Программирование на C++

Минцифры России

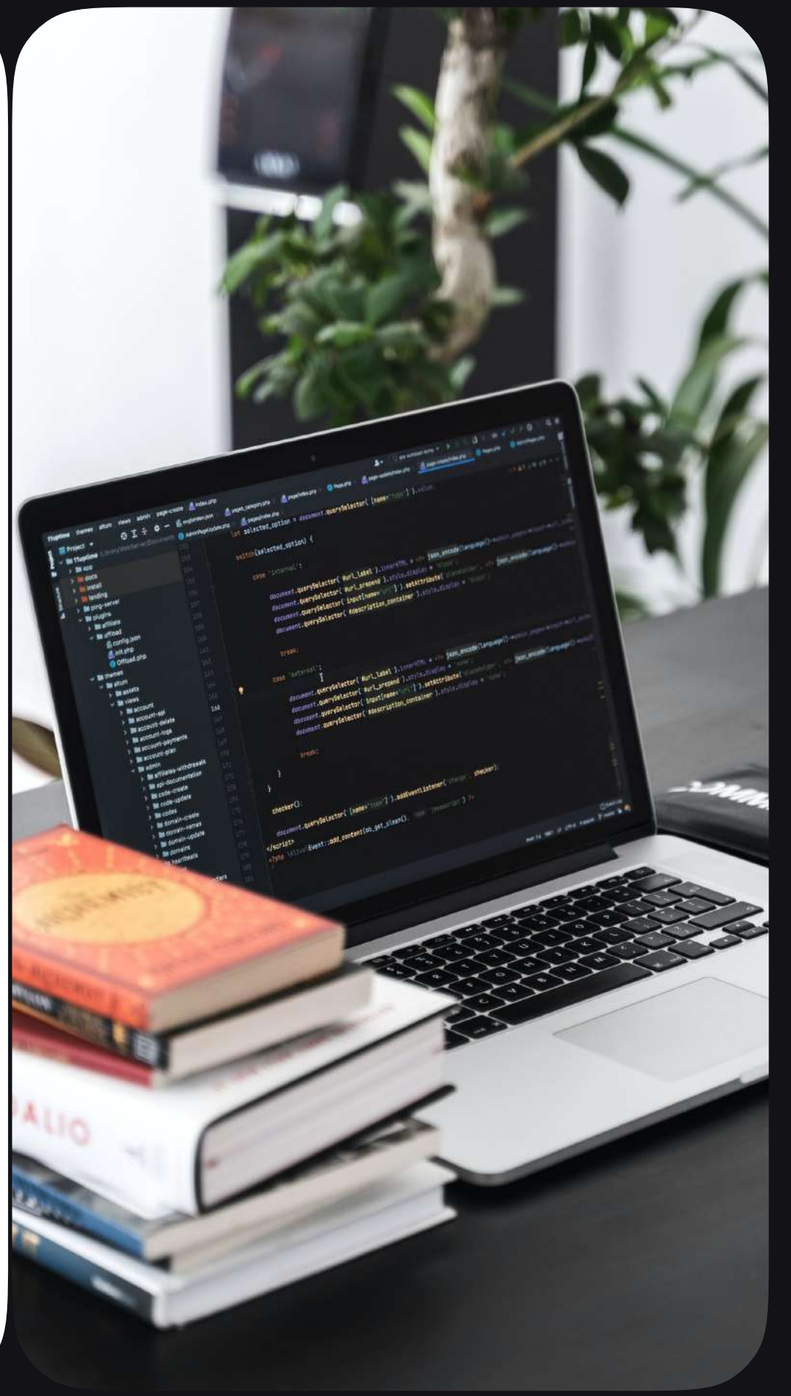
UCHi DOMA

20.35 УНИВЕРСИТЕТ

Урок 6 Модуль 4

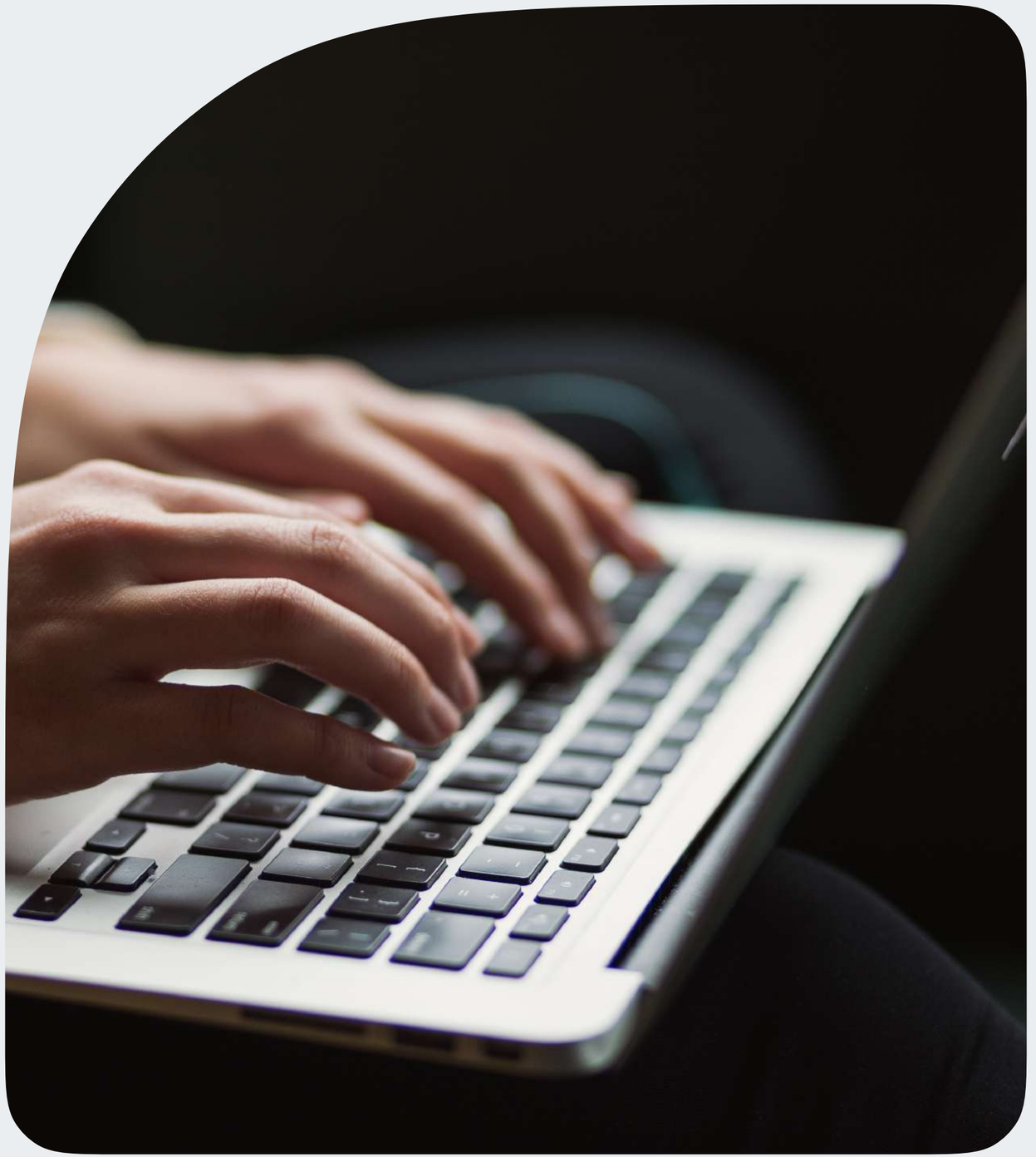
Визуальные компоненты. Продолжение

Полезные материалы

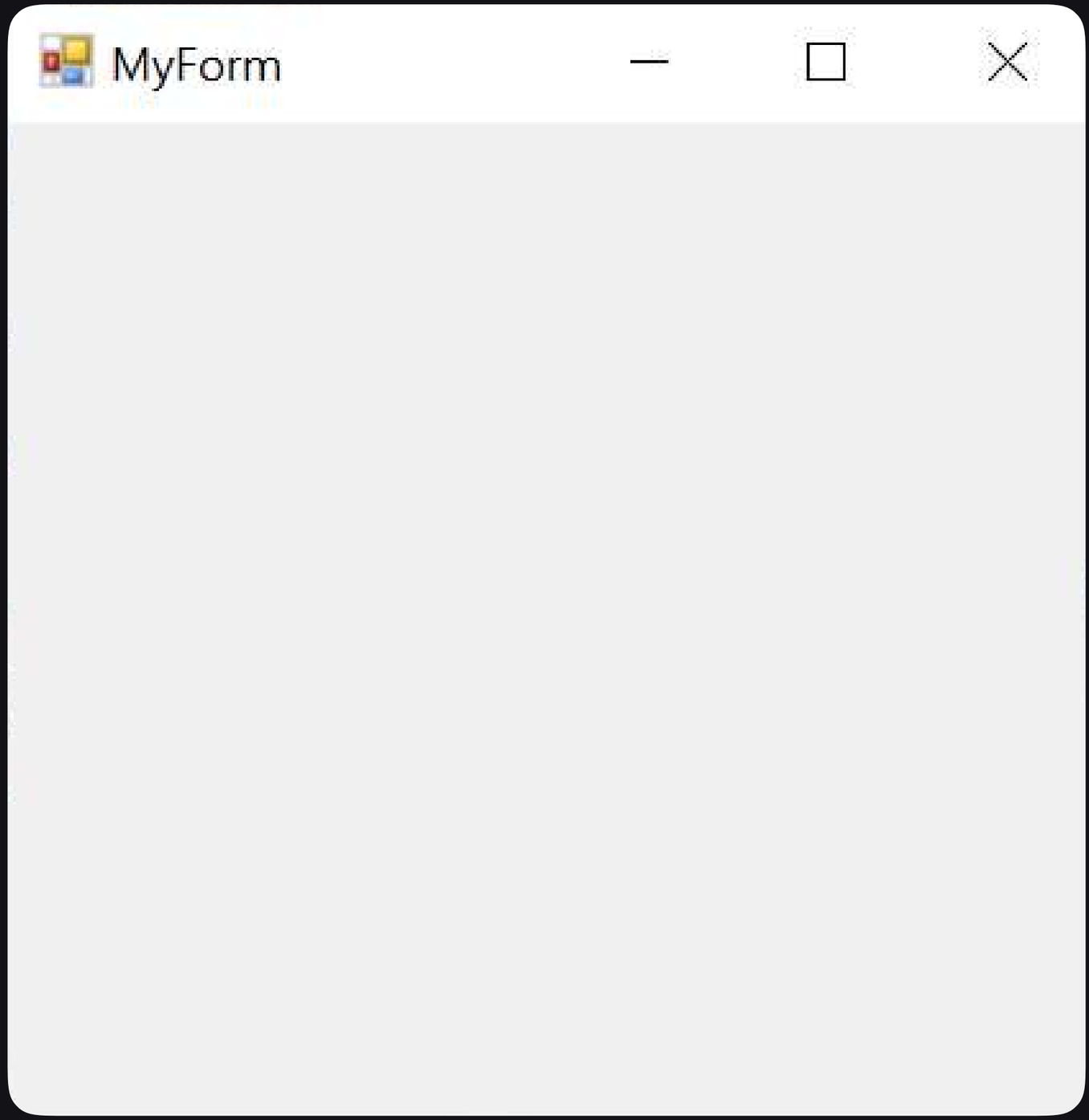


Цели урока

- Познакомится с инструментами проектирования графического интерфейса
- Отработать на практике написание приложений с графическим интерфейсом на C++



На прошлом уроке мы создали приложение с одним пустым окном.



