

Н. Культин

C#

в задачах и примерах

Санкт-Петербург

«БХВ-Петербург»

2007

УДК 681.3.06
ББК 32.973.26-018.1
К90

Культин Н. Б.

К90 С# в задачах и примерах. — СПб.:
БХВ-Петербург, 2007. — 240 с.: ил. + CD-ROM
ISBN 978-5-9775-0115-6

Книга представляет собой сборник задач и программ на языке С#. Примеры и задачи различной сложности — от простейших до приложений работы с графикой, мультимедиа и базами данных — демонстрируют назначение базовых компонентов, раскрывают тонкости разработки .NET-приложений в Borland Developer Studio или Turbo С#. Уделено внимание технологии ASP.NET. Справочник содержит описание базовых компонентов и часто используемых функций. Компакт-диск содержит дистрибутив Turbo С# Explorer, размещенный с разрешения Borland Software Corporation, а также проекты, рассматриваемые в книге.

Для начинающих программистов

УДК 681.3.06
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>Римма Смоляк</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Виктория Пиотровская</i>
Дизайн серии	<i>Игоря Цырульникова</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 22.05.07.

Формат 60×90^{1/16}. Печать офсетная. Усл. печ. л. 15.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

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

ISBN 978-5-9775-0115-6

© Культин Н. Б., 2007
© Оформление, издательство "БХВ-Петербург", 2007

Оглавление

Предисловие	5
Часть 1. Примеры и задачи	7
БАЗОВЫЕ КОМПОНЕНТЫ	7
Конвертер.....	8
Электроэнергия	10
Автозаправка	14
Кафе.....	16
Жалюзи.....	19
Стеклопакет	22
Калькулятор.....	25
Просмотр иллюстраций	30
Секундомер.....	34
Таймер.....	37
Угадай число.....	40
Справочная система.....	44
Работа с несколькими формами	47
ФАЙЛЫ	50
Погода	50
Средняя температура	53
Редактор текста.....	56
ГРАФИКА	66
Приветствие.....	66
Диаграмма.....	68
График.....	72
Круговая диаграмма.....	77
Использование кистей.....	83
Фоновый рисунок.....	85
Бегущая строка	87
Полет в облаках.....	90
БАЗЫ ДАННЫХ	95
Записная книжка	95
Контакты	101
Ежедневник.....	109
Чтение данных из xml-файла.....	120
Конвертер базы данных Microsoft Access.....	122
Каталог.....	126
ИГРЫ И ДРУГИЕ ПОЛЕЗНЫЕ ПРОГРАММЫ	133
Парные картинки.....	133
Собери картинку	141
Сапер.....	147
Master Mind.....	157
Будильник	169
Экзаменатор.....	174

ASP.NET	184
Добро пожаловать	184
Хорошие окна	186
Авторизация.....	188
Контакты.....	193
Часть 2. Краткий справочник.....	197
Форма.....	197
Компоненты.....	199
Button.....	199
ComboBox	201
ContextMenu.....	202
CheckBox.....	202
CheckedListBox	204
GroupBox.....	205
ImageList.....	206
Label.....	206
ListBox	207
MainMenu.....	208
NotifyIcon	209
NumericUpDown	209
OpenFileDialog	210
Panel.....	211
PictureBox.....	212
RadioButton	213
ProgressBar	215
SaveFileDialog	215
StatusBar	216
TextBox.....	218
ToolBar	219
ToolTip.....	220
Timer.....	220
Графика	221
Карандаш	221
Кисть	222
Графические примитивы	224
Типы данных.....	226
Целый тип	226
Вещественный тип	226
Символьный тип.....	227
Строковый тип.....	227
Функции	227
Функции преобразования	227
Функции манипулирования строками	229
Функции манипулирования датами и временем.....	230
Функции манипулирования каталогами и файлами	232
Математические функции.....	234
События	236
Исключения	237
Приложение. Описание прилагаемого компакт-диска.....	238
Предметный указатель.....	239



Предисловие

Среда разработки Borland® C#Builder® for the Microsoft® .NET Framework является одним из популярнейших средств разработки компьютерных программ.

Microsoft .NET — это новая технология разработки программного обеспечения. В ее основе лежит идея универсального программного кода, который может быть выполнен любым устройством (компьютером), вне зависимости от используемой этим устройством операционной системы (операционная система устройства должна поддерживать технологию .NET). Универсальность программного кода обеспечивается за счет предварительной (на этапе разработки) компиляции исходной программы в промежуточный код, который во время загрузки (выполнения) транслируется в выполняемую программу.

