

Н.Ф. Антипенко, Т.А. Санькова

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКИХ ПРИЛОЖЕНИЙ СРЕДСТВАМИ VBA



Омск • 2012

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Сибирская государственная автомобильно-дорожная
академия (СибАДИ)»

Н.Ф. Антипенко, Т.А. Санькова

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКИХ
ПРИЛОЖЕНИЙ СРЕДСТВАМИ VBA

Учебно-методическое пособие

Омск
СибАДИ
2012

УДК 004.9
ББК 73.6
А 72

Рецензенты:

канд. пед. наук, доц. *В.В. Котенко* (ОмГПУ);
канд. пед. наук, доц. *Н.В. Манюкова* (ОмЭИ)

Работа одобрена редакционно-издательским советом академии в качестве учебно-методического пособия для всех направлений и специальностей факультетов «Нефтегазовая и строительная техника», «Автомобильный транспорт».

Антипенко Н.Ф.

А 72 Разработка пользовательских приложений средствами VBA: учебно-методическое пособие / Н.Ф. Антипенко, Т.А. Санькова. – Омск: СибАДИ, 2012. – 92 с.

Данное учебно-методическое пособие предназначено для студентов 1–3 курсов различных направлений и специальностей факультетов «НСТ» и «АТ», изучающих дисциплины «Информатика», «Информационные технологии», «Прикладное программирование», «Программирование и алгоритмизация». «Информационные технологии в строительстве». Пособие содержит теоретический материал и задания к лабораторным работам для изучения основ разработки пользовательских приложений средствами языка программирования VBA. Пособие может быть использовано преподавателями в качестве дидактического материала.

Табл. 10. Ил. 92. Библиогр.: 4 назв.

© ФГБОУ ВПО «СибАДИ», 2012

ВВЕДЕНИЕ

Visual Basic For Applications (VBA) – это сочетание одного из самых простых языков программирования и всех вычислительных возможностей табличного процессора Excel. С помощью VBA можно легко и быстро создавать разнообразные приложения, даже не являясь специалистом в области программирования. VBA содержит графическую среду, позволяющую наглядно конструировать экранные формы и управляющие элементы.

1. ОСНОВНЫЕ ПОНЯТИЯ

1.1. Объектно-ориентированное программирование

Приложения (прикладные программы) являются интерактивными системами, в которых для организации взаимодействия между пользователем и программой используются методы (подпрограммы), управляемые событиями. В основе объектно-ориентированного программирования (ООП) лежит идея объединения в одной структуре данных и действий, которые выполняются над этими данными.

Совокупность данных, описывающих какое-то явление, и программного кода, описывающего его поведение, называется **объектом**. Объект объединяет в себе – **инкапсулирует** – данные и методы работы с этими данными.

С объектом неразрывно связано понятие класса. **Класс** задает свойства и поведение объектов класса. Каждый объект является экземпляром класса. Способность класса сохранять свойства и методы класса-предшественника называется **наследованием**.

Способность объекта принимать различные формы называется **полиморфизмом**.

Объектно-ориентированная программа – это совокупность объектов и способов их взаимодействия. Обмен сообщениями между объектами производится с помощью сообщений. Сообщения являются результатом появления событий.

Событие – любое воздействие от мыши или клавиатуры, либо действие, вызванное перерисовкой. События предназначены для создания пользовательской реакции на те или иные входные воздействия. Все события в VBA имеют названия, например, *Click* – щелчок левой кнопкой мыши, *DbfClick* – двойной щелчок левой

кнопкой мыши, *MouseMove* – перемещение указателя мыши, *Change* – изменение объекта и др.

При наступлении того или иного события можно запрограммировать соответствующие действия, написав процедуру обработки данного события.

Кроме процедур, которые составляет программист, VBA содержит большое количество встроенных процедур и функций. Эти процедуры и функции описаны в справочной системе или документации на систему программирования.


1.2. Разработка пользовательского интерфейса



Пользовательский интерфейс – комплекс программных средств, обеспечивающих взаимодействие пользователя с системой; средства связи между пользователем и системой.


Для того чтобы начать создавать свое собственное приложение в VBA, нужно запустить редактор VBA в Excel (**Сервис – Макрос – Редактор Visual Basic**), а затем создать форму с помощью команды меню **Вставка – UserForm (Insert – UserForm)**.

Форма в VBA – это окно в интерфейсе пользователя. На форме можно размещать различные объекты: кнопки, переключатели, списки, полосы прокрутки и т.д. Каждый компонент в программе, как и сама форма, имеет уникальное имя.

Общий вид окна VBA в режиме отображения формы и в режиме написания программного кода показан на рис. 1, 2.

 Для перехода между объектами приложения используется окно проекта, которое можно вывести на экран с помощью команды **Вид – Окно проекта (View – Project Explorer)** или соответствующей кнопки на панели инструментов.


 В окне проекта имеются кнопки для переключения между режимами просмотра проекта (**View Object**) и просмотра программного кода (**View Code**). Также перейти из режима создания программного кода в режим редактирования формы можно сделав двойной щелчок мыши по имени формы в списке объектов в окне проекта.

 Для управления свойствами объектов используется окно свойств, которое можно вывести на экран с помощью команды меню **Вид – Окно свойств (View – Properties Window)** или кнопки **Properties Window** на панели инструментов.

Кнопки переключения между формой и программой

Окно проекта

Кнопка Панель элементов

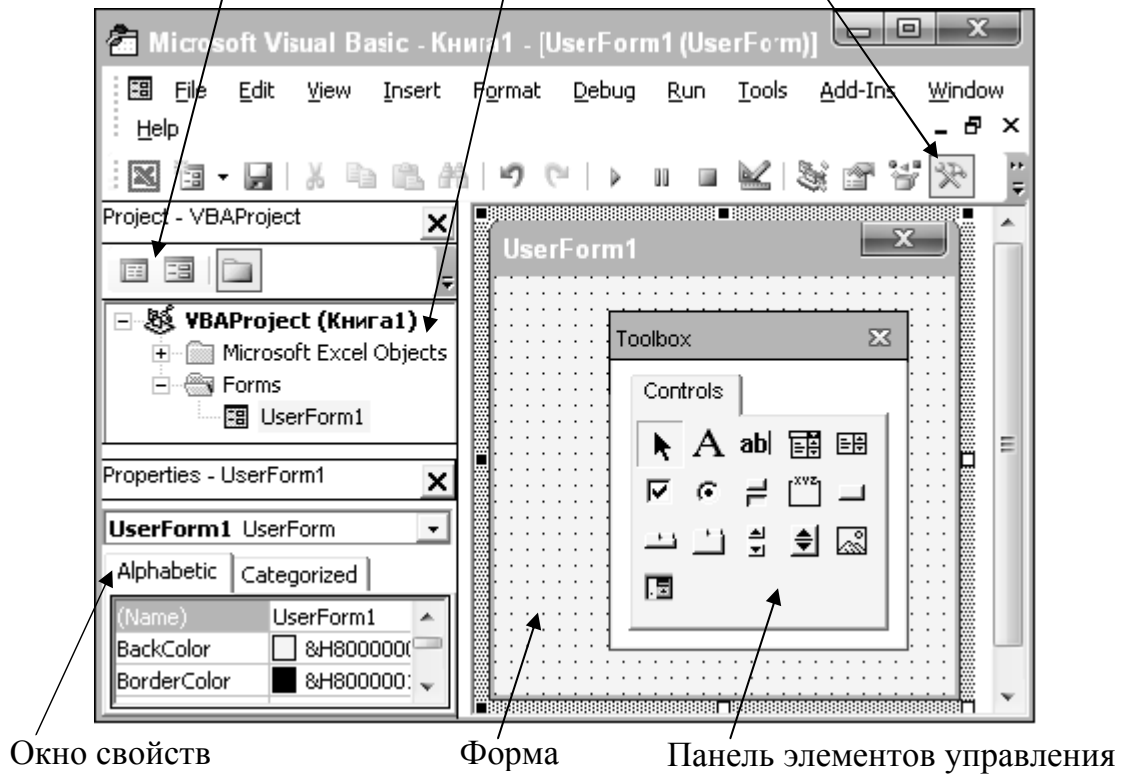


Рис. 1. Общий вид окна VBA в режиме отображения формы

Окно программного кода

Выбор объекта (Object)

Выбор события (Procedure)

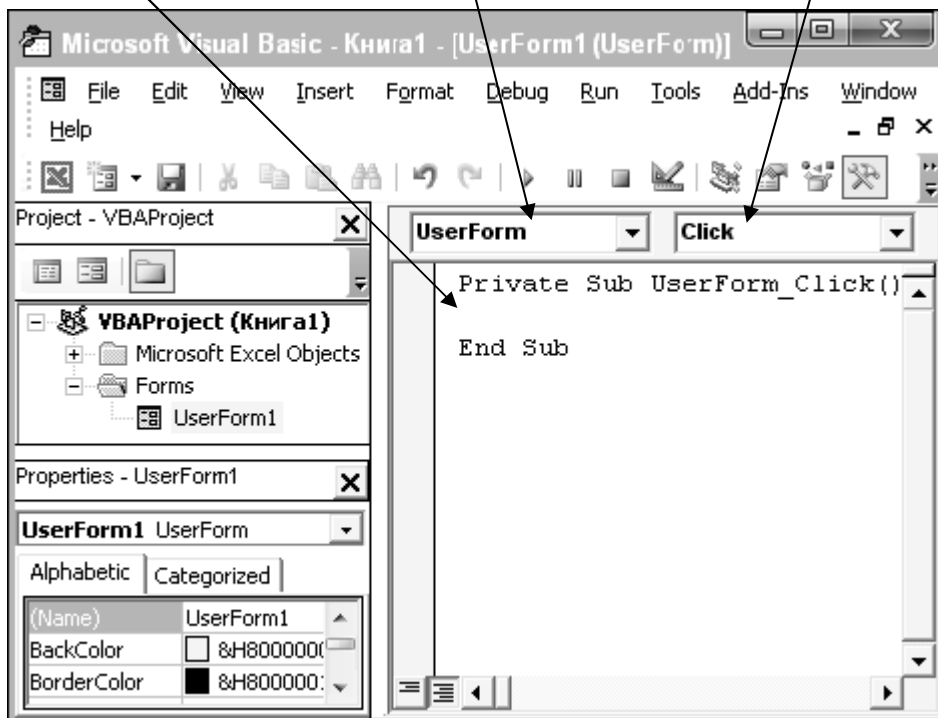


Рис. 2. Общий вид окна VBA в режиме написания программного кода

1.2.1. Элементы управления, их свойства и методы




Чтобы добавлять на форму различные элементы управления (кнопки, метки, поля, списки и т.д.) нужно использовать панель элементов управления. Вывести эту панель на экран можно с помощью кнопки **Панель элементов (Toolbox)** на панели инструментов (см. рис. 1).

Перечень основных элементов управления приведен в табл. 1.

Таблица 1

Основные элементы управления

Элемент управления	Название	Обозначение	Описание
	Метка	Label	Вывод данных на форму
	Текстовое поле	TextBox	Ввод данных
	Список	ListBox	Выбор данных из списка
	Поле со списком	ComboBox	Ввод данных или выбор из списка
	Командная кнопка	CommandButton	Выполнение каких-либо действий
	Флажок	CheckBox	Выбор какого-либо параметра (можно несколько)
	Переключатель	OptionButton	Выбор какого-либо параметра (только одного)
	Выключатель	ToggleButton	Выбор какого-либо параметра (включено/выключено)
	Рисунок	Image	Вывод на форму графических изображений
	Рамка	Frame	Объединение флажков и переключателей в группу
	Набор вкладок	TabStrip	Размещение элементов управления на разных вкладках
	Набор страниц	MultiPage	Размещение элементов управления на разных страницах
	Полоса прокрутки	ScrollBar	Выбор значения из диапазона

	Счетчик	SpinButton	Выбор значения из небольшого диапазона
---	---------	------------	--

В VBA каждый элемент пользовательского интерфейса (включая форму) имеет ряд свойств. Под **свойством** объекта подразумевается какая-либо его характеристика, например, размеры, координаты, имя.

Свойства могут быть:

- простыми (значения свойств являются числами или строками);
- перечисляемыми (значения свойств выбираются из предлагаемого списка);
- вложенными (свойства включают в себя еще набор вложенных свойств).

Свойства можно устанавливать в процессе разработки приложения в окне свойств, а также изменять в программном коде после запуска программы. Некоторые свойства объектов доступны только в программе.

Поместив на форму нужные элементы управления, можно установить свойства для каждого из них в окне свойств (см. рис. 1). Для этого следует выделить объект, найти в окне свойств нужное свойство и установить его значение. Для выделения нескольких объектов используются клавиши [Shift] или [Ctrl].

Для того чтобы обратиться к свойству объекта в программе, нужно указать имя объекта и через точку имя свойства.

Задание значения свойства объекта в программе осуществляется следующим образом:

Имя объекта.Свойство = Значение

Чтобы считать значение какого-либо свойства объекта нужно в программе написать следующий оператор:

Имя переменной = Имя объекта.Свойство

Рассмотрим основные свойства объектов:

AutoSize – автоматический подбор размера объекта.

BackColor – цвет объекта (фон).

BorderColor – цвет границы.

BorderStyle – тип границы.

Caption – надпись (заголовок).

Enabled – доступность объекта.

Font – шрифт.

ForeColor – цвет шрифта.
Height – высота объекта.
LargeChange – большой шаг изменения значения диапазона.
Left – горизонтальная координата левого верхнего угла объекта.
ListIndex – номер выбранного элемента списка.
Locked – блокировка ввода (выбора) данных.
MaxLength – максимальная длина вводимых данных.
MultiLine – использование нескольких строк в поле.
Name – имя объекта.
PasswordChar – символ, отображаемый в поле вместо вводимого.
Picture – графическое изображение (файл с рисунком).
PictureSizeMode – режим отображения рисунка.
ScrollBars – полосы прокрутки.
SmallChange – маленький шаг изменения значения диапазона.
StartPosition – стартовая позиция формы.
Text – текст в поле.
TextAlign – выравнивание текста.
TextLength – длина текстовой строки.
Top – вертикальная координата левого верхнего угла объекта.
Value – значение.
Visible – видимость объекта.
Width – ширина объекта.
WordWrap – перенос по словам.

В VBA существуют особые типы процедур и функций, называемые **методами**. Методы принадлежат конкретным объектам, так же как и свойства, и определяют те задачи, которые может выполнять данный объект.

Чтобы вызвать метод, нужно указать имя объекта и через точку имя метода, т.е. в программном коде должна быть написана команда следующего вида:

Имя объекта.Метод

Рассмотрим основные методы объектов:

AddItem (строка) – добавить элемент в конец списка.

Clear – очистить список.

Hide – скрыть форму.

RemoveItem (номер) – удалить заданный элемент из списка.

SetFocus – установить фокус на объект.

Show – показать форму.

1.2.2. Написание программного кода

Для того чтобы написать процедуру обработки события *Click* для какого-либо объекта, нужно дважды щелкнуть левой кнопкой мыши по этому объекту на форме или выделить объект и воспользоваться кнопками переключения между программой и объектом (см. рис. 1). На экране появится окно программного кода (см. рис. 2) с заготовкой процедуры обработки данного события. С помощью раскрывающихся списков «Выбор объекта» (Object) и «Выбор события» (Procedure) можно написать процедуру обработки нужного события для любого объекта (см. рис. 2).

▶ После написания программного кода (процедур обработки событий для объектов) запуск приложения осуществляется с помощью кнопки **Запуск (Run Sub)** на панели инструментов или с помощью клавиши [F5].

1.2.3. Создание элементов управления на рабочем листе Excel

Для того чтобы создавать элементы управления непосредственно на рабочих листах Excel, нужно воспользоваться панелью инструментов «Элементы управления» (команда меню **Вид – Панели инструментов – Элементы управления**). Панель инструментов «Элементы управления» содержит помимо уже знакомых элементов управления три кнопки (рис. 3).

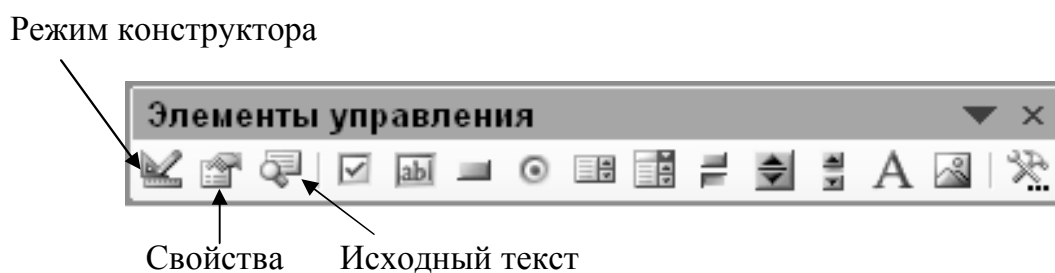


Рис. 3. Панель инструментов «Элементы управления»

Кнопка **Режим конструктора** используется для того, чтобы редактировать объекты. Чтобы выйти из режима конструктора и начать работу с объектами нужно нажать на ту же самую кнопку (см. рис. 3), которая теперь имеет название **Выход из режима конструктора**. Кнопка **Свойства** выводит на экран окно свойств.

Кнопка **Исходный текст** позволяет открыть окно программного кода для выбранного объекта.

С помощью кнопки, расположенной на рабочем листе, можно запустить Ваше приложение, не заходя в редактор Visual Basic. Для этого в окне программного кода созданной кнопки следует написать команду для вызова метода *Show* Вашей формы, например, `UserForm1.Show`.

1.2.4. Защита проекта

Для того чтобы защитить Ваше приложение от несанкционированного просмотра или редактирования можно установить защиту проекта. Тогда пользователь, не знающий пароля, не сможет открыть форму и программный код.

Для установки защиты в окне проекта (Project Explorer) следует щелкнуть правой кнопкой мыши по объекту **VBAProject** и выбрать команду **VBAProject Properties**. Затем на вкладке *Protection* установить флажок *Lock project for viewing*, ввести пароль в поле *Password* и повторить его в поле *Confirm password*, сохранить и закрыть рабочую книгу. При последующем открытии рабочей книги и попытке открыть проект в редакторе Visual Basic у пользователя будет запрашиваться пароль.

Если Вы установили защиту проекта, следует разместить на рабочем листе кнопку для запуска приложения (см. выше) или сделать открытие формы (запуск приложения) автоматическим при открытии рабочей книги. Для этого нужно:

- щелкнуть правой кнопкой мыши по объекту **Эта книга** в окне проекта (Project Explorer) и выбрать команду **View Code**;
- выбрать в списке «Выбор объекта» (Object) объект *Workbook*;
- выбрать в списке «Выбор события» (Procedure) событие *Open*;
- написать в процедуре команду открытия формы (например, `UserForm1.Show`).

1.2.5. Работа с макросами

Для того чтобы вставить в процедуру обработки события какие-либо действия над рабочим листом в Excel, например, форматирование ячеек, построение диаграммы и т.д., можно записать

макрос в Excel, а затем скопировать записанный макрос в свою процедуру.

Порядок работы:

- 1) выполнить команду **Сервис – Макрос – Начать запись**;
- 2) ввести имя макроса;
- 3) выполнить в нужном порядке действия, которые должны быть записаны в макросе;
- 4) выполнить команду **Сервис – Макрос – Остановить запись**;
- 5) выполнить команду **Сервис – Макрос – Макросы**, выбрать имя макроса и нажать кнопку **Изменить**;
- 6) скопировать команды из макроса в свою программу.

1.3. Основы языка Visual Basic

Приведем краткий обзор основных команд и операторов языка VBA. Основные типы данных описаны в табл. 2.

Таблица 2

Основные типы данных VBA

Тип	Содержимое переменной	Диапазон допустимых значений
Byte	Достаточно малое целое неотрицательное число	От 0 до 255
Integer	Целое число	От -32768 до 32767
Long	Большое целое число	От -2147483648 до 2147483647
Single	Вещественное число Значение одинарной точности с плавающей запятой	От -3,402823E38 до -1,401298E-45 для отрицательных значений и от 1,401298E-45 до 3,402823E38 для положительных значений
Double	Вещественное число Значение двойной точности с плавающей запятой	От -1,79769313486231E308 до 1,79769313486232E308
String	Последовательность символов	От 0 до 65400 символов
Boolean	Логический тип данных	Истина (-1) или Ложь (0)
Date	Дата/Время	Даты с 1.01.100 по 31.12.9999 и время с 0:00:00 по 23:59:59
Type	Набор данных разного типа, имеющих общее имя (запись)	Определяется пользователем

Основные математические функции, используемые в VBA, приведены в табл. 3. Перечень операций приведен в табл. 4 (в порядке уменьшения приоритета операции).

Таблица 3

Математические функции VBA

Функция	Значение
Abs(число)	Абсолютное значение числа (модуль числа)
Cos(число)	Косинус числа
Exp(число)	Возводит константу <i>e</i> в указанную степень
Int(число)	Целая часть числа
Log(число)	Натуральный логарифм числа
Rnd	Выдает случайное число от 0 до 1
Round(число, точность)	Округляет число с указанной точностью
Sin(число)	Синус числа
Sqr(число)	Квадратный корень из числа
Tan(число)	Тангенс числа

Таблица 4

Порядок выполнения операций в VBA

Знаки операций	Операции
Арифметические операции	
^	Возведение в степень
-	Отрицание
* или /	Умножение или деление
\	Вычисление целой части от деления
Mod	Вычисление остатка от деления
+ или -	Сложение или вычитание
Операции сравнения	
=	Равенство
<>	Неравенство (не равно)
<	Меньше
>	Больше
<=	Меньше или равно
>=	Больше или равно
Логические операции	
Not	Логическое НЕ
And	Логическое И
Or	Логическое ИЛИ

Оператор описания переменной

Dim Имя переменной **As** Тип

Оператор присваивания

Имя переменной = Выражение
Оператор ввода

Имя переменной = **InputBox** (“Сообщение”)

Оператор вывода

MsgBox (“Сообщение” & Имя переменной)

Оператор перехода на рабочий лист Excel

Sheets(“Имя листа”).**Select**

Sheets(Номер листа).**Select**

Считывание данных из ячейки рабочего листа Excel

Имя переменной = **Cells**(номер строки, номер столбца)

Вывод данных в ячейку рабочего листа Excel

Cells(номер строки, номер столбца) = Значение

Условный оператор

Неполная форма:

1. **If** Условие **Then** Оператор

2. **If** Условие **Then**
 Блок операторов
End If

Полная форма:

1. **If** Условие **Then** Оператор1 **Else** Оператор2

2. **If** Условие **Then**

Блок операторов 1
Else
Блок операторов 2
End If

Оператор выбора

Select Case Тестируемое выражение
Case Значение 1
Блок операторов 1
Case Значение 2
Блок операторов 2
.....
Case Значение n
Блок операторов n
Case Else
Блок операторов
End Select

Оператор цикла с параметром

For Параметр = Начало **To** Конец **Step** Шаг
Блок операторов
Next Параметр

Операторы циклов с предусловием

1. **Do While** Условие
Блок операторов
Loop
2. **Do Until** Условие
Блок операторов
Loop
3. **While** Условие
Блок операторов
Wend

Операторы циклов с постусловием

1. Do

Блок операторов
Loop While Условие

2. Do

Блок операторов
Loop Until Условие

2. ПРИМЕРЫ СОЗДАНИЯ ПРИЛОЖЕНИЙ

Пример 1

На форме расположены метки с текстом «Hello!» и «Внимание!», кнопка «Изменить текст» и временно невидимая кнопка «Убрать цвет». Щелчок левой кнопкой мыши по первой метке изменяет ее цвет на красный и отображает кнопку «Убрать цвет». Двойной щелчок мыши по первой метке изменяет ее цвет на синий и отображает кнопку «Убрать цвет». Кнопка «Убрать цвет» возвращает метке первоначальный цвет и скрывает себя.