Файл MyForm.h

```
#pragma once
namespace WinApp1 {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    public ref class MyForm : public System::Windows::Forms::Form
    {
    public:
        MyForm(void)
        {
            InitializeComponent();
        }
    protected:
        ~MyForm()
         {
             if (components)
             {
                delete components;
             }
         }
    private:
         System::ComponentModel::Container ^components;
#pragma region Windows Form Designer generated code
         void InitializeComponent(void)
         {
             this->components = gcnew System::ComponentModel::Container();
             this->Size = System::Drawing::Size(300,300);
             this->Text = L"MyForm";
             this->Padding = System::Windows::Forms::Padding(0);
             this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
          }
#pragma endregion
    };
}
```

Файл MyForm.cpp

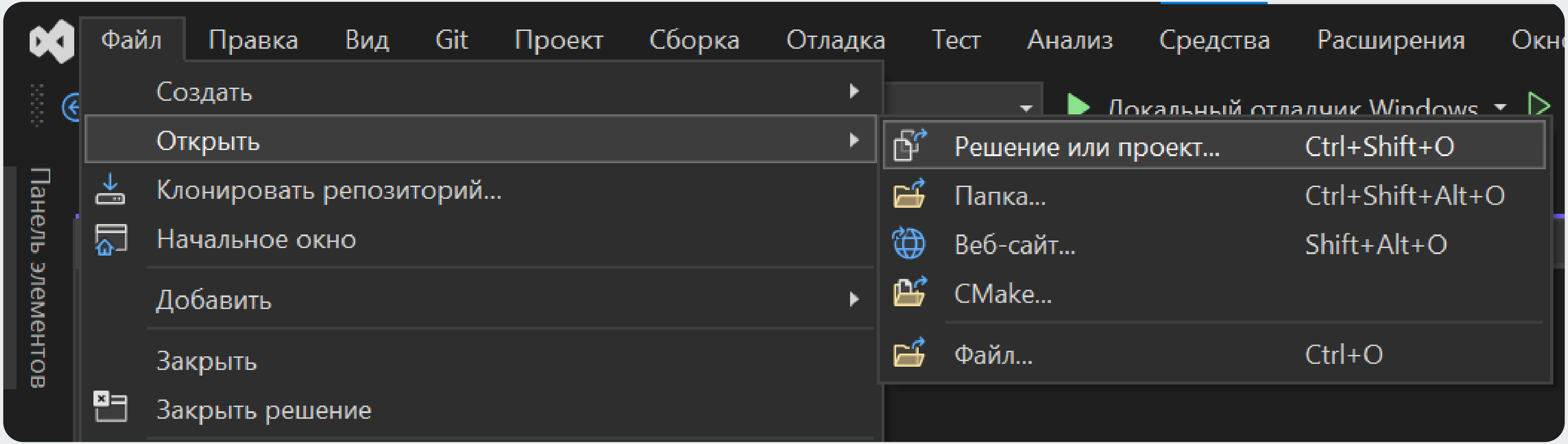
```
#include "MyForm.h"

using namespace System;
using namespace System::Windows::Forms;

[STAThreadAttribute]

int main(array<String^>^ args)
{
    Application::SetCompatibleTextRenderingDefault(false);
    Application::EnableVisualStyles();
    WinApp1::MyForm form;
    Application::Run(% form);

}
```

Установка WinForms

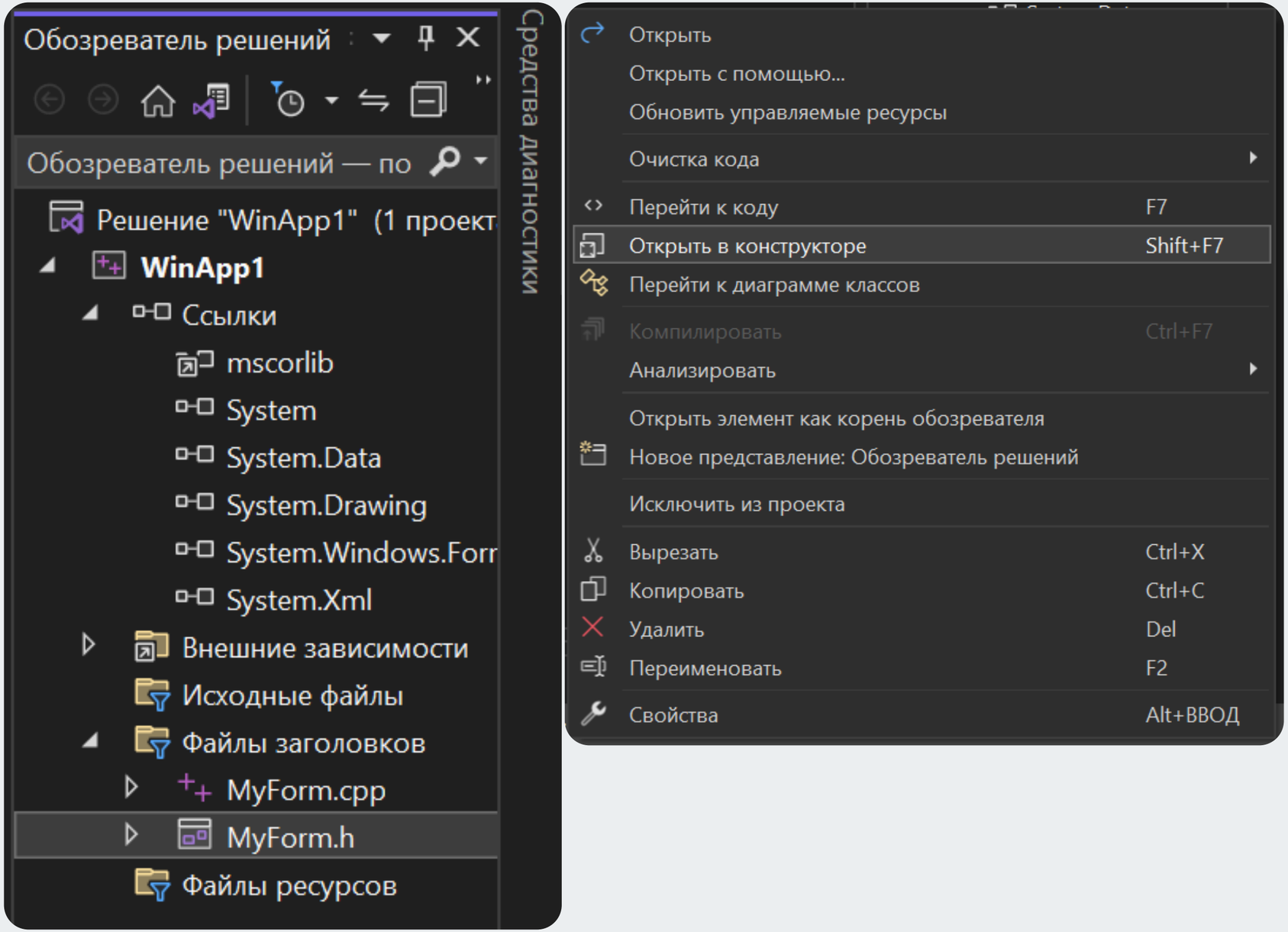
Будем работать в MS Visual Studio.