Чтобы понять, что такое .NET, какими преимуществами обладает эта технология, необходимо опробовать ее в деле. Для этого нужно изучить среду разработки, понять технологию разработки, назначение и возможности компонентов. И здесь хорошим подспорьем могут стать примеры, программы, разработанные другими программистами.

В книге, которую вы держите в руках, собраны разнообразные примеры, демонстрирующие назначение базовых компонентов, технологию работы с графикой, базами данных, разработку ASP.NET-приложений. Следует обратить внимание, что большинство примеров не являются учебными в чистом смысле, это — вполне работоспособные программы.

Состоит книга из двух частей.

Первая часть содержит примеры. Примеры представлены в виде краткого описания, диалоговых окон и хорошо документированных текстов программ.

Вторая часть книги — это краткий справочник. В нем можно найти описание базовых компонентов и наиболее часто используемых функций.

Научиться программировать можно только программируя, решая конкретные задачи. Поэтому чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Изучайте (читайте) листинги, старайтесь понять, как работают программы. Не бойтесь экспериментировать — совершенствуйте программы, вносите в них изменения.



ЧАСТЬ 1

Примеры и задачи

БАЗОВЫЕ КОМПОНЕНТЫ

В этом разделе приведены примеры, демонстрирующие назначение и технологию работы с базовыми компонентами.

Общие замечания

- Процесс создания программы состоит из двух шагов: создание формы программы (диалоговое окно) и функций обработки *событий*. Форма *приложения* (прикладная программа, работающая в Windows) создается путем добавления в форму *компонентов* и последующей их настройки.
- В форме практически любого приложения есть компоненты, обеспечивающие интерфейс (взаимодействие) между программой и пользователем. Такие компоненты называют базовыми. К базовым компонентам можно отнести:
 - Label — поле вывода текста;
 - TextBox — поле редактирования текста;
 - Button — командная кнопка;
 - CheckBox — независимая кнопка выбора;
 - RadioButton — зависимая кнопка выбора;
 - ListBox — список выбора;
 - ComboBox — комбинированный список выбора;
 - MainMenu — главное меню программы;
 - ContextMenu — контекстное меню.

- ❑ Вид компонента, его размер и поведение определяют значения *свойств* (характеристик) компонента (описание свойств базовых компонентов можно найти в справочнике — во второй части книги).
- ❑ Основную работу в программе выполняют функции обработки *событий* (описание основных событий можно найти в справочнике — во второй части книги).
- ❑ Исходную информацию программа может получить, например, из полей редактирования (компонент `Edit`), списка выбора (компонент `ListBox`), комбинированного списка (компонент `ComboBox`). Для ввода значений логического типа можно использовать компоненты `CheckBox` и `RadioButton`.
- ❑ Результат программа может вывести в поле вывода текста (компонент `Label`), в поле редактирования текста (компонент `TextBox`), в окно сообщения (функция `MessageBox.Show()`).
- ❑ Для преобразования текста (например, находящегося в поле редактирования) в целое число нужно использовать функцию `Convert.ToInt16()`, для преобразования в дробное — `Convert.ToDouble()`. Для преобразования численного значения переменной в строку нужно использовать функцию `ToString()`.

Конвертер

Программа **Конвертер** пересчитывает цену из долларов в рубли. Демонстрирует использование компонентов `TextBox` и `Label` для ввода и отображения числовых данных. Программа спроектирована таким образом, что пользователь может ввести в поля редактирования только правильные данные (числа). Форма программы приведена на рис. 1.1.

```
// нажатие клавиши в поле Цена
private void textBox1_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar) &&
        ! (Char.IsControl(e.KeyChar))) {
        if (!(e.KeyChar.ToString() == "," &&
            (textBox1.Text.IndexOf(",") == -1)))
```



```
        e.Handled = true;
    }
}

// нажатие клавиши в поле Курс
private void textBox2_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar) &&
        !(Char.IsControl(e.KeyChar))) {
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox2.Text.IndexOf(",") == -1))
            e.Handled = true;
    }
}

// щелчок на кнопке Пересчет
private void button1_Click(object sender, System.EventArgs e)
{
    double k;    // курс ( отношение рубля к доллару )
    double usd; // цена в долларах
    double rub; // цена в рублях

    label4.Text = "";

    // конструкция try { ... } catch { ... }
    // позволяет отлавливать возникающие программные ошибки
    try{
        // исходные данные
        usd = System.Convert.ToDouble(textBox1.Text);
        k   = System.Convert.ToDouble(textBox2.Text);

        // пересчет цены из долларов в рубли
        rub = usd * k;

        // вывод результата
        label4.Text = usd.ToString("N") +
            " USD = " + rub.ToString("C");
    }
}
```

```

catch{
    if ((textBox1.Text == "") || (textBox2.Text == "")){
        MessageBox.Show("Ошибка исходных данных.\n" +
            "Необходимо ввести данные в оба поля.",
            "Конвертер", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    else
        MessageBox.Show("Ошибка исходных данных.\n" +
            "Неверный формат данных в одном из полей.",
            "Конвертер", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
}

// щелчок на кнопке Завершить
private void button2_Click(object sender, System.EventArgs e)
{
    this.Close();
}

