

Программируем на C# - Windows Forms

Создание проекта.

Создание проекта.

Создание проекта

Поиск шаблонов (ALT+“B”) Очистить все

Последние шаблоны проектов

- Консольное приложение (.NET Framework) C#
- Консольное приложение C++
- Приложение WPF (.NET Framework) C#
- Приложение Windows Forms (.NET Framework) C#
- Классическое приложение Windows C++
- Консольное приложение (.NET Core) C#
- Веб-приложение ASP.NET (.NET Framework) C#
- WPF Custom Control Library (.NET Core) C#
- Библиотека классов (.NET Framework) C#
- Библиотека элементов управления Windows Forms (.NET Framework) C#
- Библиотека динамической компоновки (DLL) C++
- Проект общих элементов C++
- Мастер классических приложений Windows C++
- Проект Makefile C++
- Статическая библиотека C++

Служба рабочих ролей
Шаблон проекта для создания рабочей службы с использованием
C# Linux macOS Windows Облако Служба

Проект тестов MSTest (.NET Core)
Проект, содержащий модульные тесты MSTest, которые можно
платформе .NET Core в Windows, Linux и MacOS.
C# Linux macOS Windows Тестирование

Тестовый проект NUnit (.NET Core)
Проект с тестами NUnit, которые могут выполняться на базе .NET
Windows, Linux и MacOS.
C# Linux macOS Windows Рабочий стол Тести

Веб

Приложение Windows Forms (.NET Framework)
Проект для создания приложения с пользовательским интерфейсом
Windows Forms (WinForms)
C# Windows Рабочий стол

Приложение WPF (.NET Framework)
Клиентское приложение Windows Presentation Foundation
C# Windows Рабочий стол

WPF App (.NET Core)
Клиентское приложение Windows Presentation Foundation
C# Windows Рабочий стол

WPF Custom Control Library (.NET Core)
Библиотека настраиваемых элементов управления Windows Presentation
Foundation
C# Windows Рабочий стол Библиотека

WPF User Control Library (.NET Core)
Библиотека пользовательских элементов управления Windows Presentation