Запустите Visual Studio.

Открываем проект

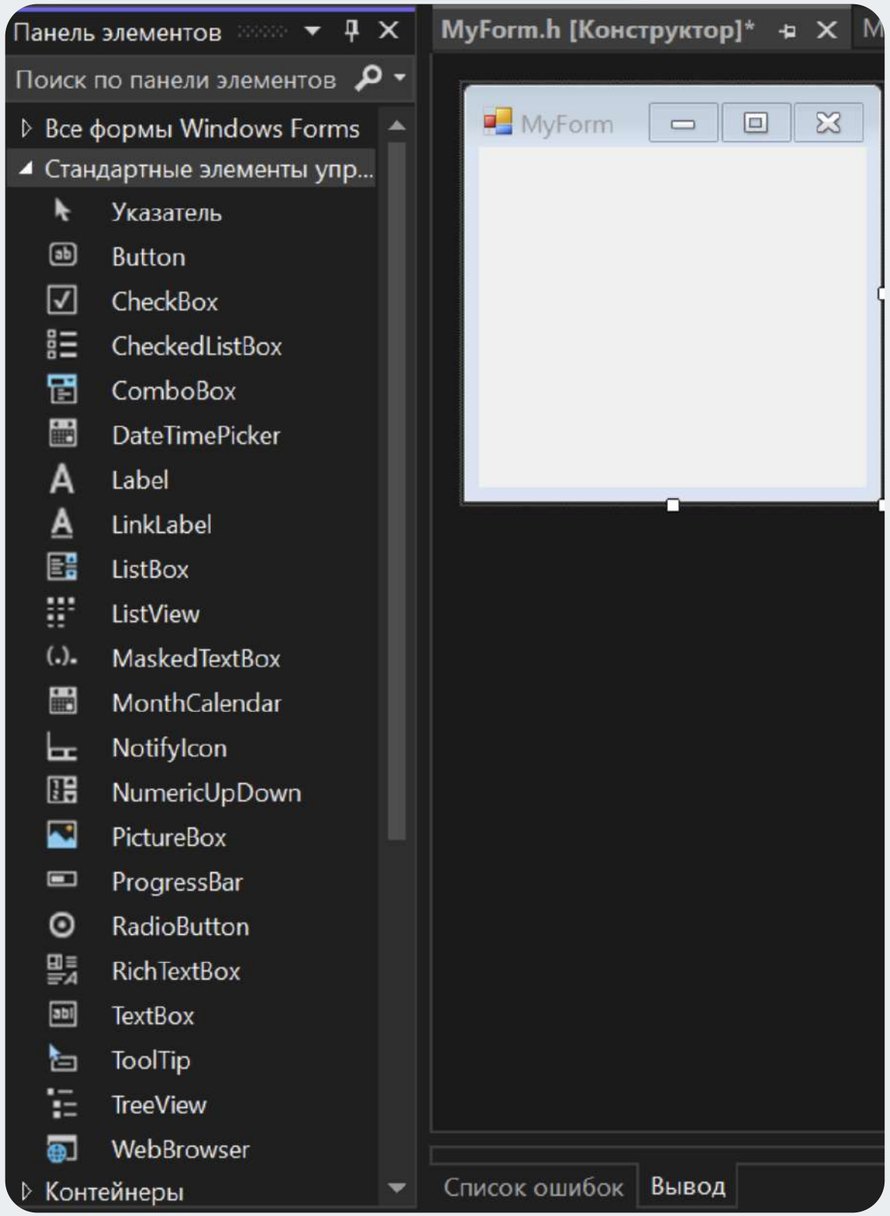


Откройте конструктор



Два раза кликните на файл MyForm.h или кликните правой кнопкой мыши и выберите «Открыть в конструкторе»

