

Никита Культин

Microsoft®

Visual C#

в задачах и примерах

Санкт-Петербург

«БХВ-Петербург»

2009

УДК 681.3.068
ББК 32.973.26-018.1
К90

Культин Н. Б.

К90 Microsoft Visual C# в задачах и примерах. — СПб.:
БХВ-Петербург, 2009. — 320 с.: ил. + CD-ROM

ISBN 978-5-9775-0410-2

Книга представляет собой сборник программ и задач для самостоятельного решения. Примеры различной степени сложности – от простейших до приложений работы с графикой и базами данных, в том числе Microsoft Access и Microsoft SQL Server Compact Edition – демонстрируют назначение базовых компонентов, раскрывают тонкости разработки приложений Windows Forms в Microsoft Visual C#. Уделено внимание использованию технологии LINQ. Справочник, входящий в книгу, содержит описание базовых компонентов, событий, исключений и наиболее часто используемых функций.

На прилагаемом компакт-диске находятся проекты, представленные в книге.

Для начинающих программистов

УДК 681.3.068
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>Леонид Кочин</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</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 31.03.09.

Формат 60×90^{1/16}. Печать офсетная. Усл. печ. л. 20.

Тираж 2500 экз. Заказ №

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

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.003650.04.08 от 14.04.2008 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

Оглавление

Предисловие.....	1
Часть 1. Примеры и задачи.....	5
Базовые компоненты.....	5
Общие замечания.....	5
Мили-километры.....	6
Фунты-килограммы.....	10
Конвертор.....	14
Фото.....	17
Комплектация автомобиля.....	20
Жалюзи.....	23
Калькулятор.....	27
Просмотр иллюстраций.....	33
Просмотр иллюстраций-2.....	39
Секундомер.....	46
Таймер.....	50
Параметры пользователя.....	54
Справочная информация.....	57
Файлы.....	61
Курс.....	61
Котировки.....	65
Редактор текста.....	68
<i>TaskDialog</i>	78
Графика.....	86
Общие замечания.....	87
Вывод текста.....	87
Диаграмма.....	90
График.....	95
Круговая диаграмма.....	101

Кисти.....	108
Бегущая строка	111
Полет.....	115
Базы данных.....	121
Общие замечания.....	121
Контакты	121
Контакты-2	129
Контакты-3	134
SQL Server Compact Edition	145
Игры и другие полезные программы	156
Парные картинки	156
Собери картинку.....	168
Сапер.....	176
Будильник.....	190
Экзаменатор	196
LINQ	209
Общие замечания.....	209
Поиск в массиве.....	209
Поиск в массиве-2	212
Обработка массива	215
Обработка массива записей.....	218
Работа с XML-документами.....	221
Отображение XML-документа	226
Экзаменатор-2.....	230
ЧАСТЬ 2. Краткий справочник	249
Форма	249
Компоненты	251
<i>Button</i>	251
<i>ComboBox</i>	253
<i>ContextMenuStrip</i>	255
<i>CheckBox</i>	256
<i>CheckedListBox</i>	258
<i>GroupBox</i>	259
<i>ImageList</i>	260
<i>Label</i>	260
<i>ListBox</i>	262

<i>MenuStrip</i>	263
<i>NotifyIcon</i>	264
<i>NumericUpDown</i>	265
<i>OpenFileDialog</i>	266
<i>Panel</i>	267
<i>PictureBox</i>	268
<i>RadioButton</i>	270
<i>ProgressBar</i>	272
<i>SaveFileDialog</i>	272
<i>TextBox</i>	274
<i>ToolTip</i>	276
<i>Timer</i>	276
Графика	277
Графические примитивы	277
Карандаш	279
Кисть	281
Типы данных	284
Целый тип	284
Вещественный тип	285
Символьный и строковый типы	285
Функции	285
Функции преобразования	285
Функции манипулирования строками	287
Функции манипулирования датами и временем	289
Функции манипулирования каталогами и файлами	291
Математические функции	294
События	296
Исключения	297
Задачи для самостоятельного решения	299
Приложение. Описание компакт-диска	303
Предметный указатель	305

Предисловие

В последнее время в общем объеме вновь создаваемого программного обеспечения различного назначения увеличивается доля .NET-приложений — программ, ориентированных на платформу Microsoft .NET. Это объясняется, прежде всего, новыми возможностями, которые предоставляет платформа прикладным программам, а также тем, что технология .NET поддерживается новейшими операционными системами.