**Выбираем
приложение
Windows
Forms**

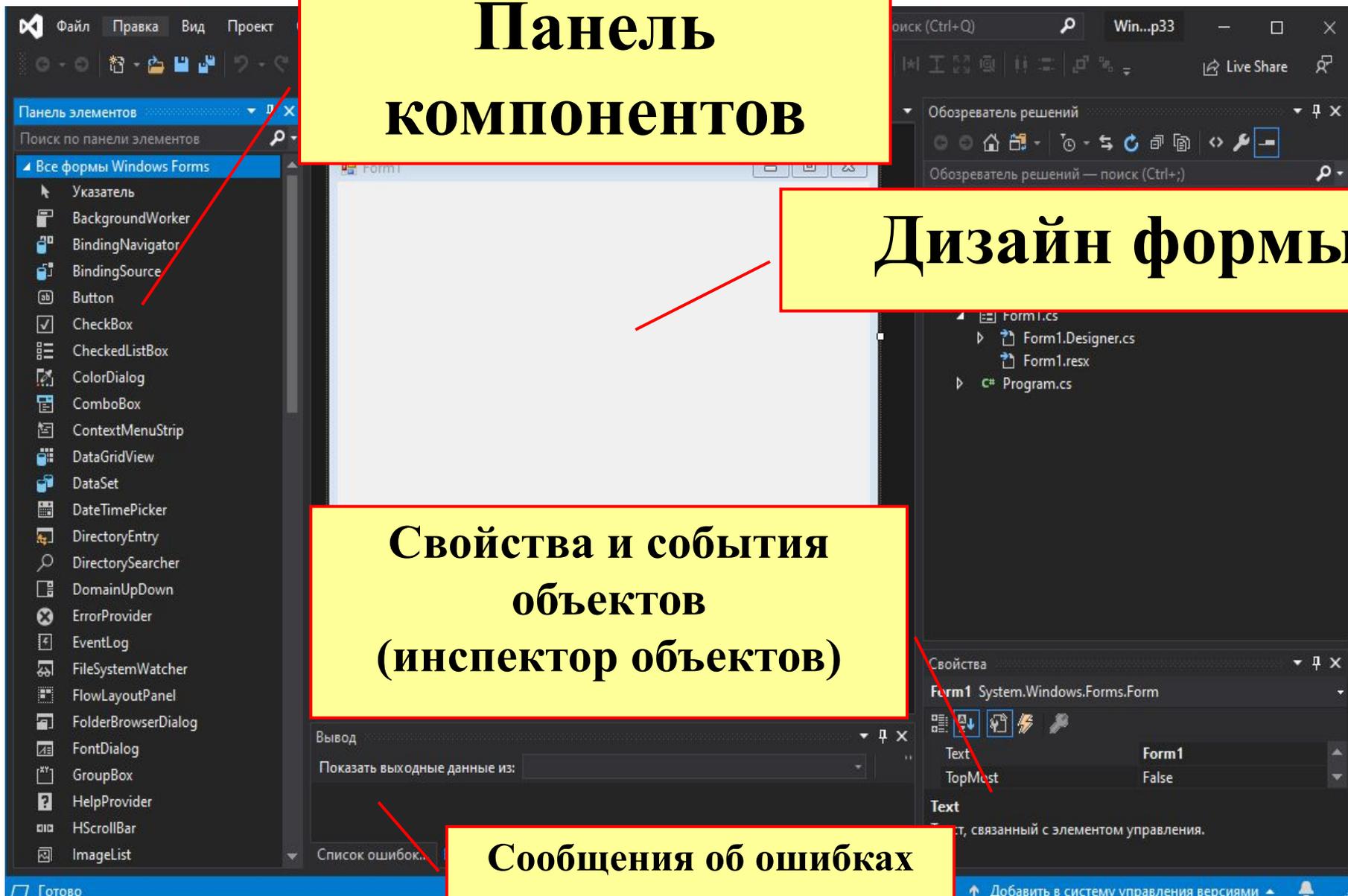
Элементы среды программирования

**Панель
КОМПОНЕНТОВ**

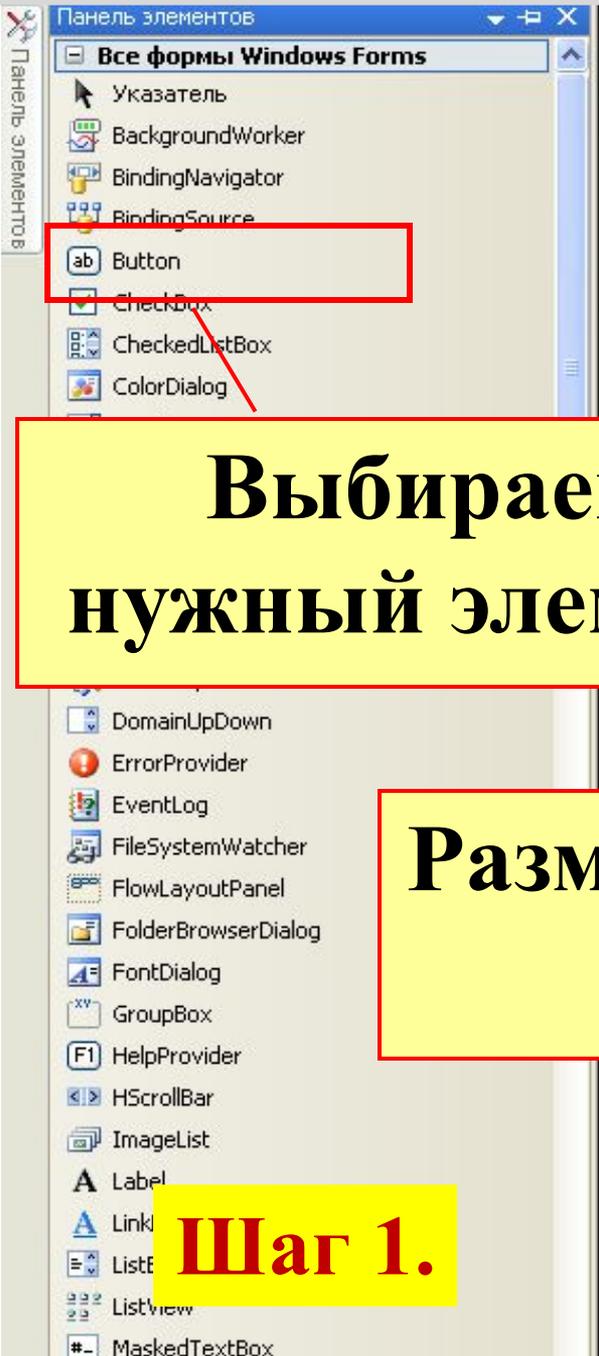
Дизайн формы

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

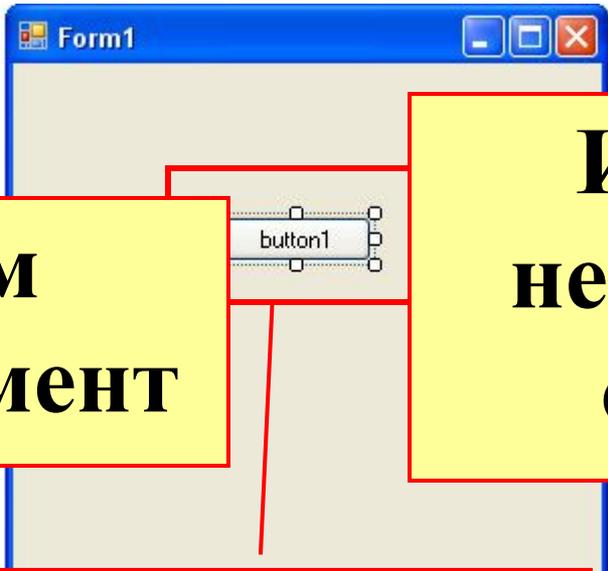
Сообщения об ошибках



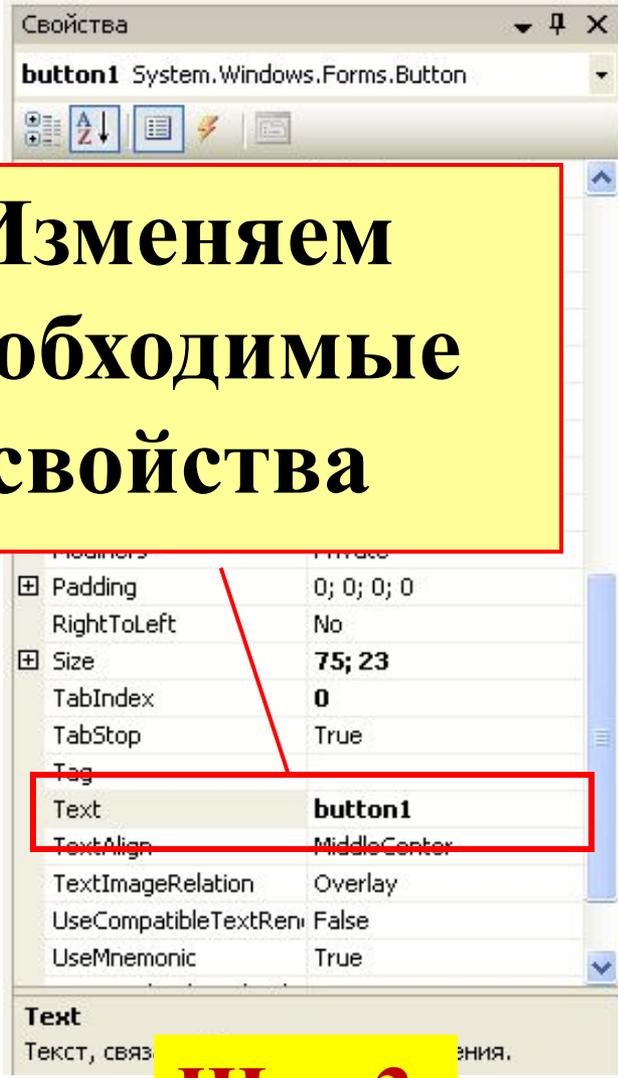
Конструирование интерфейса



**Выбираем
нужный элемент**



**Размещаем его на
форме**



**Изменяем
необходимые
свойства**

Шаг 1.

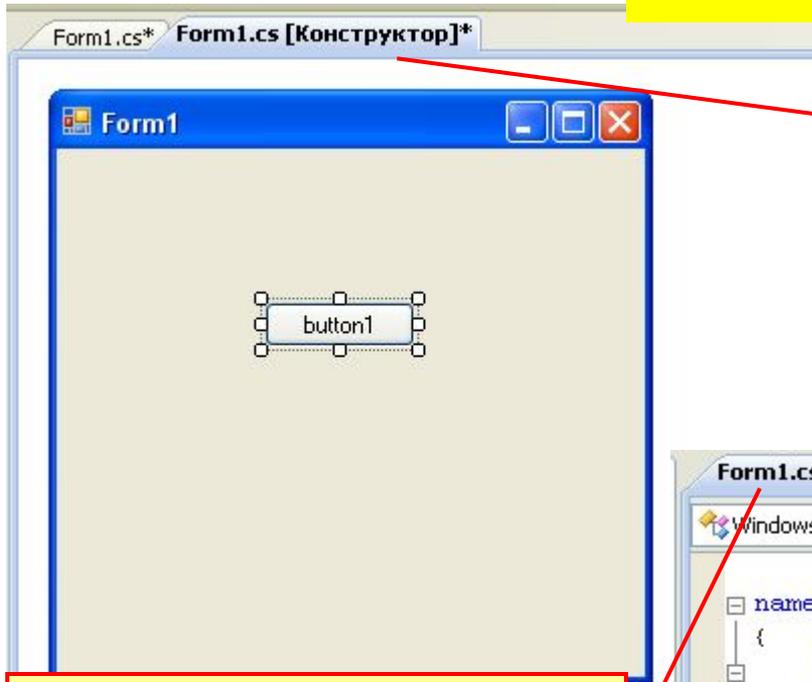
Шаг 2.