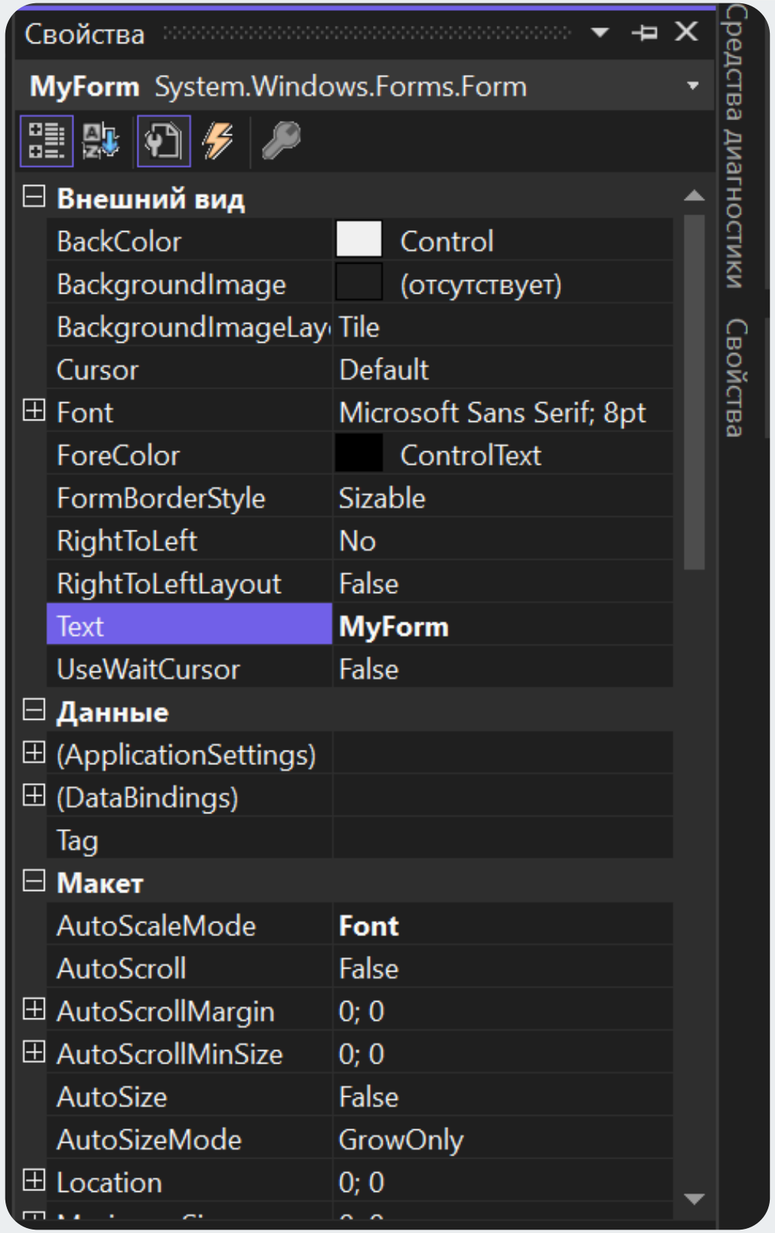
Конструктор



В панели инструментов отображаются все визуальные компоненты.

Компоненты добавляются перетаскиванием мышкой на форму.

Свойства компонентов



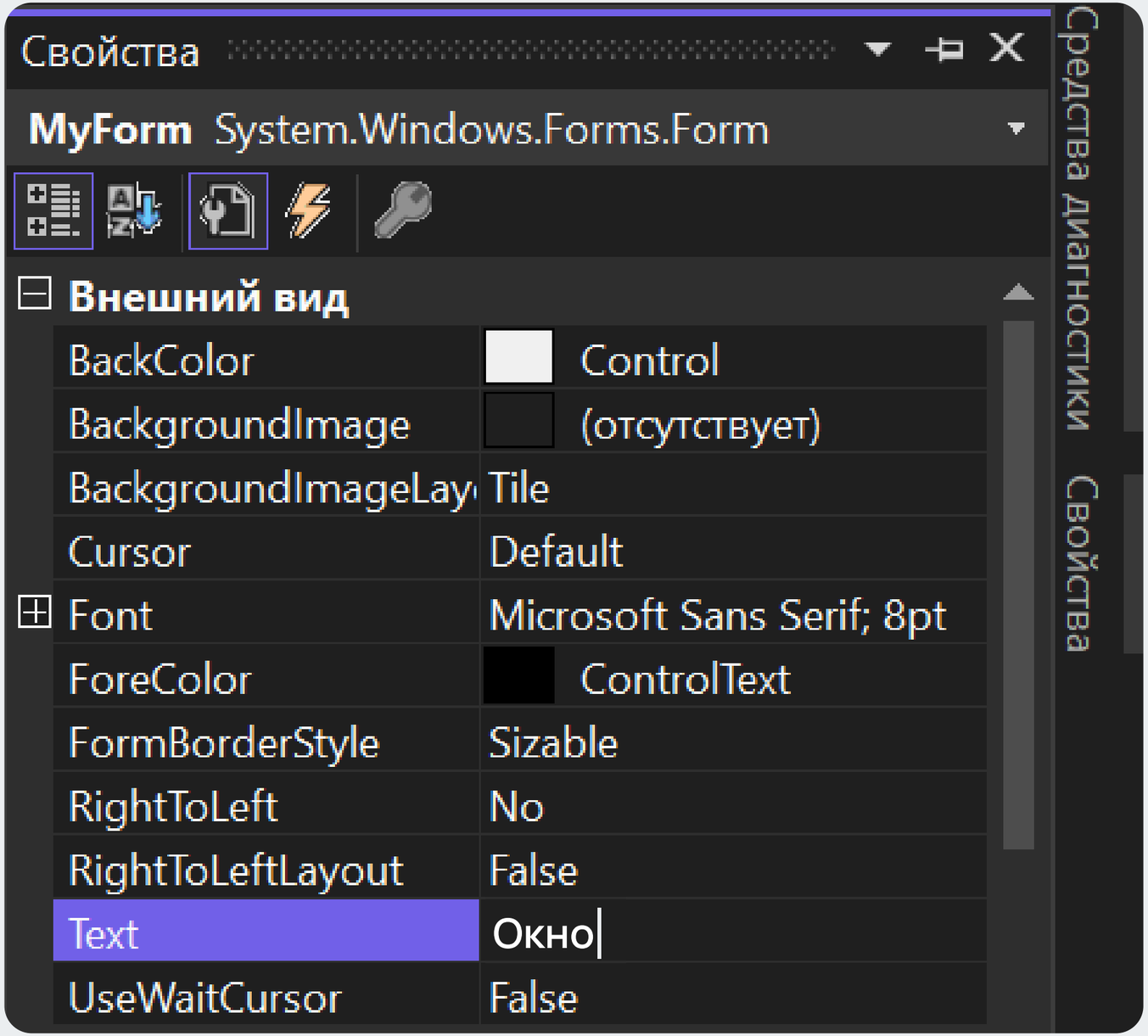
На вкладке «Свойства» отображаются и настраиваются свойства выбранного в редакторе объекта.