Microsoft .NET — это технология, в основе которой лежит идея универсального программного кода, который может быть выполнен любым компьютером, вне зависимости от используемой операционной системы. Универсальность программного кода обеспечивается за счет предварительной (выполняемой на этапе разработки) компиляции исходной программы в *универсальный промежуточный код* (CIL-код, Common Intermediate Language), который во время загрузки транслируется в *выполняемый*. Преобразование промежуточного кода в выполняемый осуществляет JIT-компилятор (от Just In Time — в тот же момент, "на лету"), являющийся элементом виртуальной выполняющей системы (Virtual Execution System, VES). Работу .NET-приложений в операционной системе Windows обеспечивает Microsoft .NET Framework.

Чтобы понять, что такое .NET, и какие возможности она предоставляет программисту, необходимо опробовать ее в деле. Для этого нужно изучить среду и технологию разработки, назначение и возможности компонентов, их свойства и методы. И здесь хорошим подспорьем могут стать примеры, программы, разработанные другими программистами.

Среда разработки **Microsoft Visual C#** является инструментом разработки .NET-приложений для Windows. В ней интегрированы удобный дизайнер форм, специализированный редактор кода,

отладчик, мастер публикации и другие инструменты, необходимые программисту.

Книга, которую вы держите в руках, посвящена практике программирования в Microsoft Visual C#. В ней собраны разнообразные примеры, демонстрирующие назначение базовых компонентов, технологии работы с файлами, графикой, и базами данных. Следует обратить внимание, что большинство примеров не являются учебными в общепринятом смысле, это — вполне работоспособные, законченные программы.

Книга состоит из двух частей. Первая часть содержит примеры программ. Примеры представлены в виде краткого описания, диалоговых окон и хорошо документированных листингов.

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

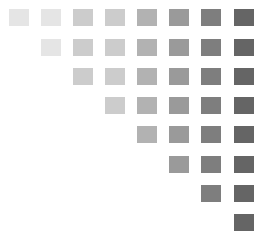
Научиться программировать можно, только программируя, решая конкретные задачи. Поэтому, чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Изучайте листинги, старайтесь понять, как работают программы. Не бойтесь экспериментировать — совершенствуйте программы, вносите в них изменения. Чем больше вы сделаете самостоятельно, тем большему научитесь!

Часть 1



ПРИМЕРЫ

И ЗАДАЧИ



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: сначала создается форма, затем — функции обработки *событий*.
- Форма создается путем добавления необходимых компонентов и последующей их настройки.
- В форме практически любого приложения есть компоненты, обеспечивающие взаимодействие программы с пользователем. Такие компоненты называют базовыми.
- К базовым компонентам можно отнести:
 - Label — поле отображения текста;
 - TextBox — поле редактирования текста;
 - Button — командную кнопку;
 - CheckBox — флажок;

- `RadioButton` — селектор (радиокнопка);
 - `ListBox` — список выбора;
 - `ComboBox` — поле редактирования со списком выбора.
- ❑ Вид и поведение компонента определяют значения его *свойств* (характеристик) (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 из полей редактирования (компонент `TextBox`), списка (компонент `ListBox`), комбинированного списка (компонент `ComboBox`).
- ❑ Ввести значения логического типа можно с помощью компонентов `CheckBox` и `RadioButton`.
- ❑ Результат программа может вывести в поле отображения текста (компонент `Label`), в поле редактирования или в окно сообщения (метод `MessageBox.Show()`).
- ❑ Для преобразования текста, находящегося, например, в поле редактирования, в целое число нужно использовать функцию `Convert.ToInt32()`, в дробное число — `Convert.ToDouble()`.
- ❑ Преобразовать численное значение в строку позволяет метод `ToString()`. В качестве параметра метода можно указать формат отображения: "c" — денежный с разделителями групп разрядов и обозначением валюты (`currency`); "N" — числовой с разделителями групп разрядов (`numeric`); "F" — числовой без разделителей групп разрядов (`fixed`).

