

**МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РФ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ**

Кафедра информационных технологий и моделирования

**О.А. Карасева**

## **Программирование в MS Office**

Учебно-методические указания по проведению лабораторно-практических

занятий для студентов в среде VBA

направления 080500.62- Бизнес-информатика

дневной, очно-заочной и заочной формы обучения

**ЕКАТЕРИНБУРГ**

**2015**

## Введение

**Microsoft Excel** – первое средство разработки информационных систем, совмещающее в себе преимущества как электронных таблиц, так и средств визуального программирования. Действительно Excel – это первая электронная таблица, предлагающая визуальное средство разработки – язык **Visual Basic for Applications**, более известное как VBA. VBA – это фактически результат внедрения первого визуального средства разработки Microsoft – языка Visual Basic – в электронную таблицу Excel (и другие приложения Microsoft Office). Он обеспечивает полный доступ из программ пользователя ко всем функциям анализа данных, имеющимся в электронной таблице.

Если мы рассмотрим Excel, как средство разработки информационных систем, то весьма важно понять, что Excel – это не просто электронная таблица. Excel содержит также мощную библиотеку, которая включает свыше ста объектов для обработки данных. С помощью VBA разработчики могут объединять объекты Excel с целью создания развитых информационных систем. Поскольку объекты Excel имеют широкую область применения, разработчики могут создавать гибкие системы так, что каждая система удовлетворяет требованиям многих пользователей. Excel также поддерживает механизм OLE и является как сервером, так и клиентом Animation. Такая поддержка механизма OLE позволяет разработчикам легко интегрировать объекты Excel в системы, созданные на базе других инструментальных средств.

### Лабораторная работа 1. Свойства и методы объектов Range и Selection

Программа VBA в Excel состоит из подпрограмм VBA. Существует два типа подпрограмм: процедуры и функции. Подпрограмма - это основной строительный блок приложений VBA, т. е. отдельный сегмент кода, содержащий несколько операторов VBA.

Процедура создания программы в VBA состоит в следующем. Откройте в Excel новую пустую рабочую книгу и загрузите редактор Visual Basic (VBE). для этого нужно выбрать команду *Сервис, Макрос, Редактор Visual Basic*, после чего на экране появится окно Редактора VBE, которое имеет следующие основные компоненты:

1. Окно проекта – содержит иерархический список элементов пользовательского проекта VBA. Это могут быть объекты Excel высокого уровня, такие как рабочие листы, диаграммы и объект ThisWorkbook (ссылка на рабочую книгу, в которой находится проект). Список может также включать модули, классы модулей

- (специальный тип модуля, позволяющий разработчику определять собственные объекты) и пользовательские диалоговые окна.
2. Окно свойств – позволяет просматривать и устанавливать свойства различных объектов проекта.
  3. Окно редактора кода – место ввода и редактирования кода VBA. Два раскрывающихся списка помогают ориентироваться в программе.
  4. Окно редактора пользовательских форм – позволяет создавать пользовательские диалоговые окна, помещать на них элементы управления ActiveX и тестировать их.
  5. Окно просмотра объектов – помогает находить и использовать объекты, необходимые пользовательским приложениям. Для вывода окна можно использовать команду *Вид, Просмотр объектов*.
  6. Окно отладки, окно локальных переменных и окно контрольных значений – все эти окна помогают при отладке программы. Каждое можно вывести с помощью команд меню *Вид*.

На экране не будет некоторых элементов окна, пока нет модулей с кодом.

**Вставка модуля VBA** осуществляется командой *Вставка, Модуль*. После чего появится новый пустой модуль. Имя нового модуля отобразится и в окне проекта.

**Ввод и запуск подпрограмм VBA.** Подпрограмма VBA должна начинаться с ключевого слова *Sub* и заканчиваться ключевым словом *End Sub*.

**Символ продолжения строки в VBA.** В языке VBA символ ( *\_* ) – пробел с подчеркиванием, говорит о том, что код будет продолжаться на следующей строке. Одна строка кода может содержать до 1024 символов, но разбиение строки может сделать код более читаемым.

В данном методическом указании объекты Excel: *Application, Workbook, Worksheets* и *Range*.

Все объекты Excel имеют так называемые "свойства" и "методы", а VBA является инструментом, посредством которого можно управлять объектами Excel с помощью их свойств и методов.

Используя VBA можно выполнить два типа действий со свойствами. Можно получать свойство (т. е. получать ранее установленное значение свойства) и можно устанавливать свойство (т. е. устанавливать значение этого свойства). При выполнении как того, так и другого действия нужно указывать имя объекта и название свойства, отделяя их друг от друга точкой (.).

В VBA синтаксис вызова методов отличается от синтаксиса установки и получения свойств. Все, что необходимо для вызова метода – это указание объекта и метода. Кроме этого, большинство методов в Excel имеют набор

аргументов, т. е. дополнительных блоков информации, которые служат для уточнения того, как должен быть выполнен метод.

Объект Range представляет собой адресуемую ячейку, строку, столбец или несколько выделенных ячеек. Объект Selection является результатом выбора ячейки или диапазона с помощью метода Select либо результатом применения свойства Selection. К нему применимы свойства и методы объекта Range.

Применение свойств и методов объектов Range и Selection рассмотрим на примере.

### Задание 1.

1. Откроем приложение Excel.
2. Перейдем в режим конструктора.
3. На рабочем листе поместим кнопку (рис.1.1).
4. В окне свойств (правой кнопкой мыши) назначим кнопке имя (*Name*) – *Щелкнуть, надпись (Caption) – Нажми.*
5. Назначим для этой кнопки макрос. Правая кнопка мыши, *Исходный текст.*
6. В окне редактирования запишем программный код, приведенный ниже.

```
Private Sub Щелкнуть_Click()  
Range("A1").Value = "Свойства и методы объекта Range"  
'выделяем ячейку A1  
Range("A1").Select  
With Selection  
MsgBox "Адрес активной ячейки - " & .Address()  
MsgBox "Значение активной ячейки - " & .Value  
End With  
'Изменяем параметры шрифта активной ячейки  
With Selection  
.Font.Size = 16  
End With  
Range("A2").Value = 2  
Range("B2").Value = 4  
Range("C2").Formula = "=A2^B2"  
'количество строк в диапазоне  
MsgBox "Количество строк в области A1:C2=" & Range("A1:C2").Rows.Count  
'очисть диапазон  
Range("A1:C2").Clear  
Cells(1, 1) = "Вы познакомились с методами и свойствами объекта Range"  
End Sub
```

7. Запустим код на выполнение. Проследим, какие действия выполняет программа. Сопоставим команды программы и действия.

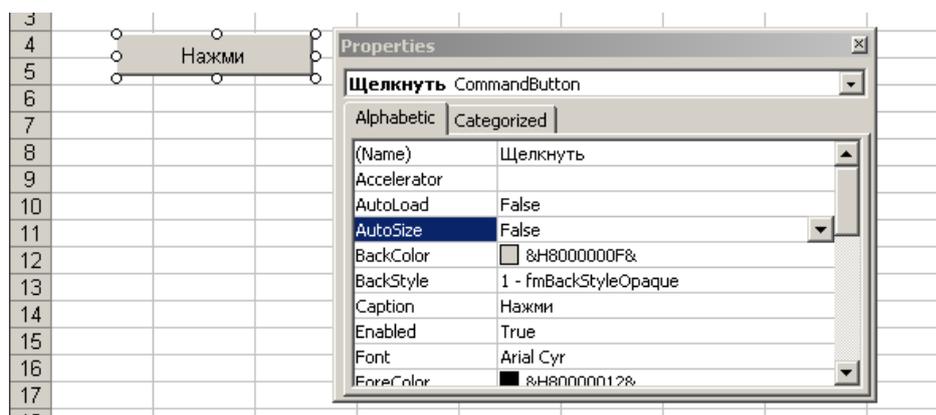


Рис.1. 1

## Задание 2. Форматирование ячеек рабочего листа

Пусть на рабочем листе имеется таблица (рис.2). Количество строк таблицы заранее неизвестно. Необходимо создать диалоговое окно, которое позволит отформатировать таблицу. Для создания приложения выполните следующие действия.