Настроим форму

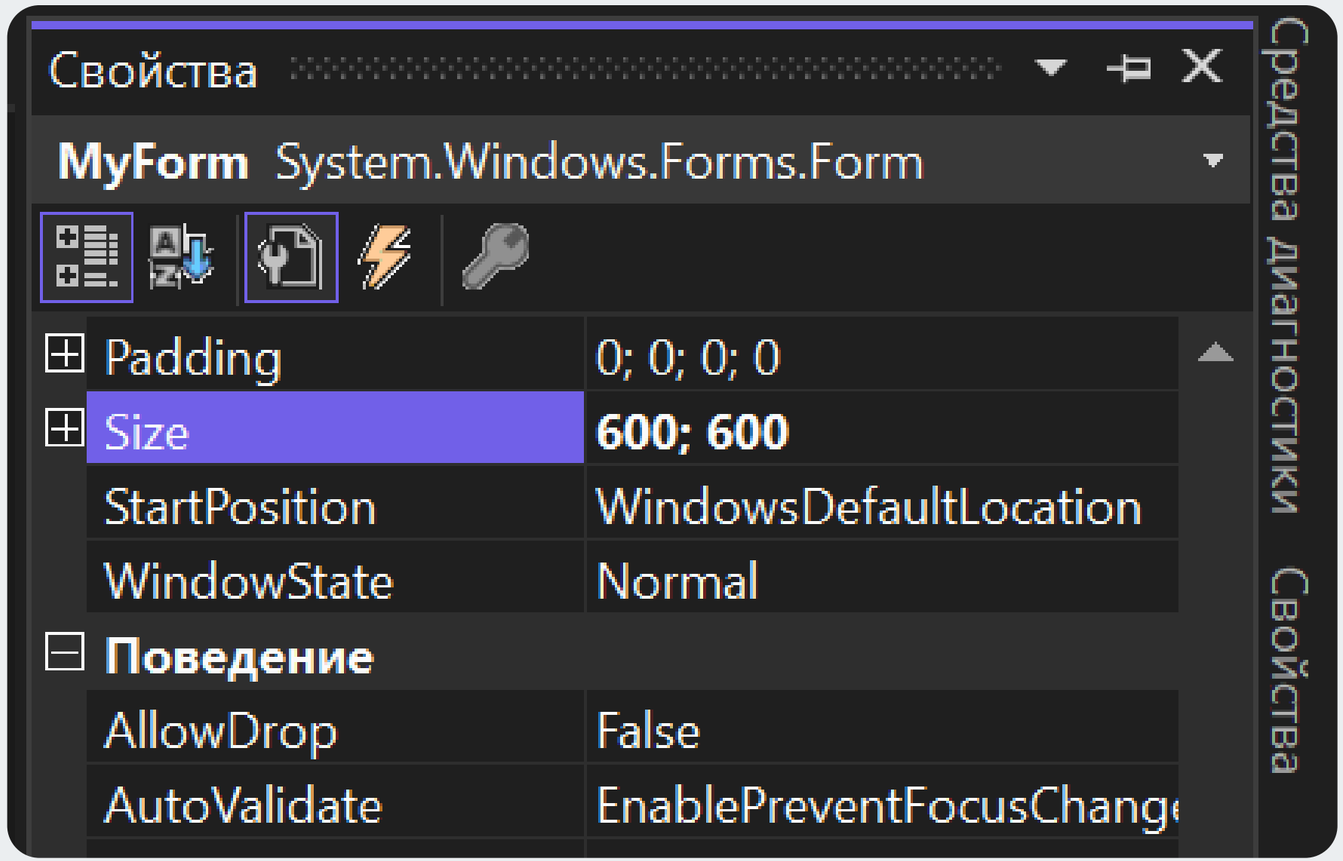
Выберите в конструкторе форму (кликните на неё), установите свойства:

- Заголовок окна (Text) «Окно»
- Размер окна 600x600
- Установите цвет окна (BackColor)
- Установите свою иконку (Icon)