Мили-километры

Программа **Мили-километры** (рис. 1.1, листинг 1.1) пересчитывает расстояние из миль в километры. Демонстрирует использование компонента `TextBox` для ввода данных и компонента `Label`

для отображения числовой информации. Программа спроектирована таким образом, что в поле редактирования можно ввести только дробное число. Значения свойств формы приведены в табл. 1.1.

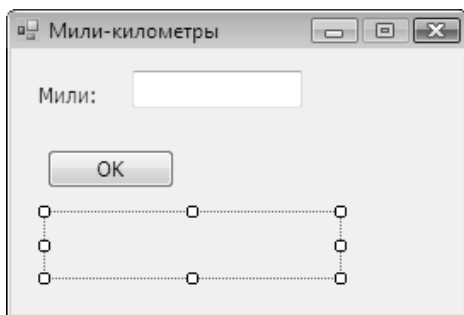


Рис. 1.1. Форма программы

Таблица 1.1. Значения свойств формы

Свойство	Значение	Комментарий
Text	Мили-километры	Текст заголовка
StartPosition	CenterScreen	Начальное положение окна — в центре экрана
FormBorderStyle	FixedSingle	Тонкая граница окна. Пользователь не сможет изменить размер окна путем перемещения его границы
MaximizeBox	False	Кнопка Развернуть окно недоступна. Пользователь не сможет развернуть окно программы на весь экран
Font	Tahoma; 9pt	Шрифт, наследуемый компонентами формы

Листинг 1.1. Модуль формы программы Мили-километры

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace m2k
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // нажатие клавиши в поле редактирования
            private void textBox1_KeyPress(object sender,
                KeyPressEventArgs e)
            {
                // Правильными символами считаются цифры,
                // запятая, <Enter> и <Backspace>.
                // Будем считать правильным символом
                // также точку, но заменим ее запятой.
                // Остальные символы запрещены.
                // Чтобы запрещенный символ не отображался
                // в поле редактирования, присвоим
                // значение true свойству Handled параметра e

                if ((e.KeyChar >= '0') && (e.KeyChar <= '9'))
                {
```

```
        // цифра
        return;
    }

    if (e.KeyChar == '.')
    {
        // точку заменим запятой
        e.KeyChar = ',';
    }

    if (e.KeyChar == ',')
    {
        if (textBox1.Text.IndexOf(',') != -1)
        {
            // запятая уже есть в поле редактирования
            e.Handled = true;
        }
        return;
    }

    if ( Char.IsControl (e.KeyChar) )
    {
        // <Enter>, <Backspace>, <Esc>
        if ( e.KeyChar == (char) Keys.Enter)
            // нажата клавиша <Enter>
            // установить курсор на кнопку ОК
            button1.Focus();
        return;
    }

    // остальные символы запрещены
    e.Handled = true;
}
```

```
// щелчок на кнопке ОК
private void button1_Click(object sender, EventArgs e)
{
    double mile; // расстояние в милях
    double km;   // расстояние в километрах

    // Если в поле редактирования нет данных,
    // то при попытке преобразовать пустую
    // строку в число возникает исключение.
    try
    {
        mile = Convert.ToDouble(textBox1.Text);

        km = mile * 1.609344;