```

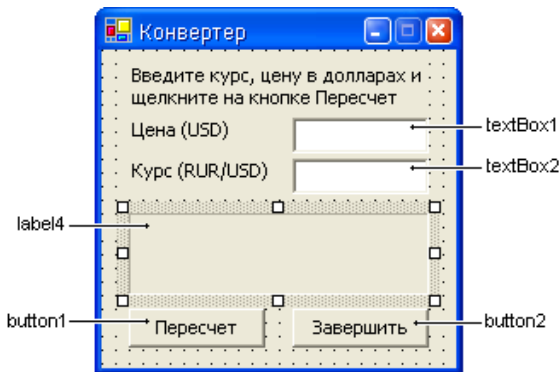


Рис. 1.1. Форма программы Конвертер

Электроэнергия

Программа **Электроэнергия** показывает, как одна функция может обрабатывать события разных, но однотипных компонентов. В данном примере — это поля ввода цены за кВт электроэнергии

(предыдущее и текущее показания счетчика). Обработываемое событие — это событие отпускания клавиши `KeyUp`. Функция обработки события `KeyUp` создается для поля `textBox1`, а затем назначается как функция обработки событий для полей `textBox2` и `textBox3`. Форма программы приведена на рис. 1.2.

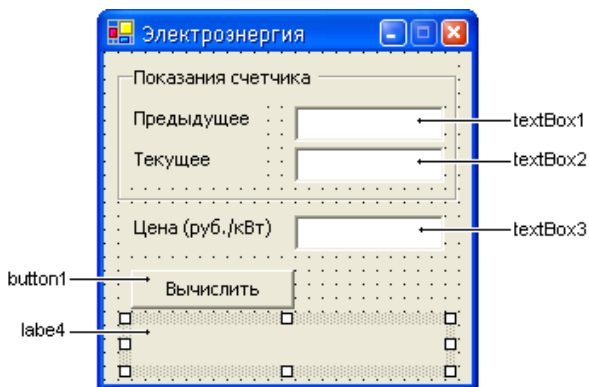


Рис. 1.2. Форма программы Электроэнергия

```
// загрузка формы
private void WinForm_Load(object sender, System.EventArgs e)
{
    // блокируем кнопку Вычислить. Она становится доступной
    // только тогда, когда введены данные во все поля
    button1.Enabled = false;
}

// нажатие клавиши в поле Предыдущее показание счетчика
private void textBox1_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)
{
    // контроль правильности вводимых данных
    if (!Char.IsDigit(e.KeyChar) &&
        !(Char.IsControl(e.KeyChar))) {
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox1.Text.IndexOf(",") == -1))
            e.Handled = true;
    }
}
```

```
// если нажата клавиша <Enter> - фокусируемся
// на поле ввода Текущего показания счетчика
if (e.KeyChar.Equals((char)13))
    textBox2.Focus();
}

// нажатие клавиши в поле Текущее показание счетчика
private void textBox2_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)
{
    // контроль правильности вводимых данных
    if (!Char.IsDigit(e.KeyChar) &&
        !(Char.IsControl(e.KeyChar))) {
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox2.Text.IndexOf(",") == -1))
            e.Handled = true;
    }
    // если нажата клавиша <Enter> - фокусируемся
    // на поле ввода Цены за кВт электроэнергии
    if (e.KeyChar.Equals((char)13))
        textBox3.Focus();
}

// нажатие клавиши в поле Цена (руб./кВт)
private void textBox3_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)
{
    // контроль правильности вводимых данных
    if (!Char.IsDigit(e.KeyChar) &&
        !(Char.IsControl(e.KeyChar))) {
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox3.Text.IndexOf(",") == -1))
            e.Handled = true;
    }
    // если нажата клавиша <Enter> -
    // фокусируемся на кнопке Вычислить
    if (e.KeyChar.Equals((char)13))
        button1.Focus();
}
```

```
// обработка события отпущения клавиши для полей Цена,  
// Предыдущее показание счетчика, Текущее показание счетчика  
private void textBox1_KeyUp(object sender,  
    System.Windows.Forms.KeyEventArgs e)  
{  
    // контроль доступности кнопки Вычислить  
    if ((textBox1.Text.Length > 0) &&  
        (textBox2.Text.Length > 0) &&  
        (textBox3.Text.Length > 0))  
        button1.Enabled = true;  
    else button1.Enabled = false;  
}  
  
// щелчок на кнопке Вычислить  
private void button1_Click(object sender, System.EventArgs e)  
{  
    float curr;           // текущее показание счетчика  
    float prev;          // предыдущее показание счетчика  
    float traf;          // цена за кВт  
    float price;        // сумма к оплате  
  
    label4.Text = "";  
  
    try{  
        // исходные данные  
        prev = Convert.ToSingle(textBox1.Text);  
        curr = Convert.ToSingle(textBox2.Text);  
        traf = Convert.ToSingle(textBox3.Text);  
  
        if(curr >= prev){  
            // вычисляем сумму к оплате  
            price = (curr - prev) * traf;  
  
            // вывод результата  
            label4.Text = "Сумма к оплате: " +  
                price.ToString("C");  
        }  
        else{  
            MessageBox.Show("Ошибка исходных данных.\n" +  
                "Текущее значение показания счетчика\n" +  
                "меньше предыдущего.", "Электроэнергия",
```

```

        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
catch (Exception exc) {
    MessageBox.Show("Ошибка исходных данных.\n" +
        "Исходные данные имеют неверный формат.\n" +
        exc.Message, "Электроэнергия",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
}
}

