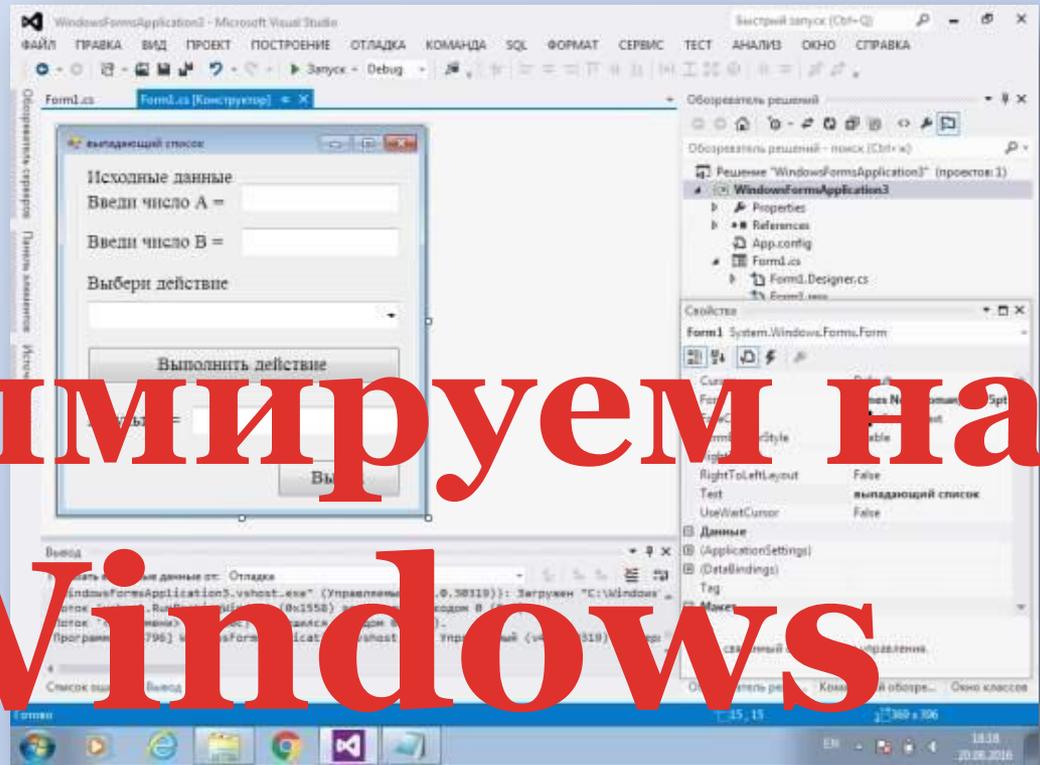


Версия для Microsoft Visual C#



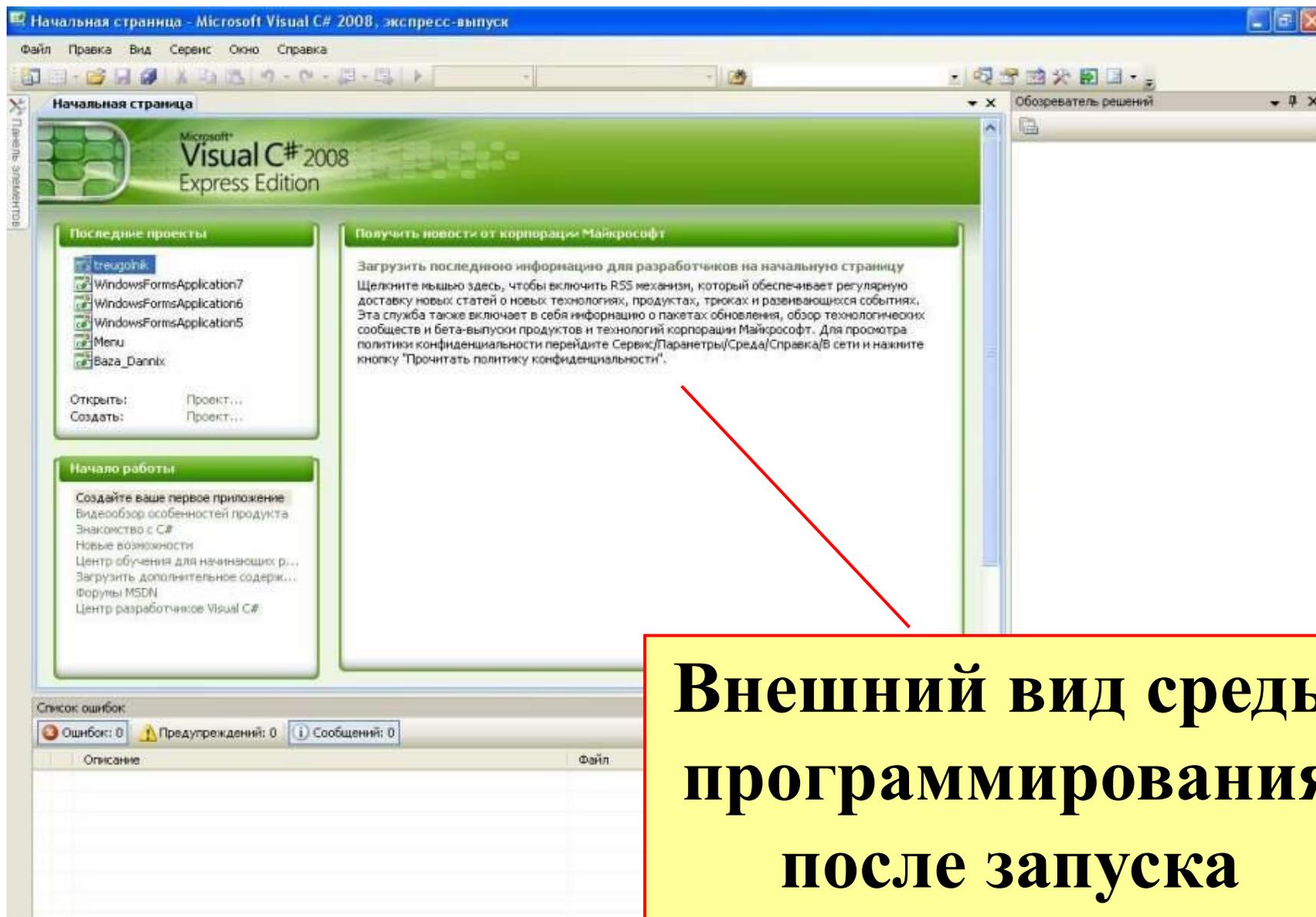
Программируем на C# и Windows Forms

Презентация к уроку информатики
11 класс

Автор: Юдин Андрей Борисович
МКОУ Плесская СОШ