Меняем заголовок окна



Меняем размер окна

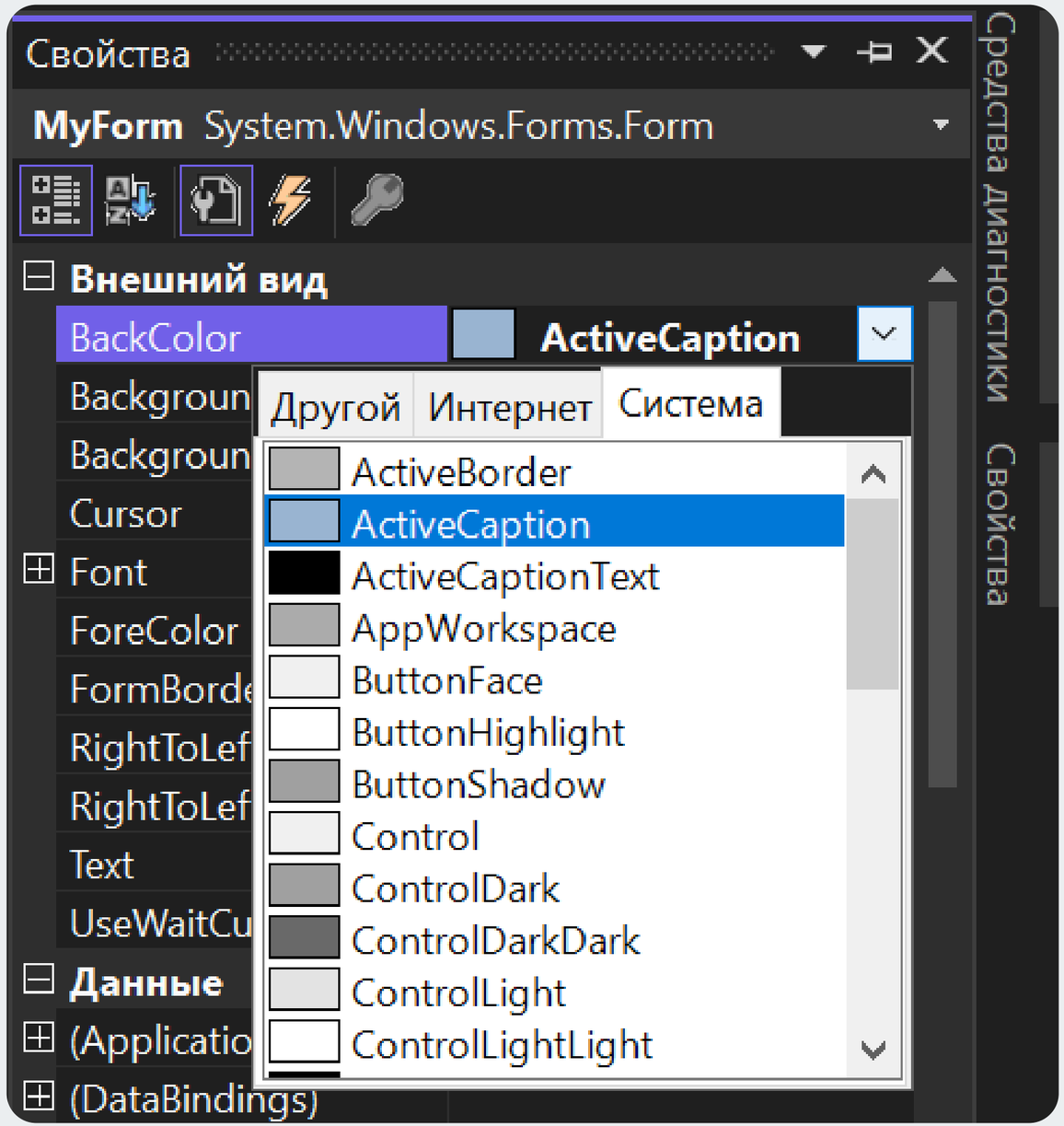


Размер окна можно поменять в коде файла MyForm.h.

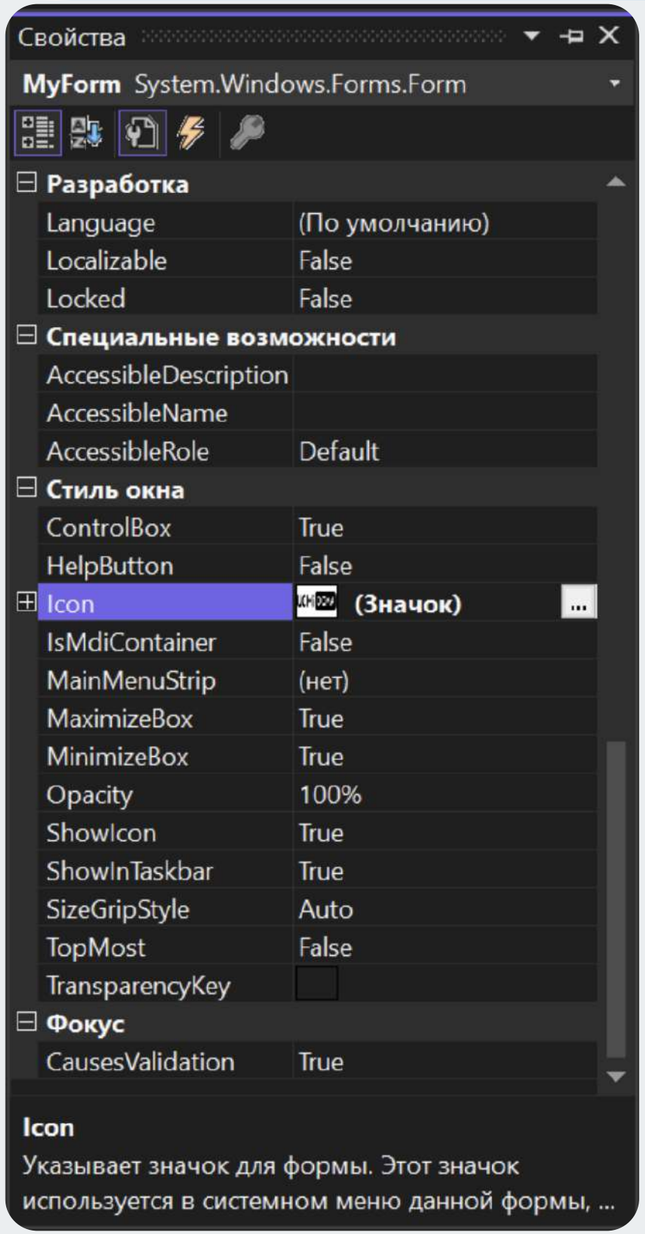
В конструкторе — растянув окно мышкой до нужных размеров.

В овне «Свойства» — задав нужные значения свойству Size.