```

Автозаправка

Программа **Автозаправка** вычисляет количество литров бензина, которое можно купить на заданную сумму. Демонстрирует работу компонента `ComboBox`. Форма программы приведена на рис. 1.3.

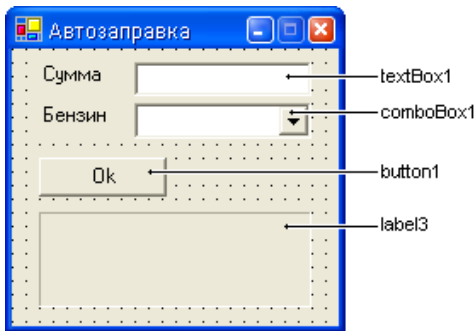


Рис. 1.3. Форма программы **Автозаправка**

```

// загрузка формы
private void WinForm_Load(object sender, EventArgs e)
{
    // установим стиль списка Бензин, DropDownList
    comboBox1.DropDownStyle =
        System.Windows.Forms.ComboBoxStyle.DropDownList;
}

```

```
// добавим элементы в список Бензин
comboBox1.Items.Add("92");
comboBox1.Items.Add("95");
comboBox1.Items.Add("98");
comboBox1.Items.Add("ДТ");

// кнопка Ok недоступна
button1.Enabled = false;
}

// нажатие клавиши в поле Сумма
private void textBox1_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)
{
    // контроль правильности вводимых данных
    if (!Char.IsDigit(e.KeyChar) &&
        !Char.IsControl(e.KeyChar)) {
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox1.Text.IndexOf(",") == -1))
            e.Handled = true;
    }
}

// содержимое поля Сумма изменилось
private void textBox1_TextChanged(object sender,
    System.EventArgs e)
{
    if ((textBox1.Text != ",") && (textBox1.TextLength > 0) &&
        (comboBox1.SelectedIndex != -1))
        button1.Enabled = true;
    else
        if (!button1.Enabled) button1.Enabled = false;
}

// изменился индекс выбранного типа топлива
private void comboBox1_SelectedIndexChanged(object sender,
    System.EventArgs e)
{
    if ((textBox1.Text != ",") && (textBox1.TextLength > 0))
        button1.Enabled = true;
}
}
```

```

// щелчок на кнопке Ok
private void button1_Click(object sender, System.EventArgs e)
{
    double cl = 17.12F; // цена за литр
                        // (указанное значение
                        // инициализирует переменную)
    double lt;        // количество литров
    double cash;     // наличные
    double ch;       // сдача

    // цена за литр для выбранного типа топлива
    switch (comboBox1.SelectedIndex)
    {
        case 0: cl = 17.12F; break;
        case 1: cl = 19.45F; break;
        case 2: cl = 20.25F; break;
        case 3: cl = 17.00F; break;
    }

    // количество литров считается с точностью до 0,1 л
    cash = Convert.ToSingle(textBox1.Text);
    lt = (double)Decimal.Truncate(
        (Decimal)(cash * 10 / cl)) / 10;
    ch = cash - lt * cl;

    label3.Text = "Литров: " + lt.ToString("N") +
        "\nСумма: " + cash.ToString("C") +
        "\nСдача: " + ch.ToString("C") +
        "\nЦена за литр: " + cl.ToString("C");
}