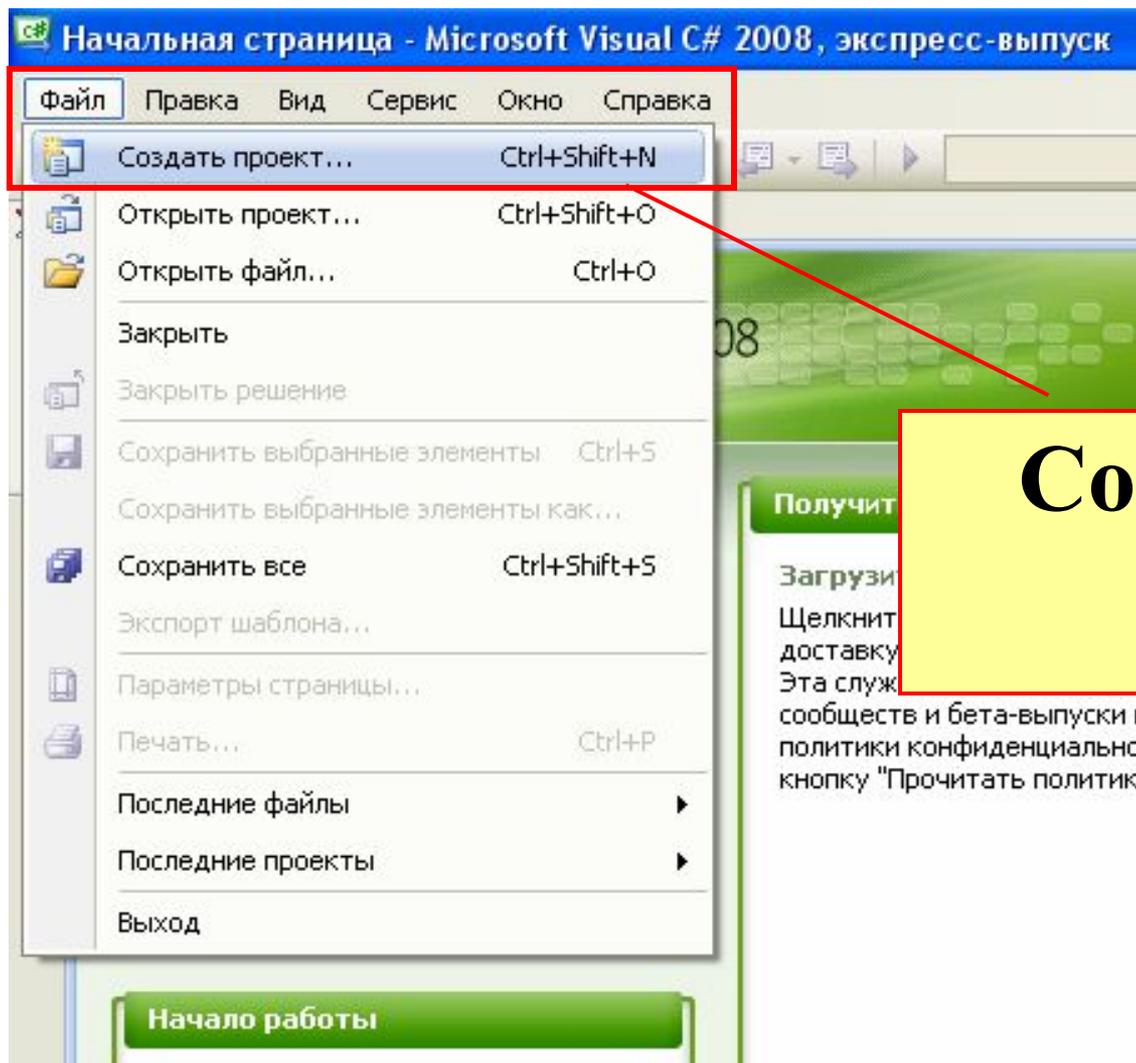
Создание проекта.

Создание проекта. Шаг 1.



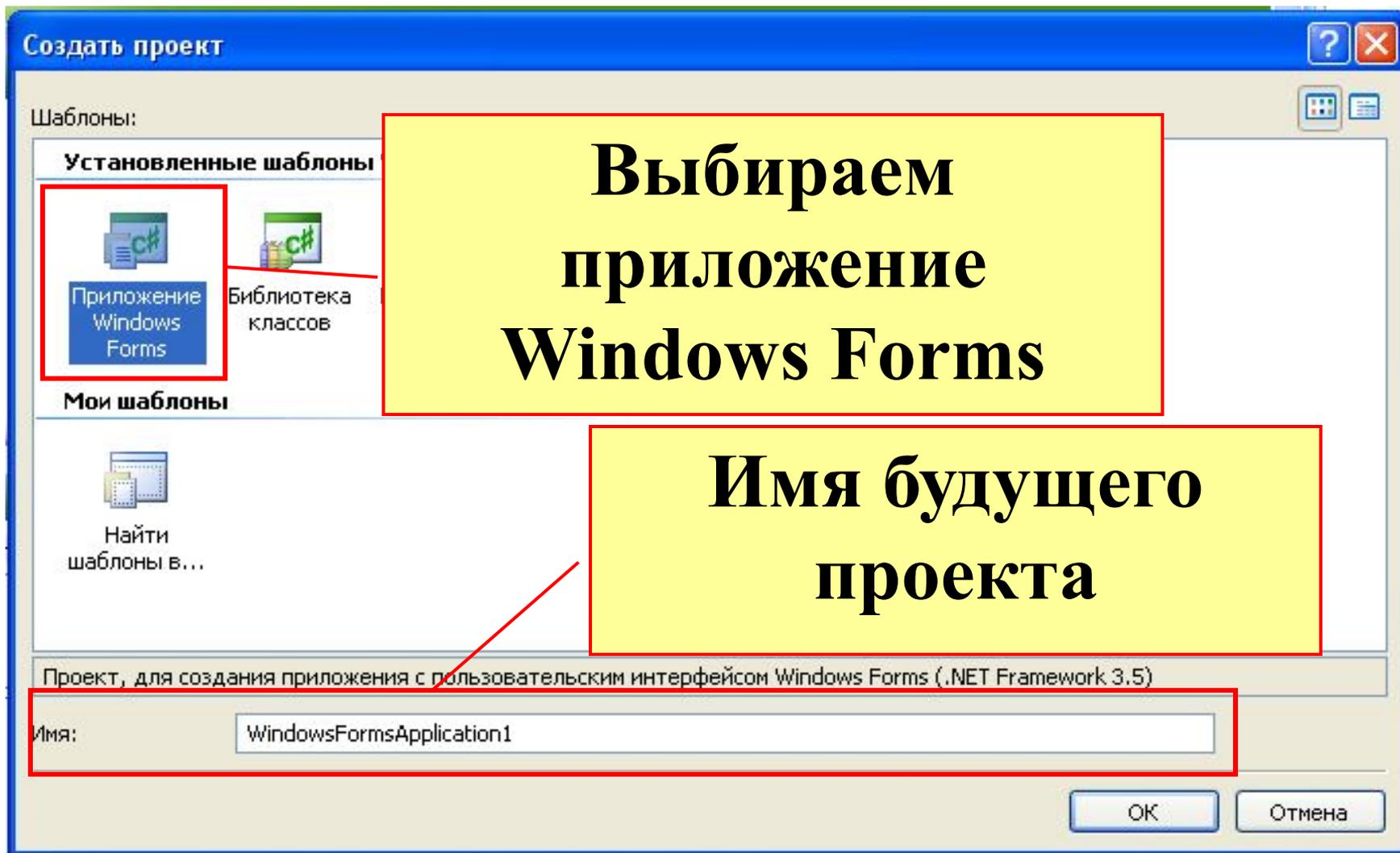
**Внешний вид среды
программирования
после запуска**

Создание проекта. Шаг 2.



**Создаем новый
проект**

Создание проекта. Шаг 3.



