

Никита Культин

# Visual Basic

**ОСВОЙ НА ПРИМЕРАХ**

Санкт-Петербург

«БХВ-Петербург»

2004

УДК 681.3.068+800.92VisualBasic  
ББК 32.973.26-018.1  
К90

**Культин Н. Б.**

К90 Visual Basic. Освой на примерах. — СПб.: БХВ-Петербург, 2004. — 288 с.: ил.

ISBN 5-94157-521-1

Рассмотрены примеры на языке Visual Basic — от простейших до приложений работы с графикой, мультимедиа и базами данных — которые демонстрируют назначение компонентов и раскрывают тонкости процесса программирования.

Справочник содержит описания базовых компонентов и наиболее часто используемых функций. На прилагаемом компакт-диске находятся исходные тексты программ.

*Для начинающих программистов*

УДК 681.3.068+800.92VisualBasic  
ББК 32.973.26-018.1

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Владимир Красильников</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Виктория Пиотровская</i>
Дизайн обложки	<i>Игоря Цырульниковой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 25.06.04.

Формат 60×90<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 18.

Тираж 5000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-521-1

© Культин Н. Б.  
© Оформление, издательство "БХВ-Петербург", 2004

# Содержание

<b>Предисловие .....</b>	<b>1</b>
<b>ЧАСТЬ I. ПРИМЕРЫ.....</b>	<b>3</b>
<b>Глава 1. Базовые компоненты .....</b>	<b>5</b>
Общие замечания .....	5
<b>Глава 2. Графика.....</b>	<b>55</b>
Общие замечания .....	55
<b>Глава 3. Мультимедиа .....</b>	<b>99</b>
Общие замечания .....	99
<b>Глава 4. Файлы .....</b>	<b>129</b>
Общие замечания .....	129
<b>Глава 5. Игры и полезные программы.....</b>	<b>141</b>
<b>Глава 6. Базы данных .....</b>	<b>207</b>
Общие замечания .....	207
<b>ЧАСТЬ II. КРАТКИЙ СПРАВОЧНИК.....</b>	<b>227</b>
<b>Глава 7. Компоненты .....</b>	<b>229</b>
Форма.....	229
<i>Label</i> .....	231
<i>TextBox</i> .....	233

<i>CommandButton</i> .....	234
<i>CheckBox</i> .....	236
<i>OptionButton</i> .....	237
<i>ListBox</i> .....	238
<i>ComboBox</i> .....	239
<i>Timer</i> .....	241
<i>DriveListBox</i> .....	242
<i>DirListBox</i> .....	243
<i>FileListBox</i> .....	244
<i>PictureBox</i> .....	246
<i>Image</i> .....	248
<i>Shape</i> .....	249
<i>Line</i> .....	251
<i>UpDown</i> .....	252
<i>CommonDialog</i> .....	254
<i>MMControl</i> .....	255
<b>Глава 8. Графика</b> .....	<b>257</b>
<i>Print</i> .....	257
<i>Line</i> .....	258
<i>Circle</i> .....	259
<i>RGB</i> .....	260
<b>Глава 9. Функции</b> .....	<b>264</b>
Ввод и вывод.....	264
Математические функции .....	266
Преобразование данных .....	267
Работа со строками.....	268
Работа с файлами.....	273
<b>Глава 10. События</b> .....	<b>278</b>
<b>Приложение. Содержание компакт-диска, прилагаемого к книге Культина Н. Б. "Visual Basic. Освой на примерах"</b> .....	<b>281</b>
<b>Предметный указатель</b> .....	<b>283</b>

# Предисловие

В последнее время резко возрос интерес к программированию. Это связано с развитием и внедрением в повседневную жизнь общества информационных технологий. Если человек имеет дело с компьютером, то рано или поздно у него возникает желание, а иногда и необходимость программировать.

Бурное развитие вычислительной техники, потребность в эффективных средствах разработки программного обеспечения, привели к появлению систем программирования, ориентированных на так называемую "быструю разработку". В основе идеологии систем быстрой разработки или *RAD-систем* (Rapid Application Development — среда быстрой разработки приложений) лежат технологии визуального проектирования и событийного объектно-ориентированного программирования. Суть этих технологий заключается в том, что среда разработки берет на себя большую часть рутинной работы по формированию программного кода, оставляя программисту решение задач по конструированию диалоговых окон и созданию функций обработки событий. В связи с этим, производительность процесса программирования, при использовании *RAD-систем*, фантастическая!

Среди *RAD-систем* особо выделяется среда Microsoft Visual Basic, которая позволяет создавать различные программы от простейших однооконных приложений, до программ управления базами данных. В качестве языка программирования в среде Microsoft Visual Basic используется Visual Basic.

