Имя", "Адрес", "Срок кредита", "Стака", _  
                "Сумма кредита", "Платежи", "Тип")  
            .Range("C:C").ColumnWidth = 25  
            .Range("D:D").ColumnWidth = 12  
            .Range("E:E").ColumnWidth = 7  
            .Range("F:F").ColumnWidth = 13  
            .Range("G:G").ColumnWidth = 10  
            .Range("G:G").ColumnWidth = 10  
            .Range("2:2").Select  
            .Range("A2").Select  
        End If  
    End With  
    UserForm1.Show  
End Sub
```

3. В Общей области декларируйте переменные:

```
Option Explicit  
Dim Фамилия As String  
Dim Имя As String  
Dim Пол As String  
Dim Адрес As String  
Dim ВидЗалог As String  
Dim СуммаЗалог As String  
Dim Срок As Integer  
Dim Ставка As Integer  
Dim Кредит As Double  
Dim Платежи As Double  
Dim Тип As String
```

```
Dim КогдаПлатим As String
Dim Дата As Date
```

4. В процедуру обработки события *Initialize* формы введите код:

```
Private Sub UserForm_Initialize()
    Application.DisplayFormulaBar = False
    With UserForm1
        'Устанавливаем переключатель "муж"
        .OptMale = True
        'Присваиваем значения спискам
        .ComCity.List = Array("Воронеж", "Эртиль", _
            "Богучар", "Новохоперск", "Лиски", _
            "Бобров", "Анна", "Россошь", "Калач")
        .ComPawn.List = Array("Ценные бумаги", _
            "Недвижимость", "Автомобиль")
        .ComRate.List = Array("3", "4", "5", "6", _
            "7", "8", "9", "10", "11", "12", "13", _
            "14", "15")
        'Устанавливаем выключатель
        .TogType.Value = False
        .OKButton.SetFocus
        .OKButton.ControlTipText = "Ввод в базу _
            данных"
        .CancelButton.ControlTipText = "Отменить _
            ввод"
        .CancelButton.Cancel = True
        .QuitButton.ControlTipText = "Закончить _
            ввод"
    End With
    'Вызываем процедуру Заголовки
    Заголовки
End Sub
```

5. В процедуру обработки события *Click* кнопки *OK* введите код:

```
Private Sub OKButton_Click()
    Dim НомерСтроки As Integer
    'Вычисляем номер первой незаполненной строки
    таблицы
    НомерСтроки = Application.CountA(ActiveSheet. _
        Columns(1)) + 1
    'Проверка ошибок ввода
    With UserForm1
```

```

If .TextLastname = "" Then
    MsgBox "Вы забыли указать фамилию", _
        vbExclamation
    Exit Sub
End If
If .TextName = "" Then
    MsgBox "Вы забыли указать имя", _
        vbExclamation
    Exit Sub
End If
If .ComCity.Value = "" Then
    MsgBox "Вы забыли указать город", _
        vbExclamation
    Exit Sub
End If
If .ComRate.Value = "" Then
    MsgBox "Вы забыли указать ставку _
        кредитования", vbExclamation
    Exit Sub
End If
If .TextStr = "" Then
    MsgBox "Вы забыли указать улицу", _
        vbExclamation
    Exit Sub
End If
If .TextBld = "" Then
    MsgBox "Вы забыли указать номер дома", _
        vbExclamation
    Exit Sub
End If
If .TextTerm = "" Then
    MsgBox "Вы забыли указать срок кредита", _
        vbExclamation
    Exit Sub
End If
If .TextAmount = "" Then
    MsgBox "Вы забыли указать сумму кредита" _
        , vbExclamation
    Exit Sub
End If
If .ComPawn.Value = "" Then
    MsgBox "Вы забыли указать вид залога", _
        vbExclamation
    Exit Sub

```

```

End If
If .TextPawn = "" Then
    MsgBox "Вы забыли указать сумму залога", _
        vbExclamation
    Exit Sub
End If
If IsNumeric(.TextAmount.Text) = False Then
    MsgBox "Сумма кредита введена неверно", _
        vbCritical
    Exit Sub
End If
If IsNumeric(.TextTerm.Text) = False Then
    MsgBox "Срок кредита введен неверно", _
        vbCritical
    Exit Sub
End If
If IsNumeric(.TextPawn.Text) = False Then
    MsgBox "Сумма залога введена неверно", _
        vbCritical
    Exit Sub
End If
'Присваиваем переменным значения
Фамилия = .TextLastname.Text
Имя = .TextName.Text
If .OptFemale.Value = True Then
    Пол = "Женщина"
End If
If .OptMale.Value = True Then
    Пол = "Мужчина"
End If
Адрес = .ComCity.Value + ", " + _
        .TextStr.Text + ", " + .TextBld.Text + _
        ", " + .TextFlat.Text
ВидЗалог = .ComPawn.Value
СуммаЗалог = .TextPawn.Text
Срок = .TextTerm.Text
Ставка = CInt(.ComRate.Value)
Кредит = CDb1(.TextAmount.Text)
If .TogType.Value = False Then
    Тип = 0
    КогдаПлатим = "В конце месяца"
End If
If .TogType.Value = True Then
    Тип = 1

```

```

        КогдаПлатим = "В начале месяца"
        .TogType.Caption = "В начале месяца"
    End If
    'Определяем размер платежей
    Платежи = Pmt(Ставка / 12, Срок, -Кредит, _
        0, Тип)
    'Определяем текущую дату и время
    Дата = Now
End With
'Записываем в ячейки текущей строки значения
переменных
With ActiveSheet
    .Cells(НомерСтроки, 1).Value = Фамилия
    'Формируем примечание
    .Cells(НомерСтроки, 1).AddComment
    .Cells(НомерСтроки, 1).Comment.Visible = _
        False
    .Cells(НомерСтроки, 1).Comment.Text Text:= _
        "Заложил " & ВидЗалог & " на сумму " _
        & СуммаЗалог & " " & " " & Дата & _
        " " & Пол
    .Cells(НомерСтроки, 2).Value = Имя
    .Cells(НомерСтроки, 3).Value = Адрес
    .Cells(НомерСтроки, 4).Value = Срок
    .Cells(НомерСтроки, 5).Value = Ставка
    .Cells(НомерСтроки, 6).Value = Кредит
    .Cells(НомерСтроки, 7).Value = Платежи
    .Cells(НомерСтроки, 8).Value = КогдаПлатим
End With
End Sub

```

6. Для процедуры обработки события Click кнопки Отмена запишите следующий код:

```

Private Sub CancelButton_Click()
    Dim НомерСтроки As Integer
    'Вычисляем номер последней заполненной строки
    НомерСтроки = Application.CountA(ActiveSheet _
        .Columns(1))
    With ActiveSheet
        .Cells(НомерСтроки, 1).Value = ""
        'Удаляем примечание
        .Cells(НомерСтроки, 1).ClearComments
        .Cells(НомерСтроки, 2).Value = ""
        .Cells(НомерСтроки, 3).Value = ""
    End With

```

```

        .Cells(НомерСтроки, 4).Value = ""
        .Cells(НомерСтроки, 5).Value = ""
        .Cells(НомерСтроки, 6).Value = ""
        .Cells(НомерСтроки, 7).Value = ""
        .Cells(НомерСтроки, 8).Value = ""
    End With
End Sub

```

7. В процедуру обработки события *Click* кнопки «Завершить работу» введите следующий код:

```

Private Sub QuitButton_Click()
    Sheets("Кредиты").Activate
End
End Sub

```

8. В окне проекта редактора *VB* выберите лист «ЭтаКнига» («*ThisWorkbook*») и введите следующую процедуру:

```

Private Sub Workbook_WindowActivate(ByVal Wn _
    As Window)
    Dim СтрокаМеню As CommandBar
    Dim Форма As CommandBarPopup
    Dim Вывод As CommandBarButton
    'Создаем строку меню
    Set СтрокаМеню = Application.CommandBars.Add _
        ("ФОРМА", msoBarTop, True, True)
    СтрокаМеню.Visible = True
    'Создаем подменю
    Set Форма = СтрокаМеню.Controls.Add _
        (msoControlPopup)
    With Форма
        .Caption = "&Вывести форму"
    End With
    Set Вывод = Форма.Controls.Add _
        (msoControlButton)
    With Вывод
        .Caption = "Вывести форму"
        'Назначаем действие команде подменю - запуск
        макроса Меню
        .OnAction = "Меню"
    End With
End Sub

```

## 9. Запишите макрос «Меню»

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

**Порядок выполнения приложения «Формирование базы данных «Кредиты»» следующий.**

1. Запускаем приложение с помощью созданной строки меню.
2. На рабочем листе откроется форма (рис. 4.2). Заполните поля формы данными по образцу для проведения вычислений.
3. После нажатия на кнопку «OK» на рабочий лист «Кредиты» будет выведена очередная запись в базу данных по учету выданных кредитов.

Оформление кредита

Фамилия: Лукин

Имя: Станислав

Пол: ☒ муж ☐ жен

Адрес:

Город: Воронеж

Улица: Димитрова

Дом: 142 Кв: 302

Залог:

Вид: Ценные бумаги

Сумма: 5300

Срок кредита (в мес): 18

Сумма кредита: 10000

Ставка: 3

В начале месяца ☐

OK

Отмена

Завершить работу

Рис. 4.2. Заполненная форма для создания очередной записи БД

1	Фамилия	Имя	Залог	Срок кредита	Ставка	Сумма кредита	Платежи	Тип
2	Лукин	Станислав	Ценные бумаги на сумму 5300	18	3	10000	2036.68974	В начале месяца
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								

Рис. 4.3. Заполнение рабочего листа Excel «Кредиты» очередной записью базы данных

**Задача 4.2.** Разработать приложение для создания базы данных “*Страхование*” с использованием формы пользователя. В базе данных должны отражаться следующие сведения:

- фамилия и имя страхователя;
- адрес (город, улица, дом, квартира);
- пол страхователя;
- срок страхования;
- вид страхования.

Разработка приложения для создания базы данных по учету страхователей и вида страхования начинается с разработки формы в виде пользовательского диалогового окна.

Вид формы пользователя с элементами управления для создания необходимой базы данных представлен на рис. 4.4.

Страхование

Фамилия

Имя

Адрес

Город

Улица

Дом

Кв

Пол

☒ Муж

☐ Жен

Вид страхования

Жизни

Убытков

Имущества

Срок страхования

OK

Отмена

Рис. 4.4. Форма «Страхование»



### Программный код приложения.

На рабочем листе Excel с именем «Лист 1» разместите элемент «Кнопка» панели инструментов *Visual Basic* с надписью на ней «Страхование». С помощью этой кнопки будет запускаться разрабатываемое пользовательское диалоговое окно. Для процедуры обработки события *Click* этой кнопки запишем следующий код:

```
Private Sub CommandButton1_Click()  
    Страхование.Show  
End Sub
```

Рабочему листу с именем «Лист 2» присваиваем имя «База данных». На этом листе будет формироваться база данных по учету страхователей и вида страхования.

Ниже приведен листинг программного кода приложения, расположенного в модуле формы «Страхование» и автоматизирующего процедуру создания базы данных по учету страхователей и вида страхования.

В общей области объявляем вспомогательные переменные:

```
Dim Пол As String * 3  
Dim НомерСтроки As Integer
```

Для формирования заголовков столбцов базы данных создается процедура Sub пользователя, которая вызывается из процедуры обработки события формы *Initialize*. Код данной процедуры следующий:

```
Public Sub ЗаголовокРабочегоЛиста()  
    If Range("A1").Value = "Фамилия" Then  
        Range("A2").Select  
        Exit Sub  
    End If  
    ActiveSheet.Cells.Clear  
    Range("A1:I1").Value = Array("Фамилия", _ "Имя",  
    "Пол", "Город", "Улица", "Дом", _ "Квартира",  
    "Вид страхования", "Срок")  
    Range("2:2").Select  
    ActiveWindow.FreezePanes = True  
    Range("A2").Select  
    Range("A1").AddComment  
    Range("A1").Comment.Visible = False  
    Range("A1").Comment.Text Text:="Фамилия _  
        клиента"  
    Range("B1").AddComment
```

```

Range("B1").Comment.Visible = False
Range("B1").Comment.Text Text:="Имя клиента"
Range("C1").AddComment
Range("C1").Comment.Visible = False
Range("C1").Comment.Text Text:="Пол клиента"
Range("D1").AddComment
Range("D1").Comment.Visible = False
Range("D1").Comment.Text Text:="Адрес клиента"
End Sub

```

Код обработки события формы Initialize:

```

Private Sub UserForm_Initialize()
    Worksheets("База Данных").Activate
    ЗаголовокРабочегоЛиста
    Application.Caption = "Регистрация"
    Application.DisplayFormulaBar = False
    With КнопкаОК
        .Default = True
        .ControlTipText = "Ввод данных в _
        базу данных"
    End With
    With КнопкаОтмена
        .Cancel = True
        .ControlTipText = "Кнопка отмены"
    End With
    Список.List = Array("Жизни", "Убытков", _
        "Имущества")
    ПолМуж.Value = True
End Sub

```

Для активных элементов формы записывается соответствующий код.  
Код процедуры обработки события Click кнопки ОК:

```

Private Sub КнопкаОК_Click()
    НомерСтроки = _
    Application.CountA(ActiveSheet.Columns(1)) + 1
    With Страхование
        If .ПолМуж.Value = True Then
            Пол = "Муж"
        Else
            Пол = "Жен"
        End If
    End With

```

```

With ActiveSheet
    .Cells(НомерСтроки, 1).Value = ПолеФамилия
    .Cells(НомерСтроки, 2).Value = ПолеИмя
    .Cells(НомерСтроки, 3).Value = Пол
    .Cells(НомерСтроки, 4).Value = ПолеГород
    .Cells(НомерСтроки, 5).Value = ПолеУлица
    .Cells(НомерСтроки, 6).Value = ПолеДом
    .Cells(НомерСтроки, 7).Value = _
        ПолеКвартира
    .Cells(НомерСтроки, 8).Value = _
        Список.Value
    .Cells(НомерСтроки, 9).Value = ПолеСрок
End With
End Sub

```

Код процедуры обработки события Click кнопки Отмена:

```

Private Sub КнопкаОтмена_Click()
    Application.Caption = Empty
End
End Sub

```

Код процедуры обработки события Change элемента Счетчик:

```

Private Sub Счетчик_Change()
    With Страхование
        .ПолеСрок.Text = CStr(.Счетчик.Value)
    End With
End Sub

```

Код процедуры обработки события Change элемента ПолеСрок:

```

Private Sub ПолеСрок_Change()
    With Страхование
        .Счетчик.Value = CInt(.ПолеСрок.Text)
    End With
End Sub

```

**Порядок и демонстрация выполнения приложения следующие:**

1. Запустите приложение путем нажатия на кнопку “*Страхование*”, расположенную на рабочем листе «*Лист 1*» табличного процессора *Excel* (рис. 4.5).

2. На рабочем листе откроется форма (рис. 4.6).

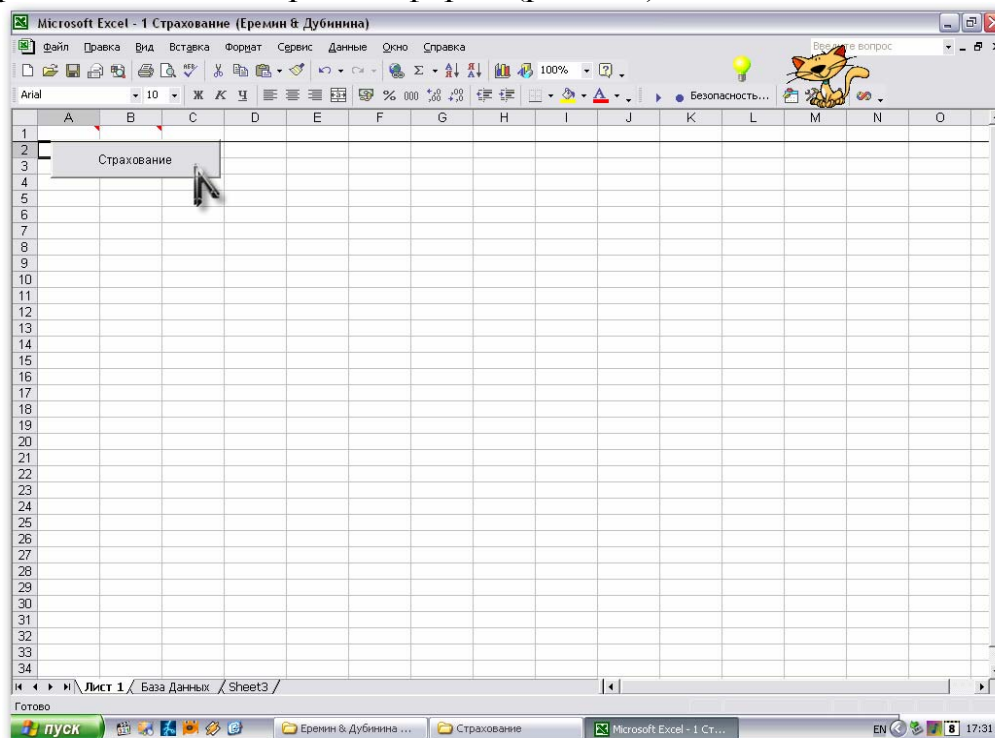


Рис. 4.5. Рабочий лист Excel с кнопкой запуска приложения

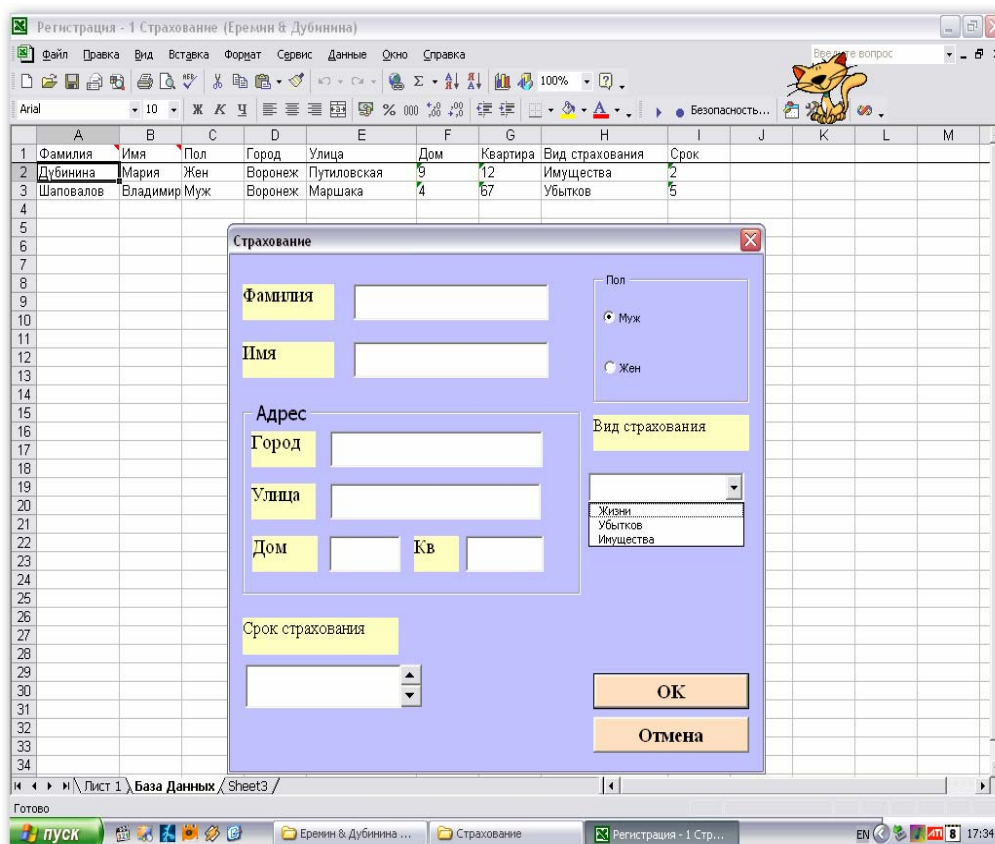


Рис. 4.6. Форма для ввода очередной записи в базу данных

3. Заполните поля формы данными по образцу на рис. 4.7. и нажмите на кнопку «OK».

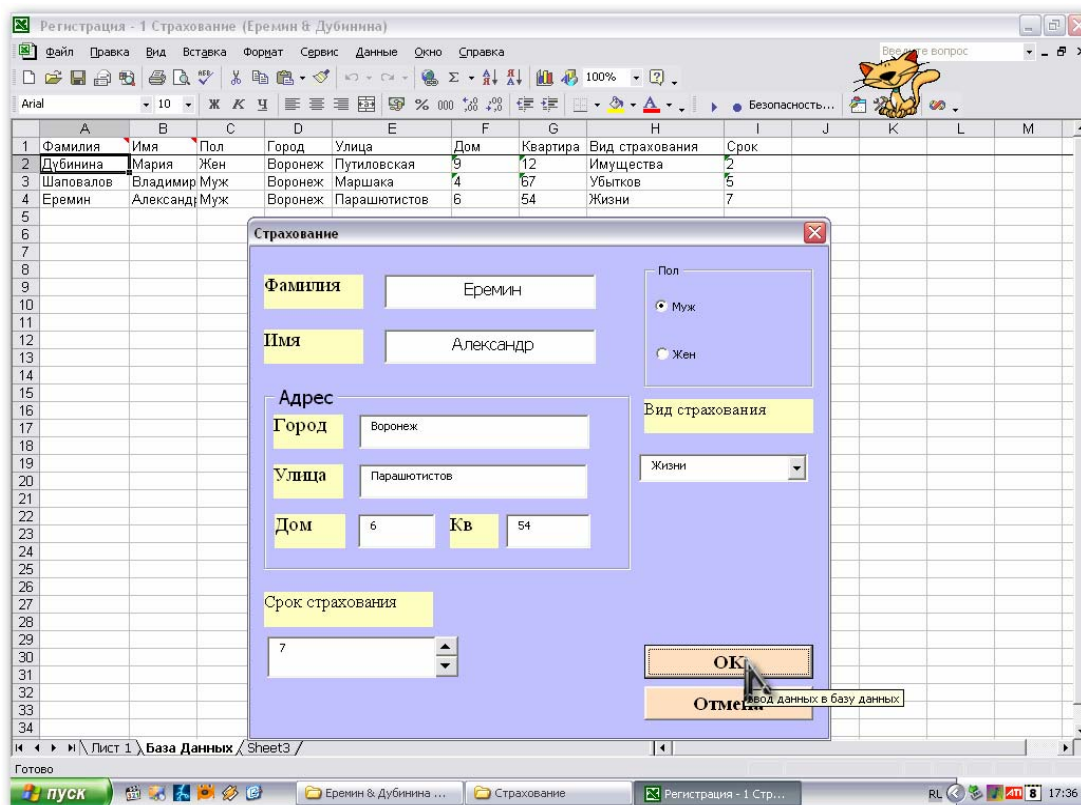


Рис. 4.7. Сформированная база данных «Страхование» на рабочем листе «База данных»

Содержимое записи будет добавлено в соответствующие поля базы данных расположенной на рабочем листе «База данных» Excel (рис. 4.7).

**Задача 4.3.** Разработать приложение с пользовательским диалоговым окном для создания на рабочем листе Excel базы данных по учету вкладов клиентов отделений банка. Приложение реализует три основных функции.

1. Принимает пять параметров: фамилия вкладчика, сумма вклада, вид вклада, отделение банка и примечание.
2. Исходные данные и расчетные значения выводит в виде записи в базу данных на рабочий лист «База». При этом проверяет, если база создается впервые, то перед выводом первой записи происходит обращение к процедуре «ЗаголовокРабочегоЛиста», которая выводит заголовки полей базы данных.
3. Запускает приложение с выводом формы на рабочий лист Excel с помощью кнопки расположенной на рабочем листе «Меню».

Разработка приложения начинается с задания имен рабочим листам: «Меню», «База», «Операции» и с разработки формы в виде пользователь-

ского диалогового окна. Форма имеет имя (*Name*) «Вклад», а название (*Caption*) – «Прием вклада».

Вид формы пользователя с элементами управления для ввода данных в поля очередной записи базы данных представлен на рис. 4.8.

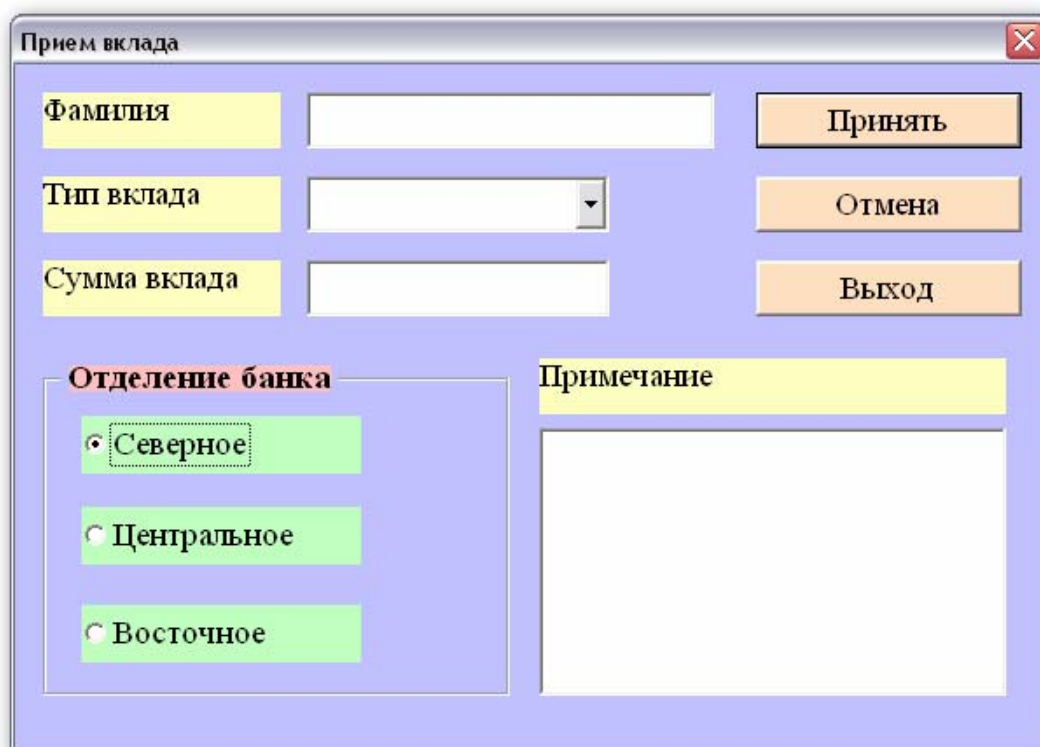


Рис. 4.8. Форма «Вклад»

### Программный код приложения.

1. Код модуля запуска приложения с выводом формы на рабочий лист Excel.

На рабочем листе Excel создается элемент панели управления «Кнопка» панели инструментов *Visual Basic* с надписью на ней “Прием вклада”. С помощью этой кнопки будет запускаться разрабатываемое диалоговое окно. Для процедуры обработки события *Click* этой кнопки записывается следующий код:

```
Private Sub ПриемВклада_Click()  
    Вклад.Show  
End Sub
```

2. Код модуля формы “Вклад”

Код обработки события формы *Initialize*:

```

Private Sub UserForm_Initialize()
    With Вклад
        .Северное.Value = True
        .ТипВклада.ListRows = 3
        .ТипВклада.List = Array("Срочный", _
            "Депозит", "Текущий")
        .Принять.SetFocus
    End With
    Application.Caption = "Регистрация. База _
        данных Банк"
    Application.DisplayFormulaBar = False
    With Принять
        .Default = True
        .ControlTipText = "Ввод данных в _
            базу данных"
    End With
    With Отмена
        .Cancel = True
        .ControlTipText = "Кнопка отмены"
    End With
    ЗаголовокРабочегоЛиста
End Sub

```

Для формирования заголовков полей базы данных создайте процедуру *Sub* пользователя, которая вызывается из процедуры обработки события формы *Initialize*. Код данной процедуры имеет следующий вид:

```

Private Sub ЗаголовокРабочегоЛиста()
    Application.Worksheets("База").Activate
    With ActiveSheet
        If .Range("A1").Value = "Фамилия" Then
            .Range("A2").Select
        Else
            ActiveSheet.Cells.Clear
            .Range("A1:E1").Value = Array("Фами- _
                лия", "Тип", "Сумма", "Отделе- _
                ние", "Примечание")
            .Range("2:2").Select
            ActiveWindow.FreezePanels = True
            .Range("A2").Select
            .Range("A1").AddComment
            .Range("A1").Comment.Visible = False
            .Range("A1").Comment.Text Text:="Фами- _
                лия клиента"
        End If
    End With
End Sub

```

```

        .Range("B1").AddComment
        .Range("B1").Comment.Visible = False
        .Range("B1").Comment.Text Text:="Тип _
        вклада"
        .Range("C1").AddComment
        .Range("C1").Comment.Visible = False
        .Range("C1").Comment.Text Text:="Сумма _
        вклада"
        .Range("D1").AddComment
        .Range("D1").Comment.Visible = False
        .Range("D1").Comment.Text _
        Text:="Отделение банка"
    End If
End With
Вклад.Show
End Sub

```

Для активных элементов управления расположенных на форме запишите соответствующий программный код.

Код процедуры обработки события *Click* кнопки «Принять»:

```

Private Sub Принять_Click()
    Dim Фамилия As String
    Dim ТипВклада As String
    Dim СуммаВклада As Double
    Dim Отделение As String
    Dim Примечание As String
    Dim НомерСтроки As Integer
    НомерСтроки = Application.CountA(Active _
        Sheet.Columns(1)) + 1
    With Вклад
        If .Фамилия.Text = "" Then
            MsgBox "Вы забыли указать фамилию", _
                vbExclamation
            Exit Sub
        End If
        If .ТипВклада.Value = "" Then
            MsgBox "Вы забыли указать тип вкла- _
                да", vbExclamation
            Exit Sub
        End If
        Фамилия = .Фамилия.Text
        ТипВклада = .ТипВклада.Value
        If .Северное.Value = True Then _

```



```

        Отделение = "Северное"
    If .Центральное.Value = True Then _
        Отделение = "Центральное"
    If .Восточное.Value = True Then _
        Отделение = "Восточное"
    If IsNumeric(.СуммаВклада.Text) = False _
        Then
        If .СуммаВклада.Text = "" Then
            MsgBox "Вы забыли указать сумму _
                вклада", vbExclamation
            Exit Sub
        End If
        MsgBox "Введена неверная сумма", _
            vbExclamation
        Exit Sub
    End If
    СуммаВклада = CDBl(.СуммаВклада.Text)
    Примечание = .Примечание.Text
End With
With ActiveSheet
    .Cells(НомерСтроки, 1).Value = Фамилия
    .Cells(НомерСтроки, 2).Value = ТипВклада
    .Cells(НомерСтроки, 3).Value = СуммаВклада
    .Cells(НомерСтроки, 4).Value = Отделение
    .Cells(НомерСтроки, 5).Value = Примечание
End With
End Sub

```

Код процедуры обработки события *Click* кнопки «Отмена»:

```

Private Sub Отмена_Click()
    Dim НомерСтроки As Integer
    НомерСтроки = Application.CountA(Active- _
        Sheet.Columns(1))
    With ActiveSheet
        .Cells(НомерСтроки, 1).Value = ""
        .Cells(НомерСтроки, 2).Value = ""
        .Cells(НомерСтроки, 3).Value = ""
        .Cells(НомерСтроки, 4).Value = ""
        .Cells(НомерСтроки, 5).Value = ""
    End With
End Sub