Элементы среды программирования

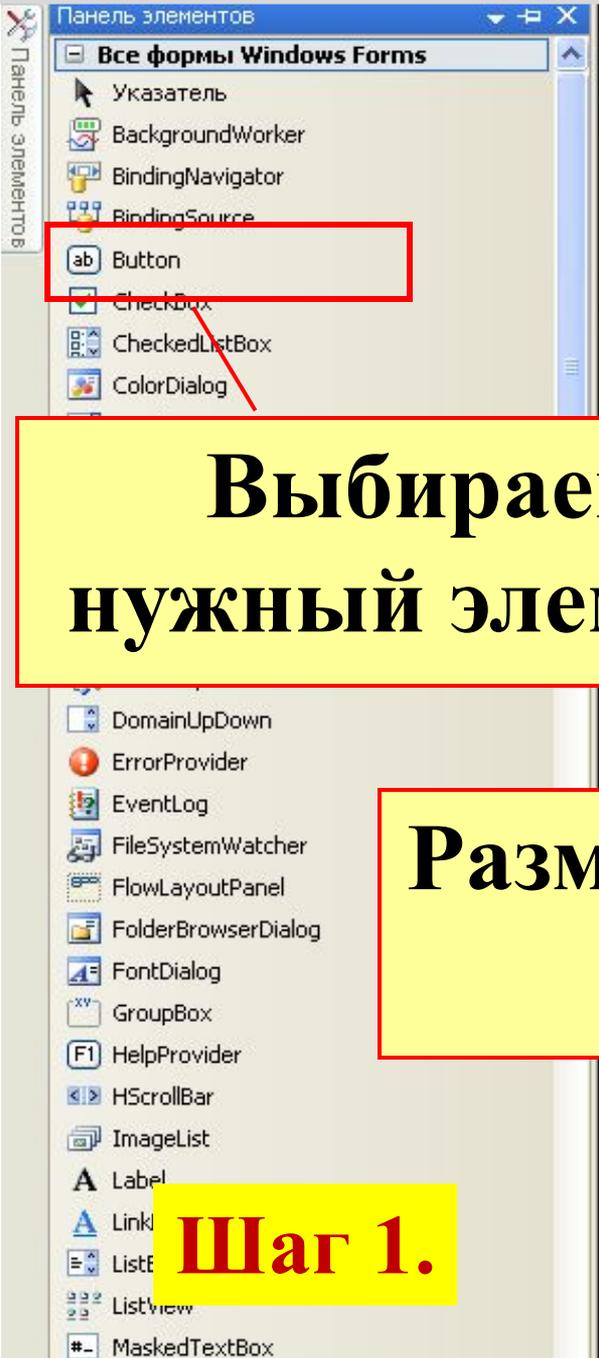
**Панель
компонентов**

Дизайнер формы

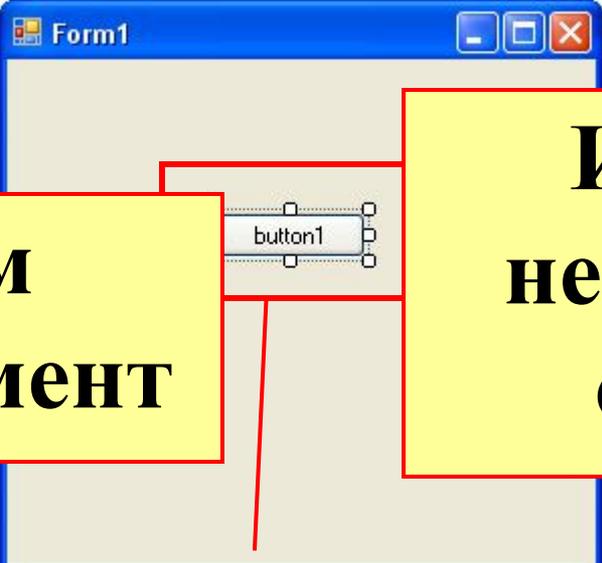
**Свойства и события
объектов
(инспектор объектов)**

**Сообщения об
ошибках**

Конструирование интерфейса



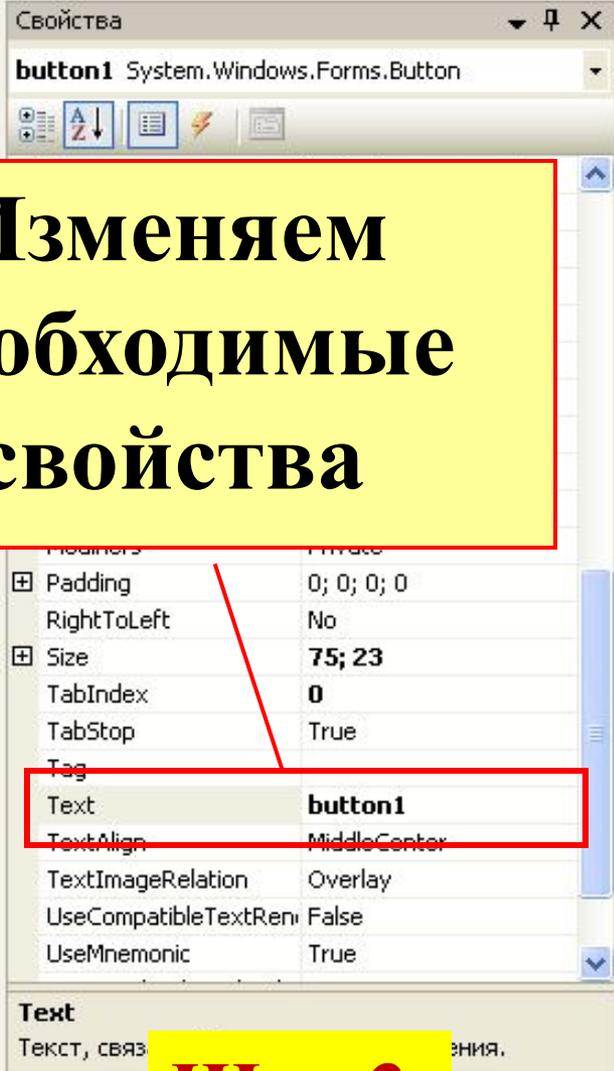
**Выбираем
нужный элемент**



**Размещаем его на
форме**

Шаг 1.

Шаг 2.

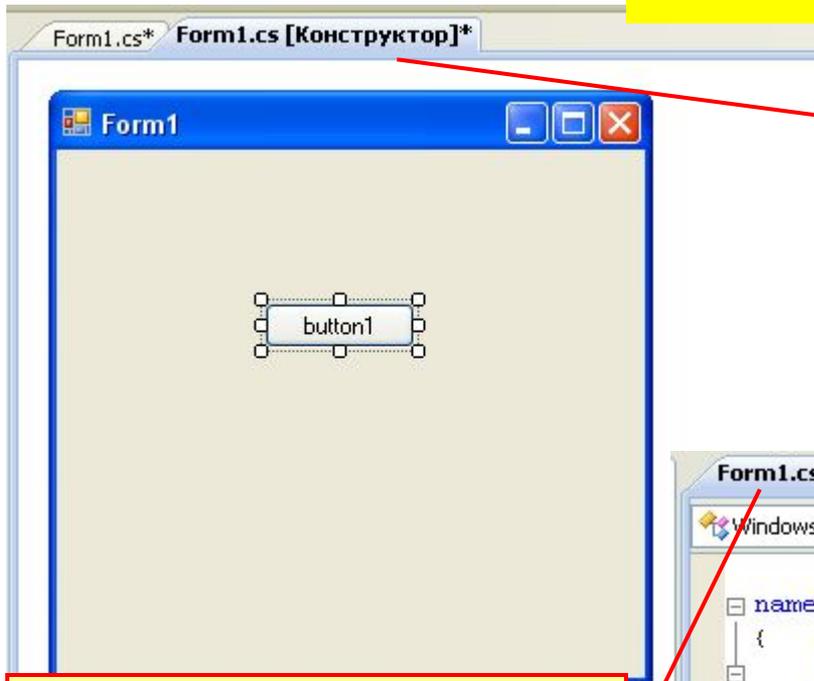


Шаг 3.

Вкладки проекта

**Вкладка
дизайнера
формы**

**Вкладка
текста
программы**



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

        private void button1_Click(object sender, EventArgs e)
        {
            |
        }
    }
}
```

Программа после создания процедуры

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        //здесь описываем глобальные переменные
        public Form1()
        {
            InitializeComponent();
            // настройки формы
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //действия происходящие при щелчке по кнопке
        }
    }
}
```

Область видимости переменных

```
public partial class
{
```

```
int n;
//здесь описываем глобальную переменную
public Form1()
{
    InitializeComponent()
    // настройки
}
```

Глобальная

**Переменная a видна на
в первой процедуре**

```
private void button1_Click(object sender, EventArgs e)
{
    int a;
    //действия при щелчке по кнопке 1
}
```

**Переменная b
видна во второй
процедуре**

```
private void button2_Click(object sender, EventArgs e)
{
    int b;
    //действия происходящие при щелчке по кнопке 2
}
```

```
}
```