Кнопка «Изменить текст» меняет текст в первой метке: заменяет текст «Hello!» на текст «Привет!» или наоборот. При нажатии левой кнопки мыши на второй метке текст в ней выделяется жирным шрифтом, а при отпускании левой кнопки мыши текст становится обычным. Внешний вид формы после щелчка по первой метке показан на рис. 4.

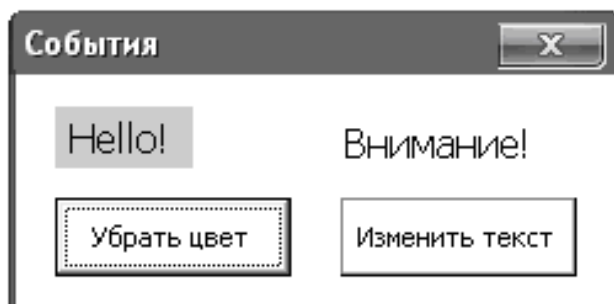


Рис. 4. Внешний вид формы

Объекты и их свойства, определяемые на этапе проектирования в окне свойств, показаны в табл. 5.

Таблица 5

Перечень объектов и их свойств

Объекты	Свойства	Значения свойств
Форма UserForm1	Caption	События
Метка Label1	Caption Font	Hello! Размер шрифта 24
Метка Label2	Caption Font	Внимание! Размер шрифта 24
Кнопка CommandButton1	Caption Visible	Убрать цвет False
Кнопка CommandButton2	Caption	Изменить текст

После создания всех объектов и задания их свойств, приступим к написанию процедур обработки нужных событий. Сделаем двойной щелчок мыши по первой метке, в окне редактора кода появится заготовка процедуры обработки события *Click* (щелчок мыши):

```
Private Sub Label1_Click()  
End Sub
```

Имя объекта Имя события

Для некоторых событий процедура содержит в скобках определенные параметры. Для краткости записи в дальнейшем мы будем их опускать.

Допишем в процедуру две команды.

```
Private Sub Label1_Click()  
Label1.BackColor = RGB(255, 0, 0) 'цвет метки красный  
CommandButton1.Visible = True 'делаем кнопку видимой  
End Sub
```

Функция RGB (Red, Green, Blue) возвращает RGB-код цвета. Значение для каждого цвета (Red – красный, Green – зеленый, Blue – синий) задаются в диапазоне от 0 до 255. Коды некоторых цветов приведены в прил. 3.

Для создания процедуры обработки события *DbClick* (двойной щелчок мыши) для первой метки выберем его из списка событий (см. рис. 2).

```
Private Sub Label1_DbClick()  
Label1.BackColor = RGB(0, 0, 255) 'цвет метки синий  
CommandButton1.Visible = True 'делаем кнопку видимой  
End Sub
```

Теперь нужно написать процедуры обработки события *Click* для двух кнопок. Сделаем двойной щелчок мышью по кнопке «Убрать цвет», в окне редактора кода появится заготовка процедуры, допишем в нее нужные команды.

```
Private Sub CommandButton1_Click()  
Label1.BackColor = &H8000000F 'исходный цвет метки  
CommandButton1.Visible = False 'делаем кнопку невидимой
```

End Sub

Код цвета объекта можно посмотреть в окне свойств и скопировать его в программный код. Аналогично создаем процедуру для второй кнопки.

```
Private Sub CommandButton2_Click()  
    Dim s As String  
    s = Label1.Caption      'считываем текст в метке  
    If s = "Hello!" Then Label1.Caption = "Привет!"    'проверяем и  
    If s = "Привет!" Then Label1.Caption = "Hello!"    'меняем текст  
End Sub
```

Создаем процедуру обработки события *MouseDown* (нажатие кнопки мыши) для второй метки. Для этого выберем из списка объектов объект *Label2*, а из списка событий событие *MouseDown*.

```
Private Sub Label2_MouseDown()  
    Label2.Font.Bold = True      'делаем жирный шрифт  
End Sub
```

Аналогично создаем процедуру для события *MouseUp* (отпустить кнопку мыши).

```
Private Sub Label2_MouseUp()  
    Label2.Font.Bold = False      'убираем жирный шрифт  
End Sub
```

Пример 2

На форме расположены метка с текстом «Введите свое имя», поле для ввода имени, пустая метка для вывода приветствия и две кнопки: «Ввод» и «Выход». Пользователь вводит в поле свое имя и нажимает на кнопку «Ввод», которая считывает введенное имя и выводит его в пустую метку в виде приветствия «Hello, имя!». Кнопка «Выход» завершает работу приложения. Внешний вид

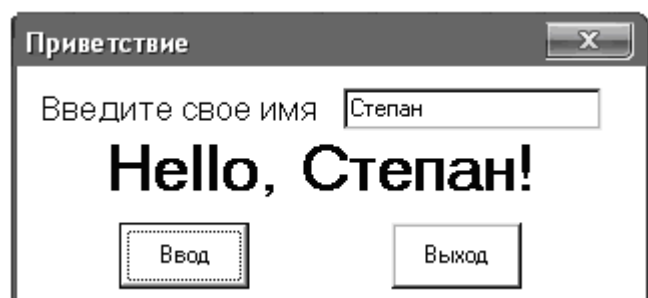


Рис. 5. Внешний вид формы

формы после нажатия кнопки «Ввод» показан на рис. 5.

После создания объектов и задания их свойств (*Caption*, *Font*) нужно написать процедуры обработки события *Click* для кнопок.

```
Private Sub CommandButton1_Click()  
    Dim s As String  
    s = TextBox1.Text      'считываем введенное имя из поля  
    'выводим в метку приветствие  
    Label1.Caption = "Hello, " & s & "!"  
End Sub
```

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

Пример 3

На форме расположены рисунок самолета и кнопка «Посадка». После нажатия на кнопку рисунок перемещается в левый нижний угол формы, и кнопка становится недоступна. Внешний вид формы показан на рис. 6.

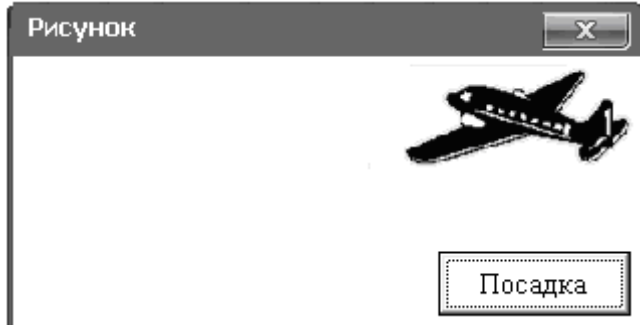


Рис. 6. Внешний вид формы
формы (без рамки и заголовка) и высотой рисунка.

Горизонтальная координата рисунка должна стать равной нулю, а вертикальная координата вычисляется как разность между истинной высотой

```
Private Sub CommandButton1_Click()  
    Image1.Left = 0  
    Image1.Top = UserForm1.InsideHeight - Image1.Height  
    CommandButton1.Enabled = False  
End Sub
```

Пример 4

На форме расположены три рисунка: станция технического осмотра, машина, канистра с бензином (временно невидимая), а также три кнопки: «На станцию» и временно недоступные «Назад» и «Заправка». Кнопка «На станцию» изменяет координаты рисунка с машиной так, чтобы «машина оказалась на станции», делает рисунок с канистрой видимым, а кнопки «Назад» и «Заправка» – доступными. Кнопка «Назад» возвращает машину на прежнее место (правый нижний угол формы), делает рисунок канистры невидимым, а кнопки «Назад» и «Заправка» – недоступными. Кнопка «Заправка» уменьшает размеры рисунка канистры на 10 пикселей, если это возможно.

Объекты и их свойства, определяемые на этапе проектирования, показаны в табл. 6.

Таблица 6

Перечень объектов и их свойств

Объекты	Свойства	Значения свойств
Форма UserForm1	Caption	ГАРАЖ
Рисунок Image1 (станция)	Picture PictureSizeMode	C:\... путь к файлу с рисунком Stretch
Рисунок Image2 (машина)	Picture PictureSizeMode	C:\... путь к файлу с рисунком Stretch
Рисунок Image3 (канистра)	Picture PictureSizeMode Visible	C:\... путь к файлу с рисунком Stretch False
Кнопка CommandButton1	Caption	На станцию
Кнопка CommandButton2	Caption Enabled	Назад False
Кнопка CommandButton3	Caption Enabled	Заправка False

Внешний вид формы на разных стадиях выполнения проекта показан на рис. 7.



Рис. 7. Вид формы при ее загрузке и после нажатия кнопки «На станцию»

Чтобы поменять порядок наложения объектов друг на друга нужно щелкнуть по объекту правой кнопкой мыши и выбрать из контекстного меню нужную команду: **Bring Forward** (на передний план) или **Send Backward** (на задний план).

Напишем процедуры обработки события *Click* для кнопок.

```
Private Sub CommandButton1_Click()
    Image2.Left = 50    'меняем координаты рисунка машины
    Image2.Top = 100
    Image3.Visible = True 'делаем видимой рисунок канистры
    CommandButton2.Enabled = True    'делаем кнопку доступной
    CommandButton3.Enabled = True
End Sub
```

```
Private Sub CommandButton2_Click()
    'перемещаем рисунок машины в правый нижний угол формы
    Image2.Left = UserForm1.InsideWidth - Image2.Width
    Image2.Top = UserForm1.InsideHeight - Image2.Height
    Image3.Visible = False    'скрываем рисунок канистры
    CommandButton2.Enabled = False    'делаем кнопки
    CommandButton3.Enabled = False    'недоступными
End Sub
```

```
Private Sub CommandButton3_Click()
    'если размеры рисунка позволяют, то уменьшаем их на 10
    If Image3.Width > 10 And Image3.Height > 10 Then
```

```

Image3.Width = Image3.Width - 10
Image3.Height = Image3.Height - 10
End If
End Sub

```

Пример 5

На форме расположены две метки с пояснительным текстом, два поля для ввода чисел, кнопка «Вычислить», вычисляющая максимальное из чисел, и метка для вывода результата. Внешний вид формы после нажатия кнопки «Вычислить» показан на рис. 8.

```

Private Sub CommandButton1_Click()
    Dim a As Integer, b As Integer
    Dim max As Integer
    a = TextBox1.Text
    b = TextBox2.Text
    If a = b Then
        Label3.Caption = "Числа равны"
    Else
        If a > b Then max = a Else Max = b
        Label3.Caption = "Максимум=" & max
    End If
End Sub

```

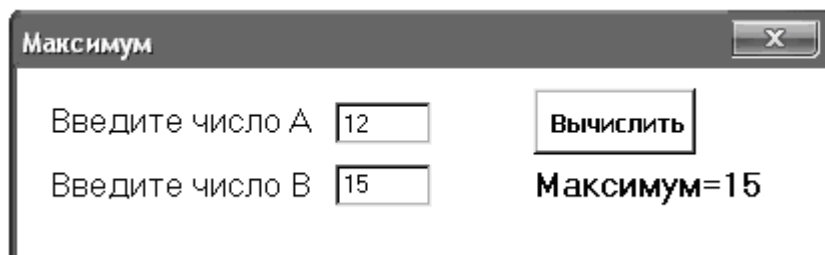


Рис. 8. Внешний вид формы

Пример 6

На форме расположены метка с текстом «Введите имя друга», поле для ввода имени, метка с текстом «Мои друзья», список для

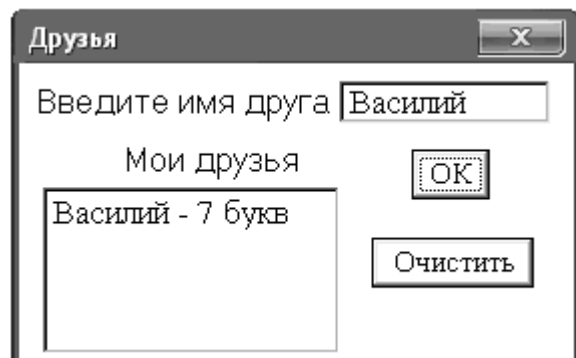


Рис. 9. Внешний вид формы

вывода в него имен и две кнопки: «ОК» и «Очистить». Пользователь вводит в поле имя и нажимает на кнопку «ОК». При этом введенное имя и количество букв в нем добавляется в список, поле очищается и в нем устанавливается текстовый курсор. Кнопка «Очистить» удаляет все элементы списка. Внешний вид формы после нажатия кнопки «ОК» показан на рис. 9.

```
Private Sub CommandButton1_Click()  
    Dim s As Byte  
    'определяем длину строки (количество букв в имени)  
    s = TextBox1.TextLength  
    'добавляем из поля в список имя и количество букв в нем  
    ListBox1.AddItem (TextBox1.Text & " – " & s & " букв")  
    TextBox1.Text = ""          'очищаем поле  
    TextBox1.SetFocus          'устанавливаем фокус в поле  
End Sub
```

```
Private Sub CommandButton2_Click()  
    ListBox1.Clear          'очищаем список  
End Sub
```

Пример 7

На форме расположены две надписи: «Товары» и «Выберите товар», список, поле со списком (комбинированный список) и три кнопки: «Добавить», «Удалить» и «Очистить». Пользователь выбирает из поля со списком товар или вводит его непосредственно в поле. При нажатии на кнопку «Добавить» выбранный товар добавляется в список. Таким образом, можно добавить в список нужные товары. Кнопка «Удалить» удаляет из списка выбранный товар. Кнопка «Очистить» удаляет из списка все товары. Внешний вид формы показан на рис. 10.



Рис. 10. Внешний вид формы

Напишем процедуру обработки события *Activate* для формы, чтобы при ее загрузке в поле со списком появился перечень товаров.

```
Private Sub UserForm_Activate()  
    ComboBox1.AddItem ("Модем")  
    ComboBox1.AddItem ("Монитор")  
    ComboBox1.AddItem ("Принтер")  
    ComboBox1.AddItem ("Сканер")  
End Sub
```

Теперь напишем процедуры обработки события *Click* для кнопок.

```
Private Sub CommandButton1_Click()  
    ListBox1.AddItem (ComboBox1.Text)  
End Sub
```

```
Private Sub CommandButton2_Click()  
    ListBox1.RemoveItem(ListBox1.ListIndex)  
End Sub
```

```
Private Sub CommandButton3_Click()  
    ListBox1.Clear  
End Sub
```

Пример 8

На форме расположены список стран, метка для вывода приветствия, метка с текстом «Выберите страну» и кнопка «Выход». После выбора пользователем страны на форме появляется приветствие на соответствующем языке. Внешний вид формы после выбора страны показан на рис. 11.

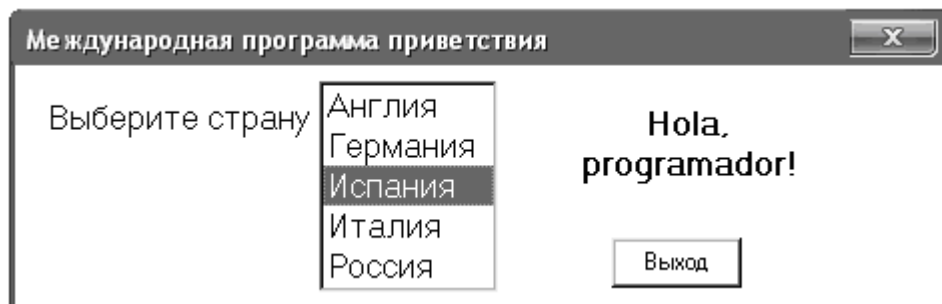


Рис. 11. Внешний вид формы

```
Private Sub UserForm_Activate()  
    ListBox1.AddItem ("Англия")  
    ListBox1.AddItem ("Германия")  
    ListBox1.AddItem ("Испания")  
    ListBox1.AddItem ("Италия")  
    ListBox1.AddItem ("Россия")  
End Sub
```

```
Private Sub ListBox1_Click()  
    Dim S As String, N As Integer  
    N = ListBox1.ListIndex    'номер выбранного элемента списка  
    'проверяем номер выбранной страны (нумерация с нуля)  
    Select Case N  
        Case 0  
            S = "Hello, programmer!"  
        Case 1  
            S = "Hello, programmierer!"  
        Case 2  
            S = "Hola, programador!"  
        Case 3  
            S = "Ciao, programmer!"  
        Case 4  
            S = "Привет, программист!"  
    End Select  
    Label1.Caption = S    'выводим текст в метку  
End Sub
```

```
Private Sub CommandButton1_Click()  
    End  
End Sub
```

Пример 9

На форме расположены три рамки: «Увлечения», «Образование», «Пол»; шесть флажков; пять переключателей, кнопка «ОК». Пользователь выбирает нужные флажки и переключатели и нажимает на кнопку «ОК». На рабочем листе выводятся выбранные данные в

виде таблицы. Внешний вид формы после выбора флажков и переключателей показан на рис. 12.

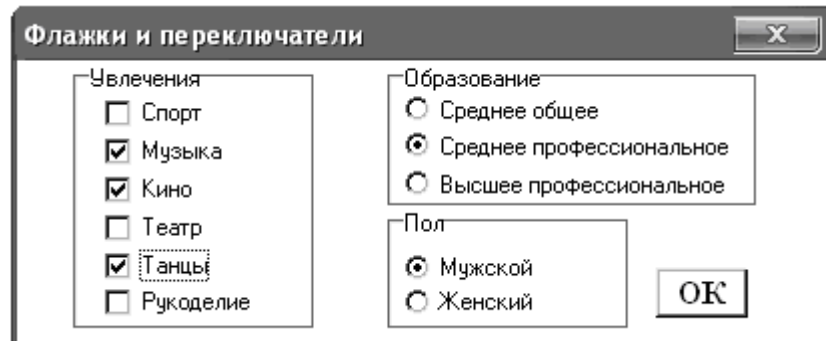


Рис. 12. Внешний вид формы

Зададим свойства *Caption* и *Font* для всех объектов, свойство *Value* для первого переключателя в группе «Пол» должно иметь значение *True* (один из переключателей обязательно должен быть выбран).

Теперь нужно написать процедуру обработки события *Click* для кнопки.

```
Private Sub CommandButton1_Click()
    Dim i As Byte, s As String
    Range("A1:C10").Clear      'очищаем ячейки листа Excel
    Cells(1, 1) = "Увлечения"
    i = 2      'номер строки
    If CheckBox1.Value = True Then 'проверяем первый флажок
        Cells(i, 1) = CheckBox1.Caption 'выводим название увлечения
        i = i + 1      'увеличиваем номер строки
    End If
    If CheckBox2.Value = True Then 'проверяем второй флажок
        Cells(i, 1) = CheckBox2.Caption
        i = i + 1
    End If
    If CheckBox3.Value = True Then
        Cells(i, 1) = CheckBox3.Caption
        i = i + 1
    End If
    If CheckBox4.Value = True Then
        Cells(i, 1) = CheckBox4.Caption
        i = i + 1
    End If
End Sub
```

```

End If
If CheckBox5.Value = True Then
    Cells(i, 1) = CheckBox5.Caption
    i = i + 1
End If
If CheckBox6.Value = True Then
    Cells(i, 1) = CheckBox6.Caption
    i = i + 1
End If
Cells(1, 2) = "Образование"
If OptionButton1.Value = True Then s = OptionButton1.Caption
If OptionButton2.Value = True Then s = OptionButton2.Caption
If OptionButton3.Value = True Then s = OptionButton3.Caption
Cells(2, 2) = s
Cells(1, 3) = "Пол"
If OptionButton4.Value = True Then
    Cells(2, 3) = "М"
Else
    Cells(2, 3) = "Ж"
End If
End Sub

```

В результате работы данной программы получим таблицу, показанную на рис. 13.

	А	В	С
1	Увлечения	Образование	Пол
2	Музыка	Среднее профессиональное	М
3	Кино		
4	Танцы		

Рис. 13. Результат работы программы

Пример 10

На рабочем листе «Товары» составлена таблица с данными о товарах (рис. 14) и расположена кнопка «Запуск» для открытия формы.

	А	В	С	Д
1	Наименование	Цена		
2	Монитор	9000		
3	Клавиатура	300		
4	Мышь	150		
5	Процессор	2500		
6	Принтер	3000		

Рис. 14. Фрагмент листа «Товары»

На форме расположены: поле со списком для выбора товара, метка для вывода цены товара, поле для ввода количества, рамка и два переключателя для выбора вида оплаты (один из них включен), рамка и два флажка для выбора услуг, кнопка «Заказать», метки с пояснительным текстом. Вычислить стоимость каждого выбранного товара с учетом вида оплаты (при безналичном расчете стоимость товара увеличивается на 2 %) и общую сумму заказа, вывести на рабочий лист «Заказ» таблицу о заказах. Внешний вид формы после ввода данных показан на рис. 15.