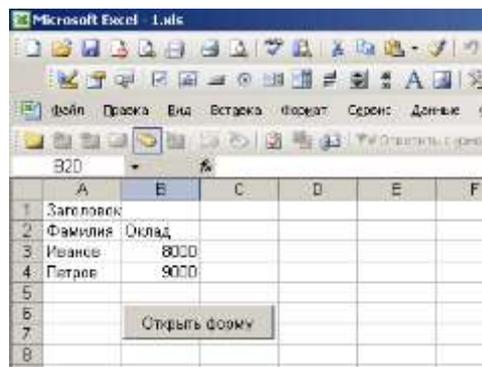


Рис. 1.2

1. Создайте форму с элементом *Надпись*, двумя кнопками и элементом *RefEdit* (Рис. 3).
2. Установите необходимые свойства элементов.

Таблица 1.1

Элемент	Имя элемента (Name)	Надпись (Caption)
Метка	Label1	Диапазон ячеек для форматирования
RefEdit	RefEdit1	
Кнопка	КнопкаФорматировать	Форматировать
Кнопка	КнопкаУбратьФормат	Убрать форматирование

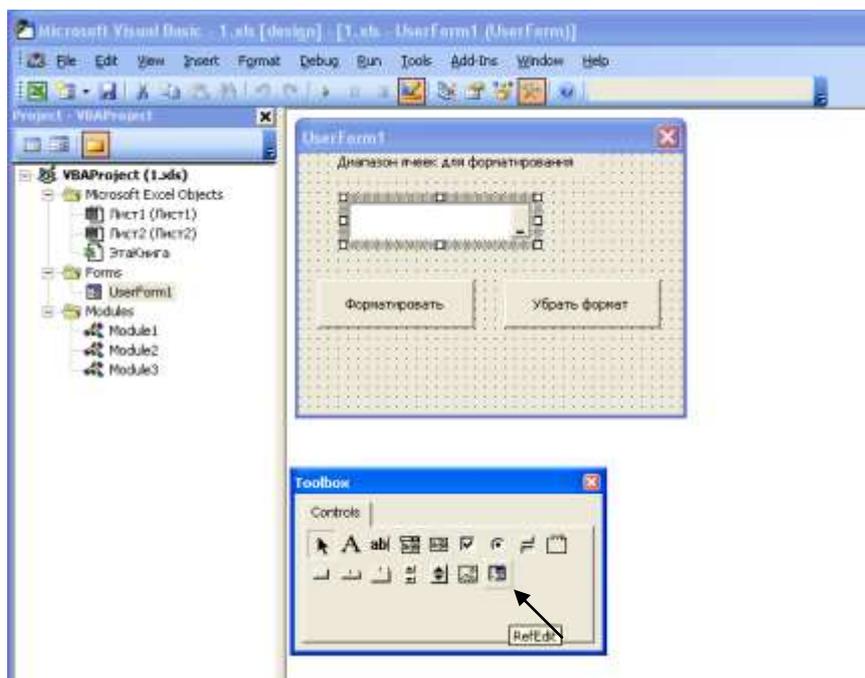


Рис. 1.3.

3. В общей области окна редактирования кода программы для формы объявите переменные:

```
Dim myR As Range
Dim Head As Range
Dim Title As Range
Dim c As Integer
Dim r As Integer
```

4. Создайте процедуры обработки события Click для кнопки КнопкаФорматировать:

```
Private Sub КнопкаФорматировать_Click()
'Присваиваем переменной myR значение (ссылка на диапазон берется
'из элемента RefEdit
Set myR = Range(RefEdit1.Text)
r = myR.Rows.Count
c = myR.Columns.Count
'ссылка на диапазон, содержащий название таблицы
Set Head = Range(myR.Cells(1, 1), Cells(1, c))
'ссылка на диапазон, содержащий заголовки столбцов таблицы
Set Title = Range(myR.Cells(2, 1), Cells(2, c))
'выделение ячеек, содержащих название таблицы
Head.Select
'Объединение ячеек и выравнивание заголовка по центру
Selection.HorizontalAlignment = xlCenterAcrossSelection
'форматирование названия таблицы
With Head.Font
.Name = "Arial Cyr"
.FontStyle = "полужирный курсив"
```

```
.Size = 14
.ColorIndex = 3
End With
'выравнивание названий колонок по центру
Title.HorizontalAlignment = xlCenter
'форматирование названий колонок
With Title.Font
.Name = "Arial Cyr"
.FontStyle = "полужирный"
.Size = 10
End With
End Sub
Private Sub CommandButton1_Click()
UserForm1.Show
End Sub
```

5. Создайте процедуры обработки события *Click* для кнопки КнопкаУбратьФормат:

```
Private Sub КнопкаУбратьФормат_Click()
'установить ссылку на объект
Set myR = Range(RefEdit1.Text)
'Очистить формат
myR.ClearFormats
End Sub
```

6. Для вызова программы разместите кнопку CommandButton1 на рабочем листе с надписью *Открыть форму* (рис.1.4).

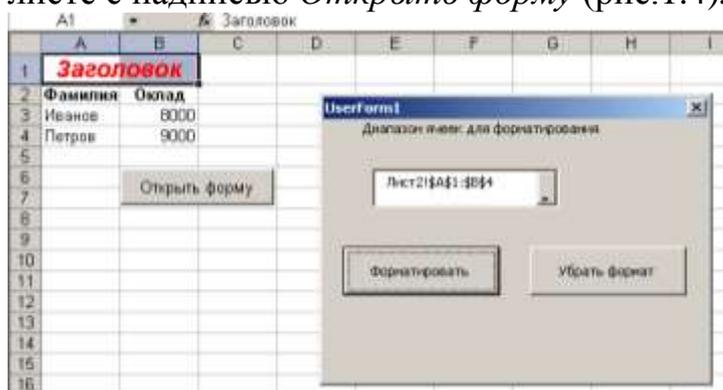


Рис.1.4.

7. Напишите код для этой кнопки (двойной щелчок по кнопке):

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

8. Запустите макрос на выполнение. Результат работы приложения представлен на рис.1.4. Сопоставьте команды программы и выполняемые ею действия.

## Лабораторная работа 2. Разработка приложения для расчета по ипотечному кредиту

Пусть исходными данными для расчета по ипотечному кредиту являются цена приобретенного имущества, величина первоначального взноса, выраженная в процентах, годовая ставка (в процентах), срок погашения ссуда (в месяцах или годах).

Требуется разработать программу с диалоговым окном, вычисляющую: величину периодических выплат, общую сумму выплат, общую сумму комиссионных, величину начального взноса. Для этого воспользуемся функцией `Pmt` (аналог ПЛТ из категории финансовых). Ее синтаксис:

`Pmt (Ставка, Кпер, Нз, [Бз [, Тип]])`

где:

Ставка – процентная ставка за период;

Кпер- количество периодов выплат для погашения кредита;

Нз – сумма, которую нужно выплатить;

Бз – баланс наличности, который нужно достигчь после последней выплаты;

Тип – признак того, когда производится выплата (0 - в конце периода; 1 – в начале периода).

Выполним разработку формы пользователя.

1. Создайте форму `Ipoteka` (рис.2.1). На рис.2.2 представлен результат работы приложения. Свойства элементов управления формы и обрабатываемых событий представлены в табл. 2.1.

# Электронный архив УГЛТУ

Расчет по ипотечному кредиту

Стоимость имущества:

Первый взнос %:  Label3

В начале периода:

Период выплат:  
 Месяц  
 Год

Месяцев:

Величина платежа: Label5

Полная сумма выплат: Label7

Комиссионные: Label9

Ставка %:

Buttons: ,

Рис. 2.1

Расчет по ипотечному кредиту

Стоимость имущества:

Первый взнос %:  161520

В начале периода:

Период выплат:  
 Месяц  
 Год

Месяцев:

Величина платежа: 1185

Полная сумма выплат: 426600

Комиссионные: 265080

Ставка %:

Buttons: ,

Рис. 2.2

Таблица 2.1.