Шаг 3.

Вкладки проекта

**Вкладка
формы**

**Вкладка
текста
программы**



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

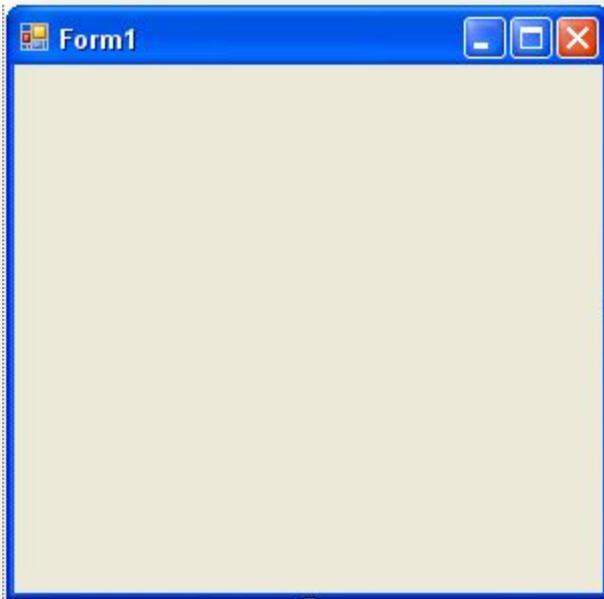
        private void button1_Click(object sender, EventArgs e)
        {
            |
        }
    }
}
```

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
        }
    }
}
```

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



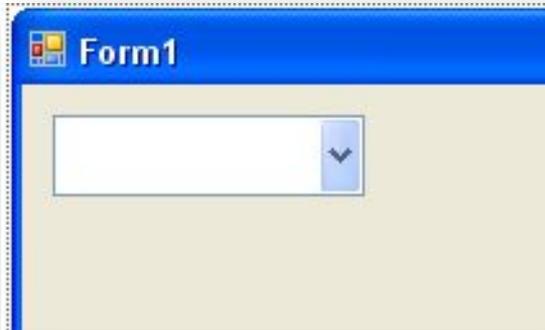
Label – надпись

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



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.

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

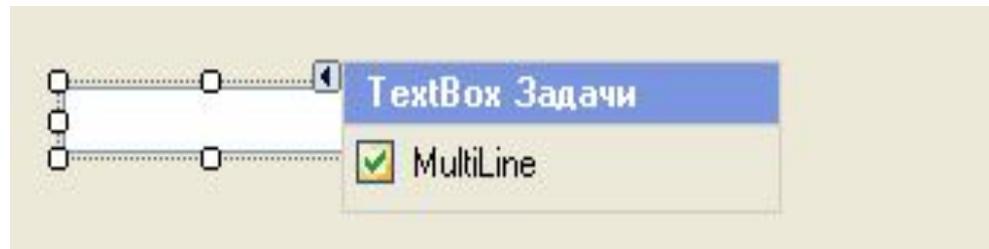
Способ 1.

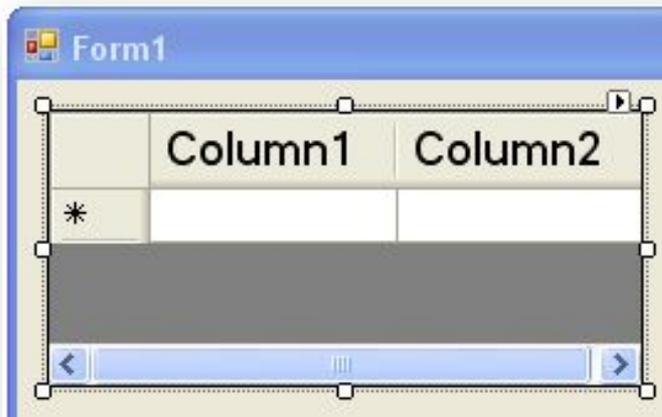


Настройка
многострочного поля
ввода через
инспектор объектов

Способ 2.

Настройка
многострочного
поля ввода через
элемент на самом
поле ввода





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

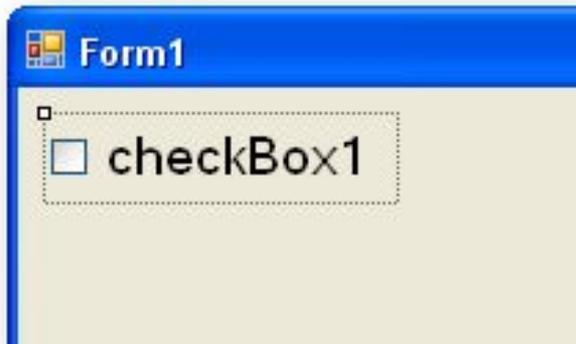
textBox3

button1

button2

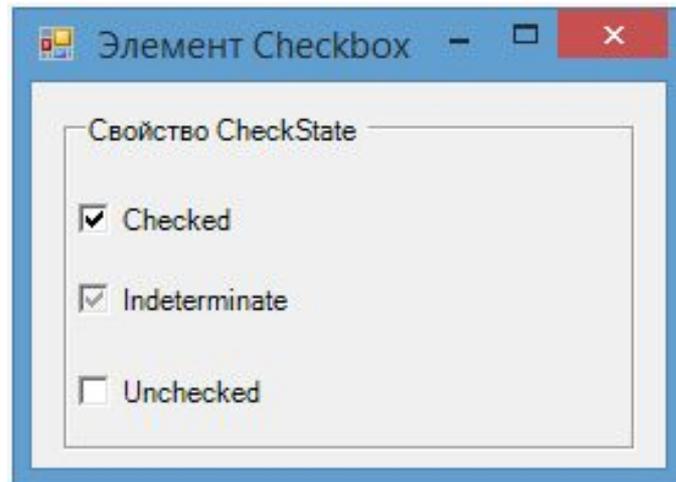
button3

button4



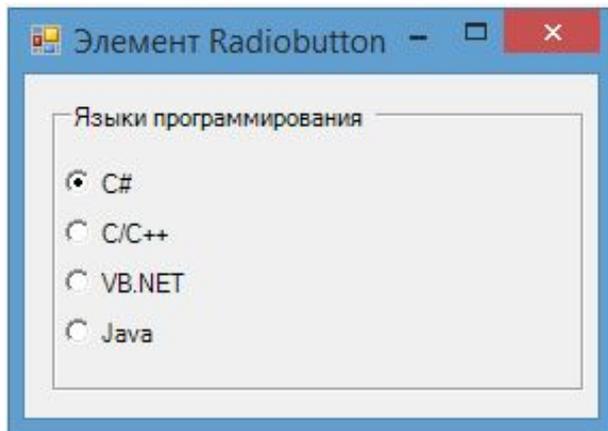
CheckBox – флажок

Свойство	Значение
Width	Ширина флажка.
Height	Высота флажка.
Text	Текст на флажка.
BackColor	Цвет фона флажка.
ForeColor	Цвет текста флажка.
Visible	Виден ли флажок.
Checked	Определяет установлен ли флажок или нет.