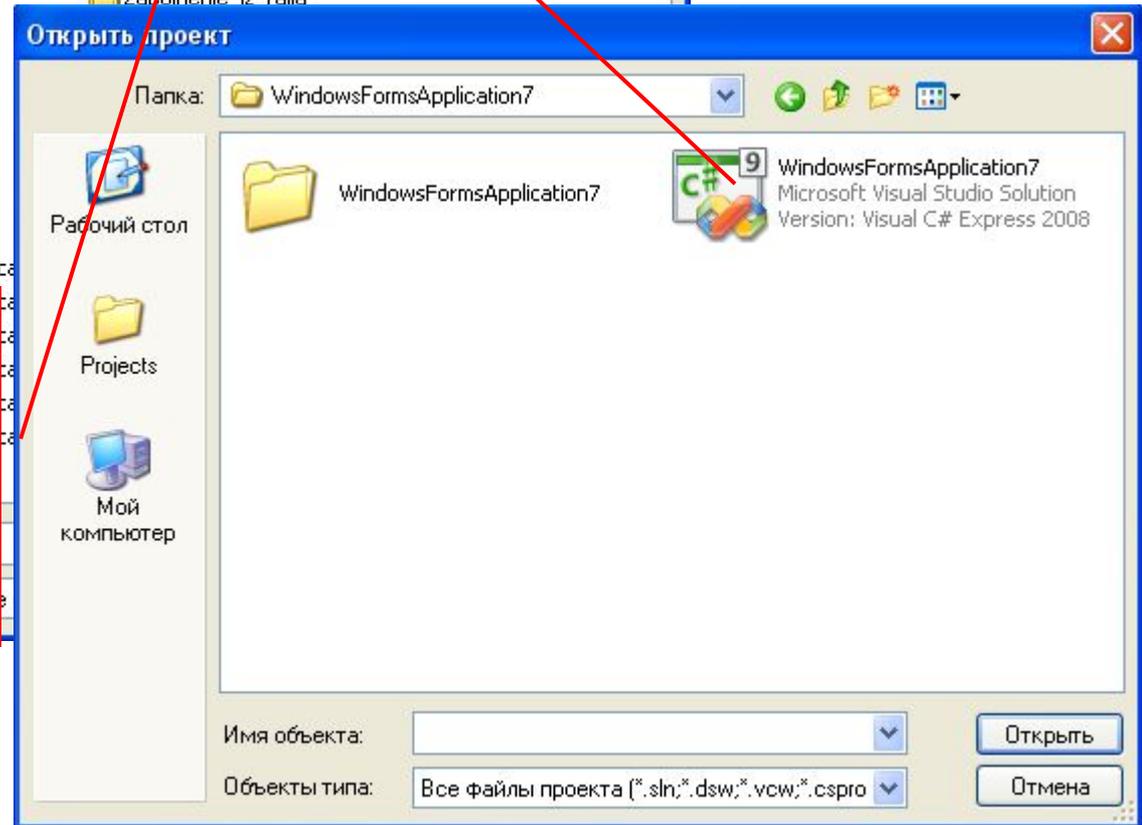
Открытие проекта.

Открытие проекта

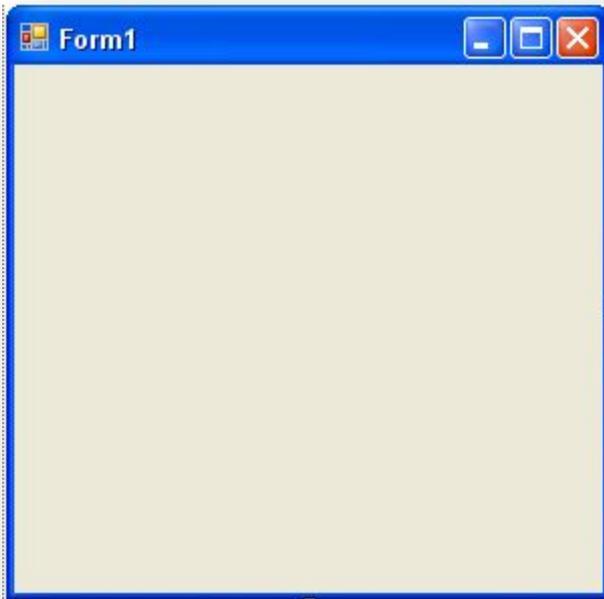
Выбираем «главный»
файл проекта

Выбираем
«Открыть проект»

Выбираем папку
с необходимым
проектом



ОСНОВНЫЕ КОМПОНЕНТЫ.



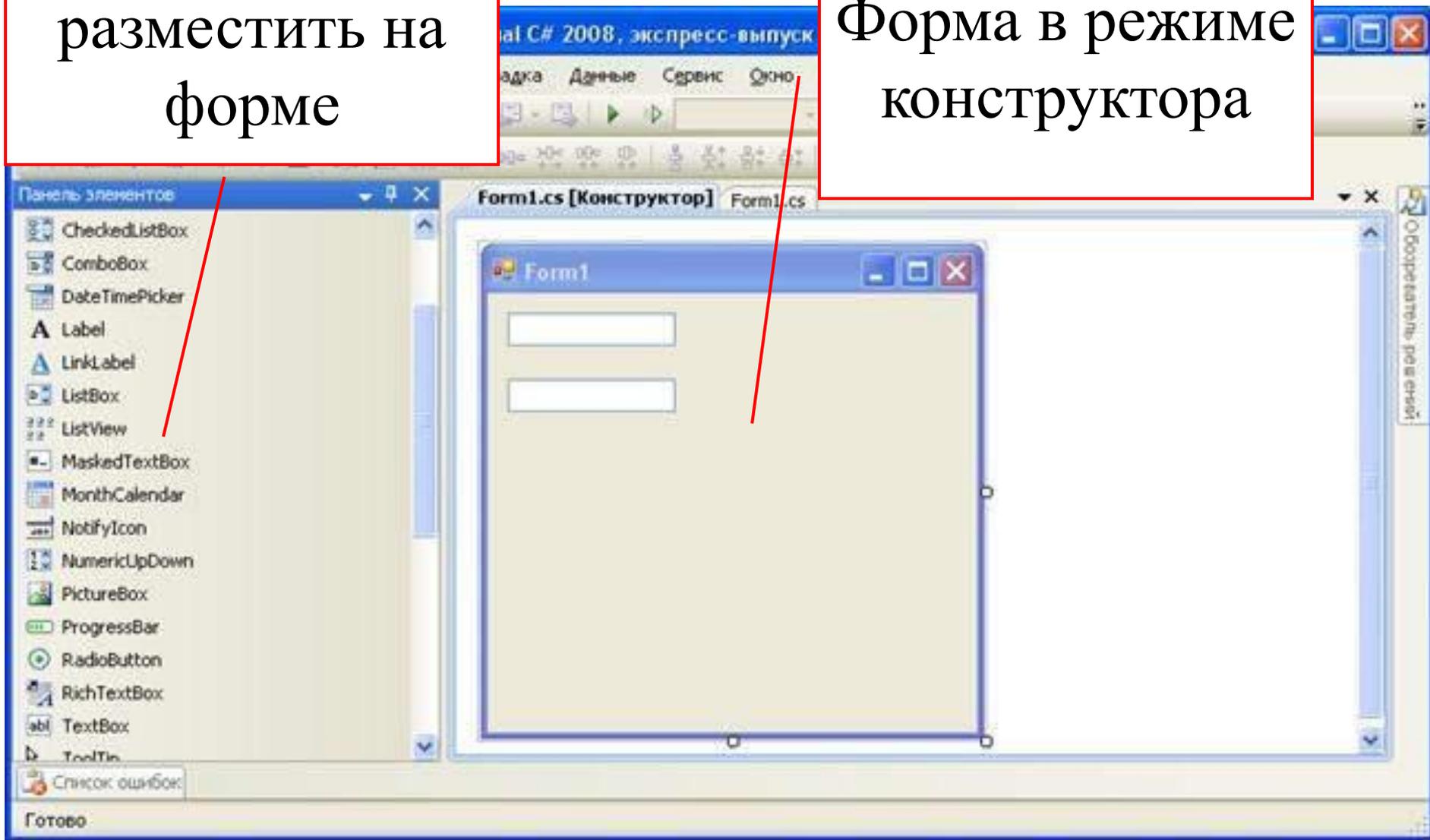
Form – форма

| Свойство | Описание |
|---------------|---|
| Text | Заголовок формы. |
| BackColor | цвет формы |
| Font | задает шрифт для всей формы и всех помещенных на нее элементов управления |
| StartPosition | начальное расположение формы |