Чтобы научиться программировать, надо программировать — писать программы, решать конкретные задачи. Для этого

необходимо изучить как язык программирования, так и среду разработки.

Освоить язык программирования Visual Basic не очень сложно. Труднее изучить среду разработки и научиться использовать ее компоненты. Хорошим подспорьем здесь могут быть программы, которые демонстрируют как назначение компонентов, так и особенности их использования.

В книге, которую вы держите в руках, собраны разнообразные примеры, которые не только демонстрируют возможности среды разработки Microsoft Visual Basic, но и знакомят с принципами организации обработки графической, звуковой информации, а также баз данных. Следует обратить внимание, что большинство примеров не являются учебными, в прямом смысле этого слова, и представляют собой вполне работоспособные программы.

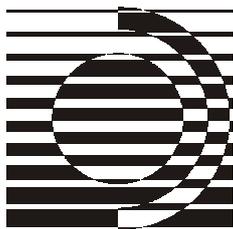
Состоит книга из двух частей и приложения.

Первая часть содержит примеры. Примеры представлены в виде краткого описания, диалоговых окон и хорошо документированных текстов программ. Для простых задач приведены только функции обработки событий. Текст остальных программ приведен полностью.

Вторая часть книги — краткий справочник, в котором можно найти описание компонентов и функций Visual Basic.

Прилагаемый к книге компакт-диск содержит проекты, которые рассмотрены в ней в качестве примеров. Каждый проект находится в отдельном каталоге. Помимо файлов проекта, в каталоге находится исполняемый файл, что позволяет, без загрузки его в среду Microsoft Visual Basic, увидеть, как работает программа.

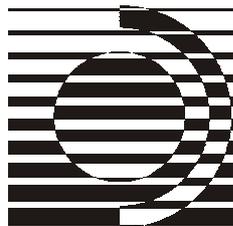
Как уже упоминалось ранее, научиться программировать можно, только решая конкретные задачи. При этом достигнутые в этой области успехи в значительной степени зависят от опыта разработчика. Поэтому, чтобы получить от книги максимальную пользу, вы должны работать с ней активно. Изучайте листинги, старайтесь понять, как работают программы. Не бойтесь экспериментировать — вносите в программы изменения. Если что-то не понятно, обратитесь к справочнику в конце книги или к справочной системе Visual Basic.



**Часть I**

**ПРИМЕРЫ**

# Глава 1



## Базовые компоненты

### Общие замечания

Процесс создания программы в Visual Basic состоит из двух шагов. Сначала нужно создать форму программы (диалоговое окно), затем написать процедуры обработки событий. Форма приложения (так принято называть прикладные программы, работающие в Windows) создается путем добавления в форму компонентов и последующей их настройки.

В форме практически любого приложения есть компоненты, которые обеспечивают интерфейс (взаимодействие) между программой и пользователем. Такие компоненты называют базовыми. К базовым компонентам можно отнести следующие:

- `Label` — поле вывода текста;
- `TextBox` — поле ввода/редактирования текста;
- `CommandButton` — командная кнопка;
- `CheckBox` — независимая кнопка выбора;
- `OptionButton` — зависимая кнопка выбора;
- `ListBox` — список выбора;
- `ComboBox` — комбинированный список выбора.

Вид компонента, его размер и поведение определяют значения *свойств* (характеристик) компонента.

Основную работу в программе выполняют процедуры обработки *событий*.

### Примечание

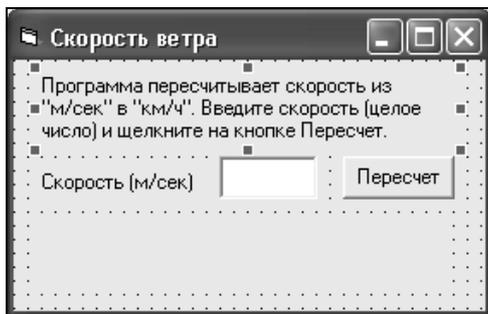
Описание свойств базовых компонентов и основных событий можно найти в справочнике, который приводится во второй части книги.

Исходную информацию программа может получить из полей ввода/редактирования (компонент — `TextBox`), списка выбора (компонент — `ListBox`) или комбинированного списка (компонент — `ComboBox`). Для ввода значений логического типа можно использовать компоненты `CheckBox` и `OptionButton`.

Конечные или промежуточные результаты программа может вывести в поле вывода текста (компонент — `Label`) или в окно сообщения (функция — `MsgBox`).

Для преобразования текста, находящегося, например, в поле ввода/редактирования, в число нужно использовать функцию `Val`. Для преобразования числа, например значения переменной, в строку можно использовать функцию `Format` или `CStr`.