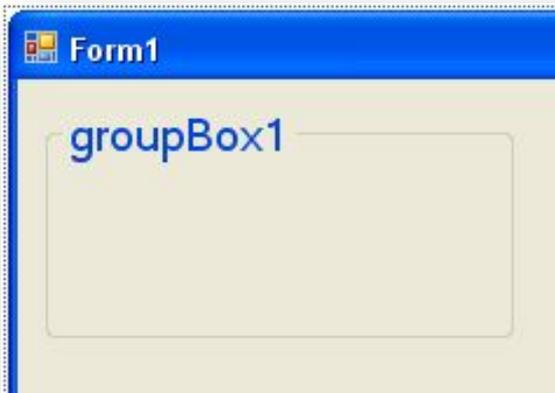




RadioButton – радио-кнопка

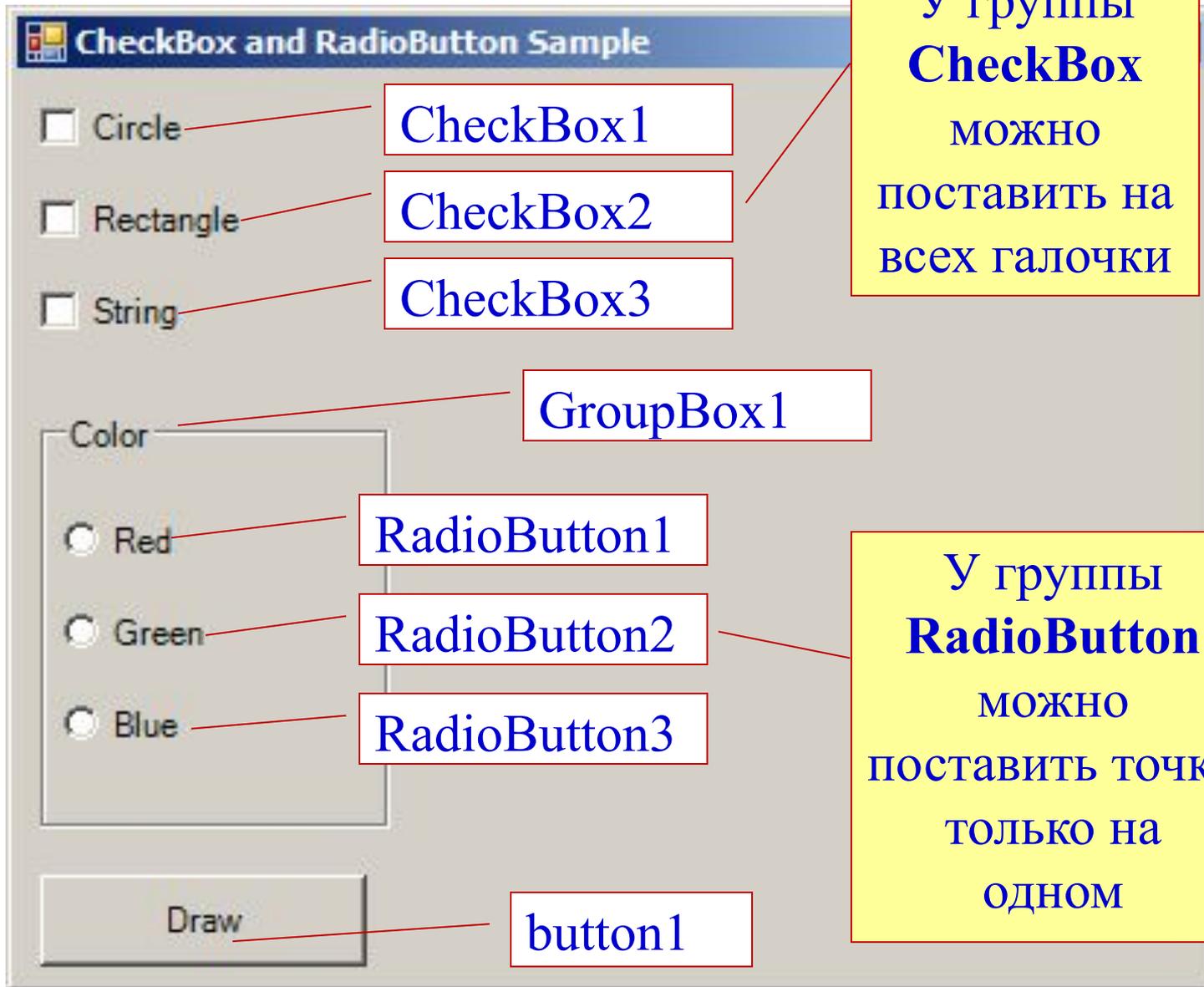


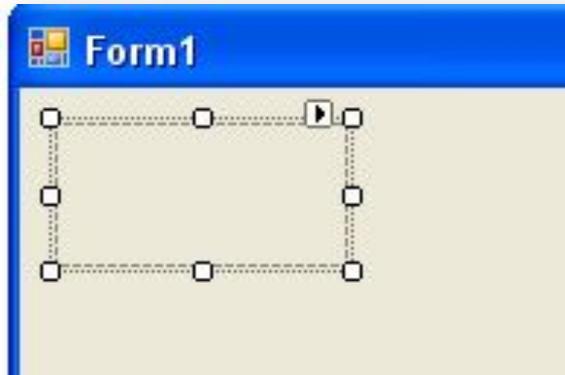
Свойство	Значение
Width	Ширина радио-кнопки.
Height	Высота радио-кнопки.
Text	Текст на радио-кнопки.
BackColor	Цвет фона радио-кнопки.
ForeColor	Цвет текста радио-кнопки.
Visible	Видна ли радио-кнопка.
Checked	Определяет установлена ли радио-кнопка или нет.