Рис. 15. Внешний вид формы

Dim i As Byte, sum As Double 'глобальные переменные

Private Sub UserForm_Activate()

Sheets("Товары").Select

For i = 2 To 6

 ComboBox1.AddItem Cells(i, 1) 'формируем список товаров

Next i

Sheets("Заказ").Select

'очищаем нужный диапазон ячеек рабочего листа

```

Range(Cells(2, 1), Cells(20, 7)).Clear
‘выводим заголовки столбцов таблицы
Cells(1, 1) = “Наименование”
Cells(1, 2) = “Цена”
Cells(1, 3) = “Количество”
Cells(1, 4) = “Стоимость”
Cells(1, 5) = “С учетом вида оплаты”
Cells(1, 6) = “Доставка”
Cells(1, 7) = “Установка”
i = 2 ‘начальное значение номера строки для вывода товаров
sum = 0 ‘начальное значение итоговой суммы
End Sub

```

```

Private Sub CommandButton1_Click()
Dim c As Double, k As Integer, st As Double
Dim stv As Double, n As Byte, t As String, v As Double
Sheets(“Товары”).Select
n = ComboBox1.ListIndex + 2 ‘номер выбранного товара
‘считываем соответствующее название и цену товара с листа
t = Cells(n, 1)
c = Cells(n, 2)
‘выводим на форму цену товара
Label3.Caption = “Цена ” & c & “ руб.”
k = TextBox1.Text ‘считываем количество из поля
‘проверяем вид оплаты и устанавливаем коэффициент
If OptionButton1.Value = True Then v = 1 Else v = 1.02
st = c * k ‘вычисляем стоимость товара
stv = st * v ‘стоимость товара с учетом вида оплаты
Sheets(“Заказ”).Select
Cells(i, 1) = t ‘выводим данные в таблицу на рабочем листе
Cells(i, 2) = c
Cells(i, 3) = k
Cells(i, 4) = st
Cells(i, 5) = stv
If CheckBox1.Value = True Then ‘проверяем услуги
Cells(i, 6) = “Доставка”
Else
Cells(i, 6) = “-”
End If
End Sub

```

```

If CheckBox2.Value = True Then
    Cells(i, 7) = "Установка"
Else
    Cells(i, 7) = "-"
End If
i = i + 1          'увеличиваем номер строки на 1
sum = sum + stv   'вычисляем итоговую сумму
Cells(i, 4) = "Итого:"
Cells(i, 5) = sum
End Sub

```

Программный код кнопки «Запуск» на рабочем листе Excel.

```

Private Sub CommandButton1_Click()
    UserForm1.Show
End Sub

```

Фрагмент рабочего листа «Заказ» после работы с программой показан на рис. 16.

	A	B	C	D	E	F	G
1	Наименование	Цена	Кол-во	Стоимость	С учетом вида оплаты	Доставка	Установка
2	Монитор	9000	2	18000	18360	Доставка	Установка
3	Клавиатура	300	2	600	600	-	-
4	Процессор	2500	1	2500	2550	-	Установка
5	Принтер	3000	1	3000	3060	Доставка	-
6				Итого:	24570		

Рис. 16. Фрагмент листа «Заказ» после работы программы

Пример 11

На форме расположены кнопка «Заполнить», заполняющая на первом рабочем листе Excel таблицу из N целых случайных чисел в диапазоне $[-10; 10]$, поле для ввода числа N , кнопка «Вычислить», вычисляющая сумму чисел и максимальное из них, и метки для вывода результатов.

Рис. 17. Внешний вид формы

Внешний вид формы показан на рис. 17.

```
Dim i As Byte, n As Byte
```

```
Private Sub CommandButton1_Click()
```

```
    n = TextBox1.Text
```

```
    Randomize
```

```
    Sheets("Лист1").Select
```

```
    'заполняем ячейки листа случайными числами
```

```
    For i = 1 To n
```

```
        Cells(1, i) = Int(Rnd * 20 - 10)
```

```
    Next i
```

```
End Sub
```

```
Private Sub CommandButton2_Click()
```

```
Dim s As Integer, max As Integer
```

```
    s = 0
```

```
    max = -100
```

```
    For i = 1 To n
```

```
        s = s + Cells(1, i) 'вычисляем сумму
```

```
        'вычисляем максимум
```

```
        If Cells(1, i) > max Then max = Cells(1, i)
```

```
    Next i
```

```
    'выводим результаты на форму
```

```
    Label2.Caption = "s=" & s
```

```
    Label3.Caption = "max=" & max
```

```
End Sub
```

Пример 12

На форме расположен набор из двух вкладок, метка и кнопка. При выборе вкладки в метке выводится текст приветствия, а на

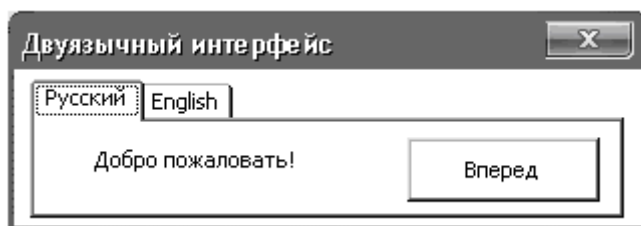


Рис. 18. Внешний вид формы

кнопке отображается надпись на соответствующем языке (в зависимости от выбранной вкладки). Внешний вид формы показан на рис. 18.

Для того чтобы переименовать вкладку, нужно щелкнуть по ней

правой кнопкой мыши и выбрать команду **Rename**. При загрузке формы должна активизироваться первая вкладка. Чтобы в момент загрузки формы наступало событие *Change* для объекта *TabStrip1* (происходила смена вкладок) следует на стадии проектирования в окне свойств установить значение свойства *Value* этого объекта, равным 1 (выбрать вторую вкладку).

```
Private Sub UserForm_Activate()  
    TabStrip1.Value = 0    'переходим на первую вкладку  
End Sub
```

```
Private Sub TabStrip1_Change()    'при выборе вкладки  
    If TabStrip1.Value = 0 Then    'проверяем вкладку  
        Label1.Caption = "Добро пожаловать!"  
        CommandButton1.Caption = "Вперед"  
    End If  
    If TabStrip1.Value = 1 Then  
        Label1.Caption = "Welcome!"  
        CommandButton1.Caption = "Next"  
    End If  
End Sub
```

Пример 13

На форме расположен набор из двух страниц. На странице «ФИО» размещены метки с пояснительным текстом, три поля для ввода данных и кнопка «Далее», считывающая введенные данные и осуществляющая переход на вторую страницу. На странице «Паспорт» размещены метки с пояснительным текстом, два поля для ввода данных и кнопка «Ввод», формирующая отчет по введенным данным на рабочем листе Excel. Внешний вид формы показан на рис. 19.

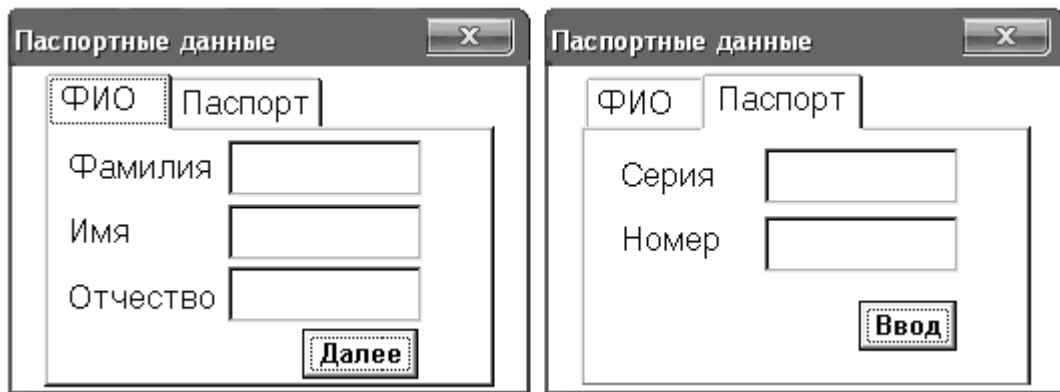


Рис. 19. Внешний вид формы для разных страниц

Dim i As Integer, a As String, b As String, c As String

```
Private Sub UserForm_Activate()
    i = 1    'счетчик номера строки
    MultiPage1.Value = 0    'выбираем первую страницу
End Sub
```

```
Private Sub CommandButton1_Click()
    'считываем данные из полей
    a = TextBox1.Text
    b = TextBox2.Text
    c = TextBox3.Text
    TextBox1.Text = ""    'очищаем поле
    TextBox2.Text = ""
    TextBox3.Text = ""
    MultiPage1.Value = 1    'переходим на вторую страницу
End Sub
```

```
Private Sub CommandButton2_Click()
    'считываем данные из полей и выводим их на рабочий лист
    Cells(i, 4) = TextBox4.Text & " " & TextBox5.Text
    Cells(i, 1) = a
    Cells(i, 2) = b
    Cells(i, 3) = c
    TextBox4.Text = ""    'очищаем поле
    TextBox5.Text = ""
    MultiPage1.Value = 0    'возвращаемся на первую страницу
    i = i + 1    'увеличиваем счетчик строк
End Sub
```

Пример 14

На форме расположены три метки, полоса прокрутки и счетчик. При прокрутке или перемещении бегунка на полосе прокрутки или при прокрутке счетчика в метках отображаются соответствующие значения свойства *Value*. Внешний вид формы после изменения счетчика и переноса бегунка на полосе вправо, а затем прокрутки полосы влево показан на рис. 20.



Рис. 20. Внешний вид формы

Объекты и их свойства, определяемые на этапе проектирования в окне свойств, показаны в табл. 7.

Таблица 7

Перечень объектов и их свойств

Объекты	Свойства	Значения свойств
Label1, Label2, Label3	Caption BorderStyle TextAlign	– 1 2
ScrollBar1	SmallChange LargeChange Max	1 5 100
SpinButton1	SmallChange Max	1 100

Событие *Change* происходит после каждого изменения значения полосы прокрутки. Если перетаскивать бегунок с помощью мыши, новое значение полосы прокрутки будет сгенерировано только после того, как отпустить кнопку мыши. Событие *Scroll* изменяет значение полосы прокрутки сразу. У счетчика событие *Scroll* отсутствует.

Dim s As Integer

‘вспомогательная переменная

```
Private Sub ScrollBar1_Change()  
    s = ScrollBar1.Value    'считываем значение полосы прокрутки  
    Label1.Caption = "Значение после события Change" & s  
End Sub
```

```
Private Sub ScrollBar1_Scroll()  
    s = ScrollBar1.Value  
    Label2.Caption = "Значение после события Scroll" & s  
End Sub
```

```
Private Sub SpinButton1_Change()  
    Label3.Caption = SpinButton1.Value  
End Sub
```

Пример 15

На форме расположены 4 выключателя, 2 рисунка, 2 метки. Выключатель «visible» делает первый рисунок видимым или не видимым. Выключатель «left» смещает второй рисунок влево на 10 пикселей и обратно. Выключатель «color» делает цвет первой надписи красным или белым. Выключатель «size» увеличивает размер шрифта во второй надписи до 14 пт или возвращает прежний размер 10 пт. Внешний вид формы показан на рис. 21.

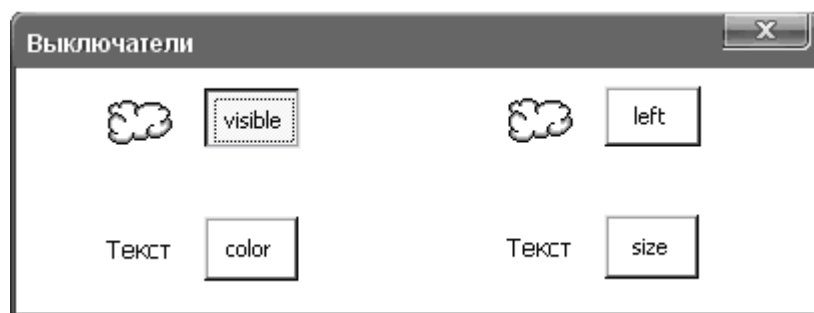


Рис. 21. Внешний вид формы

```
Private Sub ToggleButton1_Change()  
    If ToggleButton1.Value = True Then  
        Image1.Visible = True    'показываем рисунок  
    Else  
        Image1.Visible = False    'скрываем рисунок  
    End If
```

End Sub

```
Private Sub ToggleButton2_Change()  
    If ToggleButton2.Value = True Then  
        Image2.Left = Image2.Left - 10    'сдвигаем рисунок влево  
    Else  
        Image2.Left = Image2.Left + 10    'возвращаем обратно  
    End If  
End Sub
```

```
Private Sub ToggleButton3_Change()  
    If ToggleButton3.Value = True Then  
        Label1.BackColor = RGB(255, 0, 0)    'красный цвет  
    Else  
        Label1.BackColor = RGB(255, 255, 255)    'белый цвет  
    End If  
End Sub
```

```
Private Sub ToggleButton4_Change()  
    If ToggleButton4.Value = True Then  
        Label2.Font.Size = 14    'размер шрифта 14  
    Else  
        Label2.Font.Size = 10    'размер шрифта 10  
    End If  
End Sub
```

Пример 16

На форме расположены шесть кнопок для выполнения следующих действий:

- «Вычислить» – вычисление значений функции $y = \sin x$ на отрезке $[-2; 2]$ с шагом 0,3 и вывод их на рабочий лист «Таблица»;
- «Формат ячеек» – форматирование таблицы;
- «Очистить» – удаление таблицы;
- «График» – построение графика функции на отдельном листе с именем «График»;
- «Формат графика» – форматирование диаграммы;
- «Удалить» – удаление листа с графиком.

Внешний вид формы показан на рис. 22.



Рис. 22. Внешний вид формы

Напишем процедуру обработки события *Click* для первой кнопки.

```
Private Sub CommandButton1_Click()      'кнопка «Вычислить»
    Dim x As Single, y As Single, i As Byte
    Sheets("Таблица").Select
    Cells(1, 1) = "x"      'выводим заголовок таблицы
    Cells(1, 2) = "y"
    i = 2                  'счетчик номеров строк
    For x = -2 To 2 Step 0.3      'задаем значения аргумента функции
        y = sin(x)              'вычисляем значение функции
        Cells(i, 1) = x        'выводим на лист таблицу значений
        Cells(i, 2) = y        'x и y
        i = i + 1              'увеличиваем счетчик строк
    Next x
End Sub
```

Запишем макрос, выполняющий форматирование таблицы, и скопируем его команды в процедуру кнопки «Формат ячеек». Процедура может выглядеть следующим образом.

```
Private Sub CommandButton2_Click()
    Sheets("Таблица").Select      'выделяем лист
    Range("A1:B15").Select        'выделяем ячейки
    With Selection.Borders(xlEdgeLeft) 'левая граница
        .LineStyle = xlContinuous
    End With
    With Selection.Borders(xlEdgeTop) 'верхняя граница
        .LineStyle = xlContinuous
    End With
```

```

With Selection.Borders(xlEdgeBottom)  ‘нижняя граница
    .LineStyle = xlContinuous
End With
With Selection.Borders(xlEdgeRight)    ‘правая граница
    .LineStyle = xlContinuous
End With
With Selection.Borders(xlInsideVertical) ‘вертикальная сетка
    .LineStyle = xlContinuous
End With
With Selection.Borders(xlInsideHorizontal) ‘горизонтальная
    .LineStyle = xlContinuous
End With
Range(“A1:B1”).Select                ‘выделяем заголовок таблицы
‘горизонтальное и вертикальное выравнивание – по центру
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
End With
Selection.Font.Bold = True            ‘полужирное начертание
With Selection.Interior                ‘цвет и тип заливки
    .ColorIndex = 6
    .Pattern = xlSolid
End With
With Selection.Font                    ‘шрифт и размер шрифта
    .Name = “Arial Cyr”
    .Size = 14
End With
End Sub

```

Аналогично создадим процедуры для остальных кнопок.

```

Private Sub CommandButton3_Click()  ‘кнопка «Очистить»
    Sheets(“Таблица”).Select        ‘выделяем лист
    Range(“A1:B15”).Clear           ‘очищаем ячейки
End Sub

```

```

Private Sub CommandButton4_Click()  ‘кнопка «График»
    Sheets(“Таблица”).Select
    Range(“A1:B15”).Select
    Charts.Add                       ‘добавляем диаграмму

```

```

‘тип диаграммы – точечная, без маркеров
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
‘диапазон данных
ActiveChart.SetSourceData Source:=Sheets(“Таблица”) _
    .Range(“A1:B15”), PlotBy:=xlColumns
‘расположение диаграммы – новый лист “График”
ActiveChart.Location Where:=xlLocationAsNewSheet, _
    Name:= “График”
With ActiveChart
    .HasTitle = True ‘наличие заголовка
    .ChartTitle.Characters.Text = “y=sin(x)” ‘текст заголовка
    ‘подписываем ось категорий (Ox)
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = “x”
    ‘подписываем ось значений (Oy)
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = “y”
End With
With ActiveChart.Axes(xlCategory)
    .HasMajorGridlines = True ‘линии сетки по оси категорий
End With
With ActiveChart.Axes(xlValue)
    .HasMajorGridlines = True ‘линии сетки по оси значений
End With
ActiveChart.HasLegend = False ‘убираем легенду
End Sub

```

```

Private Sub CommandButton5_Click() ‘кнопка «Формат графика»
    Sheets(“График”).Select
    ActiveChart.PlotArea.Select ‘выделили область построения
    With Selection.Border ‘параметры границы
        .ColorIndex = 16 ‘цвет
        .LineStyle = xlContinuous ‘тип линии
    End With
    Selection.Interior.ColorIndex = xlNone ‘убрали заливку
    ActiveChart.SeriesCollection(1).Select ‘выделили ряд данных
    With Selection.Border
        .ColorIndex = 57 ‘цвет линии
        .Weight = xlThick ‘толщина линии
    End With

```

```

        .LineStyle = xlContinuous    'стиль линии
End With
ActiveChart.Axes(xlCategory).Select    'выделили ось категорий
With Selection.Border                  'форматируем линию оси
    .ColorIndex = 57
    .Weight = xlMedium
    .LineStyle = xlContinuous
End With
With Selection
    .MajorTickMark = xlOutside        'основные метки по оси
    .MinorTickMark = xlNone          'убрать промежуточные метки
    .TickLabelPosition = xlNextToAxis 'положение метки
End With
With ActiveChart.Axes(xlCategory)
    .MinimumScale = -2    'минимальное значение
    .MaximumScale = 2    'максимальное значение
End With
ActiveChart.Axes(xlValue).Select    'выделили ось значений
With Selection.Border                'форматируем линию оси
    .ColorIndex = 57
    .Weight = xlMedium
    .LineStyle = xlContinuous
End With
ActiveChart.ChartTitle.Select    'выделили заголовок диаграммы
Selection.Font.Bold = True        'полужирное начертание
With Selection.Font
    .Name = "Arial Cyr"          'шрифт
    .Size = 14                   'размер шрифта
End With
ActiveChart.Deselect              'снять выделение с диаграммы
End Sub

```

```

Private Sub CommandButton6_Click()    'кнопка «Удалить»
    Sheets("График").Select          'выделили лист с диаграммой
    ActiveWindow.SelectedSheets.Delete 'удалили лист
End Sub

```

Пример 17

На форме расположена метка, в которой выводится текущее время. При загрузке формы включается таймер и идет посекундный отсчет времени. Внешний вид формы показан на рис. 23.

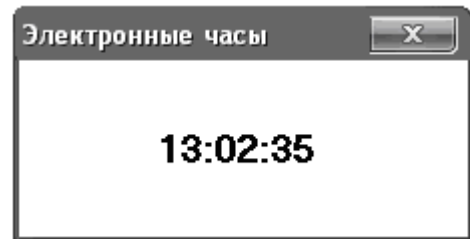


Рис. 23. Внешний вид формы

Воспользуемся функцией *Timer*, которая возвращает число секунд, прошедших с полуночи до текущего момента времени, и функцией *Time*, которая возвращает текущее время по системным часам.

```
Private Sub UserForm_Activate()
    Dim d As Date, d1 As Date
    d1 = Timer 'засекаем время
    While Timer < d1 + 100 'часы будут работать 100 секунд
        Label1.Caption = Time 'отображаем текущее время
        d = Timer
        While Timer < d + 1 'делаем паузу в 1 секунду
            DoEvents 'ожидание наступления события
        Wend
    Wend
End Sub
```

Пример 18

На форме расположены: рисунок с автомобилем, рисунок с гаражом, две кнопки «В гараж» и «Из гаража» на одном и том же месте (видна только кнопка «В гараж»). При нажатии на кнопку «В гараж» автомобиль начинает движение к гаражу, как только он доезжает до гаража, картинка гаража заменяется на картинку с открытыми воротами, автомобиль продолжает движение пока не въедет в гараж, останавливается, ворота закрываются (картинка с гаражом заменяется на прежнюю), кнопка «В гараж» становится невидимой, а кнопка «Из гаража» – видимой. Аналогичные действия происходят при нажатии на кнопку «Из гаража», только движение осуществляется в другую сторону и рисунок автомобиля повернут в другую сторону. Внешний вид формы до начала работы программы показан на рис. 24.

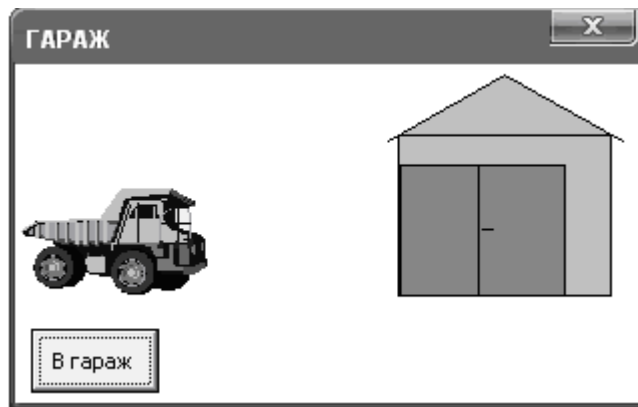


Рис. 24. Внешний вид формы

Dim start As Date

```

Private Sub CommandButton1_Click()    'кнопка «В гараж»
    'загружаем рисунок автомобиля
    Image1.Picture = LoadPicture("C:\машина1.bmp")
    'пока автомобиль не достигнет гаража, смещаем его вправо
    While Image1.Left < Image2.Left - Image1.Width
        Image1.Left = Image1.Left + 5
        'делаем паузу
        start = Timer
        While Timer < start + 0.1
            DoEvents    'ожидает наступления события
        Wend
    Wend
    'открываем ворота (меняем рисунок гаража с закрытыми
    'воротами на рисунок гаража с открытыми воротами)
    Image2.Picture = LoadPicture("C:\гараж2.bmp")
    'пока автомобиль не въедет в гараж, смещаем его вправо
    While Image1.Left < Image2.Left + 10
        Image1.Left = Image1.Left + 5
        start = Timer
        While Timer < start + 0.1
            DoEvents
        Wend
    Wend
    'возвращаем рисунок гаража с закрытыми воротами
    Image2.Picture = LoadPicture("C:\гараж1.bmp")
    Image1.Visible = False    'скрываем рисунок автомобиля

```

```

CommandButton2.Visible = True    'показываем вторую кнопку
CommandButton1.Visible = False  'скрываем первую кнопку
End Sub

```

```

Private Sub CommandButton2_Click()    'кнопка «Из гаража»
    'загружаем рисунок автомобиля
    Image1.Picture = LoadPicture("C:\машина2.bmp")
    Image1.Visible = True    'показываем рисунок
    'загружаем рисунок гаража с открытыми воротами
    Image2.Picture = LoadPicture("C:\гараж2.bmp")
    'автомобиль выезжает из гаража
    While Image1.Left > Image2.Left - Image1.Width
        Image1.Left = Image1.Left - 5
        start = Timer
        While Timer < start + 0.1
            DoEvents
        Wend
    Wend
    'загружаем рисунок гаража с закрытыми воротами
    Image2.Picture = LoadPicture("C:\гараж1.bmp")
    'продолжаем движение автомобиля до левого края формы
    While Image1.Left > 1
        Image1.Left = Image1.Left - 5
        start = Timer
        While Timer < start + 0.2
            DoEvents
        Wend
    Wend
    CommandButton1.Visible = True
    CommandButton2.Visible = False
End Sub

```

Пример 19

На форме расположены список с названиями месяцев, поле и метка с пояснительным текстом. При помощи мыши из списка перетаскивают в поле необходимый элемент, выделенный в списке. При неправильном действии, т.е. перетаскивании не в поле, происходит вывод предупреждающего сообщения. Список формируется при активации формы из значений на рабочем листе

Excel. Внешний вид формы после перемещения и диалоговое окно показаны на рис. 25.