Элемент	Присваиваемое имя	Назначение	Свойства	Событие
UserForm	Ipoteka		Caption-Расчет по ипотечному кредиту	Initialize: 1. Установка границ элемента Счетчик 2. Определение значений для элемента Список 3. Удаление текста из надписей, в которые выводится результат
TextBox1	TextBox1	Цена имущества		Change: Записывает свое значение в элемент Счетчик
TextBox2	TextBox2	Величина первого взноса		
TextBox3	TextBox3	Количество периодов выплаты		
SpinButton	SpinButton	Счетчик (для удобства установки количества периодов)	Счетчик	Change: Записывает свое значение в элемент TextBox3
OptionButton	OptionButton	Тип периода(месяц)	Переключатель	
OptionButton	OptionButton	Тип периода (Год)	Переключатель	
ToggleButton1	ToggleButton	Для указания, когда производится платеж (в начале, конце)	Выключатель	Click:
ListBox1	ListBox1	Годовая процентная	Список	

		ставка		
Label	Label	Величина платежа	Метка	
Label	Label	Полная сумма выплат	Метка	
Label	Label	Комиссионные	Метка	
CommandButton1	CommandButton1	Для управления расчетом	Кнопка	Click
CommandButton2	CommandButton2	Выход	Кнопка	Click

2. Для процедуры обработки события Initialize запишите код:

```
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
    Label11.Caption = "Месяцев"
    ToggleButton1.Caption = "В начале периода"
    Label3.Caption = ""
    Label5.Caption = ""
    Label7.Caption = ""
    Label9.Caption = ""
End Sub
```

3. Для процедуры обработки события Change элемента TextBox3 запишите код:

```
Private Sub TextBox3_Change()
    If TextBox3.Text = "" Then Exit Sub
    SpinButton1.Value = CInt(TextBox3.Text)
End Sub
```

4. Для процедуры обработки события Change элемента Счетчик запишите код:

```
Private Sub SpinButton1_Change()
    TextBox3.Value = SpinButton1.Value
End Sub
```

5. Для процедуры обработки события Click кнопки CommandButton1  
(Вычислить)

```
Private Sub CommandButton1_Click()
    Dim Ставка As Double
    Dim Кпер As Integer
    Dim Тип As Byte
```

```
Dim Hz As Double
Hz = TextBox1.Value - TextBox1.Value * TextBox2.Value / 100
Label3.Caption = Hz
Ставка = ListBox1.Value
Ставка = Ставка / 100
If OptionButton1.Value = True Then
Ставка = Ставка / 12
End If
Кпер = TextBox3.Value
If ToggleButton1 = True Then Тип = 1 Else Тип = 0
'величина патежа
Label5.Caption = Int(Pmt(Ставка, Кпер, -Hz, , Тип))
'Полная сумма выплат
Label7.Caption = Label5.Caption * Кпер
'Комиссионные
Label9.Caption = Label7.Caption - Label3.Caption
End Sub
```

6. Для процедуры обработки события Click кнопки CommandButton2  
(Завершить)

```
Private Sub CommandButton2_Click()
End
End Sub
```

7. На рабочем листе Лист1 поместите кнопку формы и напишите код для  
процедуры обработки события Click кнопки CommandButton1 (Расчет по  
ипотечному кредиту)

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

8. Перейдите на рабочий лист Лист1 и щелкните кнопку (Расчет по  
ипотечному кредиту). На экране появится форма, приведенная на рис.  
Введите данные для расчета и проверьте корректность вычислений. В  
случае необходимости выполните отладку программы.

## Лабораторная работа 3. Разработка приложения для расчета амортизации основных фондов

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

В программе предусмотрим проверку корректности вводимых исходных данных и выдачу сообщений оператору в случае обнаружения какой-либо ошибки.

Метод. Для расчета величины амортизации за период эксплуатации используем встроенную функцию VBA, имеющую имя DDB. Ее синтаксис: *DDB(ПервичнаяСтоимость, ОстаточнаяСтоимость, ВремяАмортизации, ПериодРасчета, Кратность)*

*Кратность* –указывает метод, применяемый для расчета (стандартный, если параметр опущен или метод кратного расчета, если указывается кратность, выраженная числом).

Технология работы

1. Запустим редактор VBA.
2. Выполним команду для вставки формы.
3. Расположим в окне редактирования формы управляющие элементы (рис. 3.1) и установим свойства элементам управления в соответствии с рисунком.

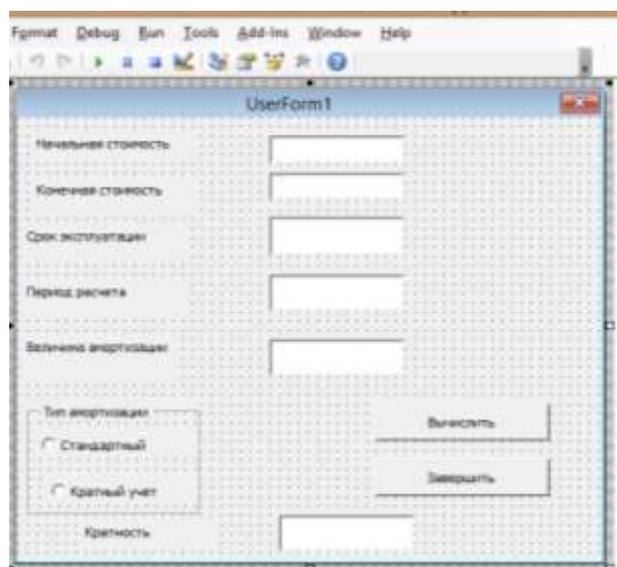


Рис.3.1. Форма

4. Создадим процедуру 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
```

5. Запишем процедуру расчета для кнопки Вычислить:

```
Private Sub CommandButton1_Click()  
If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text = "" Then  
    MsgBox "Нет данных для расчета"  
    Exit Sub  
End If  
'Преобразуем данные к числовому формату  
ПервичнаяСтоимость = CDbI(TextBox1.Text)  
ОстаточнаяСтоимость = CDbI(TextBox2.Text)  
ВремяАмортизации = CInt(TextBox3.Text)  
ПериодРасчета = CInt(TextBox4.Text)  
'Проверка корректности данных  
If ПервичнаяСтоимость < ОстаточнаяСтоимость Then  
    MsgBox "Ошибка. Остаток больше начальной стоимости"  
    TextBox1.SetFocus  
    Exit Sub  
End If  
If ВремяАмортизации < ПериодРасчета Then  
    MsgBox "Ошибка в сроке амортизации"  
    TextBox3.SetFocus  
Exit Sub  
End If  
' По какому методу производить расчет?  
If OptionButton1.Value = True Then  
    Признак = True  
Else  
    Признак = False  
End If  
If Признак = True Then  
    ВеличинаАмортизации = DDB(ПервичнаяСтоимость, ОстаточнаяСтоимость, ВремяАмортизации, ПериодРасчета)  
Else  
    TextBox6.SetFocus  
    Кратность = CInt(TextBox6.Text)  
    ВеличинаАмортизации = DDB(ПервичнаяСтоимость, ОстаточнаяСтоимость, ВремяАмортизации, _ ПериодРасчета, Кратность)  
End If  
TextBox5.Text = CStr(ВеличинаАмортизации)  
End Sub
```

6. Добавим в общий раздел (General) описания глобальных переменных (они должны быть доступны во всех процедурах):

```
Option Explicit  
Dim ПервичнаяСтоимость As Double  
Dim ОстаточнаяСтоимость As Double
```

```
Dim ВремяАмортизации As Integer  
Dim ПериодРасчета As Integer  
Dim Кратность As Integer  
Dim Признак As Boolean  
Dim ВеличинаАмортизации As Double
```

7. Запишите код для кнопки *Завершить*:

```
Private Sub CommandButton2_Click()  
    End  
End Sub
```

8. Проверьте правильность работы на контрольном примере (рис.3.2)

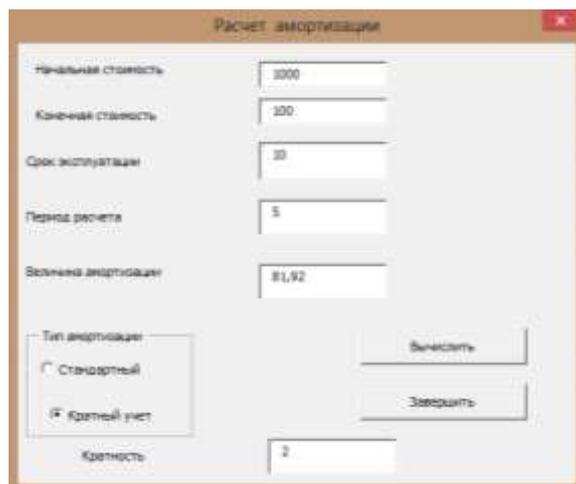


Рис.3.2. Результаты вычислений

## Лабораторная работа 4. Применение функций Excel в коде VBA

Цель работы: создать приложение для расчета суммы, которую нужно положить на счет с указанной процентной ставкой, на указанный период времени без учета капитализации процентов, чтобы получить желаемую сумму. Расчет первоначальной суммы вклада будем считать по формуле:  $P=S/(1+i*T)$ ,

Где

- S- величина вклада желаемая;
- P- исходное значение суммы вклада;
- i- номинальная процентная ставка;
- T- срок в мес.

Проект 4. 1.

В этом проекте данные будем вводить через форму.  
Технология работы (теперь не так подробно!)

1. Создайте форму (рис.4.1). Имена управляющих элементов найдите в коде кнопки `CommandButton1_Click()`.

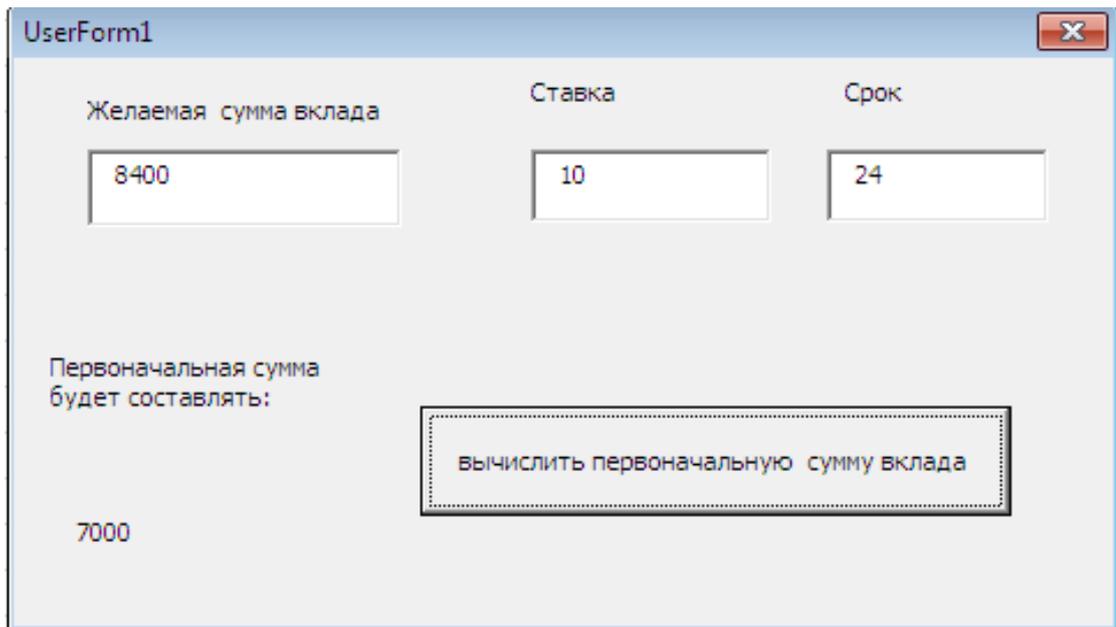


Рис.4.1 Вид приложения в режиме отображения результатов

2. Создайте в редакторе кода модуль (Вставка, Модуль).
3. Создайте в редакторе кода функцию для вычисления первоначальной суммы (Вставка, Функция).

```
Public Function СуммаВклада(суммаВклада_ж As Long, ставка As Double,
срок As Integer) As Double
'вычисление значения исходной суммы
' по формуле  $p=s/(1+iT)$ 
СуммаВклада = суммаВклада_ж / (1 + ставка * срок / 12)
End Function
```

4. Создайте в редакторе кода процедуру для расчета суммы, которая будет вызывать ранее созданную функцию:

```
Private Sub CommandButton1_Click()
Dim lsSum1 As Long, stavka1 As Double, srok1 As Integer, res1 As Double
lsSum1 = lsSum.Value
stavka1 = stavka.Value / 100
srok1 = srok.Value
res1 = СуммаВклада(lsSum1, stavka1, srok1)
res.Caption = Str(res1)
End Sub
```

5. Запустите приложение с цифрами контрольного примера (рис.4.1). Если не получен правильный результат, откорректируйте проект.

Проект 4. 2.

В этом проекте данные будем вводить в таблицу Excel, затем читать в форму, далее как в Проекте 4.1 (рис.4.2). Результат необходимо вывести из формы в столбец D. Чтение следует организовать в цикле, меняя номер строки.

Проект выполнить самостоятельно, используя опыт предыдущих проектов.

	A	B	C	D	E	F	G
1	ставка	срок	Необходимая сумма в конце	Сумма вклада	Контрольный результат		
2	10	2	101667		100000,33		
3	10	24	8400		7000		
4							

Рис.4.2. Исходные данные для чтения в форму

## Лабораторная работа 5. Разработка программы для создания базы данных Страхование

Требуется разработать программу с диалоговым окном для создания базы данных по учету страхователей и видов страхования. В базе данных должны храниться сведения:

- ФИО страхователя;
- Адрес (город, улица, дом, квартира);
- Пол;
- Срок страхования;
- Вид страхования.

Разработка формы пользователя. Вид формы Страхование представлен на рис. Для удобства ввода данных кроме элементов *Поле* в форме предусмотрены: элемент *Счетчик* для установки срока страхования, элемент поле со списком для ввода вид страхования.

Выполните действия:

1. Создайте форму пользователя и разместите в ней элементы управления в соответствии с рис. Назначьте элементам свойства (табл.5.1). Элементу *Счетчик* назначьте свойства недоступно (`Enabled=False`) и невидимо (`Visible = False`).
2. Элементам управления присвойте свойства и имена согласно табл.5.1.
3. Рабочему листу с именем «Лист2» присвойте имя «База данных».
4. На рабочем листе с именем «Лист1» поместите элемент *Кнопка*, а на ней – надпись «Страхование».

5. Для процедуры обработки события *Click* этой кнопки запишите код, приведенный ниже:

```
Private Sub CommandButton1_Click()
'Инициализация формы страхование
Страхование.Show
End Sub
```

6. Для заголовков столбцов базы данных создайте процедуру Sub пользователя, которая будет вызываться из процедуры обработки события формы Initialize. Для этого выполните команду меню *Вставка – Процедура* и присвойте ей имя *ЗаголовокРабочегоЛиста*. Процедура выполняет следующие действия:

- проверяет, заполнена ли первая строка – строка заголовков столбцов (по значению ячейки A1); если не заполнена, то не выполняет никаких действий, завершает работу и передает управление в точку вызова;
- если первая строка не заполнена, то в ячейки первой строки рабочего листа записываются заголовки граф таблицы базы данных, комментарии к ним, закрепляет первую строку и завершает работу, передавая управление в точку ее вызова.

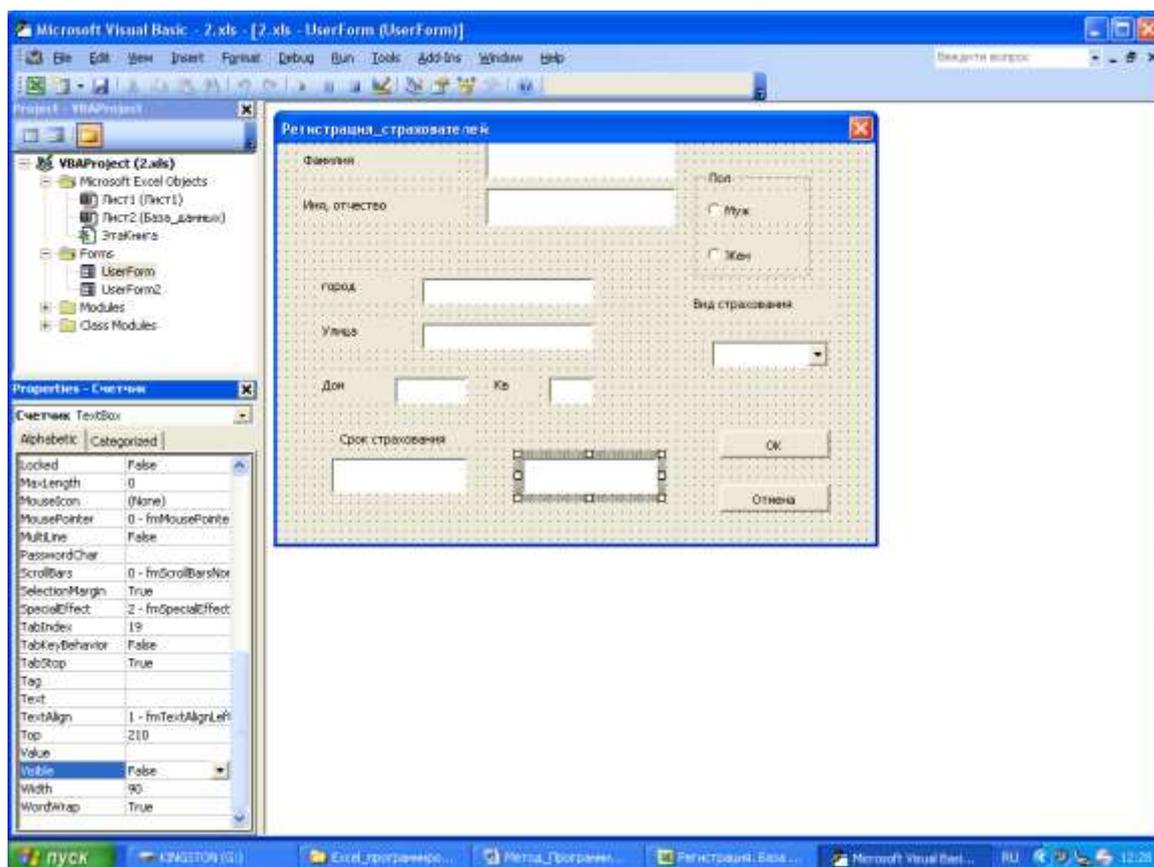


Рис.5.1.

Табл.5.1. Элементы управления формы Страхование

Элемент	Имя (Свойства)	События
Форма	UserForm	Initialize
Поле	ПолеФамилия	
Поле	ПолеИмяОтчество	
Поле	ПолеГород	
Поле	ПолеУлица	
Поле	ПолеДом	
Поле	ПолеКвартира	
Поле	ПолеСрок	Change
Счетчик	Счетчик (Enabled=False, Visible = False)	Change
Список		
Переключатель	ПолМуж	
Переключатель	ПолЖен	
Кнопка	КнопкаОК, Caption=ОК	Click
Кнопка	КнопкаОтмена, Caption=Отмена	Click

7. В окне редактирования кода введите текст программы этой процедуры:

```
Private Sub ЗаголовокРабочегоЛиста()
Application.Worksheets("База_данных").Activate
If Range("A1").Value = "Фамилия" Then
Range("A2").Select
Exit Sub
End If
ActiveSheet.Cells.Clear
Range("A1:11").Value = Array("Фамилия", "Имя", "Пол", "Город", "Улица", "Дом", _
"Квартира", "Вид страхования", "Срок")
Range("2:2").Select
ActiveWindows.FreezePanels = 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,отчество 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

8. Объявите вспомогательные переменные, поместив их объявление в общей области:

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

9. В процедуру обработки события Initialize формы введите код:

```
Private Sub UserForm_Initialize()  
'Активируем рабочий лист с именем "База данных"  
Application.Worksheets("База_данных").Activate  
  
'вызываем процедуру формирования названий граф  
ЗаголовокРабочегоЛиста  
' изменяем текст в строке заголовка приложения  
Application.Caption = "Регистрация. База данных страхования"  
'Закрываем строку формул  
Application.DisplayFormulaBar = False  
'назначаем кнопке ОК клавишу Enter и всплывающую подсказку  
With UserForm.КнопкаОК  
UserForm.КнопкаОК.Default = True  
UserForm.КнопкаОК.ControlTipText = "Ввод данных в базу данных"  
End With  
With UserForm.КнопкаОтмена  
UserForm.КнопкаОтмена.Default = False  
UserForm.КнопкаОтмена.ControlTipText = "Кнопка отмены"  
End With  
'присваиваем значения элементам списка  
Список.List = Array("Жизни", "Убытков", "Имущества")  
'устанавливаем переключатель  
ПолМуж.Value = True  
End Sub
```

10. В процедуру обработки события Click кнопки ОК введите код:

```
Private Sub КнопкаОК_Click()  
'Вычисляем номер первой незаполненной строки таблицы  
'НомерСтроки = Application.CountA(ActiveSheet.Columns(1)) + 1  
НомерСтроки = Application.CountA(Worksheets("База_данных").Columns(1)) + 1  
With UserForm  
If ПолМуж.Value = True Then  
Пол = "Муж"  
Else  
Пол = "Жен"  
End If  
End With  
'Записываем в ячейки текущей строки значения полей формы
```

```
With Application.Worksheets("База_данных")
.Cells(НомерСтроки, 1).Value = ПолеФамилия
.Cells(НомерСтроки, 2).Value = ПолеИмяОтчество
.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
```

11. Для процедуры обработки события Click кнопки Отмена введите код:

```
Private Sub КнопкаОтмена_Click()
'Восстанавливаем прежний текст в строке заголовка приложения
Application.Caption = Empty
'Завершаем выполнение программы
End
End Sub
```

12. Запустите программу на выполнение, щелкнув по кнопке, ценной на рабочем листе. Проверьте работоспособность и в случае необходимости устраните ошибки.

### **Лабораторная работа 6. Разработка приложения для создания базы данных по регистрации вкладов**

#### Задание

Требуется разработать приложение для создания базы данных Excel, использующее форму пользователя для ввода данных. БД хранит информацию:

- фамилия вкладчика;
- суммы вклада;
- виды вклада;
- отделение банка;
- примечание.

Возможный вариант формы Вклад приведен на рис 6.1. В ней для ввода типа вклада помещен элемент *Поле со списком*, а для указания отделения банка в одном контейнере три элемента *Переключатель*.

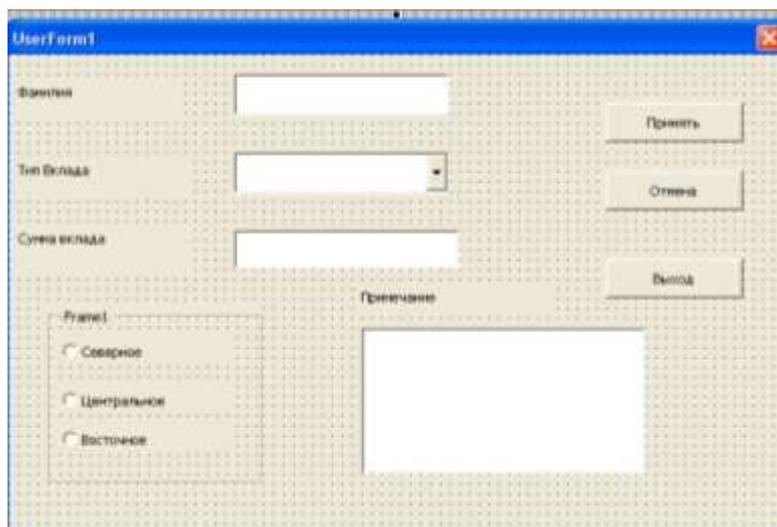


Рис. 6.1 Форма Вклад в конструкторе

Выполните следующие действия.

1. Создайте форму пользователя согласно рис. 6.1.
2. Элементам управления присвойте имена и свойства в соответствии с таблицей 6.1.

Табл.6.1. Элементы управления формы Вклад

Элемент	Имя (Свойства)	События
Форма	Вклад Caption=Прием вклада	
Поле	Фамилия	
Поле	СуммаВклада	
Поле со списком	ТипВклада	
Переключатели	Северное	
	Центральное	
	Восточное	
Поле	Примечание	
Кнопка	Принять Caption= Принять	Click
Кнопка	Отмена, Caption=Отмена	Click
Кнопка	Выход, Caption= Выход	Click

3. Рабочему листу с именем Лист2 присвойте имя База, а рабочему листу с именем Лист1 присвойте имя Меню.

```
Private Sub ПриемВклада_Click()  
'Вызываем процедуру формирования заголовков БД  
ЗаголовокРабочегоЛиста  
End Sub
```

4. Для формирования заголовков столбцов создадим процедуру, которая будет вызываться по событию кнопки Прием вклада. Для этого:
- Выделите в конструкторе объект *Меню(Лист1)*;
  - В контекстном меню (правая кнопка мыши) щелкните *Просмотр кода(View Code)*,
  - выполните команду меню *Вставка(Insert), Процедура (Procedure)* и присвоим ей имя *ЗаголовокРабочегоЛиста*.