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



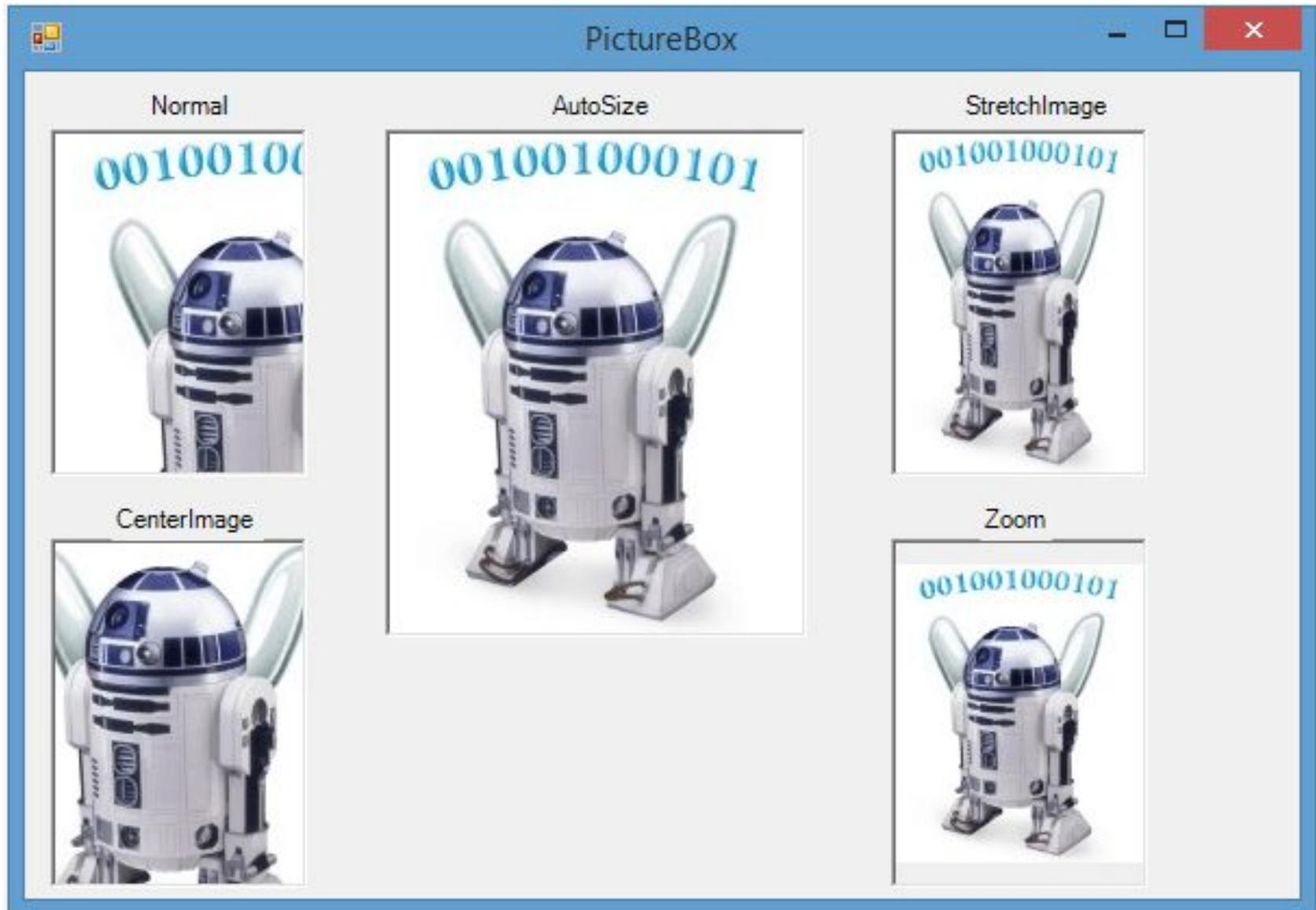


PictureBox – рисунок

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



Преобразование типов.

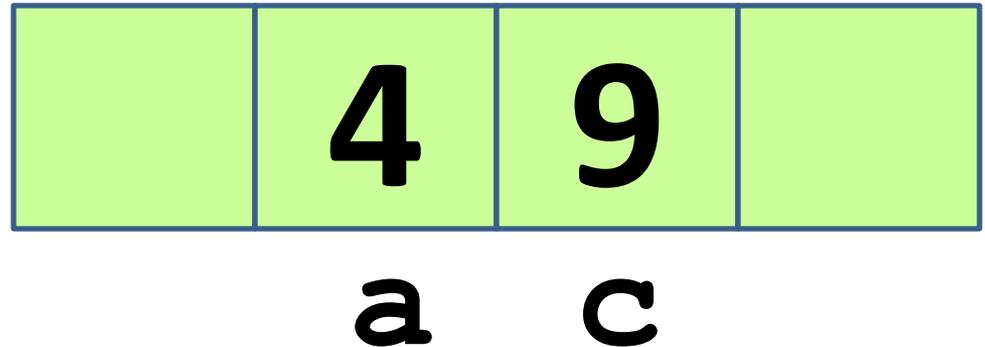
Тип String

Form1

4



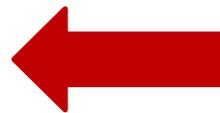
Тип Double



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

Form1

9



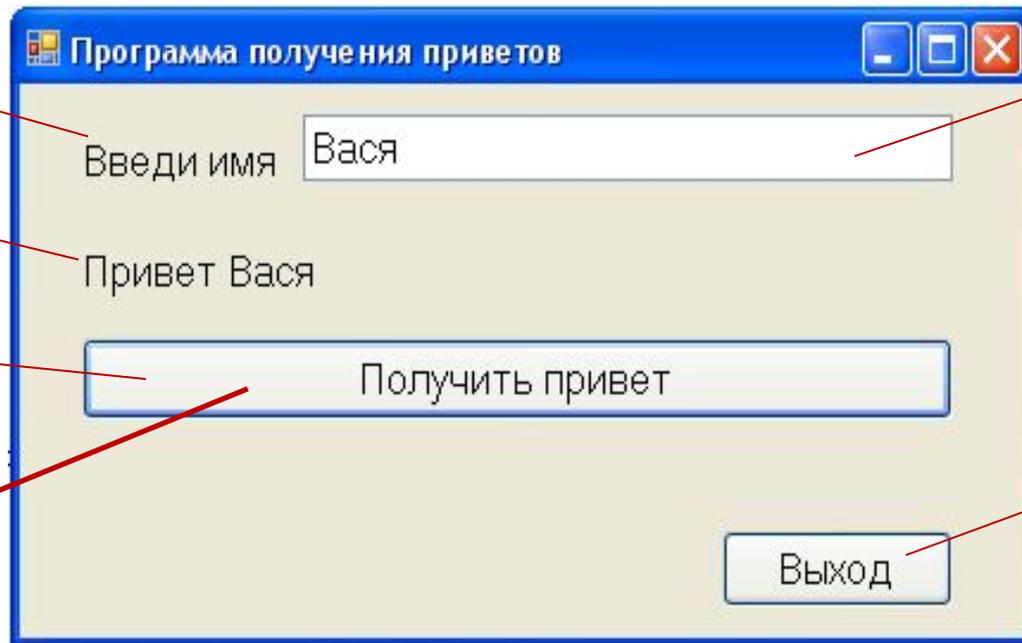
```
textBox3.Text = Convert.ToString(c);
```

Простые проекты.