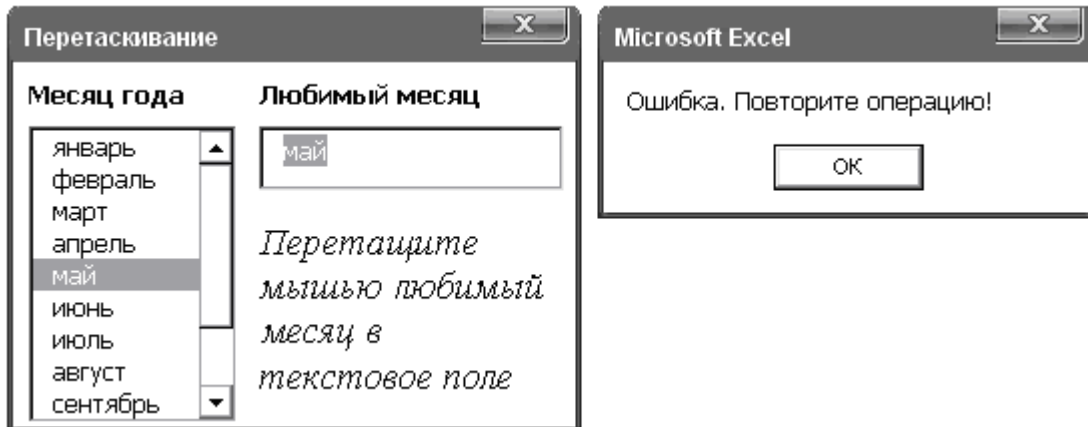


Рис. 25. Внешний вид формы и диалогового окна

Для реализации данного проекта необходимо использовать технологию **DragAndDrop** (прил. 2), позволяющую перемещать объекты. Когда пользователь захватывает выбранный им элемент списка, то возникает некоторое событие, в нашем примере это событие *MouseMove*.

```
Private Sub UserForm_Activate()
    Dim i As Byte
    'формируем список значениями из ячеек рабочего листа
    For i = 1 To 12
        ListBox1.AddItem (Cells(i, 1))
    Next i
End Sub
```

```
Private Sub ListBox1_MouseMove(ByVal Button As Integer)
    Dim D As DataObject
    Dim R As Integer
    If Button = 1 Then 'если копируем объект
        'создает новый объект и связываем его с переменной D
        Set D = New DataObject
        'копируем содержимое выбранного элемента списка
        'в текстовую строку
        D.SetText ListBox1.Value
        R = D.StartDrag 'результат операции
        'если переместили не на текстовое поле
```

```
If R = 0 Then MsgBox ("Ошибка. Повторите операцию!")  
End If  
End Sub
```

Пример 20

На форме расположена кнопка «Начать», нажатие на которую выводит на экран предупреждающее диалоговое окно с текстом «Нажмите кнопку» и кнопкой «ОК» (рис. 26).

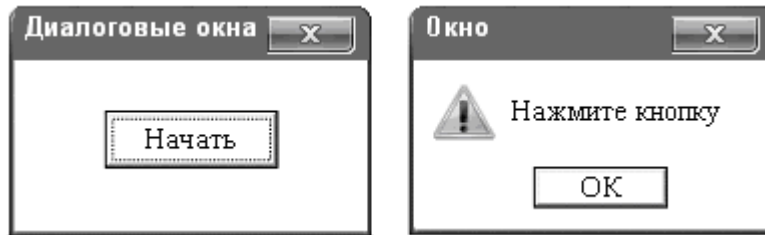


Рис. 26. Внешний вид формы и диалогового окна

После нажатия кнопки «ОК» появляется диалоговое окно с текстом «Начать?» и кнопками «Да» и «Нет». Если пользователь нажал кнопку «Да», то у него запрашивается его имя. Затем, если пользователь ввел имя, выводится информационное диалоговое окно с приветствием, в противном случае выводится предупреждающее диалоговое окно с соответствующим сообщением (рис. 27).

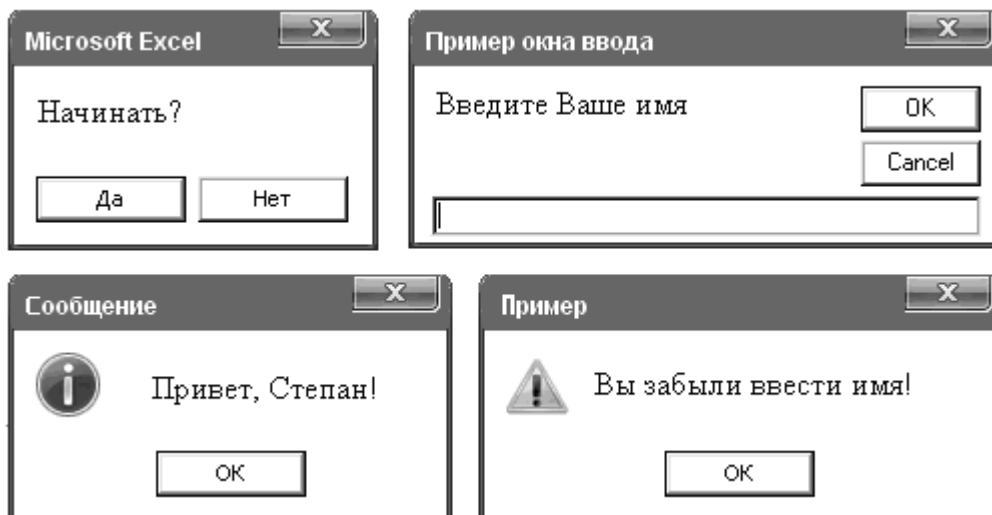


Рис. 27. Внешний вид диалоговых окон и окна ввода

Если пользователь нажал кнопку «Нет», то выводится диалоговое окно с уточнением его действий и кнопками «Да», «Нет» и «Отмена». При нажатии кнопки «Да» выводится диалоговое окно с текстом «Ну наконец-то!» и завершается работа приложения. При нажатии других кнопок выводится диалоговое окно с текстом «Ха-ха!» (рис. 28).

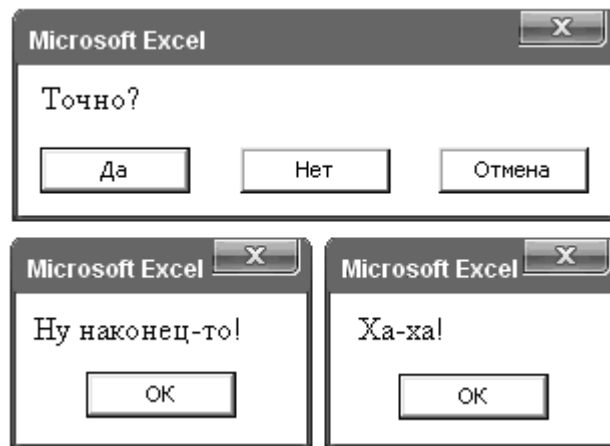


Рис. 28. Внешний вид диалоговых окон

Для реализации данного проекта необходимо использовать различные виды диалоговых окон (прил. 1).

Напишем процедуру обработки события *Click* для кнопки.

```
Private Sub CommandButton1_Click()
    Dim N As String          'переменная для имени пользователя
    'вывод предупреждающего диалогового окна
    MsgBox "Нажмите кнопку!", vbExclamation, "Окно"
    'вывод обычного диалогового окна с кнопками «Да» и «Нет»
    'и проверка нажатия кнопки «Да»
    If MsgBox("Начинать?", vbYesNo) = vbYes Then
        N = InputBox("Введите Ваше имя", "Пример окна ввода")
        If N <> "" Then      'если имя введено
            'выводим информационное окно с приветствием
            MsgBox "Привет, " & N & "!", vbInformation, "Сообщение"
        Else
            'иначе выводим предупреждающее окно
            MsgBox "Вы забыли ввести имя!", vbExclamation, "Пример"
        End If
    Else                    'иначе (нажата не кнопка «Да»)
        'выводим диалоговое окно с кнопками «Да», «Нет»,
```

```

        «Отмена» и проверяем нажатие кнопки «Да»
    If MsgBox(“Точно?”, vbYesNoCancel) = vbYes Then
        MsgBox (“Ну наконец-то!”)
    End
    Else ‘иначе (нажата кнопка «Нет» или «Отмена»)
        MsgBox (“Ха-ха!”)
    End If
End If
End Sub

```

Пример 21

Проект содержит три формы. На первой (главной) форме расположены метка и семь кнопок, образующих главное меню (видны только три основных пункта меню). Структура главного меню: первый пункт «Дата/Время», включающий подпункты «Дата» и «Время» для вывода в метку текущей даты или времени; второй пункт меню «Формат», включающий подпункты «Шрифт» и «Размер» для изменения шрифта и размера шрифта в метке; третий пункт меню «Выход» для завершения работы приложения.

На второй форме расположен список с названиями шрифтов, кнопки «ОК» и «Отмена», метка с пояснительным текстом. На третьей форме расположено поле со списком для выбора размера шрифта, кнопки «ОК» и «Отмена», метка с пояснительным текстом.

Внешний вид главной формы на разных стадиях выполнения проекта показан на рис. 29, внешний вид вспомогательных форм показан на рис. 30.

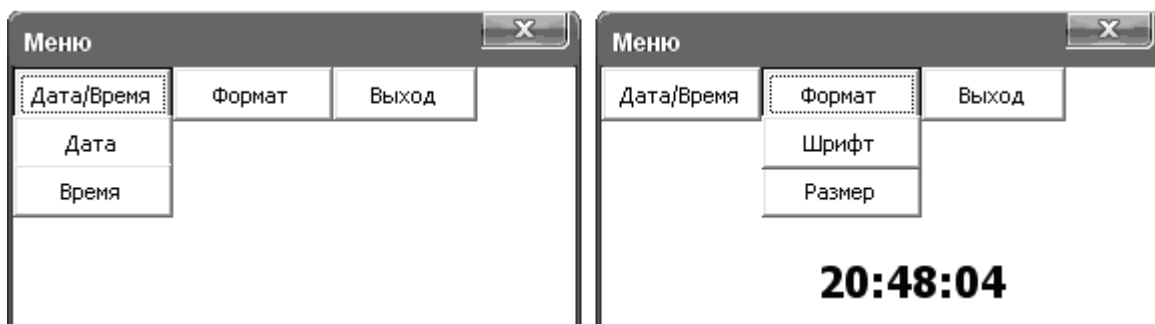


Рис. 29. Внешний вид формы на разных стадиях выполнения проекта

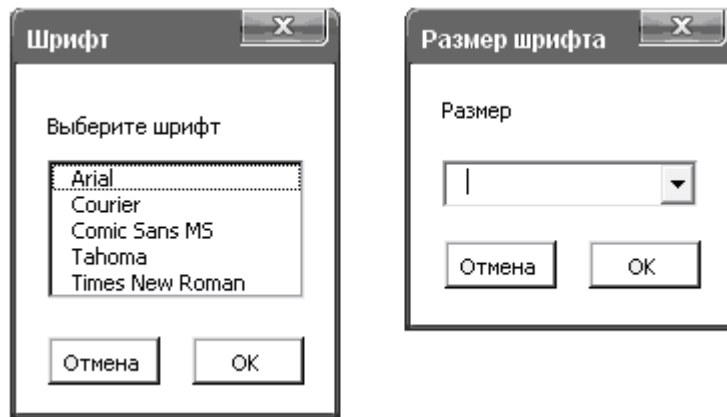


Рис. 30. Внешний вид вспомогательных форм

Напишем процедуры для кнопок на главной форме.

```
Private Sub CommandButton1_Click()    'кнопка «Дата/Время»
    CommandButton2.Visible = True    'показываем кнопку «Дата»
    CommandButton3.Visible = True    'показываем кнопку «Время»
End Sub
```

```
Private Sub CommandButton2_Click()    'кнопка «Дата»
    Label1.Caption = Date             'выводим в метку текущую дату
    CommandButton2.Visible = False    'скрываем пункты меню
    CommandButton3.Visible = False
End Sub
```

```
Private Sub CommandButton3_Click()    'кнопка «Время»
    Label1.Caption = Time             'выводим в метку текущее время
    CommandButton2.Visible = False    'скрываем пункты меню
    CommandButton3.Visible = False
End Sub
```

```
Private Sub CommandButton4_Click()    'кнопка «Формат»
    CommandButton5.Visible = True     'показываем кнопку «Шрифт»
    CommandButton6.Visible = True     'показываем кнопку «Размер»
End Sub
```

```
Private Sub CommandButton5_Click()    'кнопка «Шрифт»
    CommandButton5.Visible = False    'скрываем пункты меню
    CommandButton6.Visible = False
    UserForm2.Show                    'выводим на экран вторую форму
```


End Sub

```
Private Sub CommandButton6_Click()    'кнопка «Размер»
    CommandButton5.Visible = False    'скрываем пункты меню
    CommandButton6.Visible = False
    UserForm3.Show                    'выводим на экран третью форму
End Sub
```

```
Private Sub CommandButton7_Click()    'кнопка «Выход»
    End
End Sub
```

Напишем процедуры для кнопок на второй форме.

```
Private Sub UserForm_Activate()    'формируем список шрифтов
    ListBox1.AddItem "Arial"
    ListBox1.AddItem "Courier"
    ListBox1.AddItem "Comic Sans MS"
    ListBox1.AddItem "Tahoma"
    ListBox1.AddItem "Times New Roman"
End Sub
```

```
Private Sub CommandButton1_Click()    'кнопка «ОК»
    'меняем шрифт в метке на главной форме
    UserForm1.Label1.Font.Name = ListBox1.Text
    UserForm2.Hide                    'скрываем вторую форму
End Sub
```

```
Private Sub CommandButton2_Click()    'кнопка «Отмена»
    UserForm2.Hide
End Sub
```

Напишем процедуры для кнопок на третьей форме.

```
Private Sub UserForm_Activate()
    Dim i As Byte
    For i = 10 To 30 Step 2    'заполняем список числами
        ComboBox1.AddItem i
    Next i
End Sub
```

```
Private Sub CommandButton1_Click()    'кнопка «ОК»  
    'меняем размер шрифта в метке на главной форме  
    UserForm1.Label1.Font.Size = ComboBox1.Text  
    UserForm3.Hide    'скрываем третью форму  
End Sub
```

```
Private Sub CommandButton2_Click()    'кнопка «Отмена»  
    UserForm3.Hide  
End Sub
```

3. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Проект «Мой первый проект»

На форме расположены метка без текста и две кнопки: «Ввод» и «Закреть». При нажатии на кнопку «Ввод» у пользователя запрашивается имя, а затем в метку выводится текст вида «Hello, ИМЯ!». При нажатии на кнопку «Закреть» приложение завершает работу. Внешний вид формы после нажатия кнопки «Ввод» показан на рис. 31.

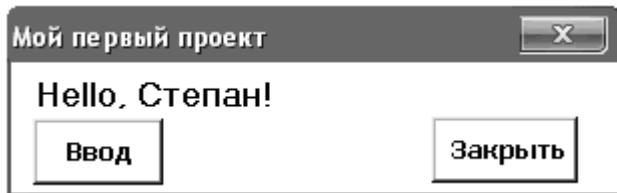


Рис. 31. Внешний вид формы

Проект «Приветствие»

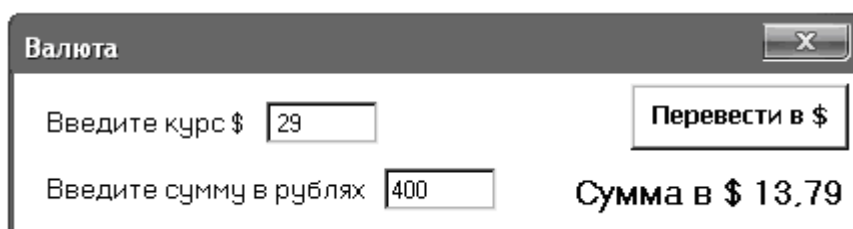
На форме расположены метка с текстом «Введите свое имя», текстовое поле для ввода имени и две кнопки: «Ввод» и «Выход». Пользователь вводит в поле свое имя и нажимает на кнопку «Ввод». Появляется диалоговое окно с приветствием вида «Hello, ИМЯ!». При нажатии на кнопку «Выход» приложение завершает работу. Внешний вид формы показан на рис. 32.



Рис. 32. Внешний вид формы

Проект «Валюта»

На форме расположены метки с пояснительным текстом, поле для ввода курса доллара, поле для ввода суммы в рублях, метка для вывода результата и кнопка «Перевести в \$» для перевода рублей в доллары. Внешний вид формы после вычислений показан на рис. 33. Создать аналогичный проект для перевода долларов в рубли.

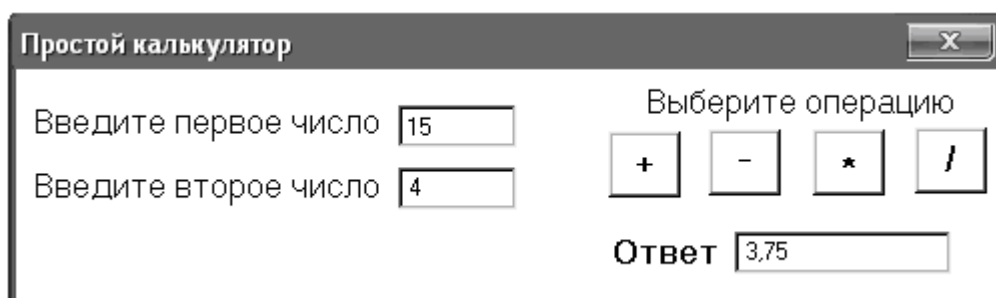


The screenshot shows a window titled "Валюта" with a close button (X) in the top right corner. Inside the window, there are two input fields and one button. The first input field is labeled "Введите курс \$" and contains the value "29". The second input field is labeled "Введите сумму в рублях" and contains the value "400". To the right of the second input field is a button labeled "Перевести в \$". Below the input fields, the text "Сумма в \$ 13.79" is displayed.

Рис. 33. Внешний вид формы

Проект «Простой калькулятор»

На форме расположены метки с пояснительным текстом, два поля для ввода чисел, поле для вывода результата и четыре кнопки с обозначениями операций. Пользователь вводит два числа и нажимает на нужную кнопку с операцией, на форме появляется ответ. Внешний вид формы после вычисления частного показан на рис. 34.



The screenshot shows a window titled "Простой калькулятор" with a close button (X) in the top right corner. Inside the window, there are two input fields for numbers, a set of operation buttons, and a result field. The first input field is labeled "Введите первое число" and contains the value "15". The second input field is labeled "Введите второе число" and contains the value "4". To the right of the input fields is a label "Выберите операцию" above four buttons: "+", "-", "*", and "/". Below the operation buttons is a label "Ответ" followed by an input field containing the value "3.75".

Рис. 34. Внешний вид формы

Проект «Вычисления»

Даны два числа. Вычислить значение выражения

$$R = \frac{a^2 + 3 \cdot b}{|a + 1| + 5}$$

На форме разместить две метки с пояснительным текстом, два поля для ввода чисел, кнопку «Вычислить» и метку для вывода результата. Результат округлить до двух знаков после запятой. Внешний вид формы после нажатия кнопки «Вычислить» показан на рис. 35.

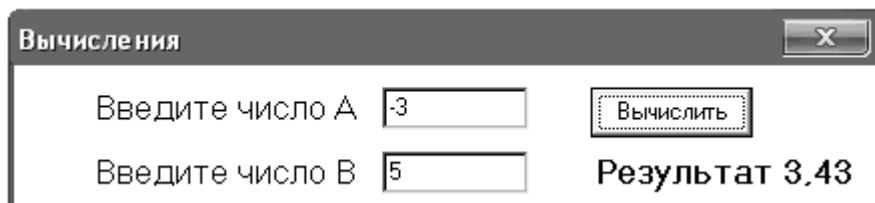


Рис. 35. Внешний вид формы

Проект «Скорость автомобиля»

На форме расположены метка с пояснительным текстом, поле для ввода скорости автомобиля (в км/ч), метка для вывода результата и

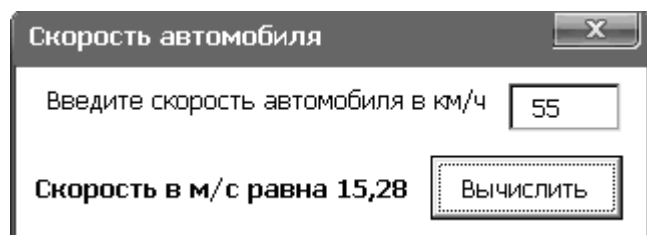


Рис. 36. Внешний вид формы

кнопка «Вычислить». При нажатии на кнопку из поля считывается значение скорости, переводится в м/с и выводится в метку. Внешний вид формы после вычислений показан на рис. 36.

Проект «Скорость»

На форме расположены метки с пояснительным текстом, рамка, три поля для ввода данных, кнопка «Вычислить» и метка для вывода результата. При нажатии на кнопку «Вычислить» из полей считываются данные, вычисляется скорость движения (в км/ч) и выводится в метку. Внешний вид формы после вычислений показан на рис. 37.