        label2.Text = km.ToString("n")
            + " км.";
    }
    catch
    {
        // обработка исключения:
        // переместить курсор в поле редактирования
        textBox1.Focus();
    }
}
}
```

Фунты-килограммы

Программа **Фунты-килограммы** (рис. 1.2, листинг 1.2) пересчитывает вес из фунтов в килограммы. Показывает, как можно управлять доступностью командной кнопки (компонент `Button`) в зависимости от наличия данных в поле редактирования.

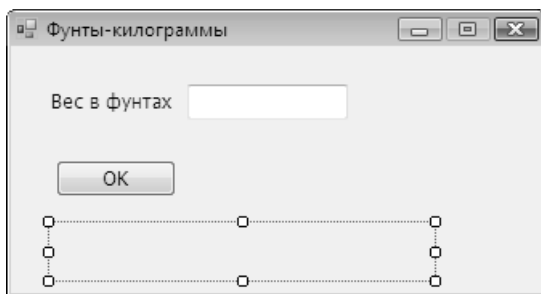


Рис. 1.2. Форма программы Фунты-килограммы

Листинг 1.2. Модуль формы программы Фунты-килограммы

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // сделать кнопку ОК недоступной
            button1.Enabled = false;
        }
    }
}
```

```
// нажатие клавиши в поле редактирования
private void textBox1_KeyPress(object sender,
                               KeyPressEventArgs e)
{
    if ((e.KeyChar >= '0') && (e.KeyChar <= '9'))
        return;

    if (e.KeyChar == '.') e.KeyChar = ',';

    if (e.KeyChar == ',')
    {
        // в поле редактирования не может
        // быть больше одной запятой и запятая
        // не может быть первым символом
        if ( (textBox1.Text.IndexOf(',') != -1) ||
            (textBox1.Text.Length == 0) )
        {
            e.Handled = true;
        }
        return;
    }

    if ( Char.IsControl (e.KeyChar) )
    {
        // <Enter>, <Backspace>, <Esc>
        if ( e.KeyChar == (char) Keys.Enter)
            // установить курсор на кнопку ОК
            button1.Focus();
        return;
    }

    // остальные символы запрещены
    e.Handled = true;
}
```



```
// текст в поле редактирования изменился
private void textBox1_TextChanged(object sender,
                                EventArgs e)
{
    label2.Text = ""; // очистить поле отображения
                       // результата расчета

    if (textBox1.Text.Length == 0)
        // в поле редактирования нет данных
        // сделать кнопку ОК недоступной
        button1.Enabled = false;
    else
        // сделать кнопку ОК доступной
        button1.Enabled = true;
}

// щелчок на кнопке ОК
private void button1_Click(object sender, EventArgs e)
{
    double funt; // вес в фунтах
    double kg;   // вес в килограммах

    funt = Convert.ToDouble(textBox1.Text);

    // 1 фунт = 409,5 грамма
    kg = funt * 0.4095;

    label2.Text = funt.ToString("N") + " ф. = "
                 + kg.ToString("N") + " кг.";
}
}
```

Конвертор

Программа **Конвертор** (рис. 1.3, листинг 1.3) демонстрирует обработку одной функцией событий от нескольких компонентов.

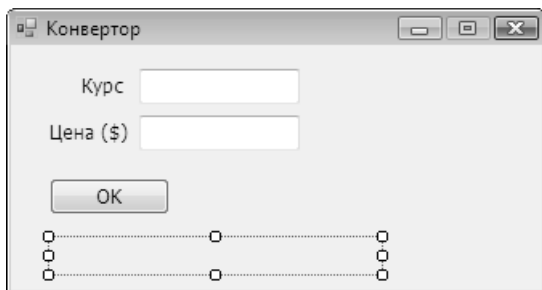


Рис. 1.3. Форма программы Конвертор

Листинг 1.3. Модуль формы программы *Конвертор*

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
```

```
        InitializeComponent();
    }

    // Обрабатывает нажатие клавиши в полях
    // редактирования Курс и Цена.
    // Сначала надо обычным образом создать функцию
    // обработки события KeyPress для компонента
    // textBox1, затем - указать ее в качестве
    // обработчика этого же события для компонента
    // textBox2
    private void textBox1_KeyPress(object sender,
                                   KeyPressEventArgs e)
    {
        if ((e.KeyChar >= '0') && (e.KeyChar <= '9'))
            return;

        if (e.KeyChar == '.') e.KeyChar = ',';

        if (e.KeyChar == ',')
        {
            if ((textBox1.Text.IndexOf(',') != -1) ||
                (textBox1.Text.Length == 0))
            {
                e.Handled = true;
            }
            return;
        }

        if (Char.IsControl(e.KeyChar))
        {
            if (e.KeyChar == (char)Keys.Enter)
            {
                if (sender.Equals(textBox1))
                    // клавиша <Enter> нажата в поле Курс
                    // переместить курсор в поле Цена
            }
        }
    }
}
```

```
        textBox2.Focus();
    }
    else
        // клавиша <Enter> нажата в поле Цена
        //
        button1.Focus();
    }
    return;
}

