

Microsoft®

Visual C#

В задачах и примерах
2-е издание

Базовые компоненты

*Программирование графики
и баз данных*

Работа с LINQ

*Справочник по компонентам
и функциям*

**В ногу
со временем!**



Материалы
на www.bhv.ru

Никита Культин

Microsoft®

Visual C#

в задачах и примерах

**2-е издание,
исправленное**

Санкт-Петербург

«БХВ-Петербург»

2014

УДК 004.438 Visual C#
ББК 32.973.26-018.1
К90

Культин Н. Б.

К90 Microsoft® Visual C# в задачах и примерах. — 2-е изд.,
исправл. — СПб.: БХВ-Петербург, 2014. — 320 с.: ил.

ISBN 978-5-9775-3323-2

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 — 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 FTP-сервере издательства находятся коды примеров из книги.

Для начинающих программистов

УДК 004.438 Visual C#
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 30.04.14.

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

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3323-2

© Культин Н. Б., 2014

© Оформление, издательство "БХВ-Петербург", 2014

Оглавление

Предисловие.....	7
ЧАСТЬ 1. ПРИМЕРЫ И ЗАДАЧИ.....	9
Базовые компоненты	11
Общие замечания	11
Мили-километры	12
Фунты-килограммы.....	16
Конвертор.....	20
Фото	23
Комплектация автомобиля	26
Жалюзи.....	29
Калькулятор	33
Просмотр иллюстраций	39
Просмотр иллюстраций-2.....	45
Секундомер	52
Таймер	56
Параметры программы.....	60
Справочная информация.....	63
Файлы.....	67
Курс.....	67
Котировки	71
Редактор текста.....	74
<i>TaskDialog</i>	84
Графика.....	92
Общие замечания	93
Вывод текста.....	93
Диаграмма.....	96
График	101
Круговая диаграмма	107
Кисти	114

Бегущая строка	117
Полет	121
Базы данных	127
Общие замечания	127
Контакты	127
Контакты-2	135
Контакты-3	140
SQL Server Compact Edition	151
Игры и другие полезные программы	162
Парные картинки	162
Собери картинку	174
Сапер	182
Будильник	196
Экзаменатор	202
LINQ	215
Общие замечания	215
Поиск в массиве	215
Поиск в массиве-2	218
Обработка массива	221
Обработка массива записей	224
Работа с XML-документами	227
Отображение XML-документа	232
Экзаменатор-2	236
ЧАСТЬ 2. КРАТКИЙ СПРАВОЧНИК	253
Форма	255
Компоненты	257
<i>Button</i>	257
<i>ComboBox</i>	259
<i>ContextMenuStrip</i>	261
<i>CheckBox</i>	262
<i>CheckedListBox</i>	264
<i>GroupBox</i>	265
<i>ImageList</i>	266
<i>Label</i>	266
<i>ListBox</i>	268
<i>MenuStrip</i>	269
<i>NotifyIcon</i>	270

<i>NumericUpDown</i>	271
<i>OpenFileDialog</i>	272
<i>Panel</i>	273
<i>PictureBox</i>	274
<i>RadioButton</i>	276
<i>ProgressBar</i>	278
<i>SaveFileDialog</i>	278
<i>TextBox</i>	280
<i>ToolTip</i>	282
<i>Timer</i>	282
Графика.....	283
Графические примитивы.....	283
Карандаш.....	285
Кисть.....	287
Типы данных.....	290
Целый тип.....	290
Вещественный тип.....	291
Символьный и строковый типы.....	291
Функции.....	291
Функции преобразования.....	291
Функции манипулирования строками.....	293
Функции манипулирования датами и временем.....	295
Функции манипулирования каталогами и файлами.....	297
Математические функции.....	300
События.....	302
Исключения.....	303
Задачи для самостоятельного решения.....	305
Приложение. Описание электронного архива.....	309
Предметный указатель.....	311

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 компиляции исходной программы в *универсальный промежуточный код*, который во время загрузки транслируется в *выполняемый код*. 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 Microsoft Windows обеспечивает платформа Microsoft .NET Framework.