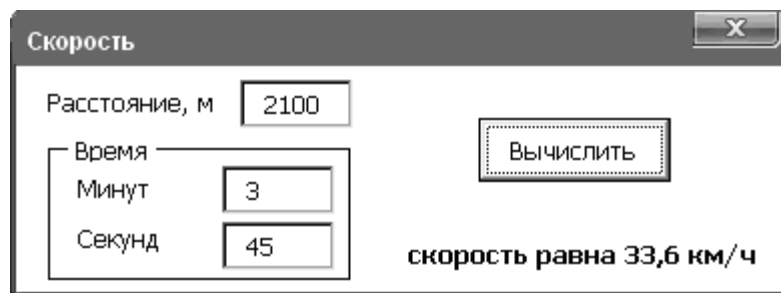


Рис. 37. Внешний вид формы

Проект «Просмотр рисунка»

На форме расположены рисунок и две кнопки: «Увеличить» и «Уменьшить». При нажатии на кнопку «Увеличить» размеры рисунка увеличиваются на 10 пикселей, рисунок смещается влево и вверх на 5 пикселей (таким образом, центр рисунка остается на своем месте). Аналогично при уменьшении рисунка его размеры уменьшаются, и он смещается вправо и вниз. Размеры рисунка не могут превысить размеры формы и не могут стать отрицательными. Внешний вид формы показан на рис. 38.



Рис. 38. Внешний вид формы

Проект «Догони!»

На форму, занимающую весь экран, загружается картинка. При наведении на нее указателя мыши картинка «убегает» – меняет свои координаты случайным образом в пределах формы. Внешний вид формы показан на рис. 39.



Рис. 39. Внешний вид формы

Проект «Самолет»

На форме расположены рисунок и девять кнопок. Щелчок по той или иной кнопке перемещает рисунок в указанном направлении (к краю формы). Координаты рисунка задаются относительно размеров и координат формы. Внешний вид формы показан на рис. 40.

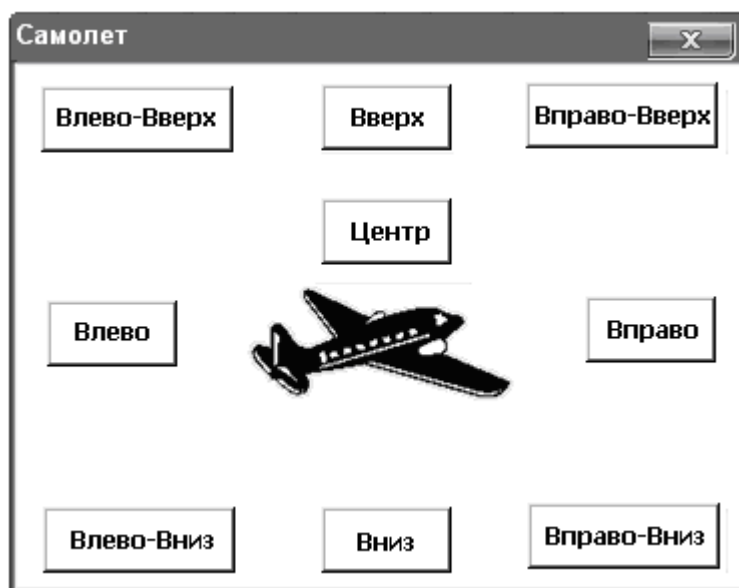


Рис. 40. Внешний вид формы

Проект «Загадка»

На форме расположены временно невидимые метка с текстом загадки, рисунок с изображением отгадки, три кнопки: «Загадка», «Выход» и временно невидимая «Отгадка». При нажатии на кнопку «Загадка» на форме появляется надпись с текстом загадки и кнопка «Отгадка». При нажатии на кнопку «Отгадка» текст загадки и кнопка

«Загадка» исчезают и на форме появляется рисунок с изображением отгадки. Внешний вид формы на разных стадиях выполнения проекта показан на рис. 41.

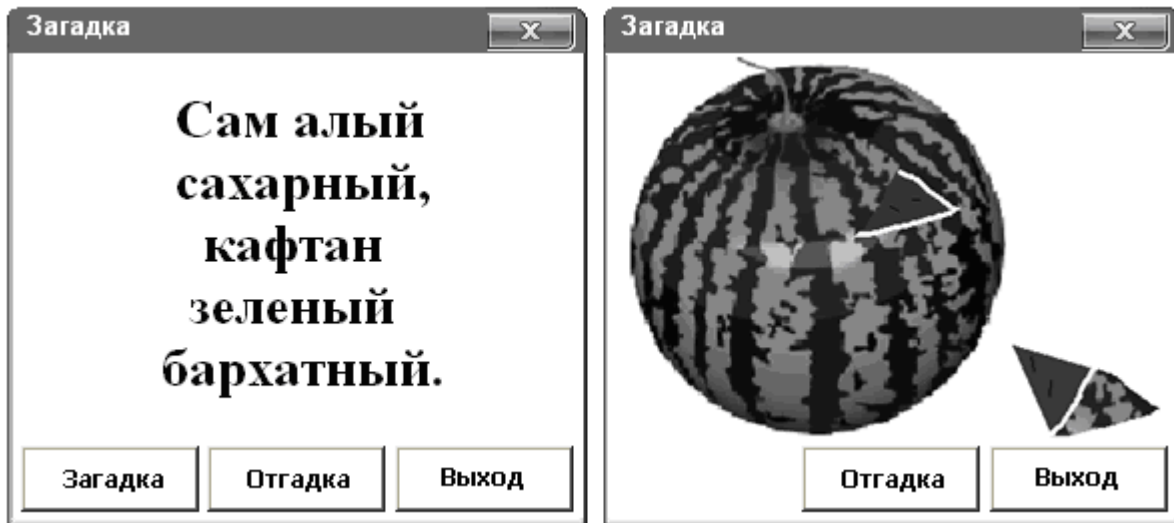


Рис. 41. Внешний вид формы на разных стадиях работы приложения

Изменить проект таким образом, чтобы у пользователя запрашивалась отгадка, и рисунок с изображением появлялся только в случае, если ответ правильный.

Проект «Магазин»

На форме расположены метки с пояснительным текстом, поле для ввода стоимости товара, поле для ввода внесенной покупателем суммы, кнопка «ОК» для вычислений. Требуется определить, хватает ли внесенной покупателем суммы для покупки данного товара, и вывести в диалоговом окне соответствующее сообщение с указанием размера сдачи или недостающей суммы. Внешний вид формы показан на рис. 42.

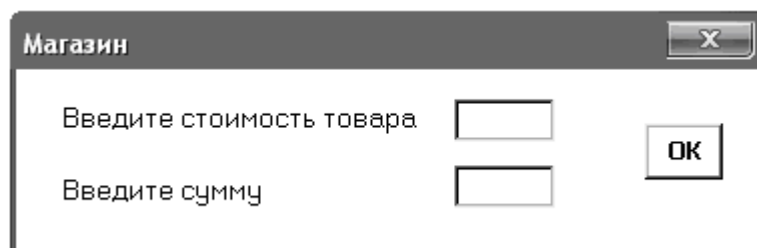


Рис. 42. Внешний вид формы

Проект «КВУР»

На форме расположены три поля для ввода коэффициентов квадратного уравнения, поле для вывода корней уравнения или сообщения о том, что действительных корней нет, метки с пояснительным текстом и две кнопки «Вычислить» и «Выход».

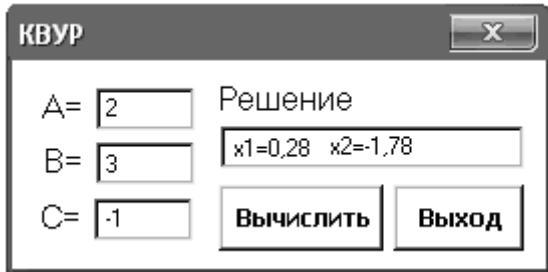


Рис. 43. Внешний вид формы

Проект «Счастливая семерка»

На форме расположены три метки для вывода чисел, метки с пояснительным текстом, метки для вывода количества попыток и побед, временно невидимый рисунок, кнопки «Вращать» и «Выход». Пользователь нажимает на кнопку «Вращать», в метках появляются три цифры (от 0 до 9), сгенерированных случайным образом. Если появилась хотя бы одна семерка, то рисунок становится видимым. Одновременно ведется подсчет количества попыток



Рис. 44. Внешний вид формы

и побед. Внешний вид формы после двух нажатий на кнопку «Вращать» показан на рис. 44.

Проект «Возраст»

На форме расположены метка с пояснительным текстом, поле для ввода возраста, кнопка «ОК» и метка для вывода сообщения, в котором указывается возраст с пояснением (год, года, лет).

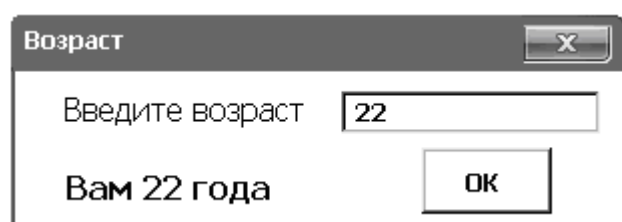


Рис. 45. Внешний вид формы

Внешний вид формы после нажатия кнопки «ОК» показан на рис. 45.

Проект «Оценка»

На форме расположены метка с пояснительным текстом, поле для ввода количества баллов, кнопка «ОК» для вычислений и метка для вывода результата. Требуется определить, какую оценку получил тестируемый, если дана следующая шкала оценки:

от 90 до 100 баллов – оценка «5»;

от 70 до 89 баллов – оценка «4»;

от 51 до 69 баллов – оценка «3»;

менее 51 балла – оценка «2».

Внешний вид формы после вычислений показан на рис. 46.

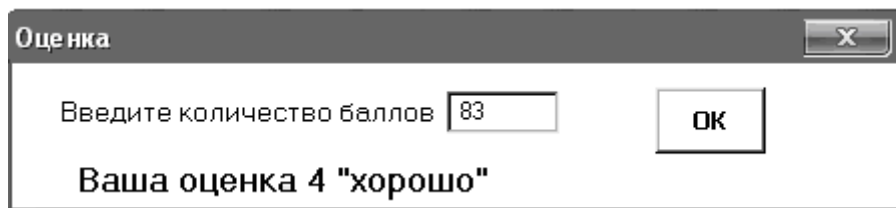


Рис. 46. Внешний вид формы

Проект «Факториал числа»

На форме расположены метка с пояснительным текстом, поле для ввода числа, кнопка «Вычислить», вычисляющая факториал введенного числа, и метка для вывода результата. Внешний вид формы после вычислений показан на рис. 47.

Изменить проект таким образом, чтобы при нажатии на кнопку вычислялась сумма и среднее арифметическое всех целых чисел от 1 до заданного числа.

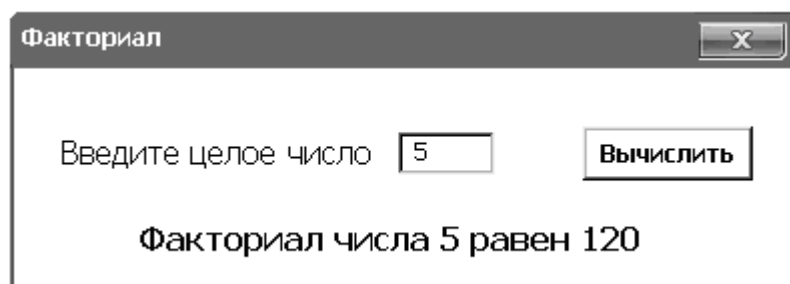


Рис. 47. Внешний вид формы

Проект «Таблица умножения»

На форме расположены метки с пояснительным текстом, поле для ввода числа, список для вывода результатов, кнопки «ОК» и «Очистить». Кнопка «ОК» выводит в список таблицу умножения для заданного числа. Кнопка «Очистить» очищает список. Внешний вид формы после вычислений показан на рис. 48.

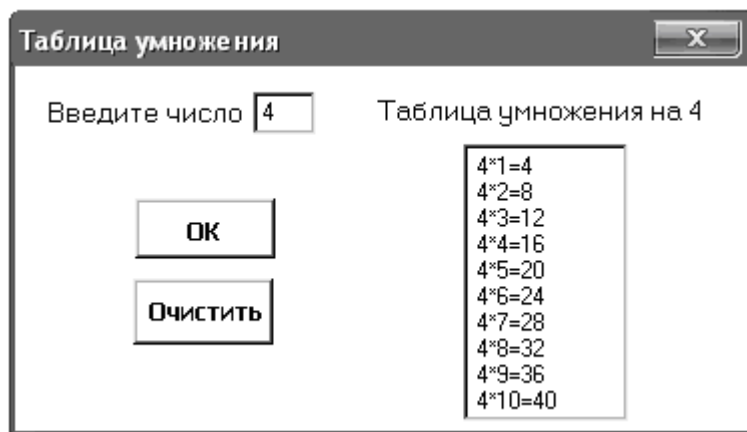


Рис. 48. Внешний вид формы

Проект «Список»

На форме расположены список, кнопки «Добавить элемент», «Удалить элемент» и «Очистить список». При нажатии на кнопку «Добавить элемент» у пользователя запрашивается значение, которое добавляется в список. Кнопка «Удалить элемент» удаляет из списка выбранный элемент. Кнопка «Очистить список» удаляет из списка все элементы. Внешний вид формы показан на рис. 49.

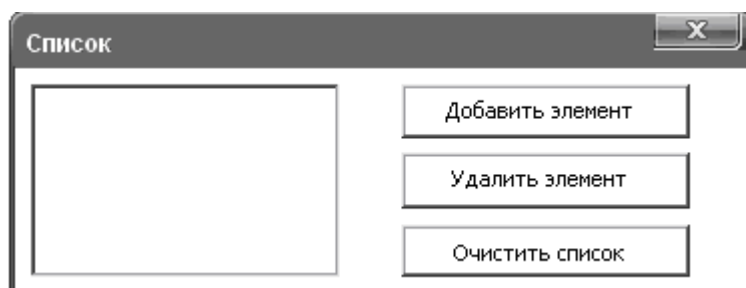


Рис. 49. Внешний вид формы

Проект «Километры – мили»

На форме расположены список и кнопка «ОК», выводящая в список таблицу перевода километров в мили для значений от 1 до 10 км (1 миля = 1,603 км). Внешний вид формы после вычислений показан на рис. 50.

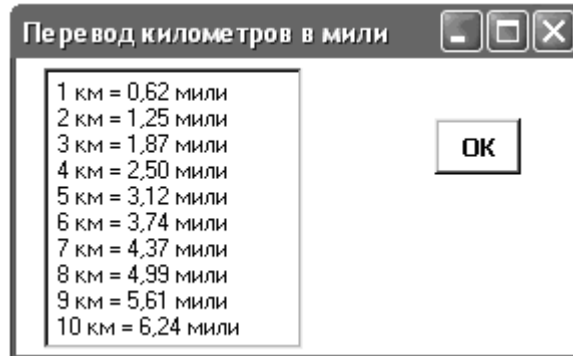


Рис. 50. Внешний вид формы

Проект «Фаренгейт»

На форме расположены метки с пояснительным текстом, поля для ввода начального, конечного значения и шага, кнопка «ОК», переводящая градусы по Цельсию (°C) в градусы по Фаренгейту (F) для всех значений из указанного диапазона ($F = °C \cdot 1,8 + 32$) и выводящая результаты на рабочий лист. Внешний вид формы показан на рис. 51.

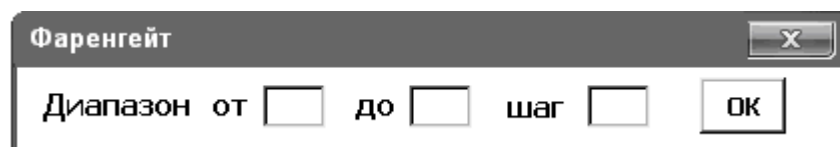


Рис. 51. Внешний вид формы

Проект «Сумма чисел»

На форме расположены метка с текстом «Числа», метка для вывода результата, список для отображения вводимых чисел и кнопка «Ввод» для запуска процесса ввода чисел и вычисления их суммы. Требуется вычислять сумму всех вводимых пользователем чисел до тех пор, пока эта сумма не станет кратной пяти. Внешний вид формы на разных стадиях выполнения проекта показан на рис. 52.

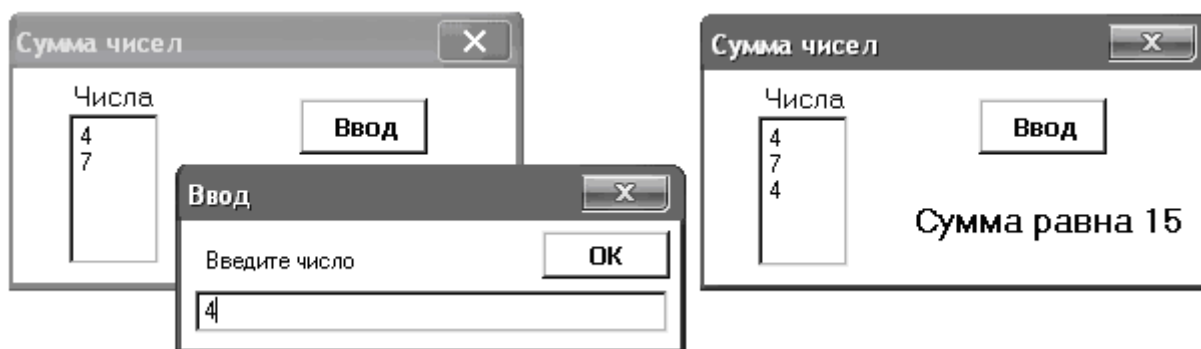


Рис. 52. Внешний вид формы на разных стадиях выполнения проекта

Проект «Угадай число»

На форме расположены кнопки «Игра 1» и «Игра 2». Пользователь пытается отгадать целое случайное число в диапазоне от 1 до 20, ведется подсчет количества попыток. В первой игре, если пользователь не угадал число, ему сообщается только то, что он не угадал и запрашивается новое число. Во второй игре при неправильном ответе дается подсказка в виде сообщения о том, что загаданное число больше или меньше введенного. Внешний вид формы и диалоговых окон на разных стадиях выполнения проекта показан на рис. 53.

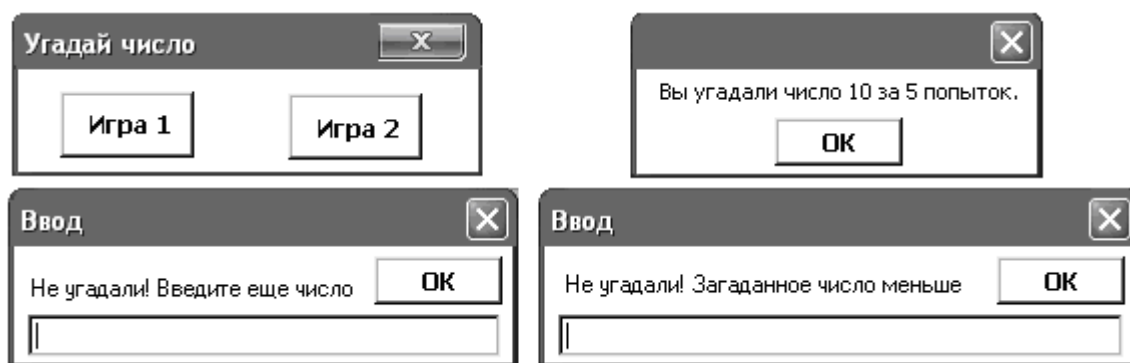


Рис. 53. Внешний вид формы и диалоговых окон

Проект «Склад»

На склад привозят груз на машинах различной грузоподъемности. Требуется вычислить количество разгруженных машин до того, как склад заполнится. На форме расположены метки с пояснительным текстом, поле для ввода вместимости склада, список для вывода данных о грузоподъемности разгруженных машин, метка для вывода количества разгруженных машин и кнопка «ОК».

Пользователь вводит вместимость склада и нажимает на кнопку «ОК». Программа запрашивает грузоподъемность каждой машины до тех пор, пока общий вес груза не превысит вместимость склада, и ведет подсчет количества разгруженных машин. Внешний вид формы после вычислений показан на рис. 54.

Рис. 54. Внешний вид формы

Проект «Скрепер»

На рабочем листе Excel составлена таблица с данными (табл. 8).

Таблица 8

Исходные данные

Марка скрепера	Емкость ковша, м ³
ДЗ-149-5	8
ДЗ-87-1	4,5
ДЗ-172-1	8,8
ДЗ-79	18

На форме расположены поле со списком для выбора марки скрепера, метка для вывода емкости ковша, поля для ввода коэффициентов, кнопка «Вычислить», метка для вывода результата, метки с пояснительным текстом. При выборе марки скрепера в метке отображается соответствующее значение емкости ковша. Кнопка «Вычислить» считывает значения коэффициентов из полей и вычисляет объем грунта в ковше V (м³) по формуле

$$V = q \cdot K_n / K_p,$$

где q – емкость ковша скрепера, м³; K_n – коэффициент наполнения ковша грунтом; K_p – коэффициент разрыхления грунта в ковше.

Внешний вид формы после вычислений показан на рис. 55.

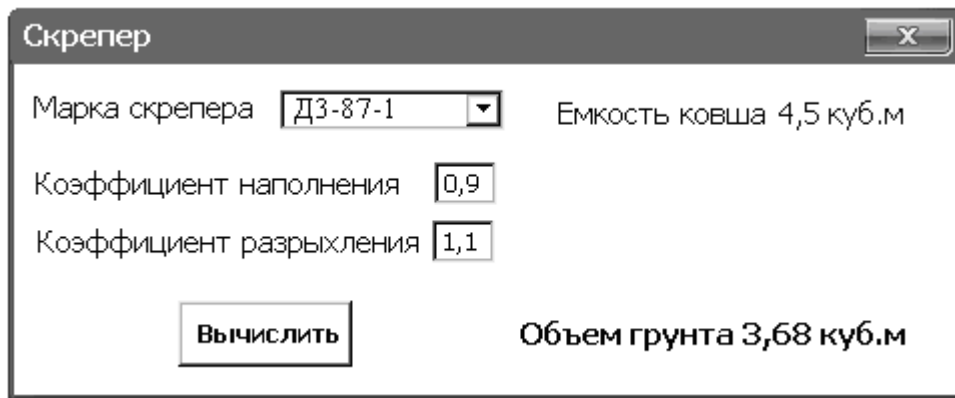


Рис. 55. Внешний вид формы

Проект «Турфирма»

На форме расположены список стран, временно недоступный комбинированный список для выбора количества суток. После выбора пользователем страны на форме появляется соответствующая картинка и стоимость путевки за сутки, комбинированный список становится доступным. При выборе из него количества суток на форме появляется общая стоимость путевки. Внешний вид формы после выбора страны и количества суток показан на рис. 56.