Элементы
которые можно
разместить на
форме

Form – форма

Форма в режиме
конструктора





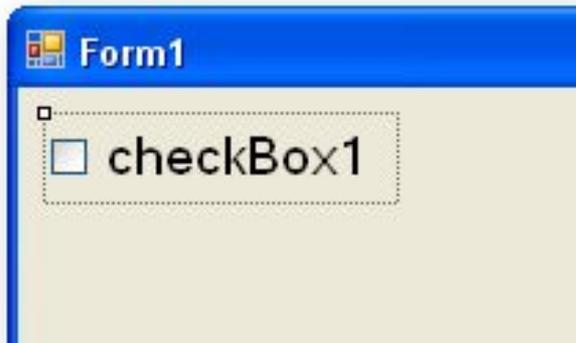
Label – надпись

| Свойство | Значение |
|-----------|----------------------|
| Width | Ширина надписи. |
| Height | Высота надписи. |
| Text | Текст на надписи. |
| BackColor | Цвет фона надписи. |
| ForeColor | Цвет текста надписи. |
| Visible | Видна ли надпись. |



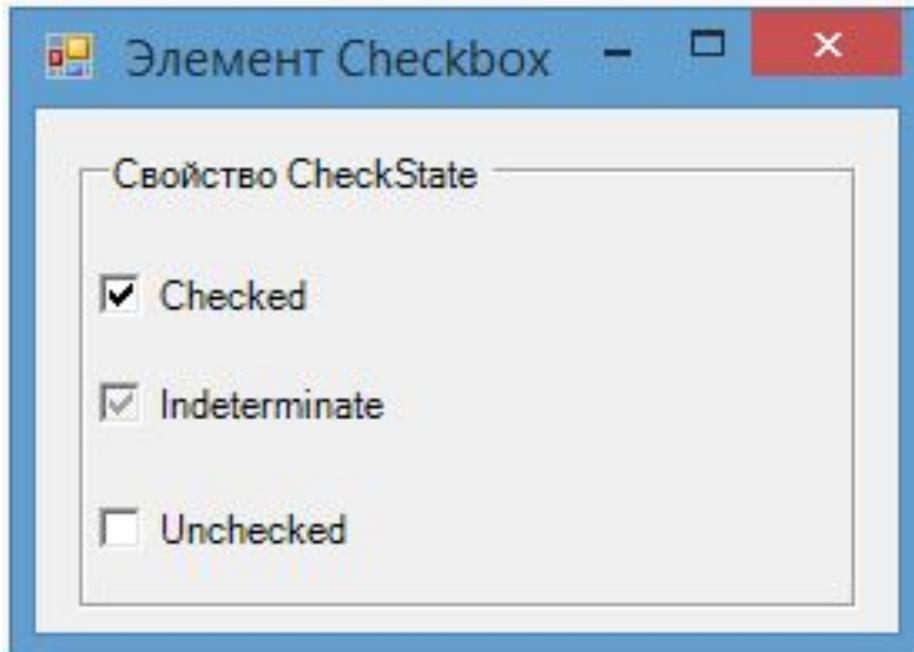
Button – кнопка

| Свойство | Значение |
|------------|------------------------------------|
| Width | Ширина кнопки. |
| Height | Высота кнопки. |
| Text | Текст на кнопке. |
| Image | Задаёт изображение для кнопки |
| ImageAlign | Выравнивание изображения на кнопке |



CheckBox – флажок

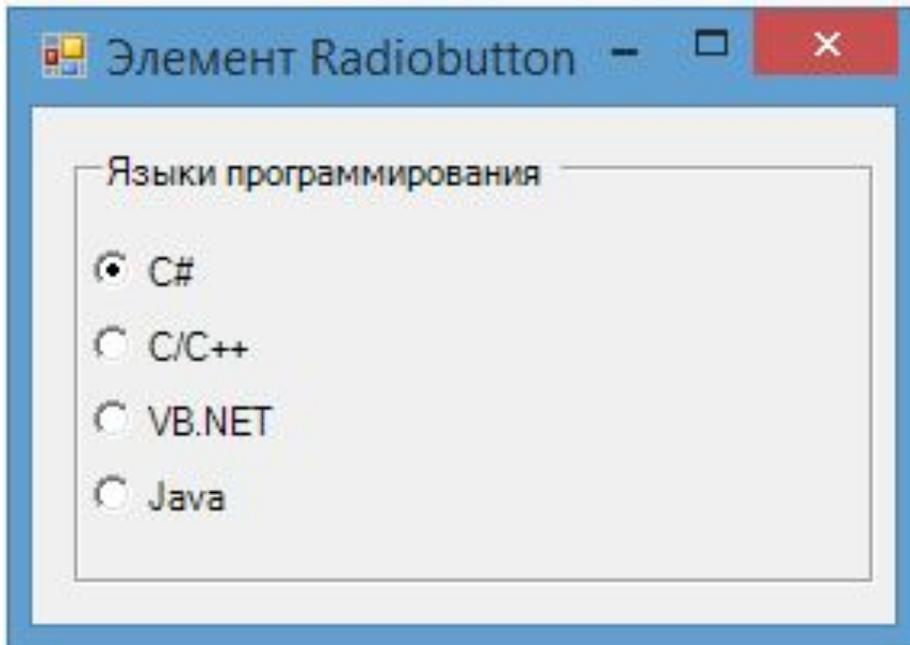
| Свойство | Значение |
|----------|--|
| Width | Ширина флажка. |
| Height | Высота флажка. |
| | Текст на флажка. |
| | Цвет фона флажка. |
| | Цвет текста флажка. |
| | Виден ли флажок. |
| | Определяет установлен ли флажок или нет. |