```

Кафе

Программа **Кафе** демонстрирует использование компонента `CheckBox`. Форма программы приведена на рис. 1.4.

```

// переменная summ объявляется внутри класса WinForm
private double summ; // стоимость заказа

// загрузка формы
private void WinForm_Load(object sender, System.EventArgs e)

```



```
{
    // сделать недоступным переключатель Соус
    checkBox3.Enabled = false;
}

// щелчок на переключателе Сэндвич
private void checkBox1_CheckedChanged(object sender,
System.EventArgs e)
{
    if (checkBox1.Checked) summ += 54.00;
    else summ -= 54.00;

    label1.Refresh();
}

// щелчок на переключателе Картошка
private void checkBox2_CheckedChanged(object sender,
System.EventArgs e)
{
    if (checkBox2.Checked) {
        summ += 24.50;

        // сделать доступным переключатель Соус
        if (!checkBox3.Enabled) checkBox3.Enabled = true;
    }
    else{
        summ -= 24.50;

        // сбросить переключатель Соус
        if (checkBox3.Checked) checkBox3.Checked = false;
        // сделать его недоступным
        checkBox3.Enabled = false;
    }

    label1.Refresh();
}

// щелчок на переключателе Соус
private void checkBox3_CheckedChanged(object sender,
System.EventArgs e)
```

```
{
    if (checkBox3.Checked) summ += 10.50;
    else summ -= 10.50;

    label1.Refresh();
}

// щелчок на переключателе Coca-Cola
private void checkBox4_CheckedChanged(object sender,
    System.EventArgs e)
{
    if (checkBox4.Checked) summ += 18.00;
    else summ -= 18.00;

    label1.Refresh();
}

// обработка события Paint компонента label1
private void label1_Paint(object sender,
    System.Windows.Forms.PaintEventArgs e)
{
    // вывести в поле компонента сумму заказа
    label1.Text = summ.ToString("C");
}

// щелчок на кнопке Ok
private void button1_Click(object sender, System.EventArgs e)
{
    if (checkBox1.Checked && checkBox2.Checked &&
        checkBox3.Checked && checkBox4.Checked) {
        // пользователь заказал полный набор,
        // предоставляется скидка 10%
        MessageBox.Show("Вам предоставляется скидка 10%\n" +
            "Стоимость заказа: " + (summ*0.9).ToString("C"),
            "Кафе");
    }
    else {
        if (checkBox1.Checked || checkBox2.Checked ||
            checkBox3.Checked || checkBox4.Checked)
```

```

        MessageBox.Show("Стоимость заказа: " +
            summ.ToString("C"), "Кафе");
    else
        MessageBox.Show("Вы ничего не заказали!",
            "Кафе");
    }
}

```

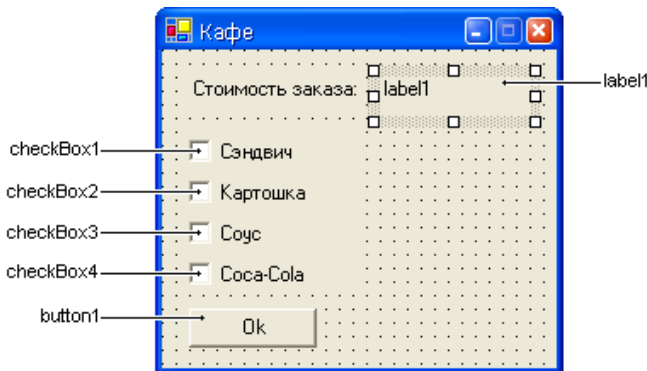


Рис. 1.4. Форма программы **Кафе**

Жалюзи

Программа **Жалюзи** демонстрирует использование компонента `RadioButton`. Форма программы приведена на рис. 1.5.

```

// загрузка формы
private void WinForm_Load(object sender, System.EventArgs e)
{
    // делаем недоступной кнопку Ok
    button1.Enabled = false;

    // материал по умолчанию - алюминий
    radioButton1.Checked = true;
}

// нажатие клавиши в поле Ширина
private void textBox1_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)