Рис. 56. Внешний вид формы

Проект «Автомобиль»

На форме расположены пять переключателей для выбора автомобиля, метка для вывода соответствующей среднетехнической скорости, поле для ввода времени, кнопка «Вычислить», метка для вывода результата, метки с пояснительным текстом. Требуется вычислить расстояние, пройденное автомобилем. Значения среднетехнической скорости (в км/ч) для автомобилей ЗИЛ, МАЗ,

КрАЗ, КамАЗ и БелАЗ соответственно равны 45, 50, 50, 44 и 55. Внешний вид формы после вычислений показан на рис. 57.

Скрепер

Марка автомобиля

Скорость 44 км/ч

ЗИЛ

МАЗ

Краз

КамАЗ

БелАЗ

Время, ч 1,1

Вычислить

Расстояние 48,4 км

Рис. 57. Внешний вид формы

Проект «Сопротивление»

На форме расположены метки с пояснительным текстом, два поля для ввода сопротивления на участках цепи, рамка с двумя переключателями для выбора типа соединения участков, метка для вывода сопротивления цепи R , кнопка «Вычислить», вычисляющая общее сопротивление цепи:

- $R=R_1+R_2$ – при последовательном соединении;
- $R=R_1 \cdot R_2 / (R_1+R_2)$ – при параллельном соединении.

Внешний вид формы после вычислений показан на рис. 58.

Сопротивление

Сопротивление R1, Ом 12

Сопротивление R2, Ом 17

Тип соединения

Последовательное

Параллельное

Вычислить

Сопротивление в сети 7,03 Ом

Рис. 58. Внешний вид формы

Проект «Тест»

На форме разместить: метки с пояснительным текстом (формулировки вопросов), три рамки для переключателей и флажков, семь переключателей и четыре флажка (варианты ответов на вопросы), кнопку «ОК» для проверки ответов и метку для вывода результата. Внешний вид формы показан на рис. 59.

За правильный ответ на вопрос добавляется один балл. Оценка за тест выставляется по следующим критериям:

- 3 балла – отлично;
- 2 балла – хорошо;
- 1 балл – удовлетворительно;
- 0 баллов – неудовлетворительно.

Тест

1. Укажите наименьшую единицу измерения информации.

- Байт
- Бит
- Бод
- Килобайт

2. Один Килобайт равен...

- 1000 байт
- 1024 байт
- 1024 бита
- 8192 бита

3. Сколько бит в одном байте?

- 10
- 2
- 8

OK

Неудовлетворительно

Рис. 59. Внешний вид формы

Проект «Кредит»

На форме расположены три поля для ввода суммы кредита, количества месяцев и размера процентной ставки, рамка, два переключателя для выбора операции, метки с пояснительным текстом, метка для вывода результата и две кнопки «Вычислить» и «Выход». Каждый месяц на сумму кредита начисляется указанный процент. Внешний вид формы после вычислений показан на рис. 60.

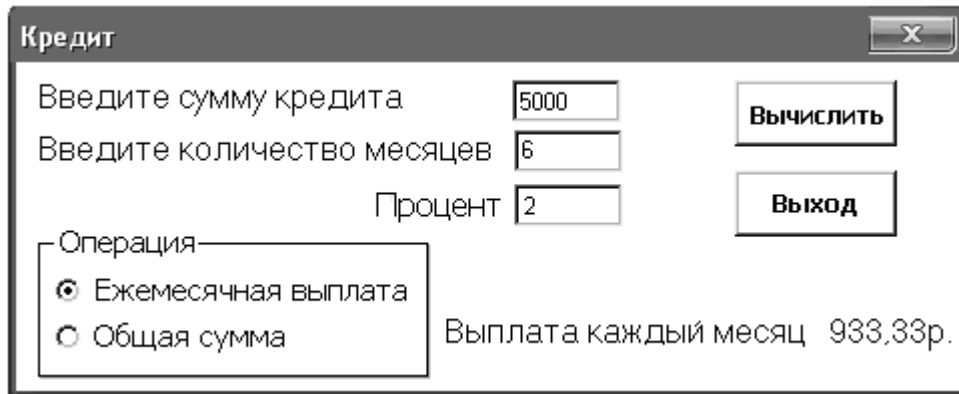


Рис. 60. Внешний вид формы

Проект «Операции»

На форме расположены метки с пояснительным текстом, два поля для ввода чисел, кнопка «Вычислить», две рамки, четыре переключателя для выбора операции, три флажка для выбора способа вывода и метка для вывода результата. Кнопка «Вычислить» считывает числа из полей, выполняет выбранную операцию и выводит выбранным способом результат операции и ее название. Внешний вид формы и фрагмент рабочего листа после вычислений показан на рис. 61.

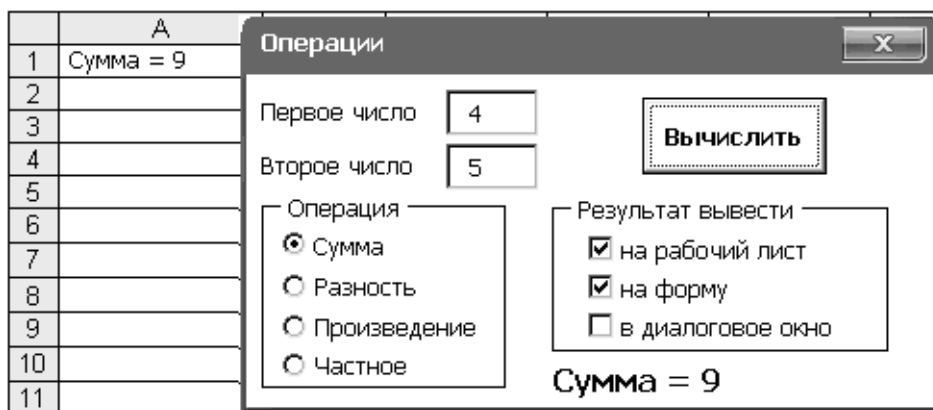


Рис. 61. Внешний вид формы и фрагмента рабочего листа

Проект «Вклад»

На форме расположены: рамки; метки с пояснительным текстом; поля для ввода фамилии, имени и отчества, внесенной суммы, срока вклада; поле со списком для выбора годовой процентной ставки (от 1 до 10); два переключателя для выбора схемы начисления процентов;

метка для вывода результата; кнопки «Вычислить» и «Заккрыть». Кнопка «Вычислить» вычисляет размер вклада на настоящий момент, выводит его в метку на форме и формирует на рабочем листе Excel отчет (жирным шрифтом выделены данные, считанные с формы). Внешний вид отчета и формы показан на рис. 62-63.

Формулы для вычисления:

1) простые проценты: $\text{Доход} = \text{Внесенная сумма} * (1 + \text{Срок}/365 * \text{Процентная ставка}/100)$;

2) сложные проценты: $\text{Доход} = \text{Внесенная сумма} * (1 + \text{Срок}/365 * \text{Процентная ставка}/100)^N$,

где N – число периодов начислений процентов (запрашивается с клавиатуры у пользователя).

	A	B	C	D	E	F	G	H
1	Отчет о доходе							
2								
3	На настоящий момент гражданин(ка)				Евсеева Ольга Константиновна			
4	имеет на своем счету по			простым	процентам			
5	при начальной сумме вложения			1000	руб.			
6	за	150	дней при	2	%-ной ставке	1008,22	руб.	

Рис. 62. Отчет на рабочем листе Excel

Рис. 63. Внешний вид формы

Проект «Перевод»

На форме расположены: рамки; метки с пояснительным текстом; поля для ввода суммы перевода (числом и прописью), сведений об отправителе и получателе перевода; флажок для выбора необходимости уведомления о получении перевода; кнопка «Печать».

После ввода данных пользователь нажимает на кнопку «Печать», на рабочем листе формируется бланк квитанции для перевода (рис. 64). Внешний вид формы после заполнения показан на рис. 65.

	A	B	C	D	E	F	G
1	Почтовый перевод на:			1200 руб. 00 коп.			
2	Одна тысяча двести руб. 00 коп.						
3	(прописью)						
4	Куда	644045		Омская Омск Волкова 3/44			
5		<i>индекс</i>		<i>адрес</i>			
6	Кому	Петровой И.Г.					
7	От кого	Смирнова П.Л.		Уведомление			
8	Адрес	432302		Саратовская Балашов Ленина 2/1/33			
9							
10		<i>индекс</i>		<i>адрес</i>			
11							

Рис. 64. Внешний вид бланка на рабочем листе

Перевод

сумма перевода

1200 руб. 00 коп.

Уведомление

Сумма прописью

Одна тысяча двести руб. 00 коп.

Куда

Индекс: 644045

Край/обл.: Омская

Город: Омск

Улица: Волкова

Дом/корп./кв.: 3/44

Откуда

Индекс: 432302

Край/обл.: Саратовская

Город: Балашов

Улица: Ленина

Дом/корп./кв.: 2/1/33

Кому

Петровой И.Г.

От кого

Смирнова П.Л.

Печать

Рис. 65. Внешний вид формы

Проект «Заказ путевок»

На первом рабочем листе создать таблицу с исходными данными (табл. 9). На форме расположены: список для выбора страны, поля для ввода количества путевок и количества суток проживания, поле со списком для выбора вида гостиничного номера, две рамки, два переключателя для выбора вида оплаты, два флажка для выбора услуг, кнопка «Ввод». Внешний вид формы показан на рис. 66.

Для вычисления общей стоимости заказанных путевок используйте формулу

$$Sum = (sp + pit \cdot ks + ex + pr \cdot ks) \cdot kp \cdot k,$$

где sp – стоимость путевки, pit – стоимость питания (за сутки), ks – количество суток проживания, ex – стоимость экскурсий, pr – стоимость проживания (за сутки), kp – количество путевок, k – коэффициент безналичного расчета.

Таблица 9

Исходные данные

Название страны	Путевка, \$	Проживание, \$ в сутки	Питание, \$ в сутки	Экскурсии, \$	Вид номера	Коэффициент безналичного расчета
Испания	100	5	2	15	одноместный	1,02
Италия	95	6	2,5	20	двухместный	
Франция	100	7	2,8	18	трехместный	
Австрия	90	5	2,6	16	люкс	
Германия	85	6	2,4	15		

The screenshot shows a window titled "Заказ путевок" with a close button (X). The form contains the following elements:

- Выбор страны:** A list box containing "Испания", "Италия", "Франция", "Австрия", and "Германия".
- Услуги:** Two checkboxes: "питание" (checked) and "экскурсии".
- Вид оплаты:** Two radio buttons: "Наличный расчет" (checked) and "Безналичный расчет".
- Количество путевок:** An empty text input field.
- Количество суток проживания:** An empty text input field.
- Выбор номера:** A dropdown menu.
- Ввод:** A button to submit the form.

Рис. 66. Внешний вид формы

Поле и поле со списком формируются при активации формы. Пользователь вводит данные и нажимает на кнопку «Ввод», вычисляется стоимость поездки и на второй рабочий лист Excel выводится таблица заказа (табл. 10).

Таблица 10

Результаты работы проекта

Страна	Германия
Кол-во путевок	2
Кол-во суток проживания	10
Номер	двухместный
Услуги	экскурсии
Вид оплаты	наличный расчет
Итого	320

Проект «Товары»

На рабочем листе Excel составлена таблица с данными (рис. 67). На форме расположены: набор вкладок, четыре надписи, четыре поля, рисунок. При выборе вкладки в полях отображаются данные о соответствующем товаре (считываются с листа), вычисляется стоимость товара и в рисунок загружается соответствующее изображение. Внешний вид формы при выборе первой вкладки показан на рис. 68.

	А	В	С
1	Наименование	Цена	Количество
2	Монитор	7000	3
3	Клавиатура	500	5
4	Принтер	3000	2

Рис. 67. Фрагмент рабочего листа

The screenshot shows a window titled "Товары" with a close button (X). On the left, there are three tabs labeled "Товар1", "Товар2", and "Товар3". The "Товар1" tab is selected. To the right of the tabs are four input fields with labels: "Наименование" (containing "Монитор"), "Цена" (containing "7000"), "Количество" (containing "3"), and "Стоимость" (containing "21000"). To the right of these fields is a small image of a CRT computer monitor.

Рис. 68. Внешний вид формы

Проект «Анкета»

На форме расположен элемент управления «Набор страниц», состоящий из двух страниц, на каждой из которых расположены соответствующие элементы управления для ввода данных. Внешний вид формы на разных стадиях выполнения проекта показан на рис. 69.

Все списки формируются при активации формы. Поле со списком «Уровень образования» содержит следующие элементы: среднее общее, начальное профессиональное, среднее профессиональное, высшее профессиональное, неоконченное высшее, студент. Список «Степень» содержит следующие элементы: бакалавр, магистр, кандидат, доктор, нет. Если уровень образования «Студент», то на форме появляется поле для ввода курса.

The image shows two screenshots of a software form titled "Анкета" (Survey). The top screenshot shows the "Персона" (Person) tab selected. It contains input fields for "Фамилия" (Surname) with the value "Иванов", "Имя" (Name) with "Иван", and "Отчество" (Patronymic) with "Иванович". There are also three dropdown menus for "Дата рождения" (Date of Birth): "День" (Day) with "3", "Месяц" (Month) with "10", and "Год" (Year) with "1991". A "Пол" (Sex) section has two radio buttons: "мужской" (male) which is selected, and "женский" (female). An "OK" button is at the bottom right. The bottom screenshot shows the "Образование" (Education) tab selected. It features a dropdown menu for "Уровень образования" (Level of education) with "Студент" (Student) selected. To its right is a text input field for "Курс" (Course) with the value "3". Below these is a list box for "Степень" (Degree) containing the items: "бакалавр" (bachelor), "магистр" (master), "кандидат" (candidate), "доктор" (doctor), and "нет" (none). An "OK" button is at the bottom right.

Рис. 69. Внешний вид формы

Пользователь вводит данные и нажимает на кнопку «ОК», которая выводит на рабочий лист Excel эти данные в виде таблицы (рис. 70), очищает поля и переключается на другую страницу. Если число или месяц рождения меньше 10, то при выводе данных на лист к ним приписывается слева ноль.

	А	В	С	Д	Е
1	Фамилия, имя, отчество	Пол	Дата рождения	Уровень образования	Степень
2	Иванов Иван Иванович	м	03.10.1991	Студент 3-го курса	Нет
3	Петрова Ирина Петровна	ж	25.06.1985	Высшее	Нет
4	Иванов Семен Петрович	м	11.10.1980	Высшее	Кандидат

Рис. 70. Фрагмент рабочего листа

Проект «Формат»

На форме расположены поле для ввода текста, кнопка «Формат» и временно невидимый набор из двух страниц. На первой странице «Шрифт» расположены метки с пояснительным текстом, список для выбора шрифта, поле со списком для выбора размера шрифта и кнопка «ОК».

На второй странице «Абзац» расположены рамка, три переключателя для выбора вида выравнивания и кнопка «ОК». После нажатия кнопки «Формат» набор страниц становится видимым, и пользователь может устанавливать нужные параметры форматирования текста в поле. Кнопка «ОК» применяет выбранные параметры форматирования и скрывает набор страниц. Внешний вид формы на разных стадиях работы проекта показан на рис. 71.

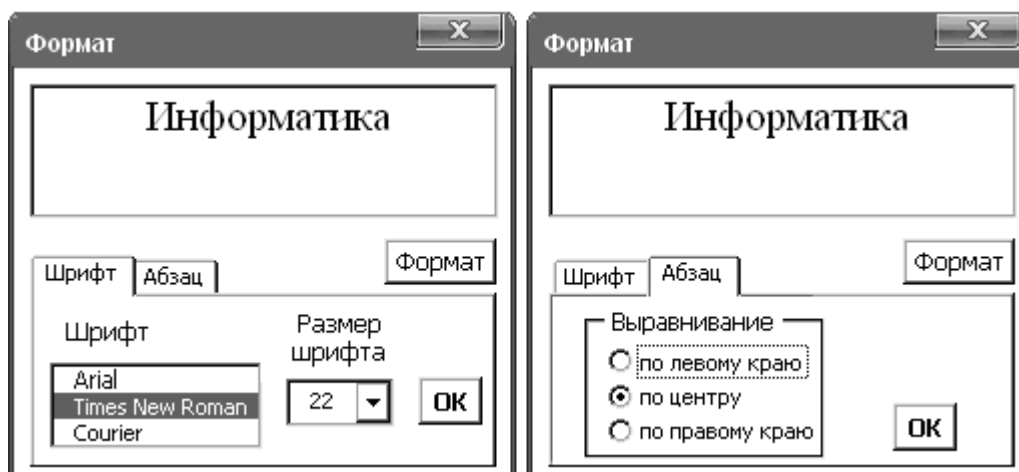


Рис. 71. Внешний вид формы

Проект «Полосы прокрутки»

На форме расположены метки с пояснительным текстом, две полосы прокрутки, два поля для отображения значений состояния полос прокрутки, метка для вывода результата и кнопка «Вычислить». Диапазон значений для полос прокрутки $[-100; 100]$, значение 0. Пользователь перемещает бегунки на полосах прокрутки, в полях отображаются соответствующие числа. После нажатия кнопки «Вычислить» в метке выводится сумма чисел. Внешний вид формы после вычислений показан на рис. 72.

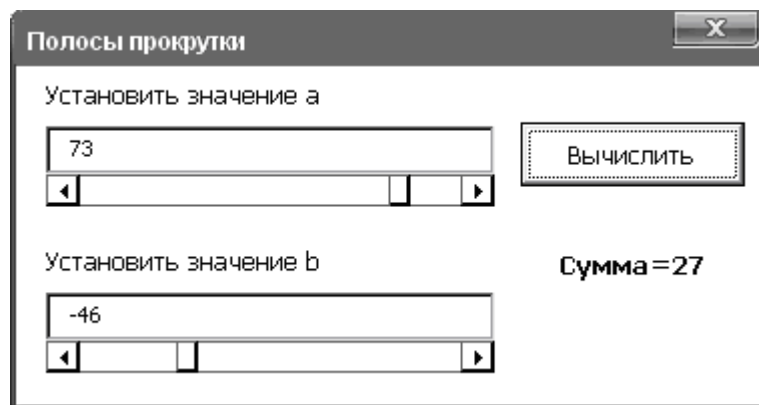


Рис. 72. Внешний вид формы

Проект «Палитра»

На форме расположен набор из трех страниц, на каждой из которых расположены метки с пояснительным текстом и область рисунка, цвет которой будет меняться в зависимости от интенсивности каждого цвета в формате RGB (Red – красный, Green – зеленый, Blue – синий).

На первой странице расположены три поля для ввода интенсивности каждого цвета и кнопка «Цвет», изменяющая цвет рисунка. На второй вкладке расположены три полосы прокрутки, при изменении которых меняется цвет рисунка. На третьей странице расположены три поля и три полосы прокрутки.

При изменении значений в полях происходят соответствующие изменения в полосах прокрутки и наоборот, а также происходит

изменение цвета рисунка. Внешний вид формы на разных стадиях работы проекта показан на рис. 73.

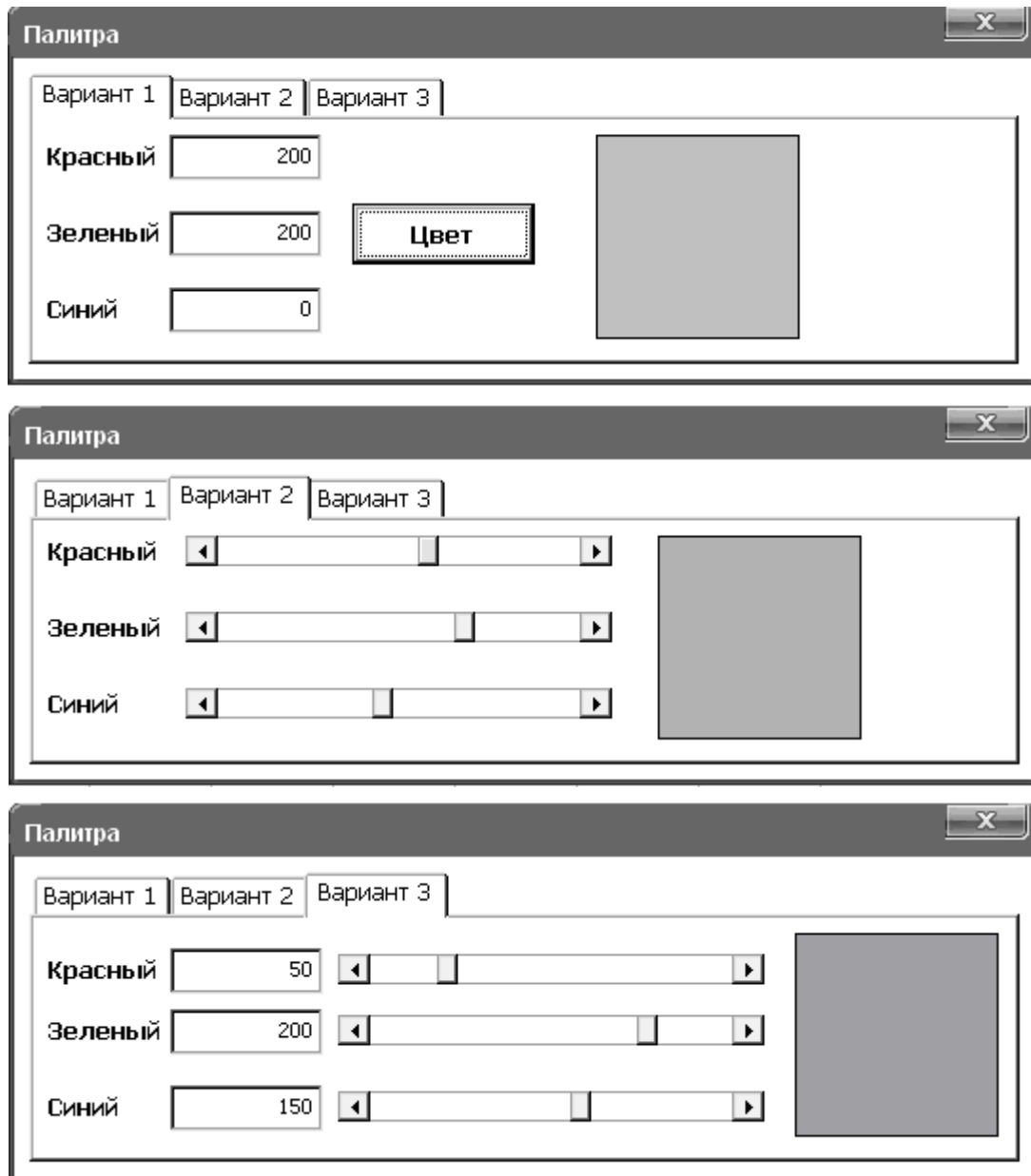


Рис. 73. Внешний вид формы

Проект «Счетчик»

На форме расположены метки с пояснительным текстом, два счетчика, два поля для вывода значений состояния счетчиков, метка для вывода результата, кнопка «Вычислить». При изменении состояния счетчика в соответствующем поле выводится значение счетчика. После нажатия кнопки «Вычислить» в метку выводится

произведение чисел. Внешний вид формы после вычислений показан на рис. 74.