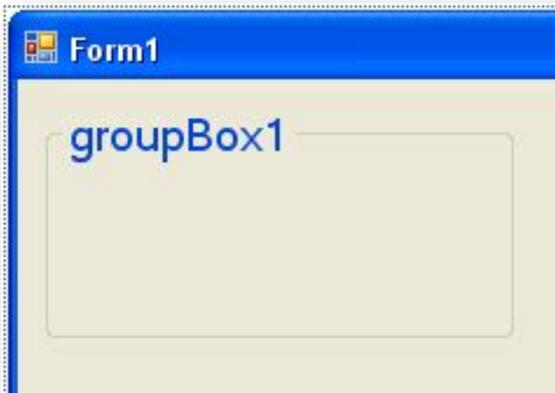




RadioButton – радио-кнопка

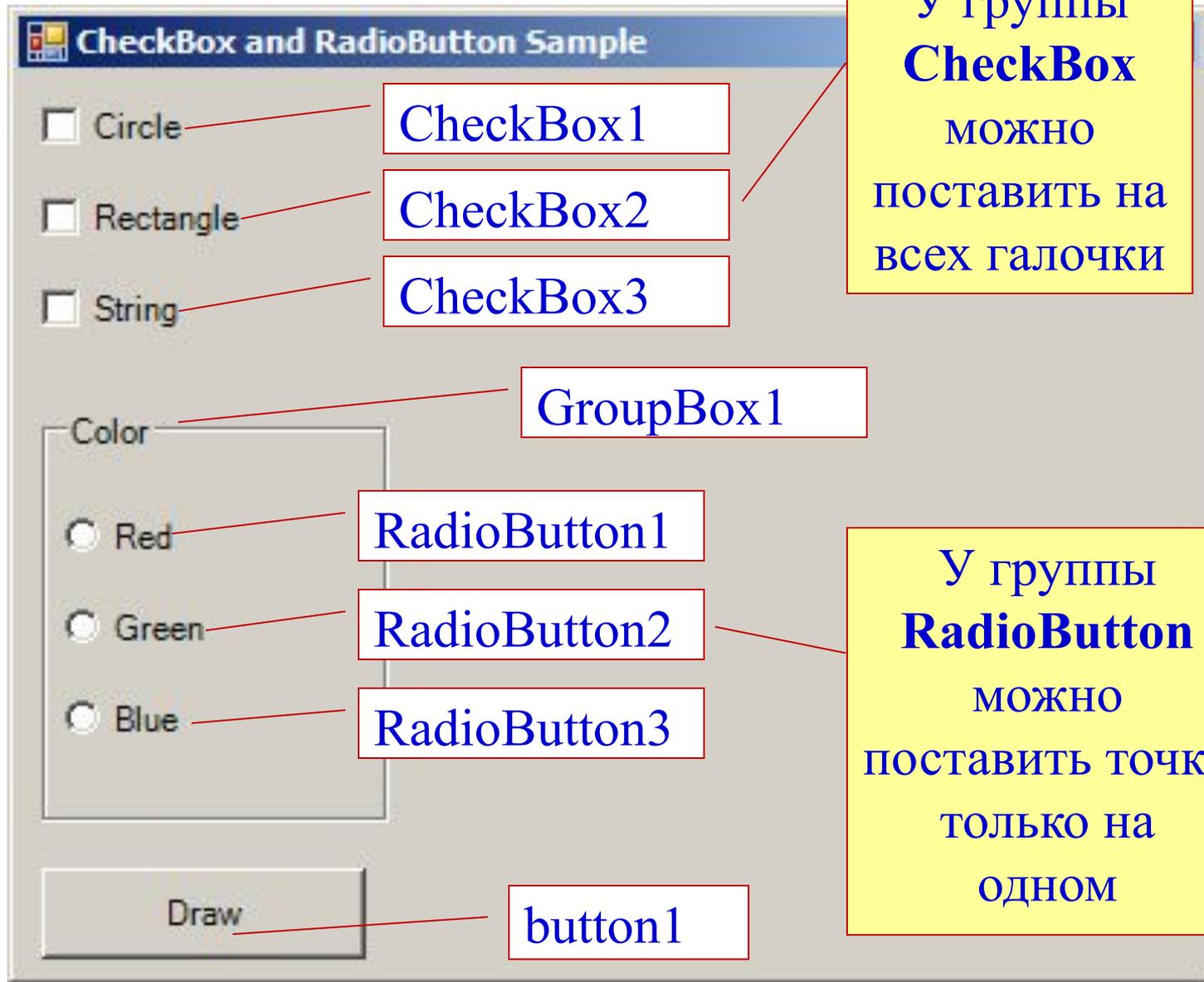
| Свойство | Значение |
|----------|---|
| Width | Ширина радио-кнопки. |
| Height | Высота радио-кнопки. |
| Text | Текст на радио-кнопки. |
| | Цвет фона радио-кнопки. |
| | Цвет текста радио-кнопки. |
| | Видна ли радио-кнопка. |
| | Определяет установлена ли радио-кнопка или нет. |



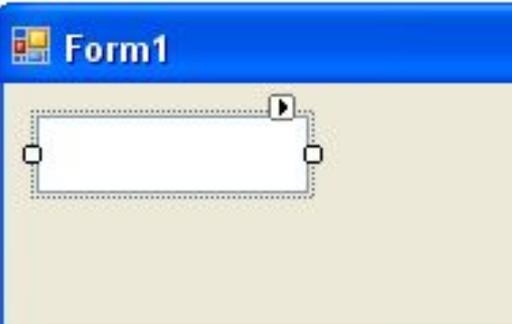


GroupBox – контейнер

| Свойство | Значение |
|------------|--------------------------------|
| Width | Ширина контейнера. |
| Height | Высота контейнера. |
| Text | Текст на контейнере. |
| BackColor | Цвет фона контейнера. |
| ForeColor | Цвет текста контейнера. |
| AutoScroll | Контейнер с полосами прокрутки |



TextBox – ОКНО ВВОДА



| Свойство | Значение |
|-----------|--------------------------|
| Width | Ширина надписи. |
| Height | Высота надписи. |
| Text | Текст на надписи. |
| BackColor | Цвет фона надписи. |
| ForeColor | Цвет текста надписи. |
| Multiline | Многострочное поле ввода |

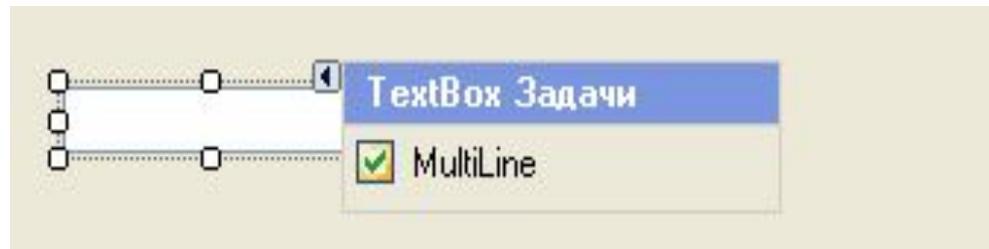
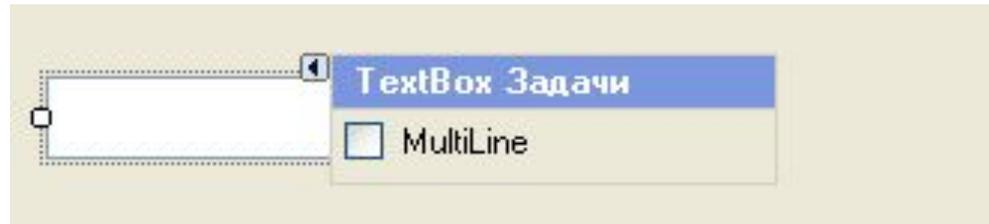
Способ 1.

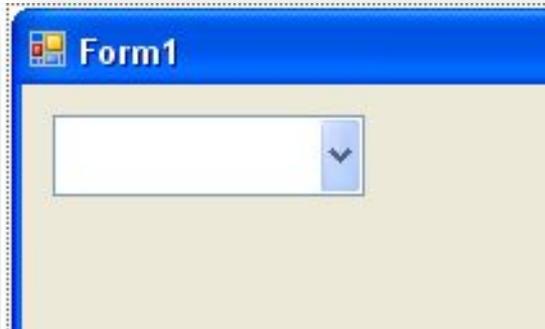


Настройка
многострочного поля
ввода через
инспектор объектов

Способ 2.

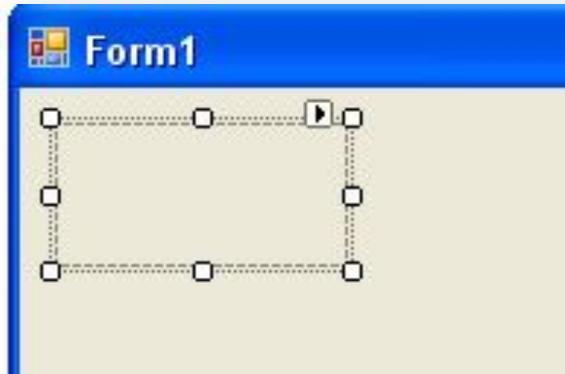
Настройка
многострочного
поля ввода через
элемент на самом
поле ввода





ComboBox – выпадающий список

| Свойство | Значение |
|----------------------|--|
| Items | хранит элементы списка |
| SelectedIndex | Индекс выбранного элемента выпадающего списка. |
| SelectedItem | Выбранный элемент выпадающего списка. |
| Items.Item[k] | К-ый элемент выпадающего списка. |



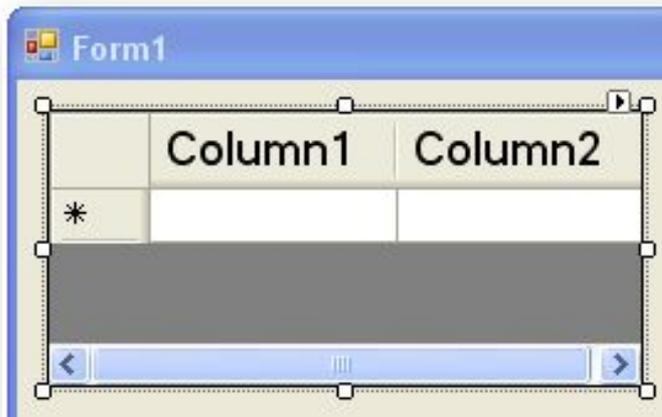
PictureBox – рисунок

| Свойство | Значение |
|----------|-------------------------------------|
| Image | Устанавливает рисунок |
| SizeMode | Настройка вида рисунка в PictureBox |