1. Программа пересчитывает скорость ветра из "метров в секунду" в "километры в час". Форма программы приведена на рис. 1.1.



**Рис. 1.1.** Форма программы **Скорость ветра**

```
' щелчок на кнопке Пересчет
Private Sub Command1_Click()
    Dim ms As Integer      ' скорость м/сек
    Dim kmh As Single     ' скорость км/ч

    ms = Val(Text1.Text)  ' ввод исходных данных
```

```
kmh = ms * 3.6          ' пересчет
' вывод результата
Label3.Caption = Format$(ms) + " м/сек - это " + _
                Format$(kmh) + " км/ч"
```

**End Sub**

2. Программа, как и в примере 1, пересчитывает скорость ветра из "метров в секунду" в "километры в час". Форма программы приведена на рис. 1.1. Программа составлена таким образом, что пользователь смог ввести в поле **Скорость [м/сек]** только целое положительное число.

```
' щелчок на кнопке Пересчет
Private Sub Command1_Click()
    Dim ms As Integer    ' скорость м/сек
    Dim kmh As Single    ' скорость км/ч

    ' проверим, ввел ли пользователь число в поле Скорость
    If Len(Text1.Text) = 0 Then
        ' если число не введено, то выводится сообщение
        ' "Нужно ввести скорость"
        Label3.Caption = " Нужно ввести скорость "
    Else
        ms = Val(Text1.Text) ' ввод исходных данных
        kmh = ms * 3.6       ' пересчет
        ' вывод результата
        Label3.Caption = Format$(ms) + " м/сек - это " + _
                        Format$(kmh) + " км/ч"
    End If
End Sub

' нажатие клавиши в поле Скорость
Private Sub Text1_KeyPress(KeyAscii As Integer)
    ' В поле Скорость можно вводить только цифры. Данная
    ' процедура проверяет, является ли введенный символ
    ' цифрой. Если нет, то введенный символ заменяется
    ' нулевым и в поле редактирования не отображается.
```

```

' KeyAscii - это код нажатой клавиши. 48 - 57 - коды
' цифр от 0 до 9, 8 - код клавиши <Backspace>.
If Not (KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8)
Then
    KeyAscii = 0
End If
End Sub

```

3. Программа пересчитывает скорость ветра из "метров в секунду" в "километры в час". Форма программы приведена на рис. 1.1. Программа составлена таким образом, что пользователь может ввести в поле **Скорость** только целое положительное число. Вычисление выполняется как в результате щелчка мышью на кнопке **Пересчет**, так и в результате нажатия клавиши <Enter>, после ввода последней цифры данных в поле **Скорость**.

```

' процедура пересчитывает скорость из м/сек в км/ч
Private Sub WindSpeed()
    Dim ms As Integer ' скорость м/сек
    Dim kmh As Single ' скорость км/ч

    ' проверим, ввел ли пользователь число в поле Скорость
    If Len(Text1.Text) = 0 Then
        ' если число не введено, то выводится сообщение
        ' "Нужно ввести скорость"
        Label3.Caption = " Нужно ввести скорость "
    Else
        ms = Val(Text1.Text) ' ввод исходных данных
        kmh = ms * 3.6 ' пересчет
        ' вывод результата
        Label3.Caption = Format$(ms) + " м/сек - это " + _
            Format$(kmh) + " км/ч"
    End If
End Sub

```

' щелчок на кнопке Пересчет

```
Private Sub Command1_Click()  
    Call WindSpeed  
End Sub  
  
' нажатие клавиши в поле Edit1  
Private Sub Text1_KeyPress(KeyAscii As Integer)  
    ' KeyAscii - код нажатой клавиши  
    ' 48 - 57 - коды цифр от 0 до 9  
    ' 8 - код клавиши <Backspace>  
    ' 13 - код клавиши <Enter>  
    Select Case KeyAscii  
        Case 48 To 57, 8 ' цифры и <Backspace>  
        Case 13          ' <Enter>  
            Call WindSpeed  
        Case Else       ' остальные символы  
            KeyAscii = 0  
    End Select  
End Sub
```

4. Программа пересчитывает вес из фунтов в килограммы (1 фунт — 409,5 г). Форма программы приведена на рис. 1.2. Программа составлена таким образом, что кнопка **Пересчет** доступна только в том случае, если пользователь ввел исходные данные.