Процедура выполняет действия:

- проверяет, заполнена ли первая строка-строка заголовка столбцов (по значению ячейки A1); если заполнена, то действия не выполняются, работа завершается и управление передается в точку вызова;
- если первая строка не заполнена, выполняется заполнение шапки после чего управление передается в точку вызова.

Текст процедуры (на Листе1):

```
Private Sub ЗаголовокРабочегоЛиста()  
'Активизируем рабочий лист  
Application.Worksheets(2).Activate  
'проверяем, есть ли названия столбцов  
With ActiveSheet  
If .Range("A1").Value = "Фамилия" Then  
.Range("A2").Select  
Else  
ActiveSheet.Cells.Clear  
.Range("A1:E1").Value = Array("Фамилия", "Вклад", "Сумма", "Отделение",  
"Примечание")  
.Range("2:2").Select  
'ActiveWindows.FreezePanels = True  
.Range("A2").Select  
.Range("A1").AddComment  
.Range("A1").Comment.Visible = False  
.Range("A1").Comment.Text Text:="Фамилия"  
.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
```

Проверьте работу процедуры по нажатию кнопки Прием вклада.

2. В процедуру обработки события Initialize формы введите код:

```
Private Sub UserForm_Initialize()  
With Вклад  
.Северное.Value = True  
'установим длину элемента Список  
.ТипВклада.ListRows = 3  
'Присвоим значения элементам списка  
.ТипВклада.List = Array("Срочный", "Депозит", "Текущий")  
.Принять.SetFocus  
'Установим фокус элементу Кнопка Принять  
End With  
End Sub
```

После этого вид при запуске будет такой:

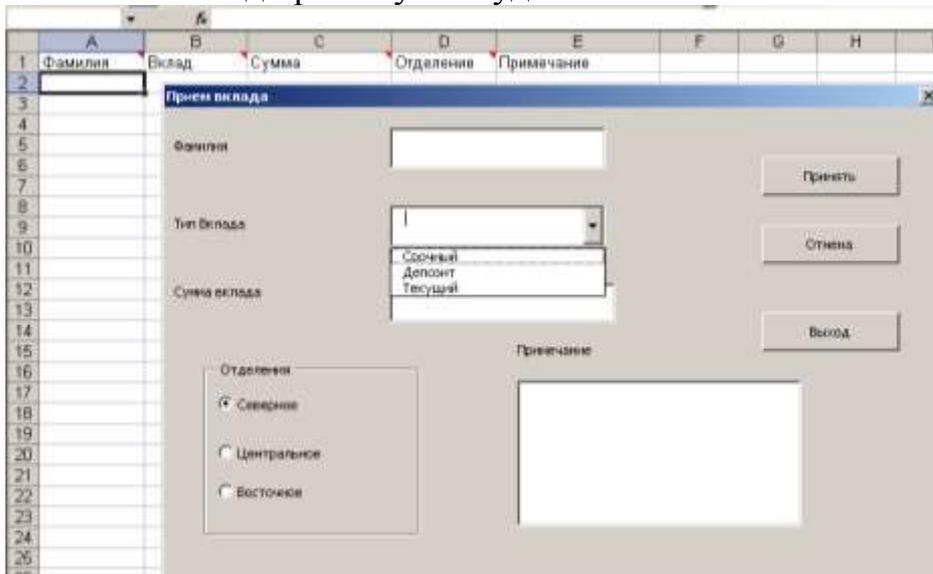


Рис.6.2. Вид при запуске.

3. В процедуру обработки события Initialize элемента Форма с именем Вклад введите код:

```
Private Sub Вклад_Initialize()  
Application.Caption = "Регистрация. База данных Банк"  
,  
  
Application.DisplayFormulaBar = False
```

```
With Принять
.Default = True
.ControlTipText = "Ввод данных в БД"
End With
With Отмена
.Cancel = True
.ControlTipText = "Кнопка Отмены"
End With
ЗаголовокРабочегоЛиста
End Sub
```

4. В процедуру обработки события Click элемента кнопка с именем Принять введите код:

```
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(ActiveSheet.Columns(1)) + 1
With Вклад
    If .Фамилия.Text = "" Then
        MsgBox "Не указана фамилия", vbExclamation
        Exit Sub
    End If
    If .ТипВклада.Value = "" Then
        MsgBox "Не указан тип вклада", vbExclamation
        Exit Sub
    End If
    Фамилия = .Фамилия.Text
    ТипВклада = .ТипВклада.Value
    If .Северное.Value = True Then Отделение = "Северное"
    If .Центральное = True Then Отделение = "Центральное"
    If .Восточное = True Then Отделение = "Восточное"
    If IsNumeric(.СуммаВклада.Text) = False Then
        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
```

5. В процедуру обработки события Click элемента кнопка с именем Отмена введите код:

```
Private Sub Отмена_Click()  
Dim НомерСтроки As Integer  
'Вычисляем номер последней строки  
НомерСтроки = Application.CountA(ActiveSheet.Columns(1))  
'Удаляем содержимое ячеек строки  
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
```

6. В процедуру обработки события Click элемента кнопка с именем Выход введите код:

```
Private Sub Выход_Click()  
'Активизируем рабочий лист с именем Меню  
Sheets(1).Activate  
End  
End Sub
```

7. Проверьте работу программы во всех режимах.

### **Лабораторная работа 7. Разработка программы для выполнения операций по вкладам**

Требуется разработать программу для выполнения операций по вкладам, сведения о которых содержатся в базе данных (предыдущая лабораторная работа).

В программе необходимо предусмотреть следующие функции:

- поиск нужной записи в БД на рабочем листе База;
- выполнение операций по приему на счет и снятие со счета;
- отмену операции по вкладу.

Задание

1. Создайте форму пользователя и разместите в ней элементы управления, как показано на рис. 8. Элементам управления присвойте имена в соответствии с табл.7.1.

Табл. 7.1. Элементы управления формы Операции по вкладам (Опер)

Элемент	Имя (Свойства)	Назначение
Форма	Опер Caption= Операции по вкладам	
Поле	Фам	Фамилия клиента
Поле	Величина	Сумма по текущей операции
Поле	Остаток	Остаток вклада
Поле	Получ	Фамилия получателя (кассира)
Поле	Дата	Дата операции
Поле со списком	Тип	Тип вклада
Поле	Прим	Примечание
Переключатели	Сев Caption=Северное	Отделение
	Центр Caption=Центральное	
	Вост Caption=Восточное	
Кнопка	Найти Caption= Найти	Работает первой (сначала нужно найти клиента по фамилии, типу вклада и отделению). Поиск, конечно, упрощен!
Кнопка	Принять Caption= Принять	Принять вклад, если клиент найден
Кнопка	Отмена, Caption=Отменить	Отменить операцию
Кнопка	Выдать, Caption= Выдать	Выдать сумму, если клиент найден
Кнопка	Выход, Caption= Выход	Закрыть приложение

# Электронный архив УГЛТУ

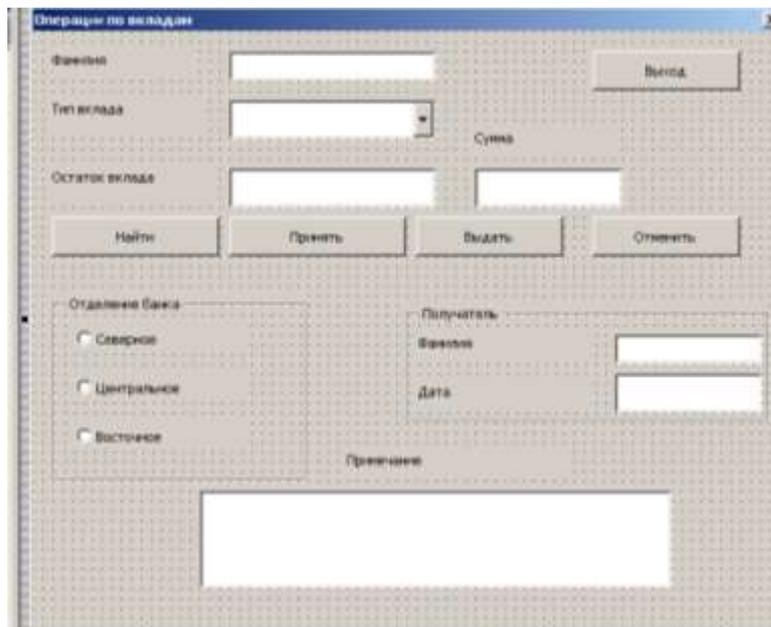


Рис. 7.1. Форма Операции по вкладам в конструкторе

2. Рабочему листу Лист3 присвойте имя Операции.
3. На рабочем листе с именем Меню поместите элемент управления *Кнопка* с надписью Операции. Занесите код:  

```
Private Sub Операции_Click()
'активизируем форму Опер
Опер.Show
End Sub
```

 Проверьте, как работает кнопка.
4. На листе операции в первой строке поместите заголовки таблицы (рис.7.2).