// остальные символы запрещены
e.Handled = true;
}

// изменился текст в поле редактирования
// textBox1 или textBox2
private void textBox1_TextChanged(object sender,
                                EventArgs e)
{
    label3.Text = "";
    if ((textBox1.Text.Length == 0) ||
        (textBox2.Text.Length == 0))
        // если какое-либо из полей не содержит
        // данных, то сделать недоступной кнопку ОК
        button1.Enabled = false;
    else
        button1.Enabled = true;
}

// щелчок на кнопке ОК
private void button1_Click(object sender,
                           EventArgs e)
{
    double usd; // цена в долларах
    double k;   // курс
    double rub; // цена в рублях
```

```
        usd = Convert.ToDouble(textBox1.Text);  
        k = Convert.ToDouble(textBox2.Text);  
  
        rub = usd * k;  
  
        label3.Text =  
            rub.ToString("C"); // финансовый формат  
    }  
}
```

Фото

Программа **Фото** (рис. 1.4, листинг 1.4) позволяет рассчитать стоимость печати фотографий. Демонстрирует использование компонента `RadioButton`.

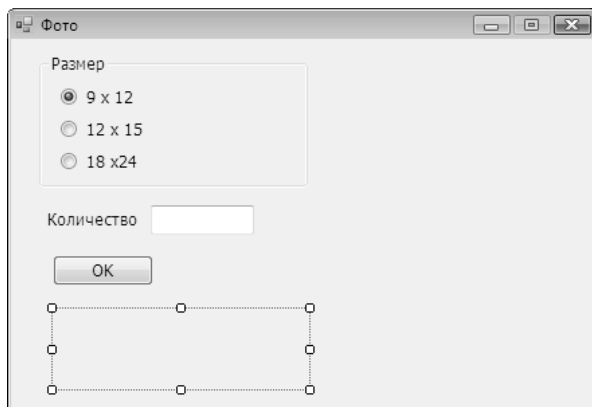


Рис. 1.4. Форма программы **Фото**

Листинг 1.4. Модуль формы программы **Фото**

```
using System;  
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // настройка компонентов
            radioButton1.Checked = true;
            button1.Enabled = false;
        }

        // щелчок на кнопке ОК
        private void button1_Click(object sender, EventArgs e)
        {
            double cena = 0 ; // цена
            int n;           // кол-во фотографий
            double sum;      // сумма

            if (radioButton1.Checked)
                cena = 8.50;
            if (radioButton2.Checked)
                cena = 10;
            if (radioButton3.Checked)
                cena = 15.5;
        }
    }
}
```

```
n = Convert.ToInt32(textBox1.Text);
sum = n * цена;

label2.Text = "Цена: " + цена.ToString("c") +
    "\nКоличество: " + n.ToString() + "шт.\n" +
    "Сумма заказа: " + sum.ToString("C");
}

// В поле Количество можно ввести только целое число
private void textBox1_KeyPress(object sender,
    KeyPressEvent e)
{
    if ((e.KeyChar >= '0') && (e.KeyChar <= '9'))
        return;

    if (Char.IsControl(e.KeyChar))
    {
        if (e.KeyChar == (char)Keys.Enter)
        {
            // нажата клавиша <Enter>
            button1.Focus();
        }
        return;
    }
    // остальные символы запрещены
    e.Handled = true;
}

private void textBox1_TextChanged(object sender,
    EventArgs e)
{
    if (textBox1.Text.Length == 0)
        button1.Enabled = false;
    else
        button1.Enabled = true;
}
```