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

Среда программирования **Microsoft Visual C#** является инструментом разработки .NET-приложений для операционной системы Microsoft Windows. В ней интегрированы удобный конструктор

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 — работоспособные, законченные программы.

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

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

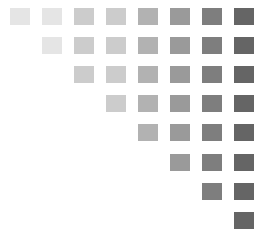
Научиться программировать можно только решая конкретные задачи. Поэтому, чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Изучайте листинги, старайтесь понять, как работают программы. Не бойтесь экспериментировать — совершенствуйте программы, вносите в них изменения. Чем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

Часть 1



ПРИМЕРЫ

И ЗАДАЧИ



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.

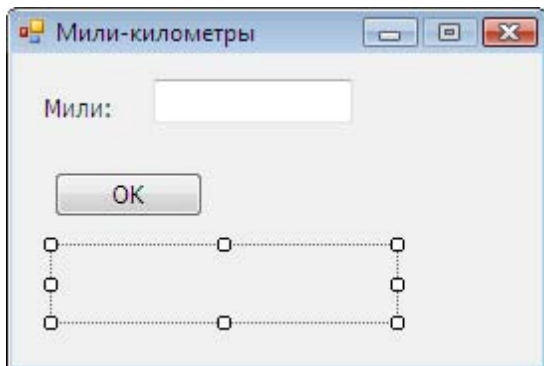


Рис. 1.1. Форма программы

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.

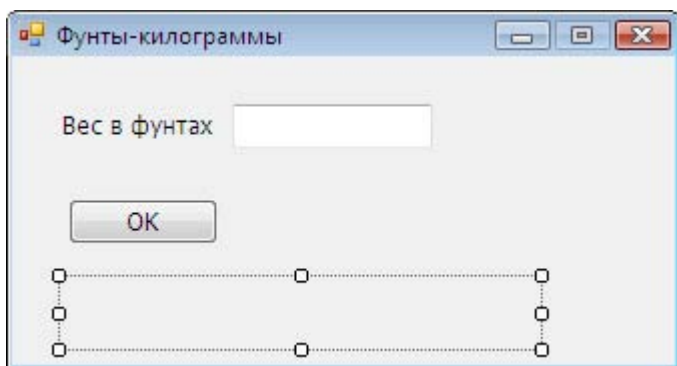


Рис. 1.2. Форма программы Фунты-килограммы

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.

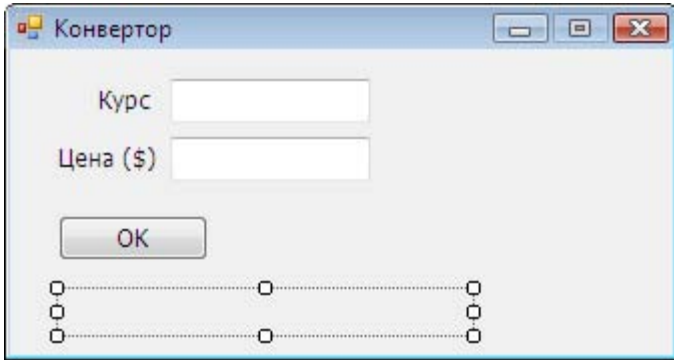


Рис. 1.3. Форма программы Конвертор

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`.