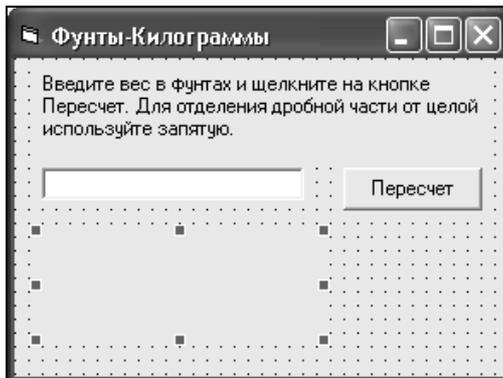


Рис. 1.2. Форма программы **Фунты-Килограммы**

```
' щелчок на кнопке Пересчет
Private Sub Command1_Click()
    Dim funt As Single    ' вес в фунтах
    Dim kg As Single     ' вес в килограммах

    ' Кнопка Пересчет доступна только в том случае, если
    ' в поле Фунты есть данные. Поэтому наличие информации
    ' в поле можно не проверять.
    funt = Val(Text1.Text)
    kg = funt * 0.4995
    Label2.Caption = Format$(funt) + " ф - это " + _
        Format$(kg) + " кг"
End Sub

' инициализация формы
Private Sub Form_Initialize()
    ' поле Фунты пусто (пользователь еще не ввел исходные
    ' данные), сделаем кнопку Пересчет недоступной
    Command1.Enabled = False
End Sub

' содержимое поля Фунты изменилось
Private Sub Text1_Change()
    ' проверим, есть ли в поле Фунты исходные данные
    If Len(Text1.Text) = 0 Then
        Command1.Enabled = False ' кнопка Пересчет недоступна
    Else
        Command1.Enabled = True  ' кнопка Пересчет доступна
    End If
End Sub

' нажатие клавиши в поле Фунты
Private Sub Text1_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
```

```
Case 48 To 57, 8 ' цифры 0 - 9 и <Backspace>
' обработка десятичного разделителя
Case 44, 46 ' 44 - код запятой, 46 - код точки
' если в поле Фунты введена запятая, то
' заменим ее на десятичный разделитель
KeyAscii = 46
' проверим, введен ли уже в поле Фунты
' десятичный разделитель
If InStr(Text1.Text, ".") <> 0 Then
    KeyAscii = 0
End If
Case Else
    KeyAscii = 0 ' остальные символы запрещены
End Select
End Sub
```

5. Программа вычисляет скорость (км/час), с которой бегун пробежал дистанцию. Форма программы приведена на рис. 1.3. Количество минут задается целым числом, количество секунд — дробным.

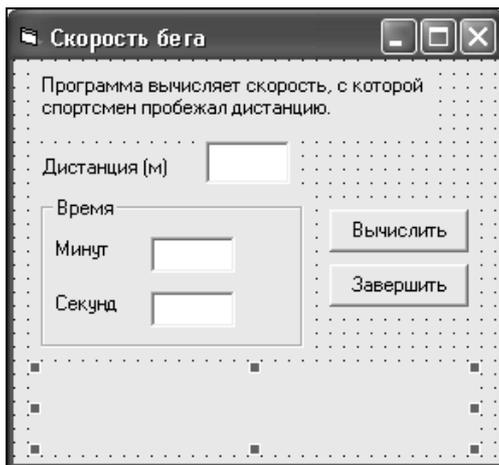


Рис. 1.3. Форма программы **Скорость бега**

```

' щелчок на кнопке Вычислить
Private Sub Command1_Click()
    Dim dist As Integer ' дистанция, метров
    Dim min As Integer ' время, минуты
    Dim sek As Single ' время, секунды
    Dim v As Single ' скорость

    ' получение исходных данных из полей ввода
    dist = Val(Text1.Text)
    min = Val(Text2.Text)
    sek = Val(Text3.Text)

    ' дистанция и время не должны быть равны нулю
    If (dist = 0) Or ((min = 0) And (sek = 0)) Then
        Label5.Caption = "Нужно задать дистанцию и время."
        Exit Sub
    End If

    ' вычисление
    v = (dist / 1000) / ((min * 60 + sek) / 3600)

    ' вывод результата
    Label5.Caption = "Дистанция: " + Format$(dist) + _
        " м" + Chr(13) + _
        "Время: " + Format$(min) + " мин " + _
        Format$(sek) + " сек " + Chr(13) + _
        "Скорость: " + Format$(v, "0.00") + _
        " км/час"

    ' функция Chr() возвращает символ по значению
    ' числового кода в Ascii
End Sub

' щелчок на кнопке Завершить
Private Sub Command2_Click()
    ' закрытие главной формы