```

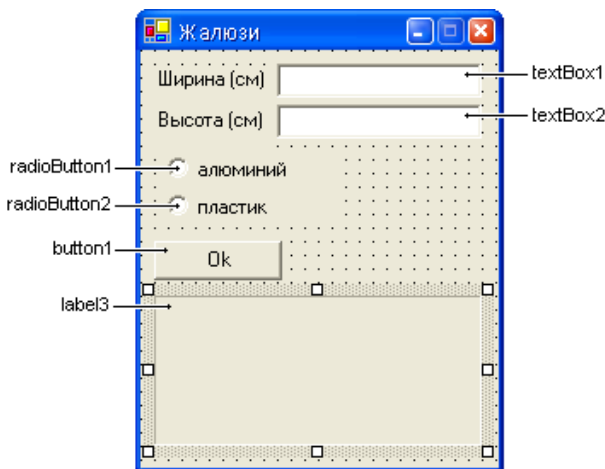


Рис. 1.5. Форма программы Жалюзи

```

{
    // контроль правильности вводимых данных
    if (!Char.IsDigit(e.KeyChar) &&
        !(Char.IsControl(e.KeyChar)))
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox1.Text.IndexOf(",") == -1))
            e.Handled = true;
}

// нажатие клавиши в поле Высота
private void textBox2_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)
{
    // контроль правильности вводимых данных
    if (!Char.IsDigit(e.KeyChar) &&
        !(Char.IsControl(e.KeyChar)))
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox2.Text.IndexOf(",") == -1))
            e.Handled = true;
}

// изменение содержимого полей Ширина и Высота;
// процедура обработки создается для поля Ширина,

```

```
// после чего назначается как процедура обработки
// и для поля Высота
private void textBox1_TextChanged(object sender,
    System.EventArgs e)
{
    // проверка, нужно ли блокировать кнопку Ok
    if ((textBox1.TextLength == 0) ||
        (textBox2.TextLength == 0) ||
        (textBox1.Text == ",") ||
        (textBox2.Text == ","))
        button1.Enabled = false;
    else button1.Enabled = true;
}

// щелчок на кнопке Ok
private void button1_Click(object sender, System.EventArgs e)
{
    Single w, h, s, // ширина, высота и площадь
           c,       // цена за 1 кв.м.
           cst;     // стоимость

    w = Convert.ToSingle(textBox1.Text);
    h = Convert.ToSingle(textBox2.Text);
    s = w * h / 10000;

    if (radioButton1.Checked)
        c = 3600; // выбран переключатель "алюминий"
    else
        c = 1800; // выбран переключатель "пластик"

    cst = s * c;

    if (radioButton1.Checked)
        label3.Text = "Размер: " + w.ToString("N") +
            " x " + h.ToString("N") + " см\n" +
            "Материал: " + radioButton1.Text +
            "\nСтоимость: " + cst.ToString("C");
    else
        label3.Text = "Размер: " + w.ToString("N") +
```

```

" x " + h.ToString("N") + " см\n" +
"Материал: " + radioButton2.Text +
"\nСтоимость: " + cst.ToString("C");
}

```

Стеклопакет

Программа **Стеклопакет** позволяет вычислить стоимость окна (стеклопакета). Демонстрирует работу с компонентами `TextBox`, `RadioButton`, `CheckBox`. Форма программы приведена на рис. 1.6.

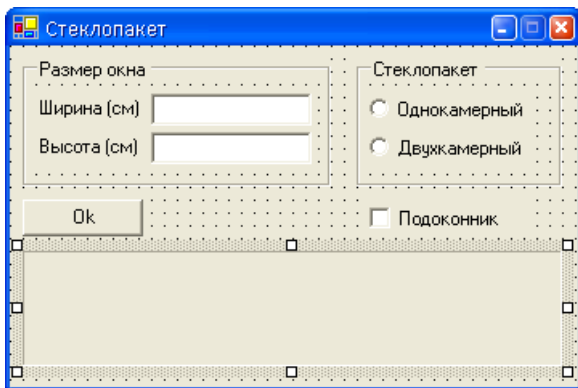


Рис. 1.6. Форма программы **Стеклопакет**

```

// загрузка формы
private void WinForm_Load(object sender, System.EventArgs e)
{
    // делаем недоступной кнопку Ok
    button1.Enabled = false;

    // по умолчанию выбранный
    // тип стеклопакета - однокамерный
    radioButton1.Checked = true;
}

// нажатие клавиши в поле Ширина
private void textBox1_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)