	А	В	С	Д	Е	Ф	Г	Н
1	Фамилия вкладчика	дата операции	Фамилия кассира	Тип вклада	Выдано	Принято	Отделение	Примечание
2								
3								
4								

Рис.7.2

5. В программном модуле, в общей области формы запишите текст объявления переменных:  

```
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
```

6. Для процедуры обработки события UserForm\_Initialize введем код:

```
Private Sub UserForm_Initialize()  
With Опер  
'устанавливаем переключатель  
.Сев.Value = True  
'количество элементов в поле со списком  
.Тип.ListRows = 3  
.Тип.List = Array("Срочный", "Депозит", "Текущий")  
'устанавливаем фокус на кнопке Найти  
.Найти.SetFocus  
'активируем рабочий лист база  
Application.Worksheets("база").Activate  
End With  
End Sub
```

7. Для процедуры обработки события кнопки Найти\_Click введем код:

```
Private Sub Найти_Click()  
'активируем рабочий лист база  
Application.Worksheets("база").Activate  
'вычисляем количество заполненных строк  
Количество = Application.CountA(ActiveSheet.Columns(1))  
Фамилия = Фам  
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  
'поиск записи  
If .Cells(Номер, 1) = Фамилия And .Cells(Номер, 2) = ТипВклада And _  
.Cells(Номер, 4) = Отделение Then Exit For  
Next  
'отображаем остаток вклада в поле Остаток  
Остаток.Text = .Cells(Номер, 3)  
End With  
If Номер = Количество + 1 Then  
MsgBox "Такого счета в базе нет", vbInformation  
End If  
' отображаем текущую дату в поле Дата  
Дата.Text = Date  
  
End Sub
```

8. Для процедуры обработки события кнопки Принять\_Click введем код:

```
Private Sub Принять_Click()  
If IsNumeric(Величина.Text) = False Then  
MsgBox "Ошибка в поле Сумма", vbInformation, "Банк"  
Exit Sub  
End If
```

```
With ActiveSheet
'записываем в ячейку новое значение остатка
.Cells(Номер, 3) = CStr(CDbl(.Cells(Номер, 3)) + CDbl(Величина.Text))
End With
'активизируем рабочий лист Операции
Worksheets("Операции").Activate
'вычисляем номер первой пувстой строки
Количество = Application.CountA(ActiveSheet.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
End With
End Sub
```

Внимание. Кнопку Принять следует щелкать после того, как вкладчик будет найден!

9. Для процедуры обработки события кнопки Отменить \_Click введем код:

```
Private Sub Отменить_Click()
Worksheets(2).Activate
With ActiveSheet
'Восстанавливаем прежний остаток в БД
.Cells(Номер, 3) = Остаток.Text
End With
Worksheets("Операции").Activate
'Удаляем данные о проведенной операции
With ActiveSheet
.Cells(Количество, 1) = ""
.Cells(Количество, 2) = ""
.Cells(Количество, 3) = ""
.Cells(Количество, 4) = ""
.Cells(Количество, 5) = ""
.Cells(Количество, 6) = ""
.Cells(Количество, 7) = ""
.Cells(Количество, 8) = ""
End With
End Sub
```

10. Для процедуры обработки события кнопки Выдать\_Click введем код:

```
Private Sub Выдать_Click()
If IsNumeric(Величина.Text) = False Then
MsgBox "Ошибка в поле Сумма", vbInformation, "Банк"
```

```
Exit Sub
End If

With ActiveSheet
'записываем в ячейку новое значение остатка
.Cells(Номер, 3) = CStr(CDbl(.Cells(Номер, 3)) - CDbl(Величина.Text))
End With
'активизируем рабочий лист Операции
Worksheets("Операции").Activate
'вычисляем номер первой пустой строки
Количество = Application.CountA(ActiveSheet.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, 5) = Величина.Text
.Cells(Количество, 7) = Отделение
.Cells(Количество, 8) = Прим.Text
End With
End Sub
```

11. Для процедуры обработки события кнопки Выход \_Click введем код:

```
Private Sub Выход_Click()
Application.Worksheets(1).Activate
End
End Sub
```

12. Проверьте работу программы.

## Лабораторная работа 8. Применение шаблонов в Word

Цель работы: создать приложение для подготовки документа на бланке на примере создания шаблона для подготовки полиса обязательного страхования гражданской ответственности.

Содержание страхового полиса в несколько сокращенном виде представлено на рис.8.1 (8.2).

Требования к создаваемому приложению:

- для удобства заполнения граф документа использовать форму, в поля которой записываются данные;
- типовые данные, такие, как марка автомобиля, хранить в поле со списком;

- занесение данных в графы документа выполнять по команде пользователя;
- после занесения информации в документ он должен автоматически идентифицироваться по имени клиента и сохраняться в определенной области файловой системы.

Разработку приложения начнем с создания макета документа. Для этого в документ поместим следующие таблицы (или надписи).

1. Таблица для записи текста *«Приложение №2 К Правилам обязательного страхования гражданской ответственности владельца транспортных средств»*. Таблица содержит одну строку и один столбец. Введем в нее соответствующий текст. Эта таблица будет иметь внутренний номер Tables(1).
2. Надпись с текстом, содержащим номер страхового полиса *«Страховой полис ААА № 100234566 Обязательного страхования гражданской ответственности владельцев транспортных средств»*
3. Таблица для записи срока страхования, включающая две строки и 11 колонок. Эта таблица будет иметь внутренний номер Tables(2). Заполним первую строку так, как показано на рис.1. Во второй строке объединим часть колонок и заполним как на рис. 8.1.
4. Таблица для записи сведений о страхователе, состоящая из двух строк и одного столбца, будет иметь внутренний номер Tables(3). Введем в первую строку текст (рис.8.1).

Приложение №2  
К Правилам обязательного  
страхования гражданской  
ответственности владельца  
транспортных средств

Страховой полис AAA № 0100234566  
Обязательного страхования гражданской ответственности  
владельцев транспортных средств

Срок страхования с		час		мин		-	-	20	г.
по 24 час 00 мин						-	-	20	г.

1. Страхователь

2. Собственник

Марка, модель транспортного средства	Идентификационный номер транспортного средства				Государственный идентификационный номер
	Паспорт ТС серия		№		

3. Лица, допущенные к управлению ТС.

№ п/п	Фамилия, имя, отчество	Водительское удостоверение (номер, серия)
1		
2		
3		

Подпись страхователя

Представитель страховщика

Рис.8.1. Макет документа в Word

- Таблица для записи сведений о собственнике, состоящая из одной строки и одного столбца, будет иметь внутренний номер Tables(4). Введем в первую строку текст (рис.8.1).
- Таблица для записи сведений о транспортном средстве, состоящая из двух строк и трех колонок, будет иметь внутренний номер Tables(5). В ячейку второго столбца и второй строки поместим еще одну таблицу, состоящую из двух строк и четырех столбцов. Она будет иметь внутренний номер Tables(1). Объединим ячейки первой строки этой

таблицы и введем в ячейки внешней и внутренней таблиц текст согласно рис.8.1.

7. Таблица для записи сведений о лиц, допущенных к управлению ТС (рис.8.1), будет иметь внутреннее имя Tables(6).
8. Таблица для записи сведений о страховщике будет иметь внутреннее имя Tables(7).
9. Закончим оформление документа в соответствии с макетом, приведенным на рис.8.1.
10. Форма для ввода данных, включающая перечисленные сведения, имеет вид, представленный на рис.8.2.