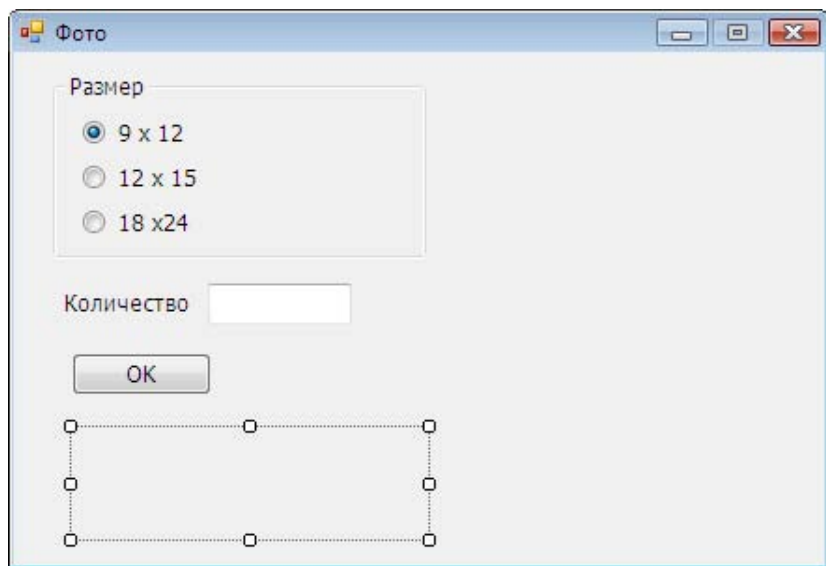


Рис. 1.4. Форма программы **Фото**

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,
    KeyPressEventArgs 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`.

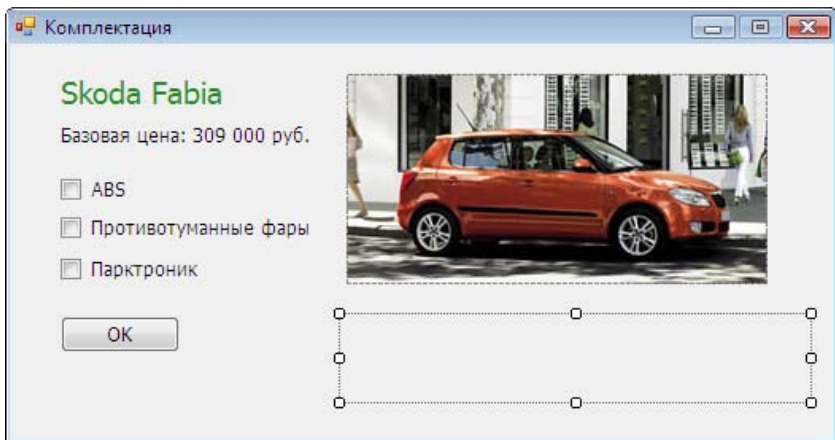


Рис. 1.5. Форма программы **Комплектация**

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).

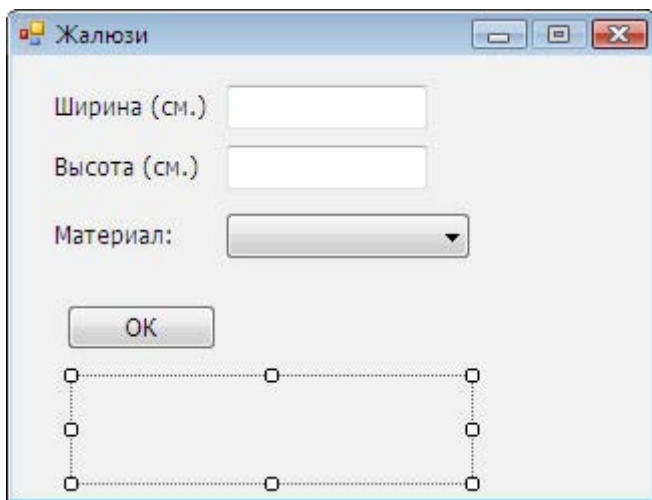


Рис. 1.6. Форма программы Жалюзи

Листинг 1.6. Модуль формы программы Жалюзи

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