Свойство **SizeMode**

- **Normal:** изображение позиционируется в левом верхнем углу PictureBox, и размер изображения не изменяется. Если PictureBox больше размеров изображения, то по справа и снизу появляются пустоты, если меньше - то изображение обрезается
- **StretchImage:** изображение растягивается или сжимается таким образом, чтобы вместиться по всей ширине и высоте элемента PictureBox
- **AutoSize:** элемент PictureBox автоматически растягивается, подстраиваясь под размеры изображения
- **CenterImage:** если PictureBox меньше изображения, то изображение обрезается по краям и выводится только его центральная часть. Если же PictureBox больше изображения, то оно позиционируется по центру.
- **Zoom:** изображение подстраивается под размеры PictureBox, сохраняя при этом пропорции





DataGridView – таблица

| Свойство | Значение |
|----------|--|
| Columns | Добавление столбцов |
| ReadOnly | Может ли пользователь редактировать ячейки |

DataGridView1

label1

label2

label3

База данных

| Id | Фамилия | Имя | Телефон |
|----|---------|-------|---------|
| 0 | Иванов | Иван | 1-22-33 |
| 1 | Петров | Петр | 1-33-22 |
| 2 | Сидоров | Сидор | 2-33-22 |
| 3 | Иванов | Петр | 3-33-44 |
| 4 | Петров | Иван | 4-33-44 |
| 5 | Сидоров | Иван | 4-33-22 |
| 6 | Петров | Сидор | 2-11-11 |
| 7 | Иванов | Сидор | 4-33-21 |
| 8 | Сидоров | Петр | 2-33-24 |

Фамилия

Имя

Телефон

Добавить

Удалить строку

Сохранить базу

Загрузить базу

textBox1

textBox2

textBox3

button1

button2

button3

button4

Преобразование типов.

Тип String

Form1

4



Тип Double



```
double a = Convert.ToSingle(textBox1.Text);
```

Form1

9



```
textBox3.Text = Convert.ToString(c);
```

Простые проекты.