```

```
Unload Form1
```

```
End Sub
```

```
' нажатие клавиши в поле Дистанция
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
    ' KeyAscii - код символа, соответствующего нажатой клавише.
```

```
    ' Если символ недопустимый, то процедура заменяет его
```

```
    ' на символ с кодом 0. В результате этого символ в поле
```

```
    ' редактирования не появляется.
```

```
    Select Case KeyAscii
```

```
        Case 48 To 57, 8 ' цифры 0 - 9 и <Backspace>
```

```
        Case 13
```

```
            ' при нажатии клавиши <Enter> курсор переводится
```

```
            ' в поле Время:Минут
```

```
            Text2.SetFocus
```

```
        Case Else
```

```
            KeyAscii = 0 ' остальные символы не отображаются
```

```
    End Select
```

```
End Sub
```

```
' нажатие клавиши в поле Время:Минут
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
```

```
    Select Case KeyAscii
```

```
        Case 48 To 57, 8 ' цифры 0 - 9 и <Backspace>
```

```
        Case 13
```

```
            ' при нажатии клавиши <Enter> курсор переводится
```

```
            ' в поле Время:Секунд
```

```
            Text3.SetFocus
```

```
        Case Else
```

```
            KeyAscii = 0 ' остальные символы не отображаются
```

```
    End Select
```

```
End Sub
```

```
' нажатие клавиши в поле Время:Секунд
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
```

```

Select Case KeyAscii
    Case 48 To 57, 8 ' цифры 0 - 9 и <Backspace>
    Case 44, 46      ' десятичный разделитель
        KeyAscii = 46
        If InStr(Text3.Text, ".") <> 0 Then
            KeyAscii = 0
        End If
    Case 13          ' клавиша <Enter>
        ' при нажатии клавиши <Enter> устанавливается
        ' фокус на кнопку Вычислить
        Command1.SetFocus
    Case Else
        KeyAscii = 0 ' остальные символы не отображаются
End Select
End Sub

```

6. Программа вычисляет доход по вкладу. Она обеспечивает расчет простых и сложных процентов. Простые проценты начисляются в конце срока вклада, сложные — ежемесячно и прибавляются к первоначальной сумме вклада. В следующем месяце проценты начисляются на новую сумму. Форма программы приведена на рис. 1.4.

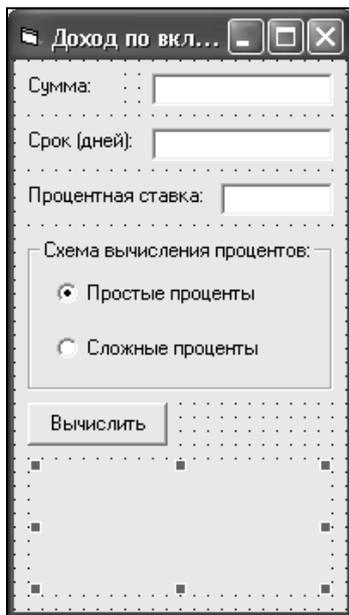
' щелчок на кнопке Вычислить

```

Private Sub Command1_Click()
    Dim sum As Single ' сумма вклада
    Dim pr As Single ' процентная ставка
    Dim srok As Integer ' срок вклада
    Dim dohod As Single ' доход по вкладу
    Dim buf As Single
    Dim i As Integer

    ' получение исходных данных
    sum = Val(Text1.Text)
    srok = Val(Text2.Text)
    pr = Val(Text3.Text)

```



**Рис. 1.4.** Форма программы **Доход по вкладу**

```
If Option1.Value = True Then
```

```
    ' выбран переключатель Простые проценты
    dohod = sum * (pr / 100) * (srok / 360)
```

```
Else
```

```
    ' выбран переключатель Сложные проценты
    buf = sum
```

```
    For i = 1 To srok
```

```
        buf = buf + buf * (pr / 100)
```

```
        ' сумма в конце срока вклада записывается в buf
        dohod = buf - sum
```

```
    Next i
```

```
End If
```

```
sum = sum + dohod
```

```
Label4.Caption = "Доход: " + Format$(dohod, "0.00") + _
    Chr(13) + _
```

```

"Сумма в конце срока вклада: " + _
Format$(sum, "0.00")

```

```
End Sub
```

```
' выбор переключателя Простые проценты
```

```
Private Sub Option1_Click()
```

```
Label2.Caption = "Срок (дней):"
```

```
Label4.Caption = ""
```

```
End Sub
```

```
' выбор переключателя Сложные проценты
```

```
Private Sub Option2_Click()
```

```
Label2.Caption = "Срок (мес.):"
```

```
Label4.Caption = ""
```

```
End Sub
```