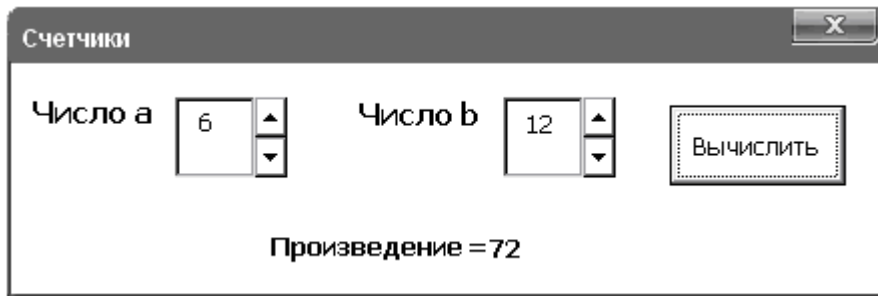


Рис. 74. Внешний вид формы

Проект «Калькулятор»

На форме расположены поле, кнопки для набора цифр и десятичной точки, кнопки со знаками арифметических операций,

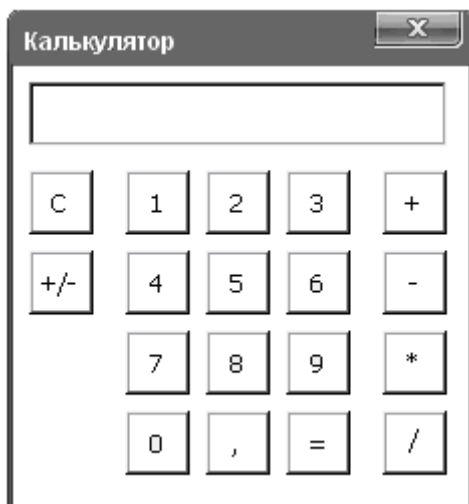


Рис. 75. Внешний вид формы

кнопка для вычисления результата, кнопка для смены знака числа, кнопка сброса (очистки поля). При нажатии кнопок с цифрами в поле дописывается справа соответствующая цифра. Считывание первого числа из поля происходит при нажатии кнопки со знаком арифметической операции, а считывание второго числа и вычисление соответствующего результата – при нажатии кнопки со знаком «=». При вводе числа необходимо отслеживать ввод десятичной точки. Внешний вид формы показан на рис. 75.

Проект «Блокнотик»

Создать простой текстовый редактор со следующими возможностями: изменение шрифта, размера шрифта, начертания (полужирное, курсив, подчеркивание), выравнивания (по левому краю, по центру, по правому краю). Поля со списками формируются при активации формы, при этом устанавливается выбранным первый элемент каждого списка, а также задаются соответствующий шрифт и

размер шрифта в поле ввода. Для изменения начертания шрифта и выравнивания использовать выключатели. При выборе одного из видов выравнивания текста другие выключатели (виды выравнивания) отключаются. Внешний вид формы показан на рис. 76.

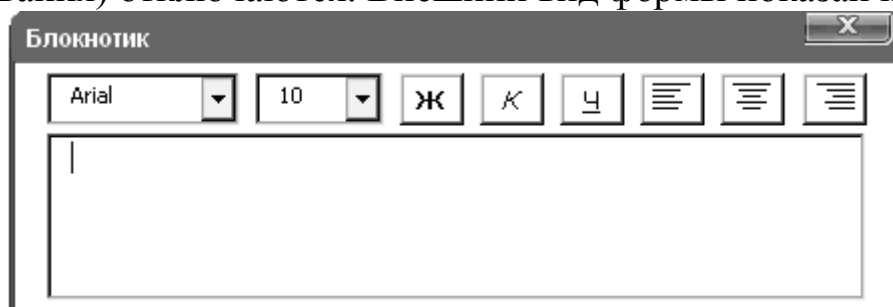


Рис. 76. Внешний вид формы

Проект «Стипендиальный фонд»

На форме расположены метки с пояснительным текстом; поля для ввода размера стипендиального фонда, количества «отличников» и «хорошистов» и размера обычной стипендии; кнопка «Расчет» для вычислений; поля для вывода результатов: размера повышенной стипендии (больше обычной на 15 %), итоговой суммы на обычную и повышенную стипендии, остаток фонда; кнопки «Диаграмма», «Очистить» и «Выход».

Кнопка «Диаграмма» строит круговую объемную диаграмму распределения стипендиального фонда на отдельном рабочем листе Excel (тип диаграммы – xIPie). Кнопка «Очистить» удаляет лист с диаграммой. Кнопка «Выход» завершает работу приложения. Внешний вид формы после вычислений показан на рис. 77.

Стипендиальный фонд

Стипендиальный фонд в размере (руб.) 16000

Количество "Отличников" 15

Количество "Хорошистов" 20

Размер обычной стипендии 400

Размер повышенной стипендии (15%) 460

Итого на обычную (руб.) 8000

Итого на повышенную (руб.) 6900

Остаток фонда (руб.) 1100

Рис. 77. Внешний вид формы

Проект «Персональный секретарь»

На форме расположены метки с пояснительным текстом, поле для ввода времени, поле для ввода сообщения и кнопка «Установить будильник и свернуть», которая «включает таймер» и сворачивает (скрывает) окно приложения. В установленное время окно программы вновь отображается. Внешний вид формы показан на рис. 78.

Персональный секретарь

Время сигнала 12:00

Сообщение Позвонить Сергею Петровичу!

Рис. 78. Внешний вид формы

Изменить проект таким образом, чтобы часы и минуты устанавливались в полях с помощью счетчиков (рис. 79).

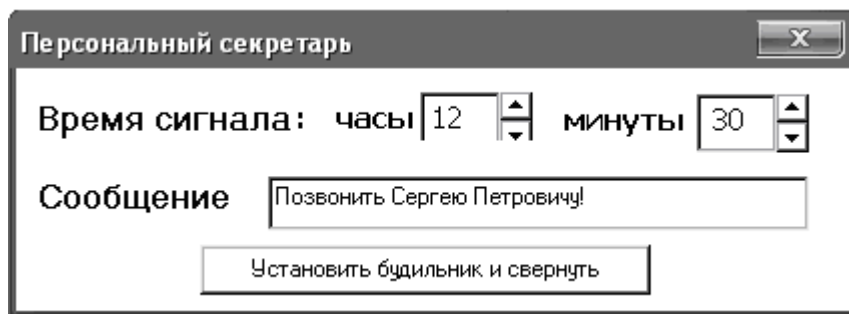


Рис. 79. Внешний вид формы

Проект «Тир»

На форме, занимающей весь экран, расположен рисунок. При активации формы рисунок начинает менять свои координаты случайным образом, перемещаясь в пределах формы, с интервалом в 1 секунду. Пользователь должен успеть щелкнуть левой кнопкой мыши по рисунку, прежде чем тот переместится в новое положение. Весь процесс длится заданное пользователем время, при этом ведется подсчет количества «попаданий» (щелчков мыши по рисунку). Внешний вид формы показан на рис. 80.



Рис. 80. Внешний вид формы

Проект «Молекула»

На форме расположены три рисунка, метка с текстом «Ходы», метка для вывода количества ходов, рисунок в центре формы (прямоугольник), кнопка «Старт». При нажатии на кнопку «Старт» рисунки (молекулы) устанавливаются в начальную позицию, а затем начинают случайным образом перемещаться по форме (случайным образом меняется горизонтальная координата), ведется подсчет

ходов. Как только все рисунки окажутся в правой части формы, игра заканчивается. Внешний вид формы на разных стадиях выполнения проекта показан на рис. 81.

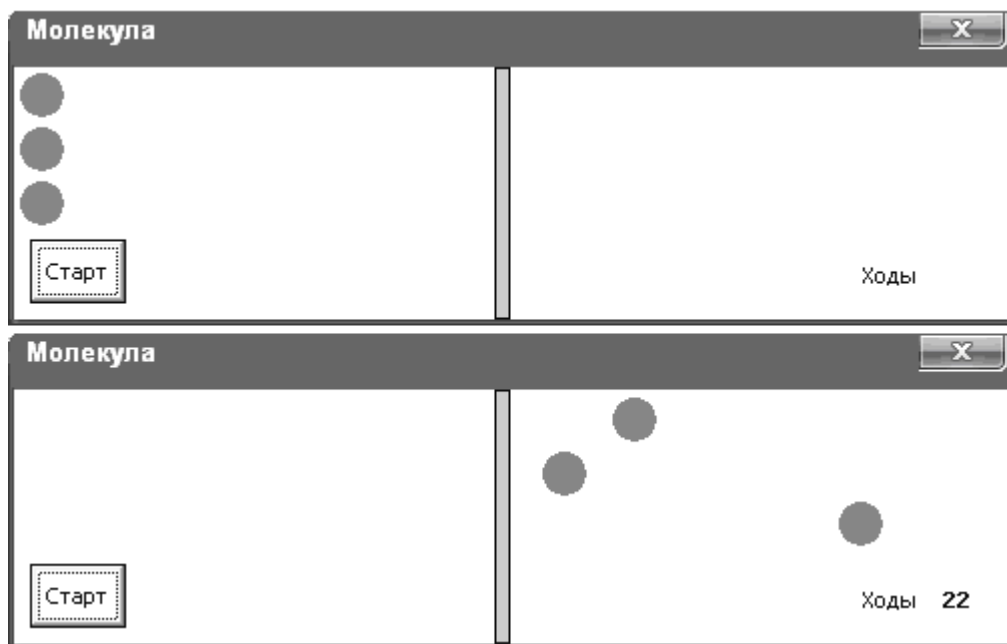


Рис. 81. Внешний вид формы до и после выполнения программы
Проект «Гонки»

На форме расположены три рисунка, метка для вывода их координат, метка для вывода номера победителя, рисунок в правой части формы – финиш, кнопка «Старт». Пользователь нажимает кнопку «Старт», рисунки начинают двигаться вправо, наращивая координаты на случайное число от 0 до 9. Как только один из рисунков оказывается полностью за финишной чертой, на форму выводится номер победителя. При повторном нажатии кнопки «Старт» рисунки возвращаются на исходную позицию, и игра начинается заново. Внешний вид формы на разных стадиях выполнения проекта показан на рис. 82.

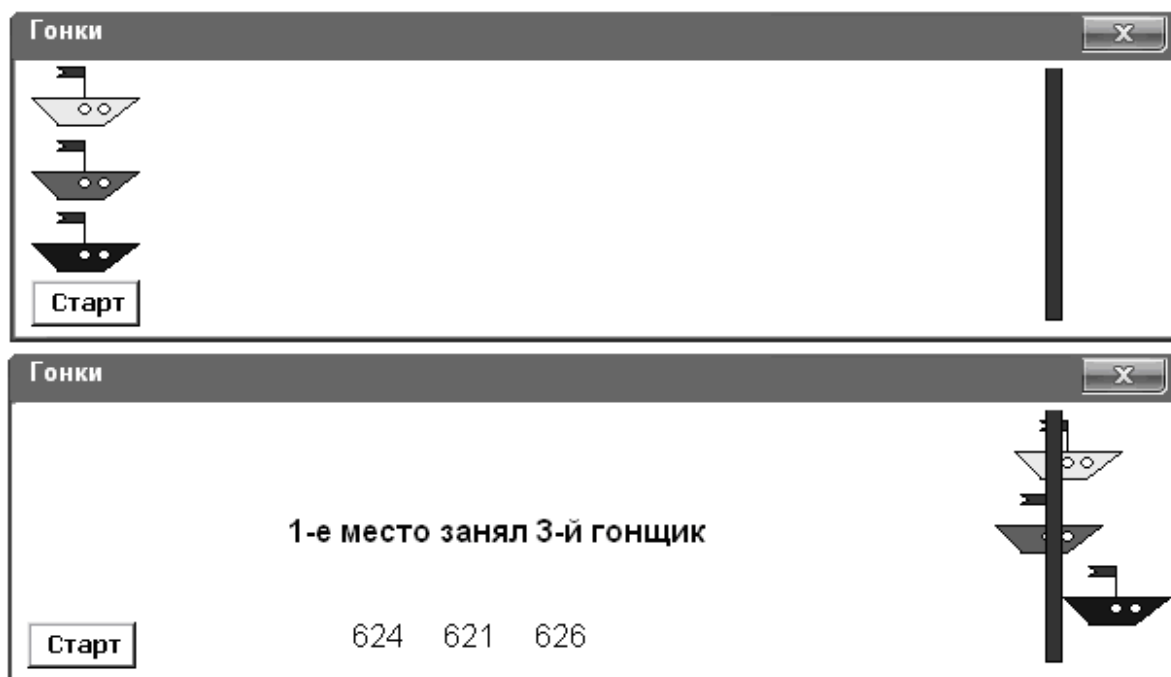


Рис. 82. Внешний вид формы до и после нажатия кнопки «Старт»

Проект «Уравнения»

В проекте создано четыре формы, на рабочем листе создана кнопка для открытия главной формы «Уравнения».

На форме «Уравнения» расположены метка с текстом, рисунок и кнопка «Далее», которая скрывает текущую форму и отображает форму «Вид уравнения».

На форме «Вид уравнения» расположены рамка, два переключателя для выбора вида уравнения, кнопки «Далее» и «Выход». Кнопка «Далее» скрывает текущую форму и в зависимости от выбранного переключателя отображает форму «Линейное уравнение» или «Квадратное уравнение». Кнопка «Выход» завершает работу приложения.

На форме «Линейное уравнение» расположены метки с пояснительным текстом, два поля для ввода значений коэффициентов уравнения, метка для вывода результата, кнопки «Вычислить» и «Назад».

На форме «Квадратное уравнение» расположены метки с пояснительным текстом, три поля для ввода значений коэффициентов уравнения, метка для вывода результата, кнопки «Вычислить» и «Назад».

На каждой из этих форм кнопка «Вычислить» осуществляет считывание коэффициентов из полей, вычисление корней уравнения и вывод результатов.

Кнопка «Назад» скрывает текущую форму и отображает форму «Вид уравнения».

Внешний вид форм на разных стадиях работы приложения показан на рис. 83.

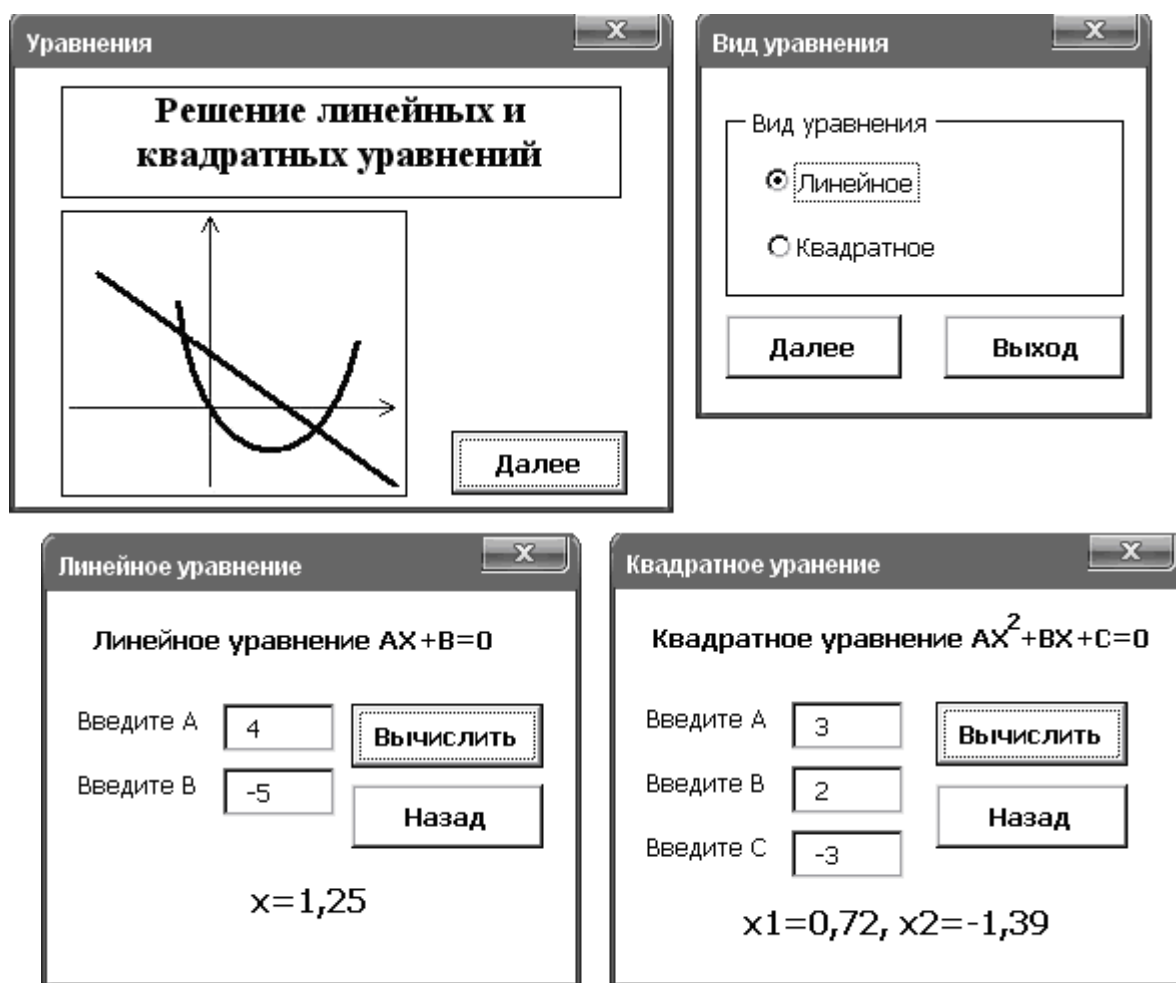


Рис. 83. Внешний вид форм
Проект «Пароль»

В проекте создано две формы: форма «Регистрация» для ввода пароля и форма «Главное окно» пользовательского приложения.

На регистрационной форме расположена метка с пояснительным текстом, поле для ввода пароля, кнопка «ОК». На ввод пароля пользователю отводится 3 попытки и 15 секунд. Если пользователь вводит правильный пароль, то появляется диалоговое окно с

приветствием, а затем на весь экран разворачивается форма «Главное окно», на которой для простоты располагается только метка с текстом «Hello!». При закрытии этой формы, происходит завершение работы приложения.

При неправильном или несвоевременном вводе пароля выводится соответствующее сообщение, и работа приложения заканчивается. Внешний вид форм и диалоговых окон на разных стадиях работы приложения показан на рис. 84.

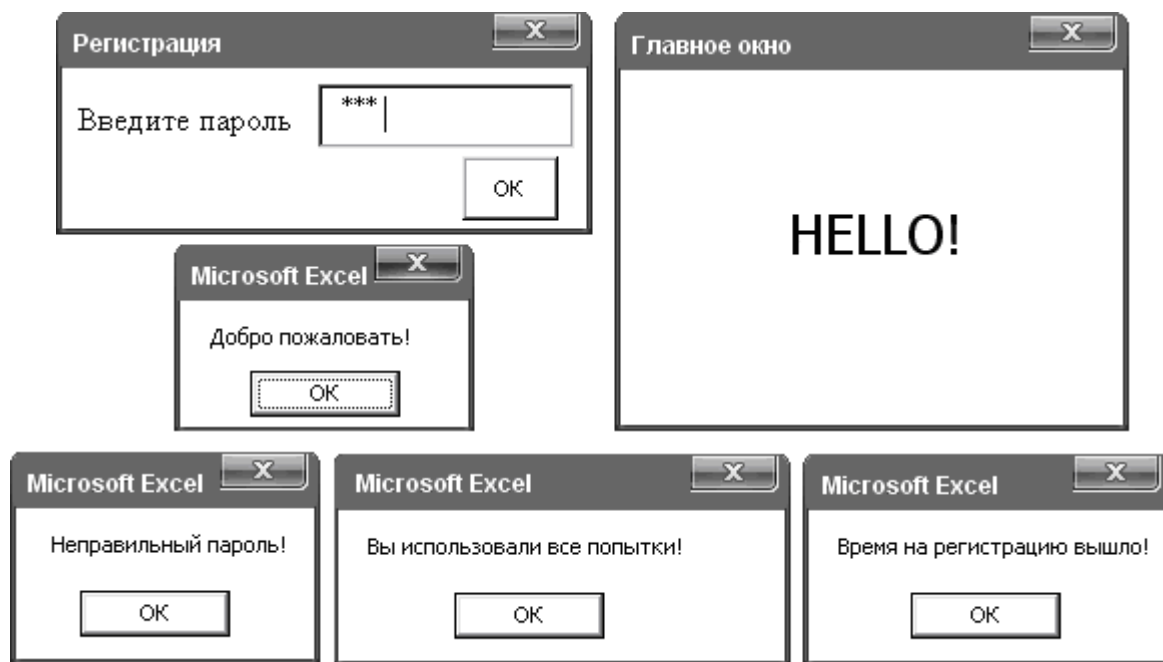


Рис. 84. Внешний вид форм и диалоговых окон

Проект «Словарь»

На форме расположены метки с пояснительным текстом и список с несколькими английскими словами. При двойном щелчке по списку анализируется состояние списка (какой элемент выбран) и в диалоговом окне выводится выбранное слово и его перевод. Внешний вид формы до и после выбора слова показан на рис. 85. Вывод данных в диалоговое окно с заголовком (см. прил. 1).

Изменить приложение следующим образом: добавить в проект еще одну форму и разместить на ней рисунок, пустую метку и поле. Двойной щелчок по списку на первой форме выводит в метку на второй форме выбранное слово, выводит в поле на второй форме

перевод слова и отображает вторую форму. Внешний вид второй формы после выбора слова показан на рис. 86.