Рис.8.2 Форма для ввода

11. В редакторе *MS Visual Basic* (<ALT>+<F11>) создадим форму и поместим в нее соответствующие элементы и присвоим им имена. Их перечень содержится в табл.8.1. Метки добавьтe сами.

Табл.8.1. Управляющие элементы для формы *Страховвание ОСАГО*

№	Элемент	Имя	Назначение	Обрабатываемое событие
1	TextBox	ЧасНач	Для ввода часа начала срока	
2	TextBox	МинНач	Для ввода минут начала срока	
3	TextBox	ДеньНач	Для ввода дня начала срока	
4	TextBox	МесНач	Для ввода месяца начала срока	
5	TextBox	ГодНач	Для ввода года начала срока	
6	TextBox	ДеньКон	Для ввода дня окончания срока	

## Электронный архив УГЛТУ

7	TextBox	МесКон	Для ввода месяца окончания срока	
8	TextBox	ГодКон	Для ввода года окончания срока	
9	TextBox	Страхователь	Для ввода ФИО страхователя	
10	TextBox	Собственник	Для ввода ФИО собственника	
11	ComboBox	Марка	Для выбора из списка марки ТС	
12	TextBox	ИденНом	Для ввода идентификационного номера ТС	
13	TextBox	ПаспортСерия	Для ввода серии и паспорта ТС	
14	TextBox	ПаспорНомер	Для ввода номера паспорта ТС	
15	TextBox	РегЗнак	Для ввода регистрационного номера ТС	
16	TextBox	ДопЛиц1	Для ввода ФИО допущенного к управлению ТС	
17	TextBox	ДопЛиц2	Для ввода ФИО допущенного к управлению ТС	
18	TextBox	ДопЛиц3	Для ввода ФИО допущенного к управлению ТС	
19	TextBox	Удост1	Для ввода номера удостоверения, допущенного к управлению ТС	
20	TextBox	Удост2	Для ввода номера удостоверения, допущенного к управлению ТС	
21	TextBox	Удост3	Для ввода номера удостоверения, допущенного к управлению ТС	
22	CommandButton	Запись	Исполняемый элемент	Click выполняет запись из формы в документ; присваивает имя документу; закрывает форму

12. В процедуру Click кнопки Запись введем код программы:

```
Private Sub Запись_Click()
ActiveDocument.Tables(2).Cell(1, 2).Select
Selection.Text = ЧасНач.Value
ActiveDocument.Tables(2).Cell(1, 4).Select
Selection.Text = МинНач.Value
ActiveDocument.Tables(2).Cell(1, 6).Select
Selection.Text = ДеньНач.Value
ActiveDocument.Tables(2).Cell(1, 8).Select
Selection.Text = МесНач.Value
```

```

ActiveDocument.Tables(2).Cell(1, 8).Select
Selection.Text = МесНач.Value
ActiveDocument.Tables(2).Cell(1, 10).Select
Selection.Text = ГодНач.Value
ActiveDocument.Tables(2).Cell(2, 2).Select
Selection.Text = ДеньКон.Value
ActiveDocument.Tables(2).Cell(2, 4).Select
Selection.Text = МесКон.Value
ActiveDocument.Tables(2).Cell(2, 6).Select
Selection.Text = ГодКон.Value
ActiveDocument.Tables(3).Cell(1, 1).Select
Selection.Text = Страхователь.Value
ActiveDocument.Tables(4).Cell(1, 1).Select
Selection.Text = Собственник.Value
ActiveDocument.Tables(5).Cell(2, 1).Select
Selection.Text = Марка.Value
ActiveDocument.Tables(5).Cell(2, 2).Tables(1).Cell(1, 1).Select
Selection.Text = ИдентНомер.Value
ActiveDocument.Tables(5).Cell(2, 2).Tables(1).Cell(2, 2).Select
Selection.Text = ПаспортСерия.Value
ActiveDocument.Tables(5).Cell(2, 2).Tables(1).Cell(2, 4).Select
Selection.Text = ПаспортНомер.Value
ActiveDocument.Tables(5).Cell(2, 3).Select
Selection.Text = ПерЗнак.Value
'занесение в документ сведений о допущенных к управлению лицах
ActiveDocument.Tables(6).Cell(2, 2).Select
Selection.Text = ДопЛиц1.Value
ActiveDocument.Tables(6).Cell(2, 3).Select
Selection.Text = Удост1.Value
'занесение в первую графу номера по порядку
ActiveDocument.Tables(6).Cell(2, 1).Select
Selection.Text = 1
If ДопЛиц2 <> "" Then
    ActiveDocument.Tables(6).Cell(3, 2).Select
    Selection.Text = ДопЛиц2.Value
    ActiveDocument.Tables(6).Cell(3, 3).Select
    Selection.Text = Удост2.Value
    ActiveDocument.Tables(6).Cell(3, 1).Select
    Selection.Text = 2
End If
If ДопЛиц3 <> "" Then
    ActiveDocument.Tables(6).Cell(4, 2).Select
    Selection.Text = ДопЛиц3.Value
    ActiveDocument.Tables(6).Cell(4, 3).Select
    Selection.Text = Удост3.Value
    ActiveDocument.Tables(6).Cell(4, 1).Select
    Selection.Text = 3
End If
ActiveDocument.Tables(7).Cell(1, 1).Select
Selection.Text = Страховщик.Value
' сохранение документа и присвоение имени файла в соответствии с фамилией
страхователя

```

```
ActiveDocument.SaveAs ("F:\Программирование в MS Office\Polis\" &  
Страхователь.Value)  
MsgBox ("Документ сохранен в папке Polis")  
End  
End Sub
```

13. При открытии приложения должен быть инициализирован список ComboBox, который содержит марки ТС. Для этого введем код на процедуру инициализации формы:

```
Private Sub UserForm_Initialize()  
    Марка.List = Array("BMW", "Ford", "Hunday Solaris")  
End Sub
```

14. При создании нового документа на основе созданного шаблона необходимо отобразить форму на экране. Для этого в процедуру обработки события создания нового документа введем инструкцию:

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

15. Сохраним созданный проект как шаблон. Для этого переключимся в окно приложения MS Word, выполним команду *Файл, Сохранить как*. Откроется диалоговое окно *Сохранение документа*. В списке *Тип файла* выберем *Шаблон документа с поддержкой макросов*, а в поле *Имя файла* введем *ПолисАвтоСтрахования*. Папку для сохранения шаблона укажите самостоятельно. После выполнения команды *Сохранить* весь проект будет сохранен в папке личных шаблонов с расширением *.dot*.

16. После сохранения проекта его можно использовать как шаблон для подготовки документов-полисов. Для этого следует открыть папку с шаблоном и щелкнуть по документу-шаблону *ПолисАвтоСтрахования*.



## Оглавление

ВВЕДЕНИЕ.....	2
ЛАБОРАТОРНАЯ РАБОТА 1. СВОЙСТВА И МЕТОДЫ ОБЪЕКТОВ RANGE И SELECTION.....	2
ЛАБОРАТОРНАЯ РАБОТА 2. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАСЧЕТА ПО ИПОТЕЧНОМУ КРЕДИТУ .....	8
ЛАБОРАТОРНАЯ РАБОТА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАСЧЕТА АМОРТИЗАЦИИ ОСНОВНЫХ ФОНДОВ .....	13
ЛАБОРАТОРНАЯ РАБОТА 4. ПРИМЕНЕНИЕ ФУНКЦИЙ EXCEL В КОДЕ VBA .....	15
ЛАБОРАТОРНАЯ РАБОТА 5. РАЗРАБОТКА ПРОГРАММЫ ДЛЯ СОЗДАНИЯ БАЗЫ ДАННЫХ СТРАХОВАНИЕ.....	17
ЛАБОРАТОРНАЯ РАБОТА 6. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ СОЗДАНИЯ БАЗЫ ДАННЫХ ПО РЕГИСТРАЦИИ ВКЛАДОВ .....	21
ЛАБОРАТОРНАЯ РАБОТА 7. РАЗРАБОТКА ПРОГРАММЫ ДЛЯ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ ПО ВКЛАДАМ .....	26
ЛАБОРАТОРНАЯ РАБОТА 8. ПРИМЕНЕНИЕ ШАБЛОНОВ В WORD.....	31