7. Программа вычисляет сопротивление электрической цепи, состоящей из двух сопротивлений. Сопротивления могут быть соединены последовательно или параллельно. Форма программы приведена на рис. 1.5. Если величина сопротивления цепи превышает 1000 Ом, результат выводится в кОм (килоомах).

```
' щелчок на кнопке Вычислить
```

```
Private Sub Command1_Click()
```

```
Dim r1, r2 As Single ' значения сопротивлений R1 и R2
```

```
Dim r As Single ' сопротивление цепи
```

```
' получение исходных данных
```

```
r1 = Val(Text1.Text)
```

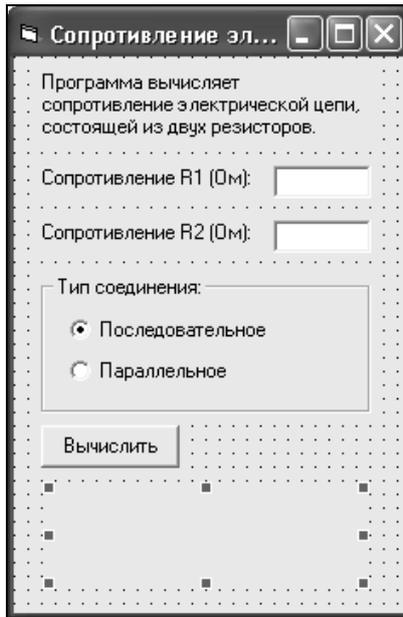
```
r2 = Val(Text2.Text)
```

```
If (r1 = 0) And (r2 = 0) Then
```

```
Label4.Caption = "Нужно задать величину хотя бы " + _
"одного сопротивления."
```

```
Exit Sub
```

```
End If
```



**Рис. 1.5.** Форма программы **Сопротивление электрической цепи**

```
' Переключатели "Последовательное соединение" и
' "Параллельное соединение" зависимы, поэтому о типе
' соединения можно судить по состоянию одного из них.
```

```
If Option1.Value = True Then
```

```
    ' выбран переключатель Последовательное соединение
    r = r1 + r2
```

```
Else
```

```
    ' выбран переключатель Параллельное соединение
    r = (r1 * r2) / (r1 + r2)
```

```
End If
```

```
Label4.Caption = "Сопротивление цепи: "
```

```
If r < 1000 Then
```

```
    Label4.Caption = Label4.Caption + _
    Format$(r, "0.00") + " Ом"
```

```

Else
    r = r / 1000
    Label4.Caption = Label4.Caption + _
    Format$(r, "0.00") + " кОм"
End If
End Sub

' щелчок на переключателе "Последовательное соединение"
Private Sub Option1_Click()
    ' пользователь изменил тип соединения,
    ' очистим поле вывода от предыдущего вычисления
    Label4.Caption = ""
End Sub

' щелчок на переключателе "Параллельное соединение"
Private Sub Option2_Click()
    Label4.Caption = ""
End Sub

```

8. Программа вычисляет силу тока, напряжение или сопротивление электрической цепи, используя закон Ома. Форма программы приведена на рис. 1.6.

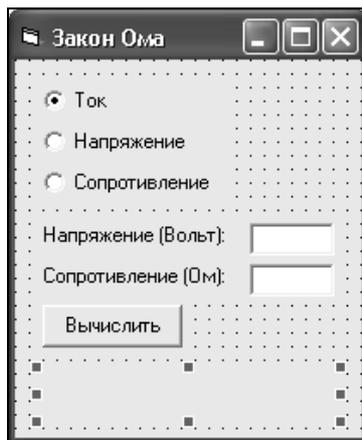


Рис. 1.6. Форма программы **Закон Ома**

*'* вычисление тока, напряжения или сопротивления

**Sub** Calculate()

**Dim** I **As** **Single**     *'* ток  
**Dim** U **As** **Single**     *'* напряжение  
**Dim** R **As** **Single**     *'* сопротивление

**If** Option1.Value = **True** **Then**

*'* ток

   U = Val(Text1.Text)

   R = Val(Text2.Text)

**If** R <> 0 **Then**

      I = U / R

      Label3.Caption = "Ток: " + Format\$(I, "0.00") + " А"

**Else**

      Label3.Caption = "Сопротивление не должно быть " + \_  
                          "равно нулю."