label1

label2

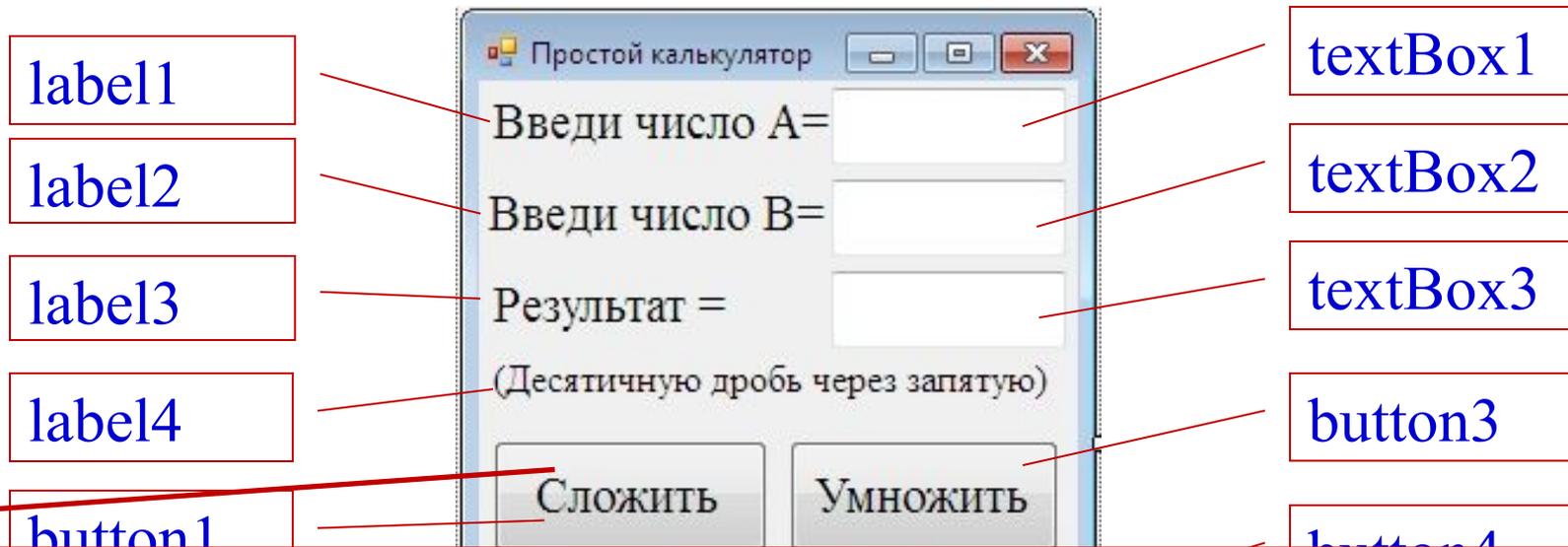
button1



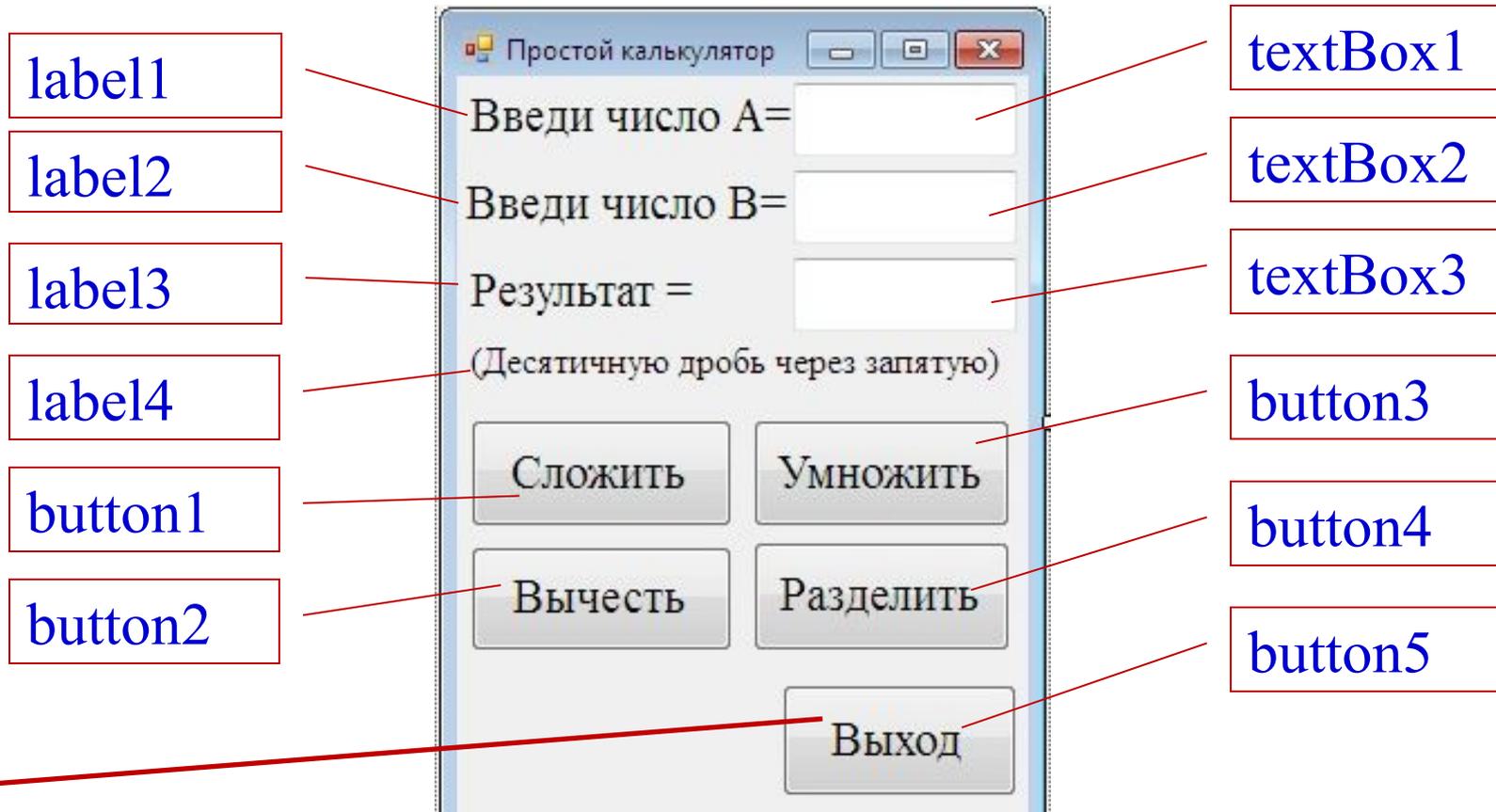
textBox1

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

textBox1

textBox2

GroupBox2

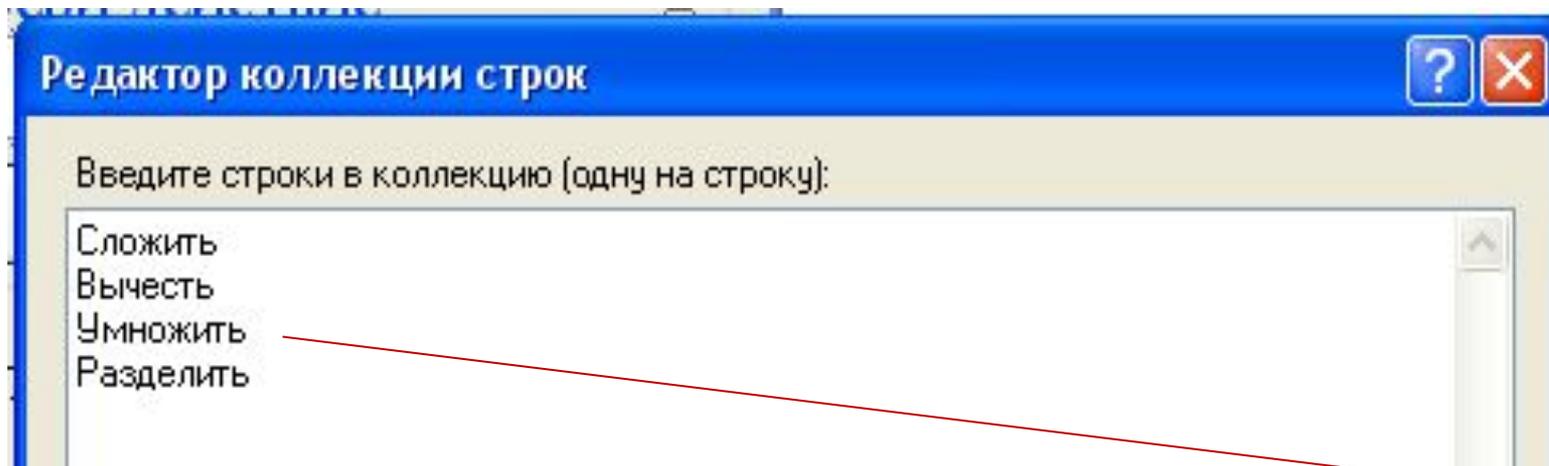
Combobox1

button1

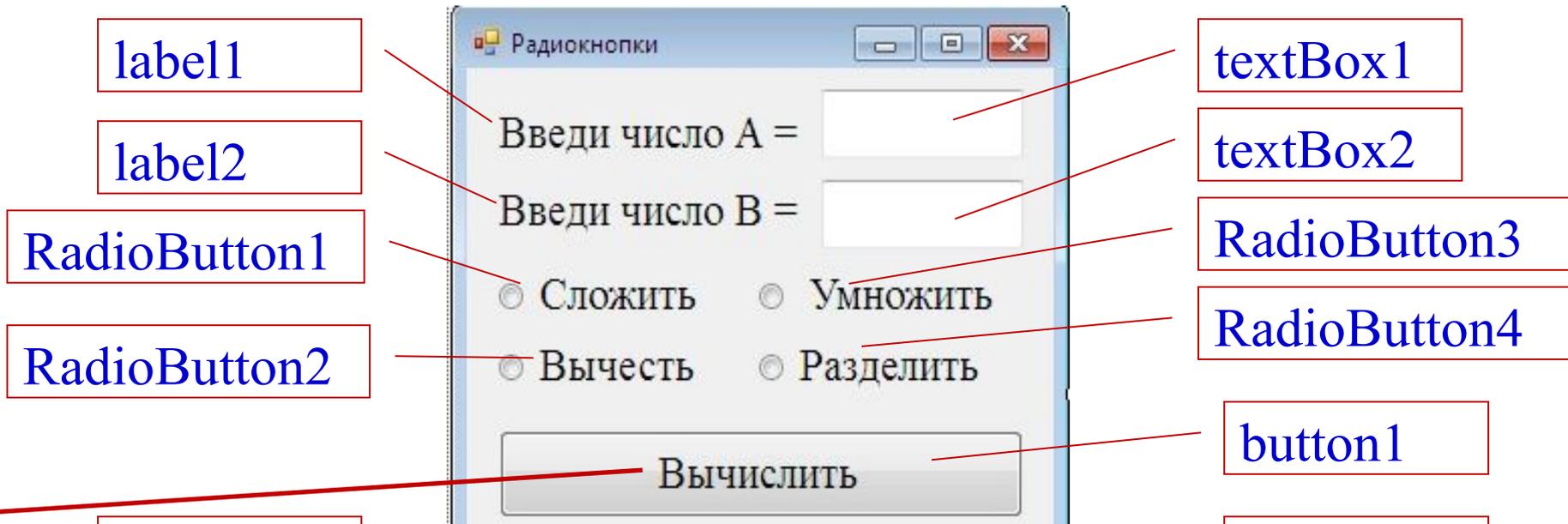
```
double a = Convert.ToSingle(textBox1.Text);
double b = Convert.ToSingle(textBox2.Text);
double c = 0;
int k = comboBox1.SelectedIndex;
if (k == 0) c = a + b;
if (k == 1) c = a - b;
if (k == 2) c = a * b;
if (k == 3) c = a / b;
textBox3.Text = Convert.ToString(c);
```



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



В открывшемся окне запишите названия операций, которые будет выполнять калькулятор



```

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);

```

1. Даны три целых числа. Найти их сумму и произведение.
2. Дана сторона квадрата a . Найти его периметр и площадь.
3. Даны три числа. Найти среднее арифметическое

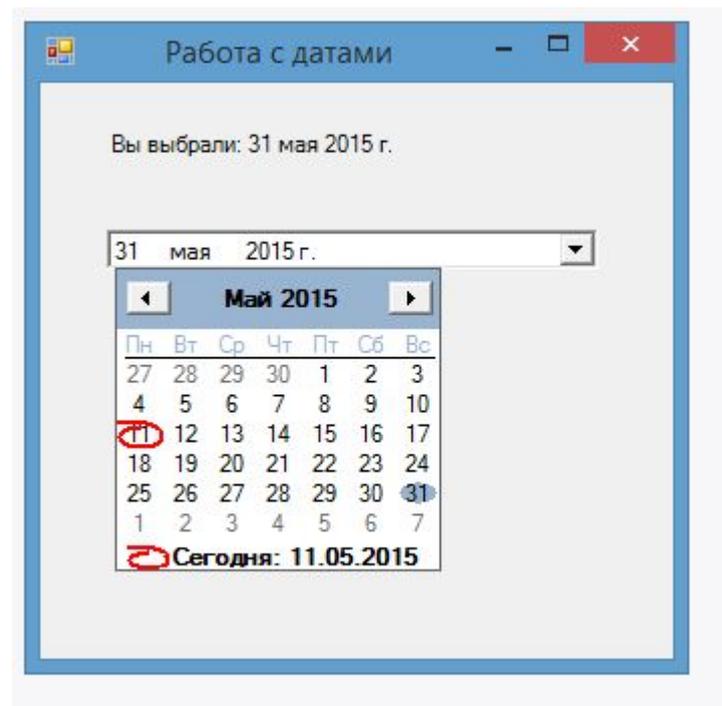
DateTimePicker и MonthCalendar

DateTimePicker

представляет раскрывающийся по нажатию календарь, в котором можно выбрать дату.

MonthCalendar

отображает календарь одного месяца



Свойства для определения дат в календаре:

- **MinDate**: определяет минимальную дату для выбора в календаре
- **MaxDate**: задает наибольшую дату для выбора в календаре
- **FirstDayOfWeek**: определяет день недели, с которого должна начинаться неделя в календаре
- **SelectionRange**: определяет диапазон выделенных дат
- **SelectionEnd**: задает конечную дату выделения
- **SelectionStart**: определяет начальную дату выделения
- **ShowToday**: при значении true отображает внизу календаря текущую дату
- **ShowTodayCircle**: при значении true текущая дата будет обведена кружочком
- **TodayDate**: определяет текущую дату. По умолчанию используется системная дата на компьютере, но с помощью данного свойства мы можем ее изменить.
- **ShowWeekNumbers**: выводит подписи дней недели и номер недели в году.