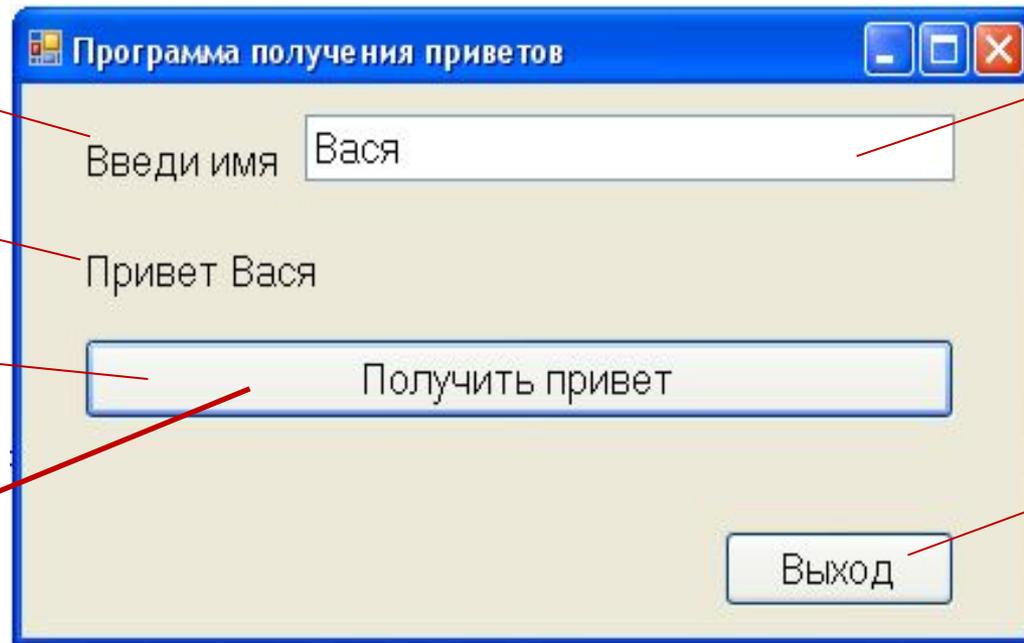
label1

label2

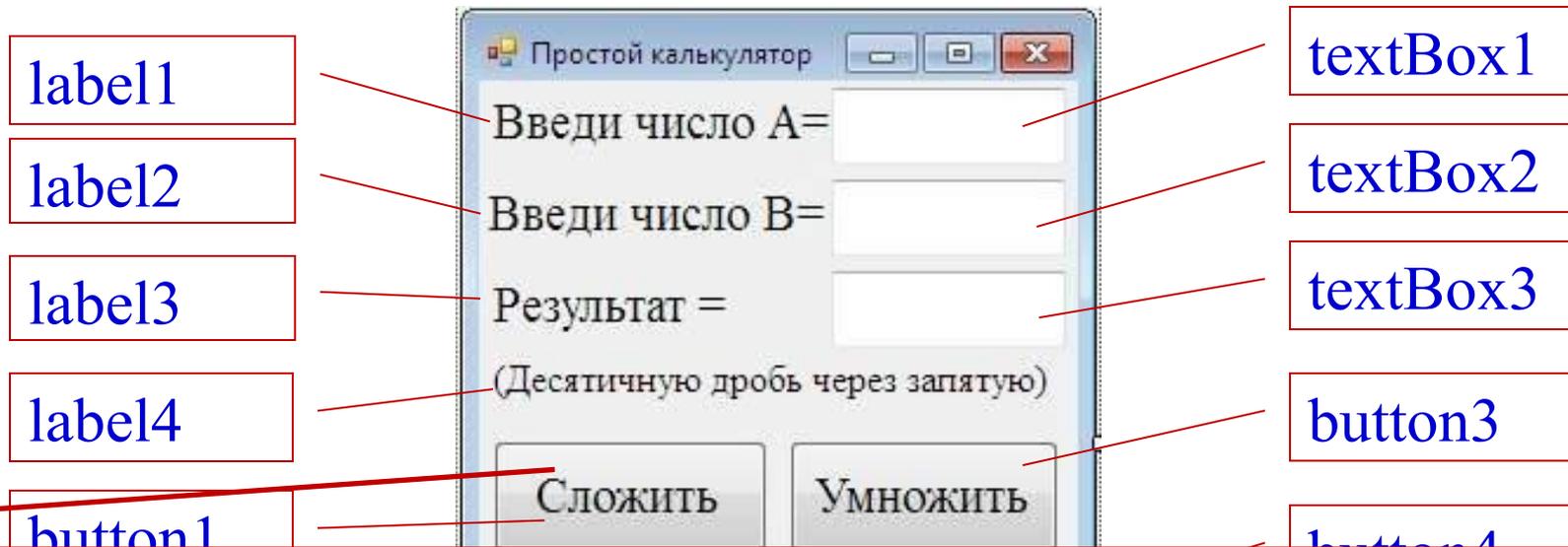
button1

textBox1

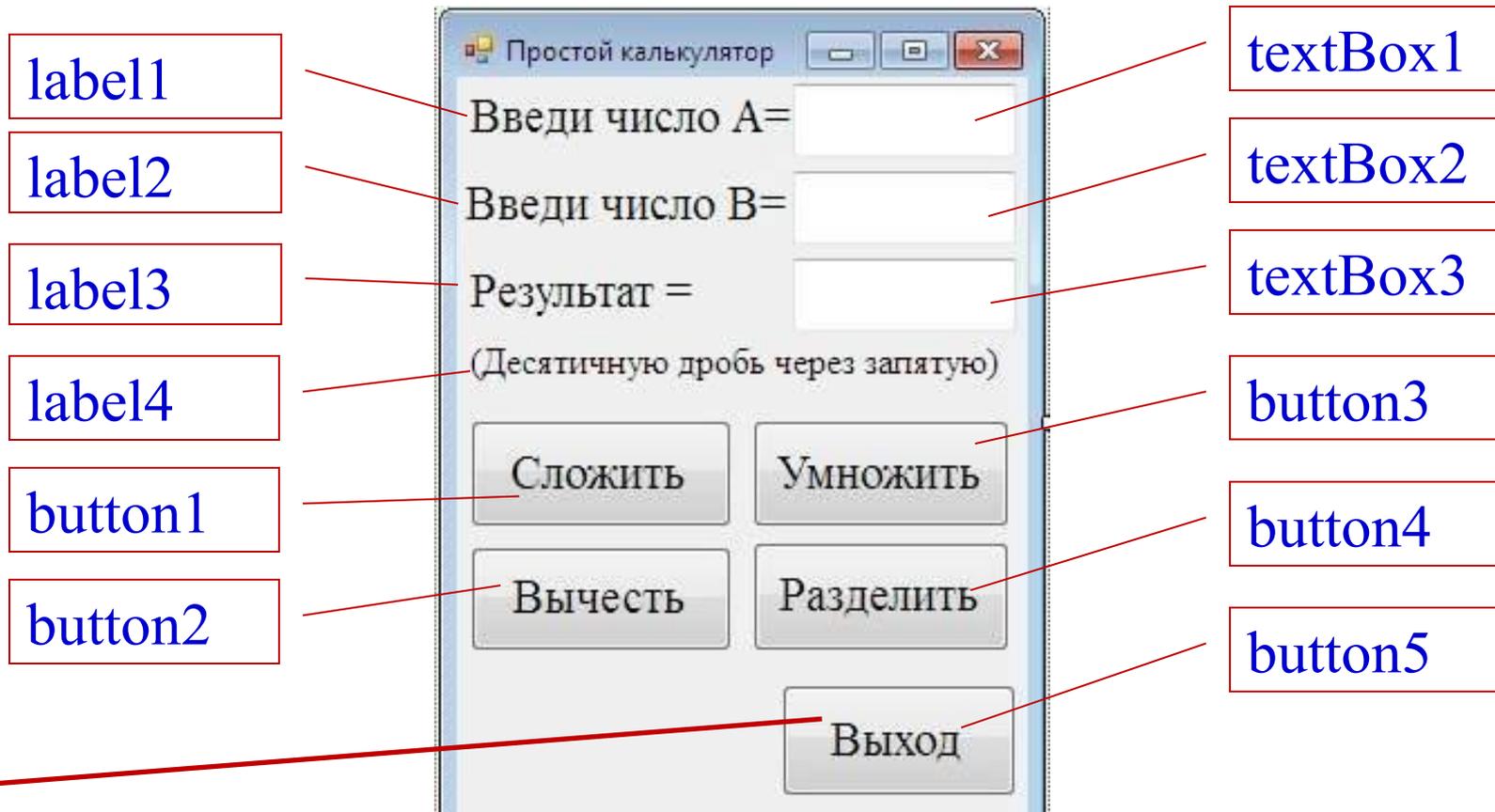
button3



```
private void button1_Click(object sender,
EventArgs e)
{
    label2.Text = "Привет " + textBox1.Text;
}
```



```
private void button1_Click(object sender,
EventArgs e)
{
    double a = Convert.ToSingle(textBox1.Text);
    double b = Convert.ToSingle(textBox2.Text);
    double c = a + b;
    textBox3.Text = Convert.ToString(c);
}
```



```
private void button2_Click(object sender,
EventArgs e)
{
    this.Close();
}
```

GroupBox1

label1

label2

Исходные данные

Введи число A =

Введи число B =

Выбери действие

Выполнить действие

textBox1

textBox2

GroupBox2

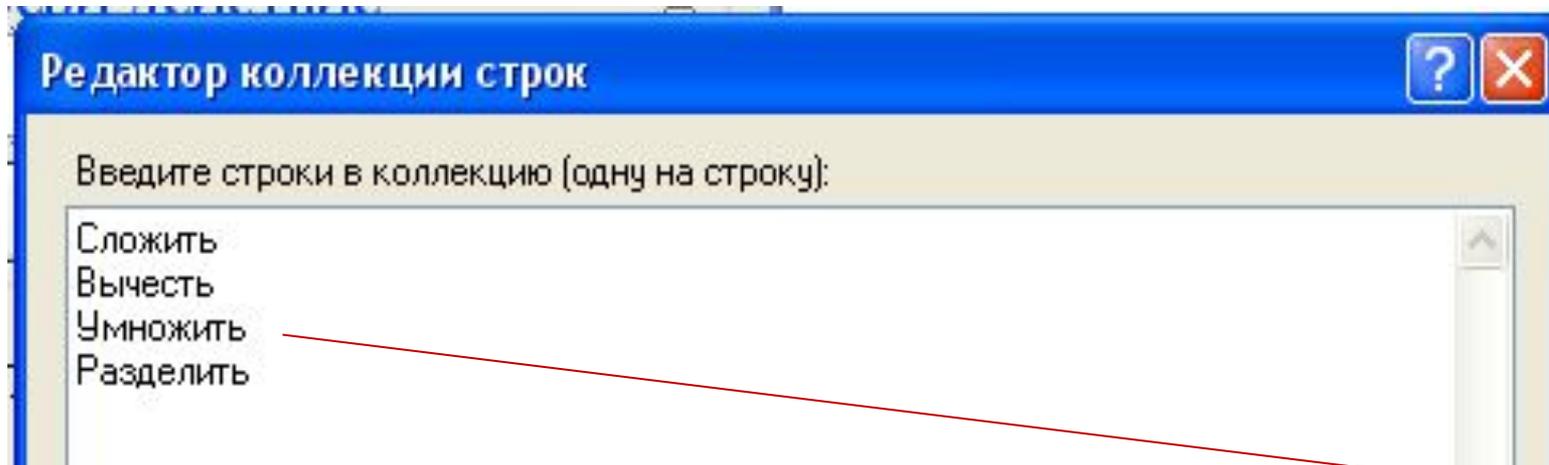
ComboBox1

button1

```
double a = Convert.ToSingle(textBox1.Text);
double b = Convert.ToSingle(textBox2.Text);
double c = 0;
int k = comboBox1.SelectedIndex;
if (k == 0) c = a + b;
if (k == 1) c = a - b;
if (k == 2) c = a * b;
if (k == 3) c = a / b;
textBox3.Text = Convert.ToString(c);
```



Для `ComboBox1` в инспекторе объектов для свойства `Items` нажмите на кнопку с тремя точками



В открывшемся окне запишите названия операций, которые будет выполнять калькулятор

label1

label2

RadioButton1

RadioButton2

textBox1

textBox2

RadioButton3

RadioButton4

button1

Введи число A =

Введи число B =

Сложить Умножить

Вычесть Разделить

Вычислить

```
double a = Convert.ToSingle(textBox1.Text);
double b = Convert.ToSingle(textBox2.Text);
double c = 0;
if (radioButton1.Checked == true) c = a + b;
if (radioButton2.Checked == true) c = a - b;
if (radioButton3.Checked == true) c = a * b;
if (radioButton4.Checked == true) c = a / b;
textBox3.Text = Convert.ToString(c);
```

Список используемой литературы:

1. В.В.Лабор. СиШарп: Создание приложений для Windows. Минск издательство Харвест,2003
2. Герберт Шилдт. С# 3.0. Полное руководство. Издательство: Вильямс, 2010
3. Чарльз Петцольд. Программирование с использованием Microsoft Windows Forms. Издательство: Русская Редакция, Питер, 2006
4. Сайт <http://metanit.com/> Сайт посвященный С# и семейству технологий .NET (ASP.NET MVC, WPF, Xamarin, EntityFramework и т.д.), технологии на базе Java (Java SE, программирование под ОС Андроид и т.д.), работа с базами данных (MS SQL Server, MySQL, MongoDB) а также WEB-технологии, такие как HTML5, AJAX, jQuery, ExtJS, AngularJS и др.