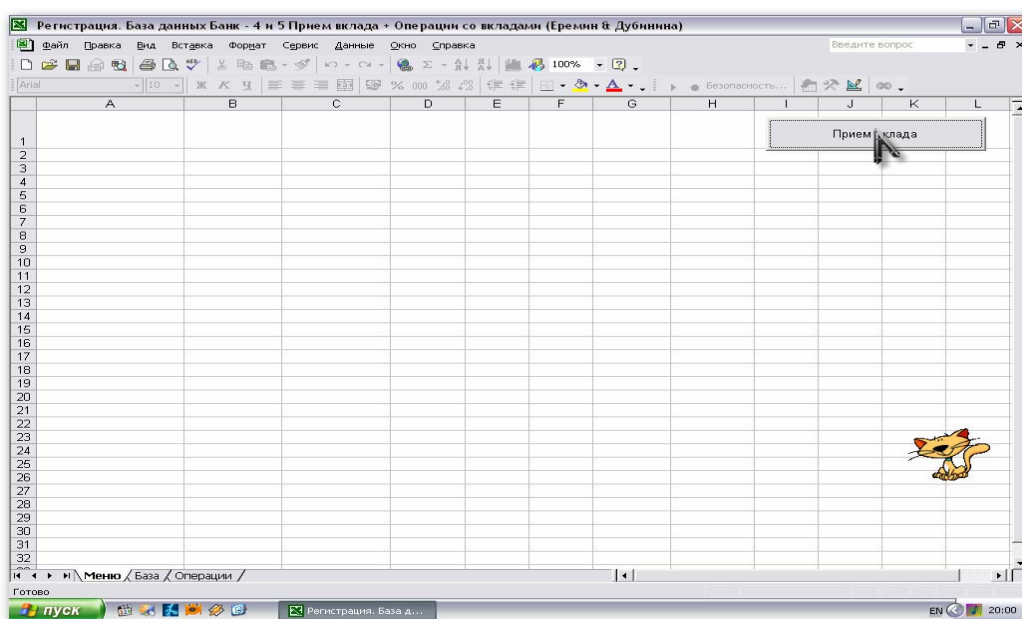
```

Код процедуры обработки события *Click* кнопки «Выход»:

```
Private Sub Выход_Click()
    Application.Worksheets("Меню").Activate
End Sub
```

**Порядок и демонстрация выполнения приложения следующие:**

1. Запустите приложение путем нажатия на кнопку «Прием вкладов», расположенную на рабочем листе «Меню» *Excel* (рис. 4.9).



*Рис. 4.9. Рабочий лист Excel с кнопкой запуска приложения*

2. На рабочем листе откроется форма (рис. 4.10). Заполните поля формы данными по образцу рис. 4.11 для внесения информации о клиенте в базу данных.
3. После нажатия на кнопку «Принять» на рабочий лист «База» будет введена очередная запись в базу данных по учету клиентов банка.

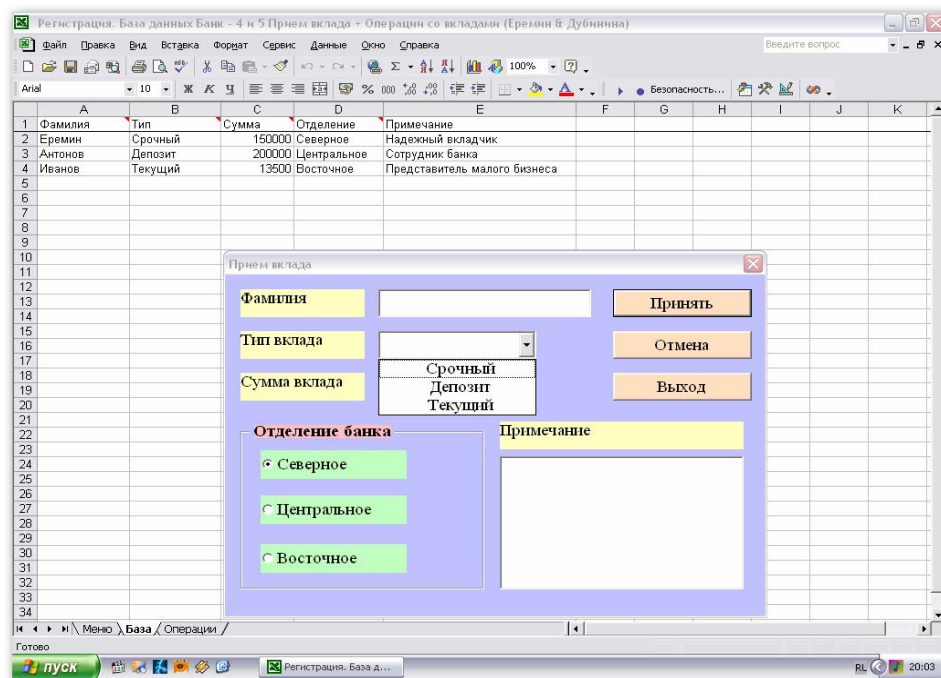


Рис. 4.10. Форма для ввода очередной записи в базу данных

На рис. 4.11. приведен образец заполненной записями базы данных по учету клиентов банка вместе с образцом заполненной формы.

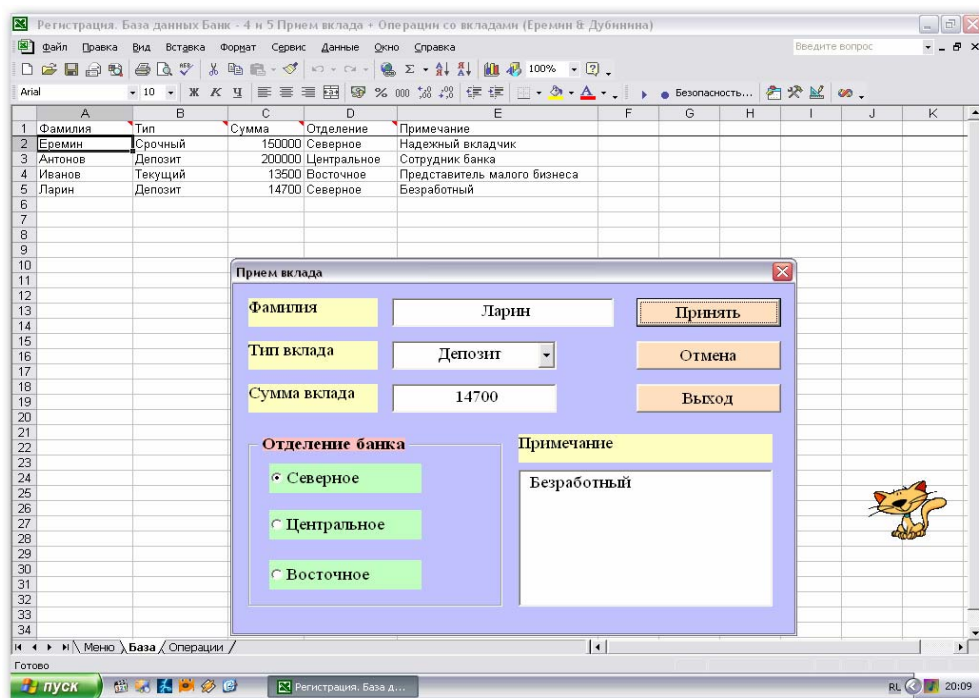


Рис. 4.11. Сформированная база данных «Регистрация вкладов»

**Задача 4.4.** Разработать приложение с пользовательским диалоговым окном для выполнения операций по вкладам клиентов банка на основе созданной базы данных «Регистрация вкладов» в задаче 4.3. База данных по учету вкладов клиентов отделений банка расположена на рабочем листе «База» Excel. Приложение реализует семь основных функций.

1. Принимает пять параметров: фамилия вкладчика, сумма, которую вкладчик берет или снимает со вклада, тип вклада, отделение банка и примечание.
2. Находит в базе данных запись по фамилии, типу вклада и названию отделения банка, где лежат деньги. Если запись найдена, то на форму выводится сумма вклада.
3. Отменяет операции по вкладу и выдает соответствующее сообщение, если не совпадает хотя бы один реквизит вкладчика.
4. Увеличивает размер вклада вкладчика на сумму, которую он хочет положить в банк на счет. Корректирует в этом случае поля записи Базы данных на рабочем листе «База».
5. Уменьшает размер вклада вкладчика на сумму, которую он хочет снять со счета в банке. Корректирует в этом случае поля записи Базы данных на рабочем листе «База».
6. Выводит исходные данные и откорректированные расчетные значения в виде записи в первую пустую строку базы данных на рабочем листе «Операции».
7. Запускает приложение с выводом формы на рабочий лист «База» Excel с помощью кнопки расположенной на рабочем листе «Операции».

Разработка приложения начинается с открытия рабочей книги, где находятся рабочие листы с именами: «Меню», «База», «Операции» и с разработки формы в виде пользовательского диалогового окна. Форма имеет имя (*Name*) «Опер», а название (*Caption*) – «Операции со вкладами». Кроме того, нужно проверить наличие базы данных на рабочем листе «База», сформированной в предыдущей задаче.

Вид формы пользователя с элементами управления для изменения данных в полях записи конкретного клиента, фамилия которого указана в форме, представлен на рис. 4.12.

#### **Программный код приложения.**

1. Код модуля запуска приложения с выводом на рабочий лист разработанного пользовательского окна.

На рабочем листе Excel создается элемент панели управления «Кнопка» панели инструментов *Visual Basic* с надписью на ней «Операции». С помощью этой кнопки будет запускаться приложение и выводиться диалоговое окно на рабочий лист «База». Для процедуры обработки события *Click* этой кнопки записывается следующий код:

```

Private Sub Операции_Click()
    Опер.Show
End Sub

```

Рис. 4.12. Форма «Опер»

## 2. Код модуля формы «Опер»

Код обработки события формы *Initialize*:

```

Private Sub UserForm_Initialize()
    With Опер
        .Сев.Value = True
        .Тип.ListRows = 3
        .Тип.List = Array("Срочный", "Депозит", _
            "Текущий")
        .Найти.SetFocus
        Application.Worksheets("База").Activate
    End With
End Sub

```

В общей области необходимо объявить вспомогательные переменные:

```
Option Explicit
Dim Фамилия As String
Dim ТипВклада As String
Dim СуммаВклада As Double
Dim Отделение As String
Dim Примечание As String
Dim Количество As Integer
Dim Номер As Integer
```

Для активных элементов формы записывается соответствующий код.  
Код процедуры обработки события *Click* кнопки «Найти»:

```
Private Sub Найти_Click()
    Worksheets("База").Activate
    Количество = Application.CountA(Active- _
        Sheet.Columns(1))
    Фамилия = Фам.Text
    ТипВклада = Тип.Value
    If Сев.Value = True Then Отделение = _
        "Северное"
    If Центр.Value = True Then Отделение = _
        "Центральное"
    If Вост.Value = True Then Отделение = _
        "Восточное"
    With ActiveSheet
        For Номер = 1 To Количество
            If .Cells(Номер, 1) = Фамилия And _
                .Cells(Номер, 2) = ТипВклада And _
                .Cells(Номер, 4) = Отделение Then _
                Exit For
        Next
        Остаток.Text = .Cells(Номер, 3)
    End With
    If Номер = Количество + 1 Then
        MsgBox "Такого счета в базе нет", _
            vbInformation
    End If
    Дата.Text = Date
End Sub
```

Код процедуры обработки события *Click* кнопки «Принять»:

```
Private Sub Принять_Click()  
    If IsNumeric(Величина.Text) = False Then  
        MsgBox "Ошибка в поле Суммы", _  
            vbInformation, "БАНК"  
        Exit Sub  
    End If  
    With ActiveSheet  
        .Cells(Номер, 3) = CStr(CDbl(.Cells _  
            (Номер, 3)) + CDbl(Величина.Text))  
    End With  
    Worksheets("Операции").Activate  
    Количество = Application.CountA(Active _  
        Sheet.Columns(1)) + 1  
    With ActiveSheet  
        .Cells(Количество, 1) = Фам.Text  
        .Cells(Количество, 2) = Дата.Text  
        .Cells(Количество, 3) = Получ.Text  
        .Cells(Количество, 4) = Тип.Text  
        .Cells(Количество, 6) = Величина.Text  
        .Cells(Количество, 7) = Отделение  
        .Cells(Количество, 8) = Прим.Text  
        .Cells(Количество, 9) = Остаток.Text + _  
            CDbl(Величина.Text)  
    End With  
End Sub
```

Код процедуры обработки события *Click* кнопки «Выдать»:

```
Private Sub Выдать_Click()  
    If IsNumeric(Величина.Text) = False Then  
        MsgBox "Ошибка в поле Суммы", _  
            vbInformation, "БАНК"  
        Exit Sub  
    End If  
    With ActiveSheet  
        .Cells(Номер, 3) = CStr(CDbl(.Cells _  
            (Номер, 3)) - CDbl(Величина.Text))  
    End With  
    Worksheets("Операции").Activate  
    Количество = Application.CountA(Active _  
        Sheet.Columns(2)) + 1  
    With ActiveSheet
```

```

        .Cells(Количество, 1) = Фам.Text
        .Cells(Количество, 2) = Дата.Text
        .Cells(Количество, 3) = Получ.Text
        .Cells(Количество, 4) = Тип.Text
        .Cells(Количество, 5) = Величина.Text
        .Cells(Количество, 7) = Отделение
        .Cells(Количество, 8) = Прим.Text
        .Cells(Количество, 9) = Остаток.Text _
            - CDb1(Величина.Text)
    End With
End Sub

```

Код процедуры обработки события *Click* кнопки «Отменить»:

```

Private Sub Отменить_Click()
    With ActiveSheet
        '.Cells(Номер, 3) = Остаток.Text
    End With
    Worksheets("Операции").Activate
    With ActiveSheet
        .Cells(Количество, 1) = ""
        .Cells(Количество, 2) = ""
        .Cells(Количество, 3) = ""
        .Cells(Количество, 4) = ""
        .Cells(Количество, 6) = ""
        .Cells(Количество, 7) = ""
        .Cells(Количество, 8) = ""
        .Cells(Количество, 9) = ""
    End With
End Sub

```

Код процедуры обработки события *Click* кнопки «Выход»:

```

Private Sub Выход_Click()
    Worksheets("Меню").Activate
    End
End Sub

```

**Порядок и демонстрация выполнения приложения следующие:**

1. Запустите приложение путем нажатия на кнопку «Операции», расположенную на рабочем листе «Операции» табличного процессора *Excel* (рис. 4.13).



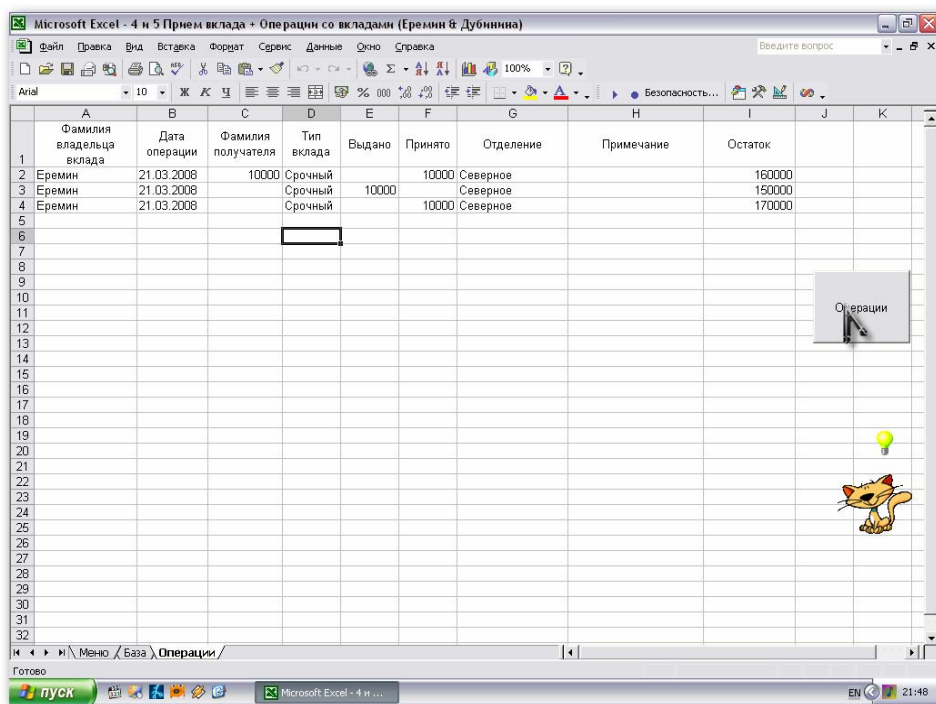


Рис. 4.13. Рабочий лист Excel с кнопкой запуска приложения

2. На рабочем листе откроется форма (рис. 4.14). Заполните поля формы данными по образцу (рис. 4.15) для поиска информации о клиенте в базе данных и нажмите кнопку «Найти».

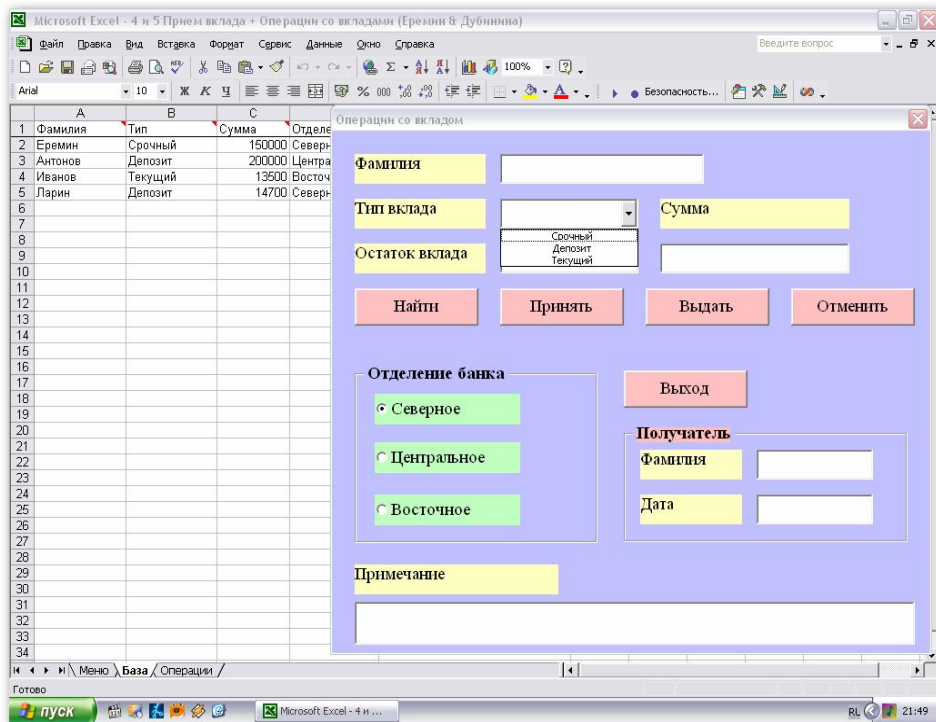


Рис. 4.14. Форма для ввода очередной записи в базу данных

3. Если запись на данного клиента будет найдена в базе данных, то остаток вклада на счете клиента будет выведен в поле «Остаток вклада» (рис. 4.15).

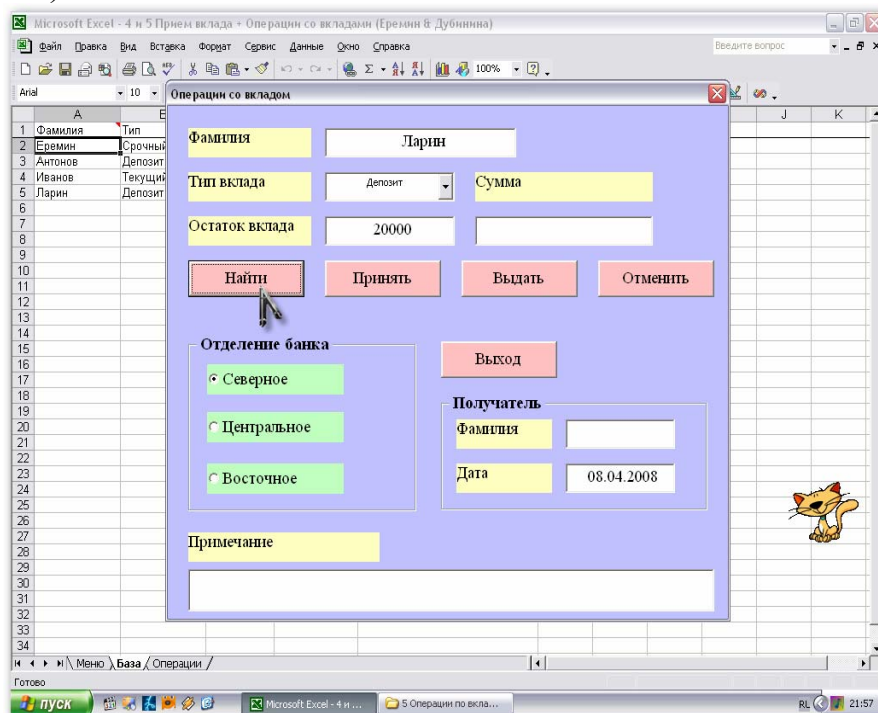


Рис. 4.15. Сформированная база данных «Регистрация вкладов»

4. Введите сумму, которую необходимо принять на счет от вкладчика, и нажмите на кнопку «Принять». На рабочем листе «Операции» Excel появляется строка с информацией о проведенной операции. Вид рабочего листа приведен на рис. 4.16.

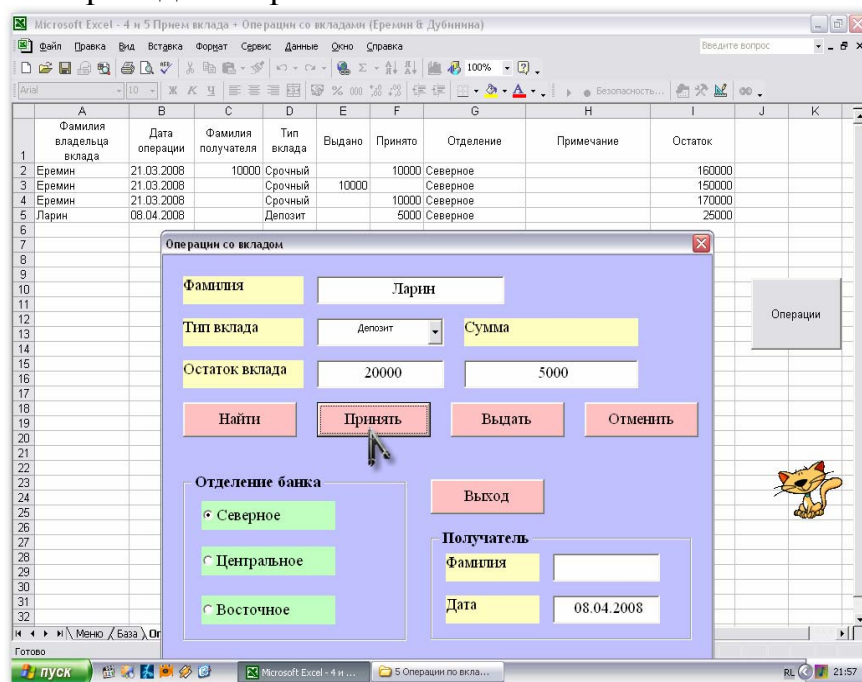


Рис. 4.16. База данных в случае добавления суммы ко вкладу клиента

5. Введите сумму, которую необходимо снять со счета вкладчика, и нажмите на кнопку «Выдать». На рабочем листе «Операции» Excel появляется строка с информацией о проведенной операции. Вид рабочего листа приведен на рис. 4.17.

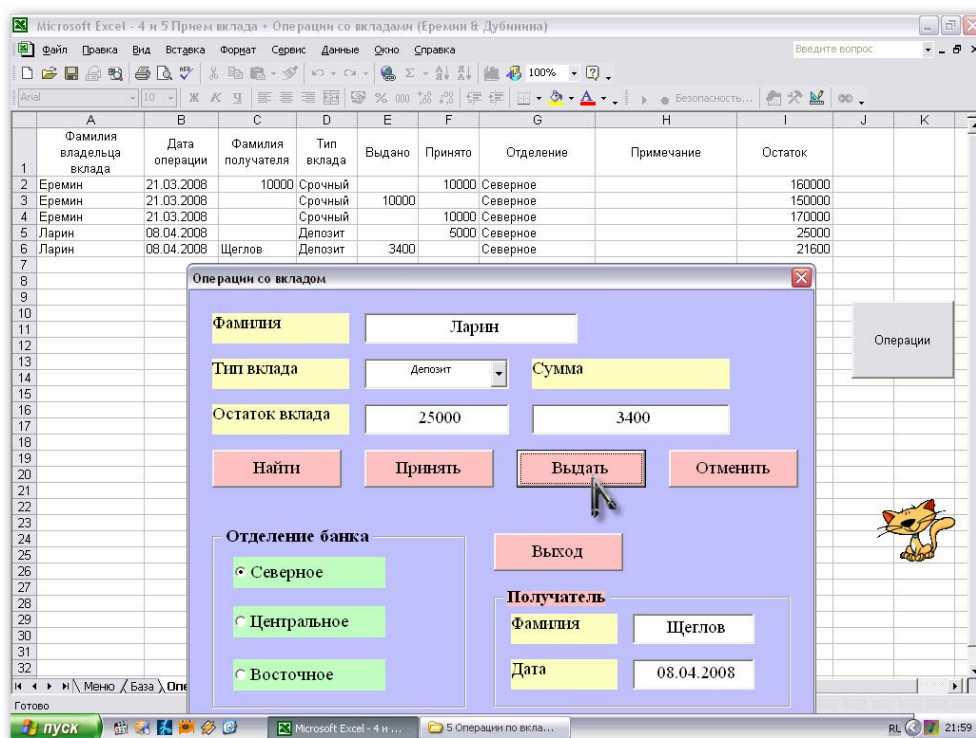


Рис. 4.17. База данных в случае снятия суммы со вклада клиента

## **Задания для самостоятельной работы по созданию приложений с использованием пользовательских форм**

**Задание 1.** Разработать приложение с диалоговым окном «Регистрация клиентов гостиницы», в котором расположить следующие элементы управления:

1. текстовые поля для ввода фамилии, имени, отчества гостя, даты рождения, места рождения, паспортных данных, места работы и должности;
2. счетчик, который управляет вводом продолжительности проживания в гостинице
3. раскрывающийся список, в котором выводятся четыре типа номеров: одноместный, двухместный, полулюкс и люкс, стоимость проживания в которых соответственно равна 850, 750, 1500, 2000 руб. в сутки;
4. если постоялец завтракает в гостинице, то суточная оплата возрастает на 150 руб.;
5. после ввода данных и нажатия на кнопку *ОК* в поле «*Стоимость проживания*» формы выводится суммарная стоимость проживания клиента, а все данные из диалогового окна и расчетные данные вводятся в базу данных, создаваемую на рабочем листе.

**Задание 2.** Разработать приложение с диалоговым окном «Информация о перевозке товаров», в котором расположить следующие элементы управления:

1. текстовое поле для ввода наименования товара;
2. поле со списком для выбора клиента, которому поставляется товар. Поле со списком в программном коде должно быть заполнено данными, взятыми с рабочего листа Excel.
3. несколько переключателей объединенных рамкой, с помощью которых определяется пункт доставки товара;
4. несколько переключателей объединенных рамкой, с помощью которых определяется метод доставки товара (поезд, самолет, грузовик);