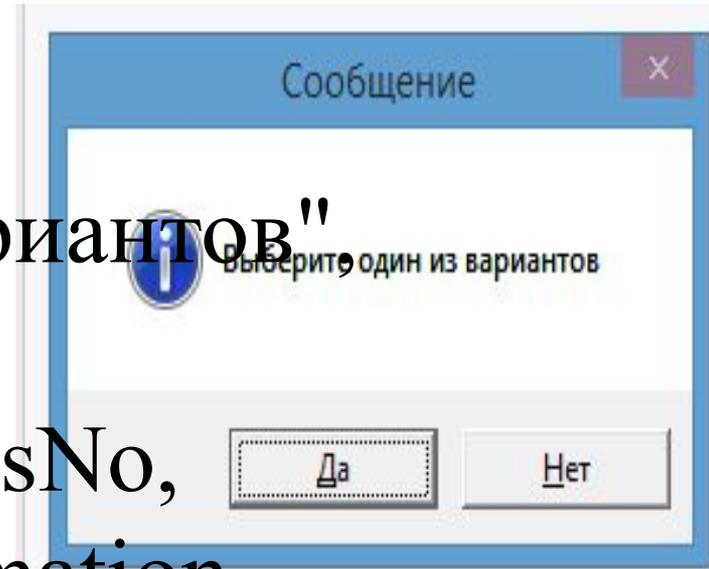
ЗАДАНИЕ:

Выполнить практическую
работу, посмотреть файл
(Календарь с#)

Окно сообщения MessageBox

для вывода сообщений
применяется элемент
MessageBox.

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show(
        "Выберите один из вариантов",
        "Сообщение",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1,
        MessageBoxOptions.DefaultDesktopOnly);
}
```



Для вывода сообщения в классе
MessageBox предусмотрен
метод **Show**, который имеет
различные версии и может
принимать ряд параметров.

```
MessageBox.Show(  
    string text,  
    string caption,  
    MessageBoxButtons buttons,  
    MessageBoxIcon icon,  
    MessageBoxDefaultButton  
defaultButton,  
    MessageBoxOptions options  
)
```

```
MessageBox.Show(  
    string text, //текст сообщения  
    string caption,  
    MessageBoxButtons buttons,  
    MessageBoxIcon icon,  
    MessageBoxDefaultButton  
defaultButton,  
    MessageBoxOptions options  
)
```

```
MessageBox.Show(  
    string text,  
    string caption, //текст заголовка окна  
сообщения  
    MessageBoxButtons buttons,  
    MessageBoxIcon icon,  
    MessageBoxDefaultButton  
defaultButton,  
    MessageBoxOptions options  
)
```

```
MessageBox.Show(  
    string text,  
    string caption,  
    MessageBoxButtons buttons, //кнопки,  
используемые в окне сообщения.  
    MessageBoxIcon icon,  
    MessageBoxDefaultButton defaultButton,  
    MessageBoxOptions options  
)
```

Принимает одно из значений перечисления :

- **AbortRetryIgnore**: три кнопки Abort (Отмена), Retry (Повтор), Ignore (Пропустить)
- **ОК**: одна кнопка ОК
- **ОКCancel**: две кнопки ОК и Cancel (Отмена)
- **RetryCancel**: две кнопки Retry (Повтор) и Cancel (Отмена)
- **YesNo**: две кнопки Yes и No
- **YesNoCancel**: три кнопки Yes, No и Cancel (Отмена)

```
MessageBox.Show(  
    string text,  
    string caption,  
    MessageBoxButtons buttons,  
    MessageBoxIcon icon, //значок окна  
сообщения.  
    MessageBoxButton  
    defaultButton,  
    MessageBoxOptions options  
)
```

Может принимать одно из следующих значений перечисления:

- **Asterisk, Information:**

значок, состоящий из буквы i в нижнем регистре, помещенной в кружок

- **Error, Hand, Stop:**

значок, состоящий из белого знака "X" на круге красного цвета.

- **Exclamation, Warning:**

значок, состоящий из восклицательного знака в желтом треугольнике

- **Question:**

значок, состоящий из вопросительного знака на периметре круга

- **None:**

значок у сообщения отсутствует

```
MessageBox.Show(  
    string text,  
    string caption,  
    MessageBoxButtons buttons,  
    MessageBoxIcon icon,  
    MessageBoxDefaultButton  
defaultButton, //кнопка, на которую по  
умолчанию устанавливается фокус.  
    MessageBoxOptions options  
)
```

Принимает одно из значений
перечисления, которые задаются
перечислением MessageVoxButtons

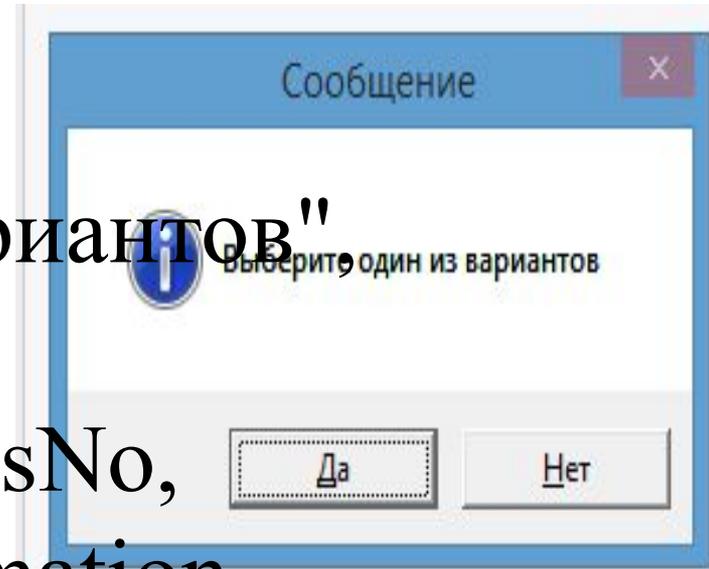
- Button1: первая кнопка
- Button2: вторая кнопка
- Button3: третья кнопка

```
MessageBox.Show(  
    string text,  
    string caption,  
    MessageBoxButtons buttons,  
    MessageBoxIcon icon,  
    MessageBoxDefaultButton  
defaultButton,  
    MessageBoxOptions options  
// параметры окна сообщения.  
)
```

Принимает одно из значений перечисления :

- **DefaultDesktopOnly**: окно сообщения отображается на активном рабочем столе.
- **RightAlign**: текст окна сообщения выравнивается по правому краю
- **RtlReading**: все элементы окна располагаются в обратном порядке справа налево

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show(
        "Выберите один из вариантов",
        "Сообщение",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1,
        MessageBoxOptions.DefaultDesktopOnly);
}
```



```
if (result == DialogResult.Yes)
    button1.BackColor=Color.Red;
```

Метод `MessageBox.Show` возвращает объект, с помощью которого мы можем узнать, какую кнопку в окне сообщения нажал пользователь. `DialogResult` представляет перечисление, в котором определены следующие значения:

- **Abort**: нажата кнопка Abort (отмена)
- **Retry**: нажата кнопка Retry (повтор)
- **Ignore**: нажата кнопка Ignore (пропустить)
- **OK**: нажата кнопка ОК
- **Cancel**: нажата кнопка Cancel (отмена)
- **None**: отсутствие результата