```

```
{
    // контроль правильности вводимых данных
    if (!Char.IsDigit(e.KeyChar) &&
        !(Char.IsControl(e.KeyChar)))
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox1.Text.IndexOf(",") == -1))
            e.Handled = true;
}

// нажатие клавиши в поле Высота
private void textBox2_KeyPress(object sender,
    System.Windows.Forms.KeyPressEventArgs e)
{
    // контроль правильности вводимых данных
    if (!Char.IsDigit(e.KeyChar) &&
        !(Char.IsControl(e.KeyChar)))
        if (!(e.KeyChar.ToString() == ",") &&
            (textBox2.Text.IndexOf(",") == -1))
            e.Handled = true;
}

// изменение типа стеклопакета, Однокамерный/Двухкамерный,
// установка/сброс флажка Подоконник;
// процедура обработки события CheckedChanged создается
// для компонента radioButton1, после чего назначается
// как процедура обработки этого же события
// и для компонентов radioButton2 и checkBox1
private void radioButton1_CheckedChanged(object sender,
    System.EventArgs e)
{
    if (label1.Text != string.Empty)
        label1.Text = string.Empty;
}

// изменение содержимого полей Ширина и Высота
private void textBox1_TextChanged(object sender,
    System.EventArgs e)
{
    // исходные данные изменились, очистим компонент
    // вывода результирующих данных label3 от данных
    // предыдущего расчета, если он производился
```

```
if (label1.Text != string.Empty)
    label1.Text = string.Empty;

// проверка, нужно ли блокировать кнопку Ok
if ((textBox1.TextLength == 0) ||
    (textBox2.TextLength == 0) ||
    (textBox1.Text == ",") ||
    (textBox2.Text == ","))
    button1.Enabled = false;
else button1.Enabled = true;
}

// щелчок на кнопке Ok
private void button1_Click(object sender, System.EventArgs e)
{
    Single w, h, s, // ширина, высота и площадь
           c,       // цена за 1 кв.м.
           cst;     // стоимость

    w = Convert.ToSingle(textBox1.Text);
    h = Convert.ToSingle(textBox2.Text);
    s = w * h / 10000;

    if (radioButton1.Checked)
        c = 5000; // однокамерный стеклопакет
    else
        c = 6000; // двухкамерный стеклопакет

    cst = s * c;

    // если установлен флажок Подоконник
    if (checkBox1.Checked) cst += 20*w;

    if (radioButton1.Checked)
        label1.Text = "Размер окна: " + w.ToString("N") +
            " x " + h.ToString("N") + " см\n" +
            "Стеклопакет: " + radioButton1.Text +
            "\nСтоимость: " + cst.ToString("C");
}
```



```

else
    labell1.Text = "Размер окна: " + w.ToString("N") +
        " x " + h.ToString("N") + " см\n" +
        "Стеклопакет: " + radioButton2.Text +
        "\nСтоимость: " + cst.ToString("C");
}

```

Калькулятор

Программа **Калькулятор** демонстрирует создание компонентов в коде программы. Кнопки калькулятора — это элементы массива компонентов `Button`. Создание и настройка кнопок осуществляется в конструкторе формы, там же назначаются процедуры обработки события `Click`. Форма и окно программы приведены на рис. 1.7.

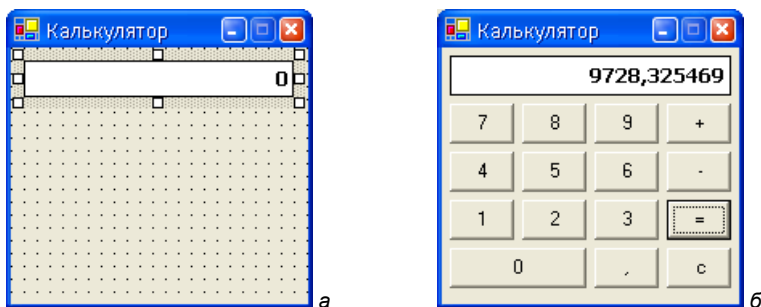


Рис. 1.7. Программа **Калькулятор** (а — форма, б — окно)

```

private const int
    bw = 40, bh = 25, // ширина и высота кнопок
    bdx = 5, bdy = 5; // расстояние между кнопками

// массив цифровых кнопок 0..9 и запятая
private Button[] d = new Button[11];

// массив операционных кнопок: +, -, =, C
private Button[] op = new Button[4];