**End If**

**Exit Sub**

**End If**

**If** Option2.Value = **True** **Then**

*'* напряжение

   I = Val(Text1.Text)

   R = Val(Text2.Text)

   U = I \* R

   Label3.Caption = "Напряжение: " + \_  
                          Format\$(U, "0.00") + " В"

**Exit Sub**

**End If**

**If** Option3.Value = **True** **Then**

*'* сопротивление

   U = Val(Text1.Text)

   I = Val(Text2.Text)

```
    If I <> 0 Then
        R = U / I
        Label3.Caption = "Сопротивление: " + _
            Format$(R, "0.00") + " Ом"

    Else
        Label3.Caption = "Ток не должен быть равен нулю."
    End If
End If
End Sub

' щелчок на кнопке Вычислить
Private Sub Command1_Click()
    If Text1.Text <> "" And Text2.Text <> "" Then
        Calculate
    Else
        Label3.Caption = "Нужно ввести исходные данные " + _
            "в оба поля."
    End If
End Sub

' выбор переключателя Ток
Private Sub Option1_Click()
    Label1.Caption = "Напряжение (Вольт):"
    Label2.Caption = "Сопротивление (Ом):"
    Label3.Caption = ""
End Sub

' выбор переключателя Напряжение
Private Sub Option2_Click()
    Label1.Caption = "Ток (Ампер):"
    Label2.Caption = "Сопротивление (Ом):"
    Label3.Caption = ""
End Sub
```

```
' выбор переключателя Сопротивление
Private Sub Option3_Click()
    Label1.Caption = "Напряжение (Вольт):"
    Label2.Caption = "Ток (Ампер):"
    Label3.Caption = ""
End Sub

' нажатие клавиши в поле Напряжение/Ток
Private Sub Text1_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 48 To 57, 8 ' цифры и <Backspace>
        Case 13         ' клавиша <Enter>
            Text2.SetFocus
        Case 44, 46     ' точка и запятая
            KeyAscii = 46
            ' не позволяет вводить знак запятой повторно
            If InStr(Text1.Text, ".") <> 0 Then
                KeyAscii = 0
            End If
        Case Else
            KeyAscii = 0 ' остальные символы не отображаются
    End Select
End Sub

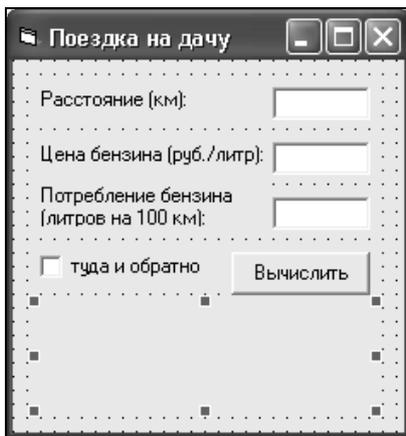
' нажатие клавиши в поле Сопротивление/Ток
Private Sub Text2_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 48 To 57, 8 ' цифры и <Backspace>
        Case 13         ' клавиша <Enter>
            Calculate
        Case 44, 46     ' точка и запятая
            KeyAscii = 46
            ' не позволяет вводить знак запятой повторно
            If InStr(Text2.Text, ".") <> 0 Then
```

```

        KeyAscii = 0
    End If
Case Else
    KeyAscii = 0 ' остальные символы не отображаются
End Select
End Sub

```

9. Программа вычисляет стоимость поездки на автомобиле, например, на дачу. Форма программы приведена на рис. 1.7.



**Рис. 1.7.** Форма программы **Поездка на дачу**

```

' При инициализации формы свойству Tag компонентов
' Text1, Text2 и Text3 присваиваются соответствующие
' значения 1, 2 и 3. Свойство Tag используется в
' процедуре KeyPress.

```

```

Private Sub Form_Initialize()
    Text1.Tag = 1 ' значение свойства Tag поля Text1
    Text2.Tag = 2 ' Tag поля Text2
    Text3.Tag = 3 ' Tag поля Text3
End Sub

```

```

' щелчок на кнопке Вычислить

```

```
Private Sub Command1_Click()
    Dim rast As Single ' расстояние
    Dim cena As Single ' цена
    Dim potr As Single ' потребление на 100 км
    Dim summ As Single ' сумма
    Dim mes As String

    ' получение исходных данных
    rast = Val(Text1.Text)
    cena = Val(Text2.Text)
    potr = Val(Text3.Text)

    ' При чтении данных из полей ввода
    ' возможен случай, когда пользователь оставит
    ' одно из полей ввода незаполненным.
    If rast = 0 Or cena = 0 Or potr = 0 Then
        Label4.Caption = "Данные нужно ввести во все поля."
        If Len(Text1.Text) = 0 Then
            Text1.SetFocus
        Else
            If Len(Text2.Text) = 0 Then
                Text2.SetFocus
            Else
                Text3.SetFocus
            End If
        End If
    End If

    Exit Sub
End If

summ = (rast / 100) * potr * cena
mes = "Поездка на дачу "

If Check1.Value = Checked Then
    summ = summ * 2
```

```

mes = mes + "и обратно "
End If

mes = mes + Chr(13) + "обойдется в " + _
    Format$(summ, "0.00") + " руб."
Label4.Caption = mes
End Sub

' Процедура KeyPress обрабатывает нажатие клавиш в полях
' Расстояние, Цена и Потребление.
' В качестве входных параметров этой процедуры используются
' код нажатой клавиши KeyAscii и имя поля, для которого эта
' процедура должна выполняться.
Sub KeyPress (KeyAscii As Integer, Text As TextBox)
    Select Case KeyAscii
        Case 48 To 57, 8 ' цифры и <Backspace>
        Case 44, 46     ' точка и запятая
            KeyAscii = 46
            ' не позволяет вводить знак запятой повторно
        If InStr(Text.Text, ".") <> 0 Then
            KeyAscii = 0
        End If
        Case 13        ' клавиша <Enter>
            Select Case Text.Tag
                Case 1      ' клавиша нажата в поле Text1
                    Text2.SetFocus
                Case 2      ' клавиша нажата в поле Text2
                    Text3.SetFocus
                Case 3      ' клавиша нажата в поле Text3
                    Command1.SetFocus
            End Select
        Case Else
            KeyAscii = 0 ' остальные символы не отображаются
    End Select