```
        label2.Text = "";  
    }  
  
    // щелчок на radioButton  
    private void radioButton1_Click(object sender,  
                                     EventArgs e)  
    {  
        label2.Text = "";  
        // установить курсор в поле Количество  
        textBox1.Focus();  
    }  
}
```

Комплектация автомобиля

Программа **Комплектация** (рис. 1.5, листинг 1.5) позволяет рассчитать стоимость автомобиля в зависимости от выбранной комплектации. Демонстрирует использование компонента `CheckBox`. Отображение картинки обеспечивает компонент `PictureBox`.



Рис. 1.5. Форма программы **Комплектация**

Листинг 1.5. Модуль формы программы *Комплектация*

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // щелчок на кнопке ОК
            private void button1_Click(object sender, EventArgs e)
            {
                double sum;        // сумма
                double discount;    // скидка
                double total;       // общая сумма

                sum = 309000;
                discount = 0;

                if (checkBox1.Checked)
                {
                    // ABS
                    sum += 8390;
                }
            }
        }
    }
}
```

```
    if (checkBox2.Checked)
    {
        // противотуманные фары
        sum += 5990;
    }

    if (checkBox3.Checked)
    {
        // парктроник
        sum += 7590;
    }

    total = sum;

    string st;
    st = "Цена в выбранной комплектации: " +
        sum.ToString("C");

    if ((checkBox1.Checked) && (checkBox2.Checked) &&
        (checkBox3.Checked))
    {
        // скидка предоставляется, если
        // выбраны все опции
        discount = sum * 0.01;
        total = total - discount;
        st += "\nСкидка (1%): " +
            discount.ToString("C") +
            "\nИтого: " + total.ToString("C");
    }
    label3.Text = st;
}

// пользователь изменил состояние переключателя
// функция обрабатывает событие CheckedChanged
// компонентов checkBox1 - checkBox3
```

```
private void checkBox1_CheckedChanged(object sender,
                                     EventArgs e)
{
    label3.Text = "";
}
}
```

Жалюзи

Программа **Жалюзи** (рис. 1.6, листинг 1.6) демонстрирует использование компонента `ComboBox`, который служит для выбора материала (пластик, алюминий, соломка, текстиль).

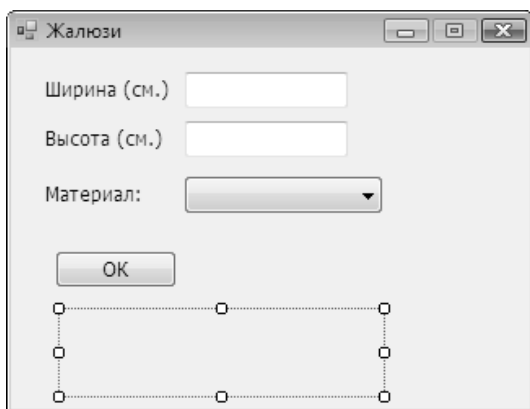


Рис. 1.6. Форма программы Жалюзи

Листинг 1.6. Модуль формы программы Жалюзи

```
using System;
using System.Collections.Generic;
```



```
{
    if ((e.KeyChar >= '0') && (e.KeyChar <= '9'))
        return;
    if (Char.IsControl(e.KeyChar))
    {
        if (e.KeyChar == (char)Keys.Enter)
        {
            if (sender.Equals(textBox1))
                // клавиша <Enter> нажата в поле Ширина
                // переместить курсор в поле Высота
                textBox2.Focus();
            else
                // клавиша <Enter> нажата в поле Высота
                // переместить фокус на comboBox1
                comboBox1.Focus();
        }
        return;
    }

    // остальные символы запрещены
    e.Handled = true;
}
}
```

```
private void textBox1_TextChanged(object sender,
                                   EventArgs e)
{
    if ((textBox1.Text.Length == 0) ||
        (textBox2.Text.Length == 0))

        button1.Enabled = false;
    else
        button1.Enabled = true;
}
```

```
        label4.Text = "";
    }

    // щелчок на кнопке ОК
    private void button1_Click(object sender, EventArgs e)
    {
        double w;
        double h;
        double cena = 0; // цена за 1 кв.м.
        double sum;

        w = Convert.ToDouble(textBox1.Text);
        h = Convert.ToDouble(textBox2.Text);

        switch (comboBox1.SelectedIndex)
        {
            case 0: cena = 50; break; // пластик
            case 1: cena = 100; break; // алюминий
            case 2: cena = 75; break; // бамбук
            case 3: cena = 70; break; // соломка
            case 4: cena = 60; break; // текстиль
        }

        sum = (w * h) / 10000 * cena;
        label4.Text =
            "Размер: " + w + "x" + h + " см.\n" +
            "Цена (р./м.кв.): " + cena.ToString("c") +
            "\nСумма: " + sum.ToString("c");
    }

    // в списке Материал пользователь
    // выбрал другой элемент
    private void comboBox1_SelectedIndexChanged
        (object sender, EventArgs e)
```