5. два флажка, которые определяют вид товара (скоропортящийся, хрупкий);
6. две кнопки «*Отправить в БД*» и «*Отмена*», одна из которых подтверждает ввод записи в базу данных, а другая отменяет ввод.
7. после ввода данных и нажатия на кнопку «*Отправить в БД*» данные из диалогового окна вводятся в базу данных, создаваемую на рабочем листе.

## Литература

1. Гарнаев А. Использование MS Excel и VBA в экономике и финансах / А. Гарнаев. – Спб.: ВHV; Санкт-Петербург, 1999. – 336 с.
2. Демидова Л.А. Программирование в среде Visual Basic for Applications: Практикум. / Л.А. Демидова, А.Н. Пылькин. – М.: Горячая линия – Телеком, 2004. – 175 с.
3. Ищенко В.А. Visual Basic. Практика (часть 2): Учебно-методическое пособие / В.А. Ищенко, Е.К. Нагина. – Воронеж, ВГУ, 2005.
4. Нагина Е.К. Visual Basic. Теория (часть 1): Учебно-методическое пособие / Е.К. Нагина, В.А. Ищенко. – Воронеж, ВГУ, 2004.
5. Нагина Е.К. Visual Basic for Applications. Теория : учебно-метод. пособие по информатике для вузов / Е. К. Нагина, В. А. Ищенко. – Воронеж : Издательско-полиграфический центр воронежского государственного университета, 2008. – 88 с.
6. Олбрайт К. Моделирование с помощью Microsoft Excel и VBA: разработка систем поддержки решений / К. Олбрайт. М.: Издательский дом “Вильямс”, 2005. – 672 с.
7. Электронный каталог научной библиотеки Воронежского государственного университета. – (<http://www.bib.vsu.ru/>).
8. Социальные и гуманитарные науки. Экономика: Библиографическая база данных. 1986-2001 гг. / ИНИОН РАН. – М., 2002. – (CD. ROM).

## Содержание

1. Создание функций пользователя .....	3
Задания для самостоятельной работы по созданию функций пользователя .....	9
2. Создание процедур .....	11
Задача 2.1 .....	11
Задача 2.2 .....	12
Задача 2.3 .....	13
Задача 2.4 .....	15
Задача 2.5 .....	19
Задача 2.6 .....	22
Задача 2.7 .....	26
Задача 2.8 .....	30
Задания для самостоятельной работы по созданию процедур с использованием управляющих структур .....	33
Задания для самостоятельной работы по созданию процедур для обработки массивов .....	40
3. Создание приложений с использованием пользовательских форм .....	44
Задача 3.1 .....	44
Задача 3.2 .....	50
4. Создание приложений для формирования и заполнения баз данных (списков) в Excel .....	56
Задача 4.1 .....	56
Задача 4.2 .....	65
Задача 4.3 .....	70
Задача 4.4 .....	77
Задания для самостоятельной работы по созданию приложений с использованием пользовательских форм .....	85
Литература .....	86

*Учебное издание*

**Нагина Елена Константиновна,  
Ищенко Валентина Александровна**

**VISUAL BASIC FOR APPLICATIONS  
ПРАКТИКА**

**Учебно-методическое пособие  
по информатике для вузов**

Подписано в печать 03.03.2009. Формат 60×84/16. Усл. печ. л. 5,2.  
Тираж 310 экз. Заказ 226.

Издательско-полиграфический центр  
Воронежского государственного университета.  
394000, г. Воронеж, пл. им. Ленина, 10. Тел. (факс) +7 (4732) 598-026  
<http://www.ppc.vsu.ru>; e-mail: [pp\\_center@ppc.vsu.ru](mailto:pp_center@ppc.vsu.ru)

Отпечатано в типографии Издательско-полиграфического центра  
Воронежского государственного университета.  
394000, г. Воронеж, ул. Пушкинская, 3