```

```
End Sub
```

```
' нажатие клавиши в поле Расстояние
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
    Call KeyPress(KeyAscii, Text1)
```

```
End Sub
```

```
' нажатие клавиши в поле Цена
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
```

```
    Call KeyPress(KeyAscii, Text2)
```

```
End Sub
```

```
' нажатие клавиши в поле Потребление
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
```

```
    Call KeyPress(KeyAscii, Text3)
```

```
End Sub
```

10. Программа "Калькулятор" выполняет сложение и вычитание. Форма программы приведена на рис. 1.8. Ниже представлены два варианта программы. В первом варианте для каждой цифровой кнопки создана отдельная процедура обработки события `Click`. Во втором варианте событие `Click` всех цифровых кнопок обрабатывает одна процедура, что позволило сократить текст программы.

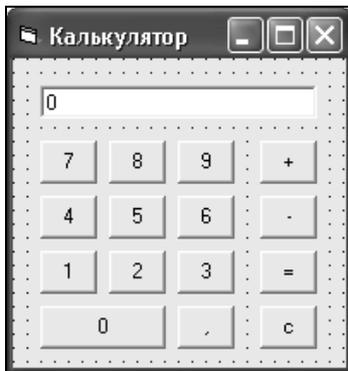


Рис. 1.8. Форма программы **Калькулятор**

```

' Вариант 1. Событие Click на каждой цифровой кнопке
' обрабатывает отдельная процедура

Dim accum As Single      ' аккумулятор
Dim oper As Integer      ' операция: 1 - "+", 2 - "-",
                          ' 0 - "выполнить" (кнопка "=")

Dim f As Integer

' f = 0 ожидание первой цифры нового числа, например, после
' выполнения операции, когда на индикаторе результат.
' f = 1 ожидание ввода остальных цифр.

' кнопка "0"
Private Sub Command0_Click()
    If f = 0 Then        ' первая цифра числа
        Text1.Text = "0"
        f = 1           ' ожидание остальных цифр
    Else

        ' Следующее условие нужно для того, чтобы на индикаторе
        ' не появлялось нескольких нулей в начале числа.
        If Text1.Text <> "0" Then
            Text1.Text = Text1.Text + "0"
        End If
    End If
End Sub

' кнопка "1"
Private Sub Command1_Click()
    If f = 0 Then        ' первая цифра числа
        Text1.Text = "1"
        f = 1           ' ожидание остальных цифр
    Else

        ' В случае, когда после выбора операции пользователь
        ' нажал "0", переменной f присваивается значение 1,

```

*' ожидается ввод остальных цифр числа. Для того чтобы  
' в поле ввода не появлялось чисел, начинающихся с нуля  
' и не являющихся дробными (например "01"), необходимо  
' выполнение следующего условия.*

```
If Text1.Text <> "0" Then  
    Text1.Text = Text1.Text + "1"  
Else: Text1.Text = "1"  
End If  
End If  
End Sub
```

```
' кнопка "2"  
Private Sub Command2_Click()  
    If f = 0 Then  
        Text1.Text = "2"  
        f = 1  
    Else  
        If Text1.Text <> "0" Then  
            Text1.Text = Text1.Text + "2"  
        Else: Text1.Text = "2"  
        End If  
    End If  
End Sub
```

```
' кнопка "3"  
Private Sub Command3_Click()  
    If f = 0 Then  
        Text1.Text = "3"  
        f = 1  
    Else  
        If Text1.Text <> "0" Then  
            Text1.Text = Text1.Text + "3"  
        Else: Text1.Text = "3"  
        End If
```