```

```
private double ac; // аккумулятор
private string co; // текущая операция
private Boolean fd; // fd == true - ждем первую цифру числа,
// например, после того как была нажата
// кнопка +;
// fd == false - ждем ввода следующей
// цифры или нажатия кнопки операции;

// конструктор формы
public WinForm()
{
    InitializeComponent();

    int x, y; // координаты размещения кнопок

    // ширина и высота рабочей области формы
    this.ClientSize =
        new System.Drawing.Size(
            4*bw + 5*bdx, 5*bh + 7*bdy);

    // параметры цифрового индикатора
    labell.SetBounds(bdx, bdy, 4*bw + 3*bdx, bh);
    labell.Text = "0";

    // создание цифровых кнопок
    for(int i = 0; i <= 10; i++) {
        d[i] = new Button();

        // цифровые кнопки 0 - 9
        if (i < 10) {
            d[i].Name = "Button" + Convert.ToString(i);
            d[i].Text = i.ToString();

            if (i != 0) {
                x = ((i-1) % 3)*bw +
                    (((i-1) % 3) + 1)*bdx;
                y = ((int)((9-i)/3) + 1)*bh +
                    ((int)((9-i)/3) + 2)*bdy;
                d[i].SetBounds(x, y, bw, bh);
            }
        }
    }
}
```

```
        else // кнопка 0
            d[i].SetBounds(
                bdx, 4*bh + 5*bdy,
                2*bw + bdx, bh);
    }
    // кнопка Знак запятой
    else {
        d[i].Name = "ButtonComma";
        d[i].Text = ",";
        d[i].SetBounds(
            2*bw + 3*bdx, 4*bh + 5*bdy,
            bw, bh);
    }

    // назначение процедуры обработки
    // события нажатия кнопки
    this.d[i].Click += new
        System.EventHandler(this.ButtonN_Click);

    // добавление сформированной кнопки на форму
    this.Controls.Add(this.d[i]);
}

// создание операционных кнопок
for(int i = 0; i <= 3; i++) {
    op[i] = new Button();

    // кнопка +
    if (i == 0) {
        op[i].Name = "ButtonPlus";
        op[i].Text = "+";
    }

    // кнопка -
    if (i == 1) {
        op[i].Name = "ButtonMinus";
        op[i].Text = "-";
    }
}
```

```
// кнопка =
if (i == 2) {
    op[i].Name = "ButtonResult";
    op[i].Text = "=";
}

// кнопка с
if (i == 3) {
    op[i].Name = "ButtonClear";
    op[i].Text = "c";
}

op[i].SetBounds(
    3*bw + 4*bdx, (i + 1)*bh + (i + 2)*bdy,
    bw, bh);

// назначение процедуры обработки
// события нажатия кнопки
this.op[i].Click += new
    System.EventHandler(this.ButtonOp_Click);

// добавление сформированной кнопки на форму
this.Controls.Add(this.op[i]);
}

// ждем первую цифру
fd = true;
co = "ButtonResult";
}

// нажатие цифровых кнопок или запятой
private void ButtonN_Click(object sender, System.EventArgs e)
{
    // нажатая кнопка
    Button btn_c = (Button)sender;

    if (btn_c.Name != "ButtonComma") {
        // цифры
        if (btn_c.Name != "Button0") {
```

```
// кнопки 1..9
if (fd) {
    labell.Text = btn_c.Text;
    fd = false;
}
else
    labell.Text += btn_c.Text;
}
else {
    // кнопка 0
    if (fd) labell.Text = btn_c.Text;
    if (labell.Text != "0")
        labell.Text += btn_c.Text;
}
}
else
    // запятая
    if (fd) {
        labell.Text = "0,";
        fd = false;
    }
    else
        if (labell.Text.IndexOf(",") == -1)
            labell.Text += btn_c.Text;
}

// нажатие операционных кнопок
private void ButtonOp_Click(object sender, System.EventArgs e)
{
    // нажатая кнопка
    Button btn_c = (Button)sender;

    // число на индикаторе
    double ind_n;

    if (btn_c.Name != "ButtonClear"){
        // кнопки +, -, =
        ind_n = Convert.ToDouble(labell.Text);
```

```
// выполняем предыдущую операцию
if (fd == false) {
    if (co.Equals("ButtonPlus"))    ac += ind_n;
    if (co.Equals("ButtonMinus"))  ac -= ind_n;
    if (co.Equals("ButtonResult")) ac  = ind_n;
}

if (btn_c.Name == "ButtonPlus")
    co = "ButtonPlus";
if (btn_c.Name == "ButtonMinus")
    co = "ButtonMinus";
if (btn_c.Name == "ButtonResult")
    co = "ButtonResult";

    labell1.Text = ac.ToString();
}
else {
    // кнопка ОЧИСТИТЬ (C)
    ac = 0;
    labell1.Text = "0";
    co = "ButtonResult";
}

fd = true;
}
```

Просмотр иллюстраций

Программа просмотра иллюстраций ImageViewer демонстрирует использование компонентов `ListBox`, `FolderBrowserDialog` и `PictureBox`. Выбор каталога просматриваемых иллюстраций осуществляется в стандартном окне выбора Windows. Отображение иллюстраций осуществляется компонентом `PictureBox`. Окно программы приведено на рис. 1.8.

```
// для получения прямого доступа к типам DirectoryInfo
// и FileInfo необходимо в директиву using программы
// добавить пространство имен System.IO
```