Для обращения к элементам управления, расположенным на другой форме следует указать имя формы и через точку имя объекта.

Изменить приложение следующим образом: на рабочем листе составить таблицу с английскими словами и соответствующим переводом. Список слов на главной форме считывается с листа при активации формы. Написать процедуру обработки события *Open* для рабочей книги *Workbook*, в которой вызвать метод *Show* для главной формы, чтобы приложение запускалось при открытии рабочей книги. Защитить проект паролем.

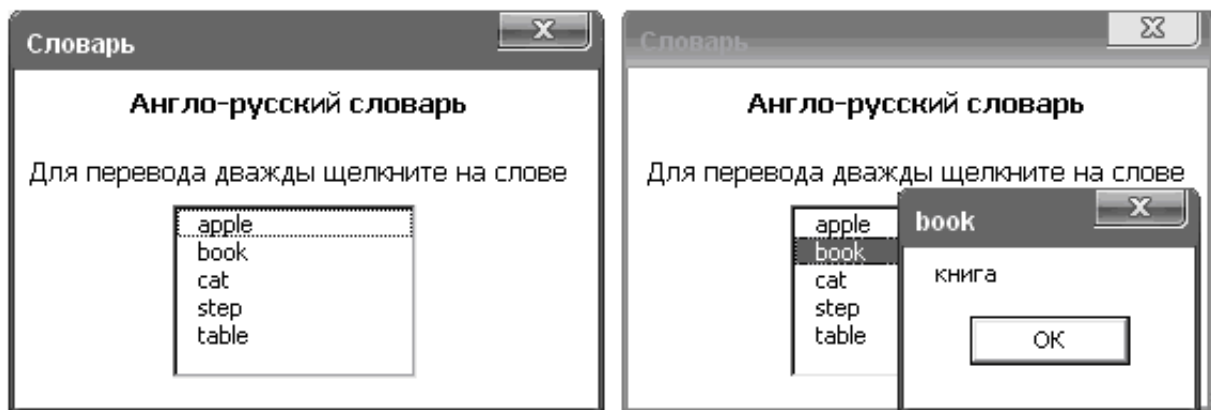


Рис. 85. Внешний вид формы и диалогового окна



Рис. 86. Внешний вид второй формы

Проект «Квадрат числа»

На форме расположена кнопка «Вычислить». При нажатии на нее у пользователя запрашивается подтверждение о намерении произвести вычисление. В случае положительного ответа запрашивается число, вычисляется его квадрат и выводится на экран в диалоговом окне. В случае отрицательного ответа программа

завершает работу. Организовать интерфейс с помощью соответствующих диалоговых окон (см. прил. 1).

Проект «Блокнот»

Проект содержит пять форм. На первой (главной) форме расположены поле для ввода текста, три временно невидимых выключателя для изменения начертания шрифта (полужирный, курсив и подчеркнутый) и восемь кнопок, образующих главное меню (рис. 87). При активации формы видны только четыре основных пункта меню: «Вид», «Вставка», «Формат» и «?». При активации «Вид» видны также «Панель инструментов», «Дата и время», «Шрифт» и «Абзац».



Рис. 87. Главная форма в режиме проектирования

Структура главного меню: первый пункт «Вид» включает подпункт «Панель инструментов», при выборе которого выключатели на форме становятся видимыми и в надписи подпункта появляется символ «+», т.е. панель инструментов видна (рис. 88). После этого можно менять начертание символов в поле с помощью выключателей. При повторном выборе данного пункта меню выключатели становятся невидимыми, и в надписи подпункта исчезает символ «+».

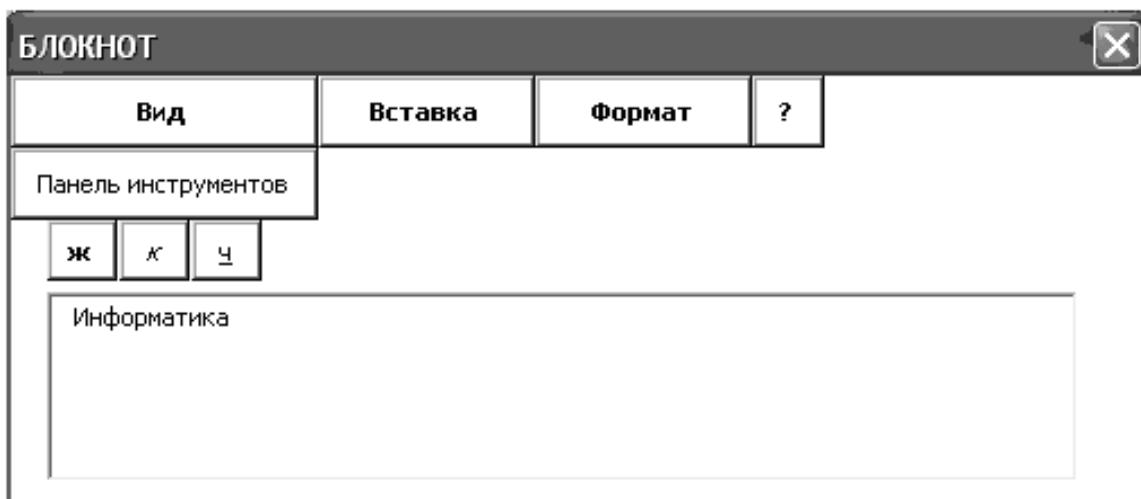


Рис. 88. Главная форма после выбора первого пункта меню

Второй пункт меню «Вставка» включает подпункт «Дата и время», при выборе которого появляется вторая форма. На второй

форме расположены два флажка и кнопки «ОК» и «Отмена» (рис. 89). При нажатии на кнопку «ОК» в поле на главной форме в конце текста добавляется текущая дата или время, в зависимости от выбранных

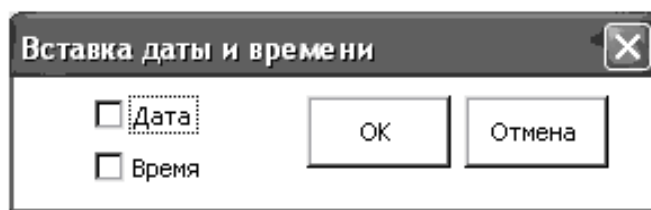


Рис. 89. Вторая форма

пользователем флажков, и вторая форма скрывается. При нажатии кнопки «Отмена» вторая форма скрывается.

Третий пункт меню «Формат» включает два подпункта «Шрифт» и «Абзац», при выборе которых появляется третья или четвертая форма соответственно.

На третьей форме расположены метки с пояснительным текстом, список со шрифтами, поле со списком для выбора размера шрифта, рамка и четыре переключателя для выбора цвета символов, кнопки «ОК» и «Отмена» (рис. 90). При нажатии на кнопку «ОК» в поле на главной форме изменяются выбранные параметры шрифта и третья форма скрывается. При нажатии кнопки «Отмена» третья форма скрывается. Списки формируются при активации формы (пять названий шрифтов, размер шрифта – числа от 8 до 30 с шагом 2), выбранными в списках устанавливаются текущие значения параметров для поля на главной форме.

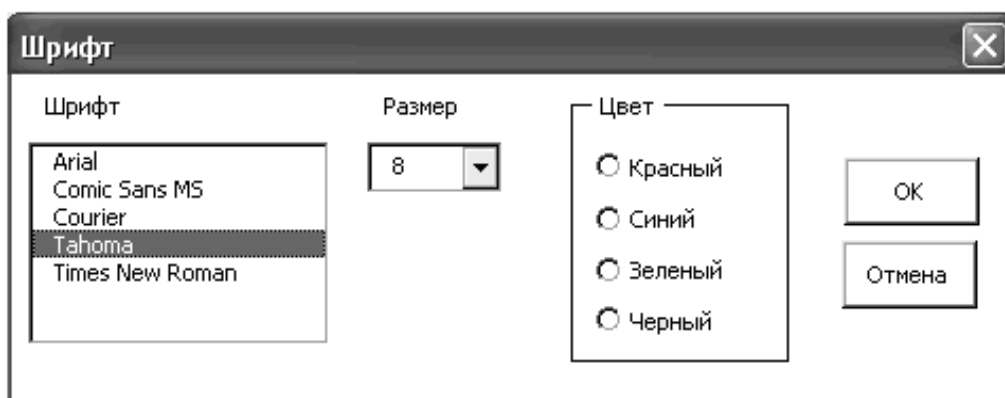


Рис. 90. Третья форма в режиме проектирования и после запуска

На четвертой форме расположен набор из двух страниц: «Выравнивание» и «Заливка» (рис. 91). На странице «Выравнивание» расположены список для выбора вида выравнивания (формируется при активации формы), кнопки «ОК» и «Отмена». При нажатии кнопки «ОК» в поле на главной форме меняется вид выравнивания текста и четвертая форма скрывается. При нажатии кнопки «Отмена» четвертая форма скрывается.

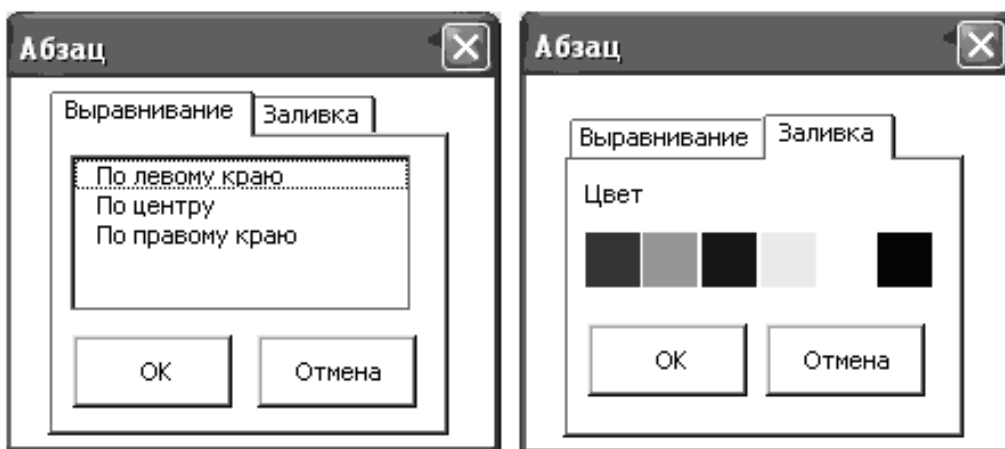


Рис. 91. Четвертая форма после запуска

На странице «Заливка» расположены метка с пояснительным текстом, кнопки «ОК» и «Отмена», шесть рисунков. Рисунки закрашены, соответственно, в красный, зеленый, синий, желтый, белый и черный цвет. Цвет границы рисунков – бирюзовый, тип границы – невидимый. При щелчке по какому-либо рисунку его граница становится видна (т.е. цвет выбран), границы остальных рисунков не видны, запоминается соответствующий код цвета. При нажатии кнопки «ОК» цвет поля на главной форме меняется в

зависимости от выбранного цвета и четвертая форма скрывается. При нажатии кнопки «Отмена» четвертая форма скрывается.

На пятой форме расположена метка с текстом о возможностях созданного приложения (рис. 92).

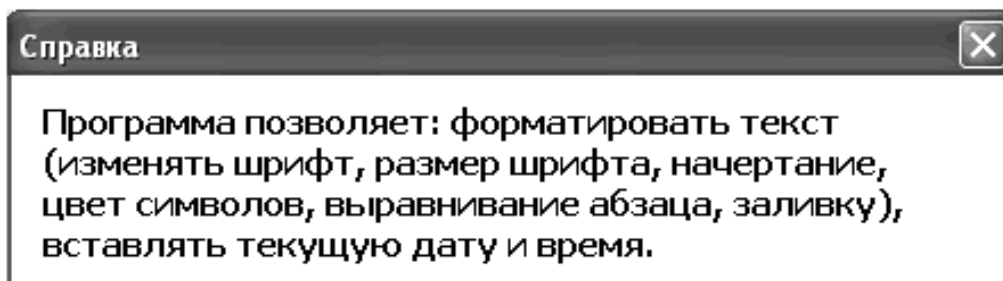


Рис. 92. Пятая форма

Проект «СТУДЕНТ»

Разработать и создать свой проект на любую тему, включающий:

- главную форму с организованным на ней меню для выполнения определенных действий или вызова вспомогательных форм;
- вспомогательные формы (минимум пять) с кнопками для перехода к главной форме и между ними;
- кнопку на рабочем листе Excel для запуска проекта с проверкой пароля.

Использовать в проекте все виды элементов управления для организации пользовательского интерфейса. Использовать различные способы ввода и вывода данных.

Библиографический список

1. *Гарбер Г. З.* Основы программирования на Visual Basic и VBA в Excel 2007 / Г. З. Гарбер. – М.: Солон-Пресс, 2008.
2. *Зиборов В.В.* Visual Basic 2010 на примерах / В.В. Зиборов. – СПб.: БХВ-Петербург, 2010.
3. *Браун С.* Visual Basic 6.0: учебный курс / С. Браун. – СПб.: Питер, 2007.
4. *Слепцова Л.Д.* Программирование на VBA в Microsoft Office 2010. Самоучитель / Л.Д. Слепцова. – М.: Диалектика, 2010.

ПРИЛОЖЕНИЕ 1

Функция **InputBox** выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа *String*, содержащее текст, введенный в поле.

Синтаксис:

InputBox(сообщение [,заголовок] [,default] [,Xpos] [,Ypos])

Аргументы:

- *сообщение* – текст, отображаемый в диалоговом окне. Для разделения строк используется символ возврата каретки (`chr(13)`);
- *заголовок* – текст, отображаемый в строке заголовка диалогового окна. Если этот аргумент опущен, в строку помещается имя приложения;
- *default* – текст, отображаемый в поле ввода как используемый по умолчанию, если пользователь не введет другую строку. Если этот аргумент опущен, поле ввода отображается пустым;
- *Xpos* – горизонтальная координата окна ввода;
- *Ypos* – вертикальная координата окна ввода.

Чтобы передать введенное в поле ввода значение программе, нужно присвоить значение функции **InputBox** переменной, например:
`S = InputBox(“Введите имя”, “Ввод”, “Иван”)`

Процедура **MsgBox** выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем, а затем возвращает значение типа *Integer*, указывающее, какая кнопка была нажата.

Синтаксис:

MsgBox(сообщение [,кнопки] [,заголовок] [,справка, раздел])

Аргументы:

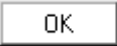
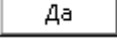
- *сообщение* – текст, отображаемый в диалоговом окне;
- *кнопки* – числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку и

модальность окна сообщения. Значение по умолчанию равно 0. Все значения данного аргумента см. в табл. П. 1.1, П. 1.2;

- *заголовок* – текст, отображаемый в строке заголовка диалогового окна. Если этот аргумент опущен, в строку помещается имя приложения;
- *справка* – имя файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот аргумент указан, необходимо наличие также аргумента *раздел*;
- *раздел* – числовое выражение, определяющее номер соответствующего раздела справочной системы.

Таблица П. 1.1

Значения аргумента кнопки процедуры MsgBox

Константа	Значение	Отображаемые кнопки
vbOkOnly	0	
vbOkCancel	1	 
vbAbortRetryIgnore	2	  
vbYesNoCancel	3	  
vbYesNo	4	 
vbRetryCancel	5	 

Примечание. Первые кнопки активны по умолчанию.

Таблица П. 1.2

Значения аргумента кнопки процедуры MsgBox

Константа	Значение	Описание	Значок сообщения
vbCritical	16	Критический	
vbQuestion	32	Вопрос	
vbExclamation	48	Предупреждение	
vbInformation	64	Информация	

Для проверки, на какой из кнопок был щелчок, можно использовать условный оператор. Например:

If MsgBox («Начинать?», vbYesNo + vbQuestion) = vbYes then

Операторы на действие этой кнопки
End If

ПРИЛОЖЕНИЕ 2

Технология **DragAndDrop** позволяет перемещать различные объекты во время работы программы. Когда пользователь захватывает выбранный им объект, возникает некоторое событие, обработчик которого вызывает метод *StartDrag* объекта *DataObject*, который хранит значение перетаскиваемого элемента.

Синтаксис метода:

```
Function StartDrag ([Effect As fmDropEffect]) As fmDropEffect
```

Эта функция обычно вызывается в операторе присваивания вида:

```
Result = объект.StartDrag([Effect As fmDropEffect])
```

Необязательный параметр *Effect* и результат выполнения функции принадлежат перечислению *fmDropEffect* (табл. П. 2.1).

Таблица П. 2.1

Значения аргумента параметра *Effect*

Константа	Значение	Описание
<i>fmDropEffectNone</i>	0	Не копировать и не передвигать опущенный исходный элемент на место назначения
<i>fmDropEffectCopy</i>	1	Копировать опущенный исходный элемент на место назначения
<i>fmDropEffectMove</i>	2	Передвинуть опущенный исходный элемент на место назначения
<i>fmDropEffectCopyOrMove</i>	3	Скопировать или передвинуть опущенный исходный элемент на место назначения

Параметр *Effect* задает цель операции и имеет по умолчанию значение 1 (*fmDropEffectCopy*). Обычно он опускается, поскольку значение по умолчанию задает наиболее вероятную цель операции. Значение, возвращаемое методом *StartDrag*, определяет результат выполнения операции. Его можно использовать для анализа того, что же произошло в результате перетаскивания на самом деле. Между запуском метода *StartDrag* в правой части оператора присваивания и присваиванием результата левой части переменной *Result* в процессе

перемещения объекта происходит много событий. Работают обработчики этих событий, и результат говорит о том, как закончился этот процесс.

ПРИЛОЖЕНИЕ 3

Таблица П. 3.1

Коды цветов в формате RGB

Цвет	RGB-код		
	Красный	Зеленый	Синий
Белый	255	255	255
Светло-серый	210	210	210
Серый	190	190	190
Темно-серый	84	84	84
Черный	0	0	0
Салатовый	143	188	143
Ярко-салатовый	0	255	127
Бледно-зеленый	152	251	152
Светло-зеленый	107	142	35
Ярко-зеленый	0	255	0
Оливково-зеленый	79	79	47
Зеленый травяной	66	111	66
Темно-зеленый	47	79	47
Морской волны	35	142	124
Бирюзовый	64	224	208
Небесно-голубой	173	234	234
Голубой	100	149	237
Ярко-голубой	50	153	204
Темно-голубой	35	107	142
Серо-голубой	192	217	217
Светло-синий	50	120	240
Синий	0	0	255

Темно-синий	0	0	128
Светло-сиреневый	199	163	199
Сиреневый	159	95	159

Окончание табл. П. 3.1

Лиловый	153	50	205
Светло-фиолетовый	143	143	189
Фиолетовый	79	47	79
Светло-желтый	255	255	224
Желтый	255	255	0
Золотой	205	127	50
Яркий золотой	217	217	25
Охра	142	107	35
Светло-оранжевый	255	218	185
Оранжевый	255	127	0
Медный	184	115	51
Светло-розовый	255	228	225
Розовый	255	110	199
Ярко-розовый	255	28	174
Розово-красный	204	50	153
Коралловый	254	105	73
Красный	255	0	0
Темно-красный	160	0	0
Красно-коричневый	142	35	35
Бордовый	109	27	68
Бежевый	245	245	220
Темно-бежевый	210	180	140
Древесный	166	128	100
Древесный, темный	133	94	66
Горчичный	198	129	0

Коричневый	100	65	0
Темно-коричневый	92	64	51

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. ОСНОВНЫЕ ПОНЯТИЯ	4
1.1. Объектно-ориентированное программирование	4
1.2. Разработка пользовательского интерфейса	5
1.3. Основы языка Visual Basic	12
2. ПРИМЕРЫ СОЗДАНИЯ ПРИЛОЖЕНИЙ	16
Пример 1	16
Пример 2	18
Пример 3	19
Пример 4	19
Пример 5	22
Пример 6	22
Пример 7	23
Пример 8	24
Пример 9	25
Пример 10	27
Пример 11	30
Пример 12	31
Пример 13	32
Пример 14	34
Пример 15	35
Пример 16	36
Пример 17	40
Пример 18	41
Пример 19	43
Пример 20	45
Пример 21	47
3. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	50
Проект «Мой первый проект»	50
Проект «Приветствие»	50
Проект «Валюта»	51
Проект «Простой калькулятор»	51
Проект «Вычисления»	51
Проект «Скорость автомобиля»	52
Проект «Скорость»	52
Проект «Просмотр рисунка»	53
Проект «Догони!»	53
Проект «Самолет»	54
Проект «Загадка»	54
Проект «Магазин»	55
Проект «КВУР»	56
Проект «Счастливая семерка»	56
Проект «Возраст»	56

Проект «Оценка».....	57
Проект «Факториал числа».....	57
Проект «Таблица умножения»	58
Проект «Список».....	58
Проект «Километры – мили».....	58
Проект «Фаренгейт»	59
Проект «Сумма чисел»	59
Проект «Угадай число».....	60
Проект «Склад».....	60
Проект «Скрепер»	61
Проект «Турфирма»	62
Проект «Автомобиль».....	62
Проект «Соппротивление»	63
Проект «Тест»	63
Проект «Кредит».....	64
Проект «Операции»	65
Проект «Вклад».....	65
Проект «Перевод»	66
Проект «Заказ путевок»	68
Проект «Товары».....	69
Проект «Анкета»	70
Проект «Калькулятор».....	74
Проект «Блокнотик»	74
Проект «Стипендиальный фонд»	75
Проект «Персональный секретарь».....	76
Проект «Тир».....	77
Проект «Молекула»	77
Проект «Гонки».....	78
Проект «Уравнения»	79
Проект «Пароль».....	80
Проект «Словарь»	81
Проект «Квадрат числа»	82
Проект «Блокнот»	83
Библиографический список.....	83
ПРИЛОЖЕНИЯ.....	84

Учебное издание

Наталья Федоровна Антипенко,
Татьяна Александровна Санькова

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКИХ
ПРИЛОЖЕНИЙ СРЕДСТВАМИ VBA

Учебно-методическое пособие

Редактор *Е.В. Садина*

Подписано к печати 02.04.2012
Формат 60×90 1/16. Бумага писчая
Оперативный способ печати
Гарнитура Times New Roman
Усл. п. л. 5,75, уч.-изд. л. 4,6
Тираж 300 экз. Заказ № ____

Издательство СибАДИ
644099, г. Омск, ул. П. Некрасова, 10

Отпечатано в ПЦ издательства СибАДИ