Меняем цвет окна

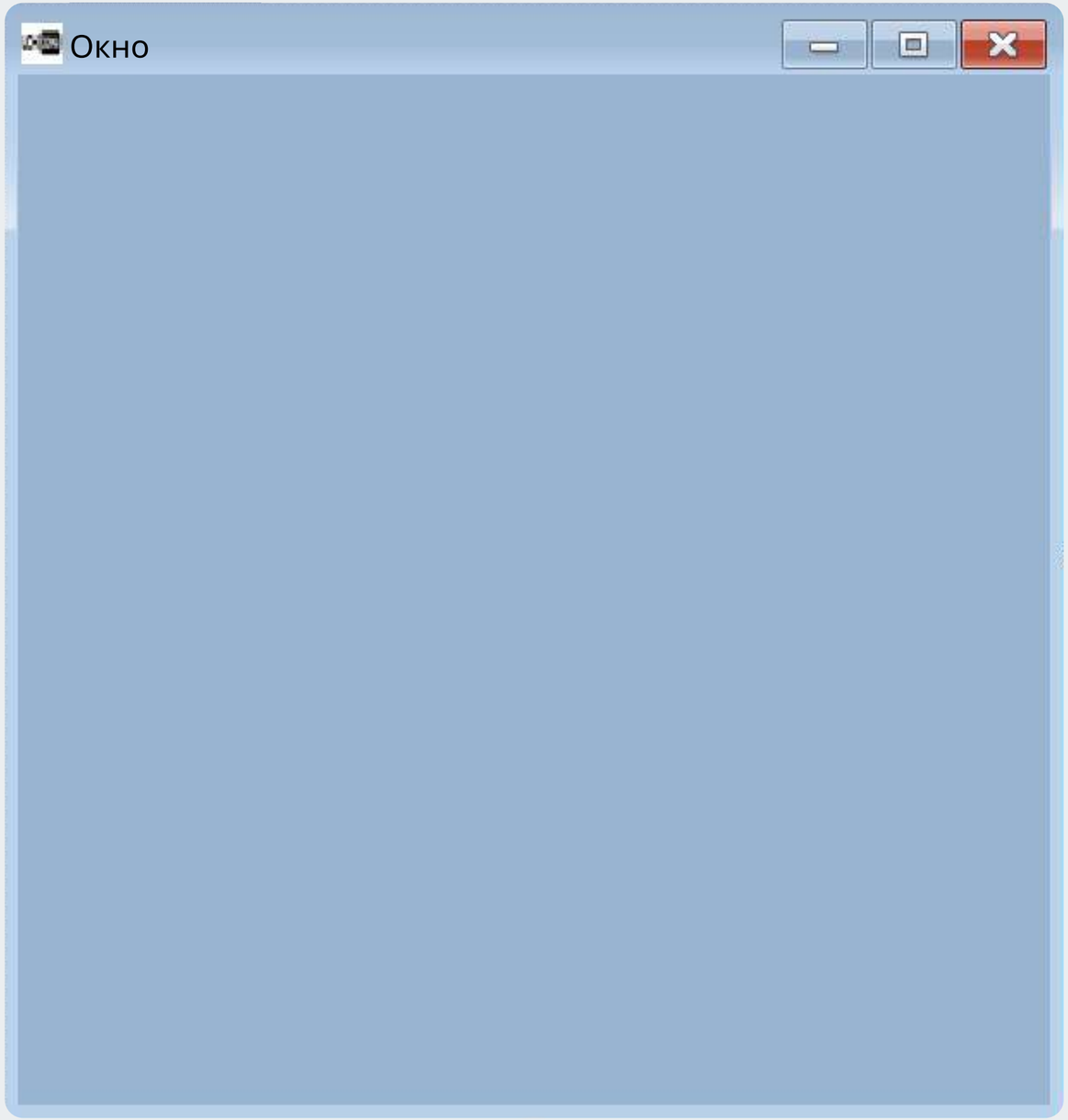


Меняем иконку



В свойстве Icon выберите новый файл иконки (.ico)

Получилось окно

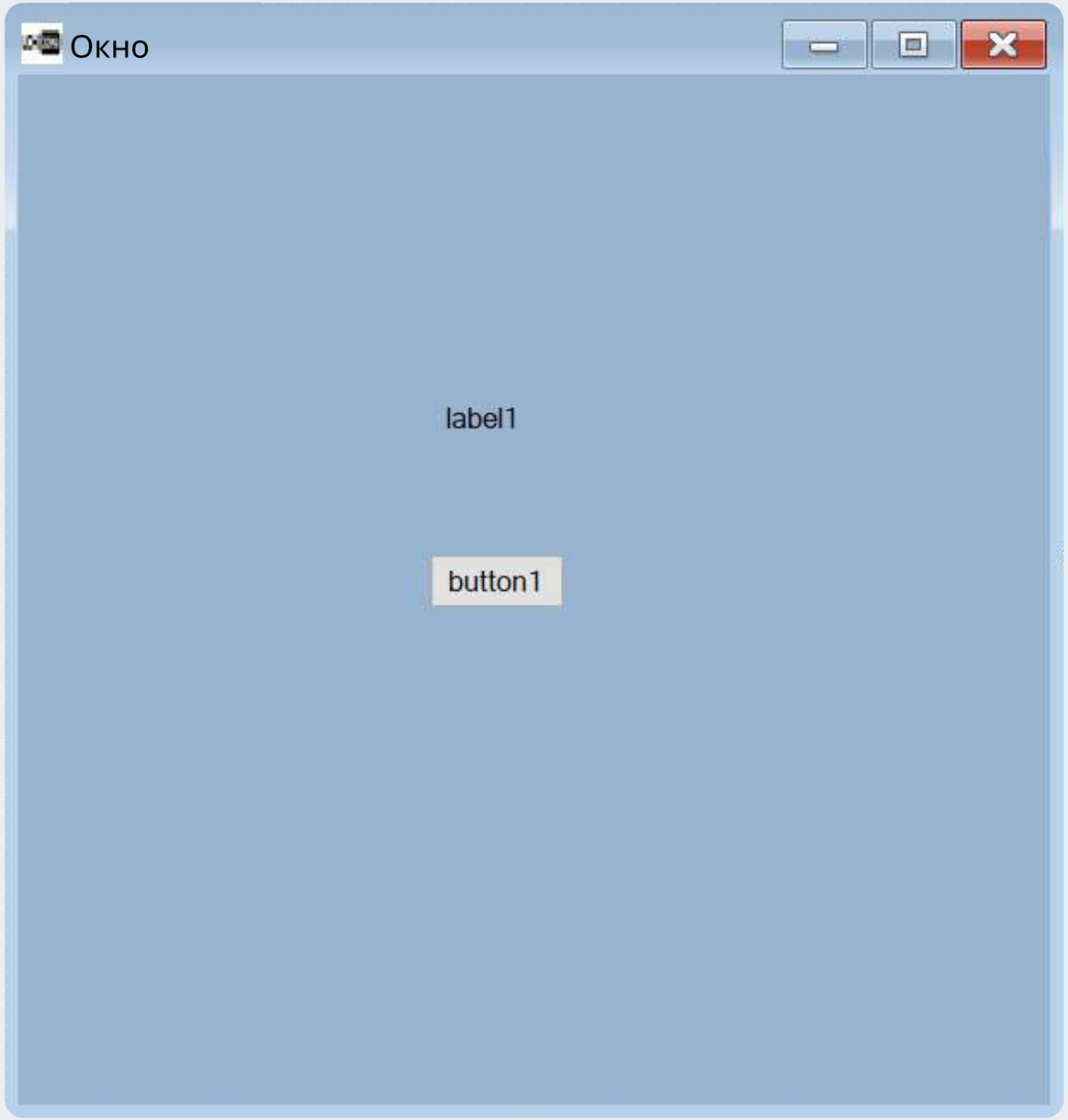


Все изменения в окне «Свойства»
и конструкторе отражаются в коде.

Откройте файл MyForm.h
и посмотрите как изменился код.

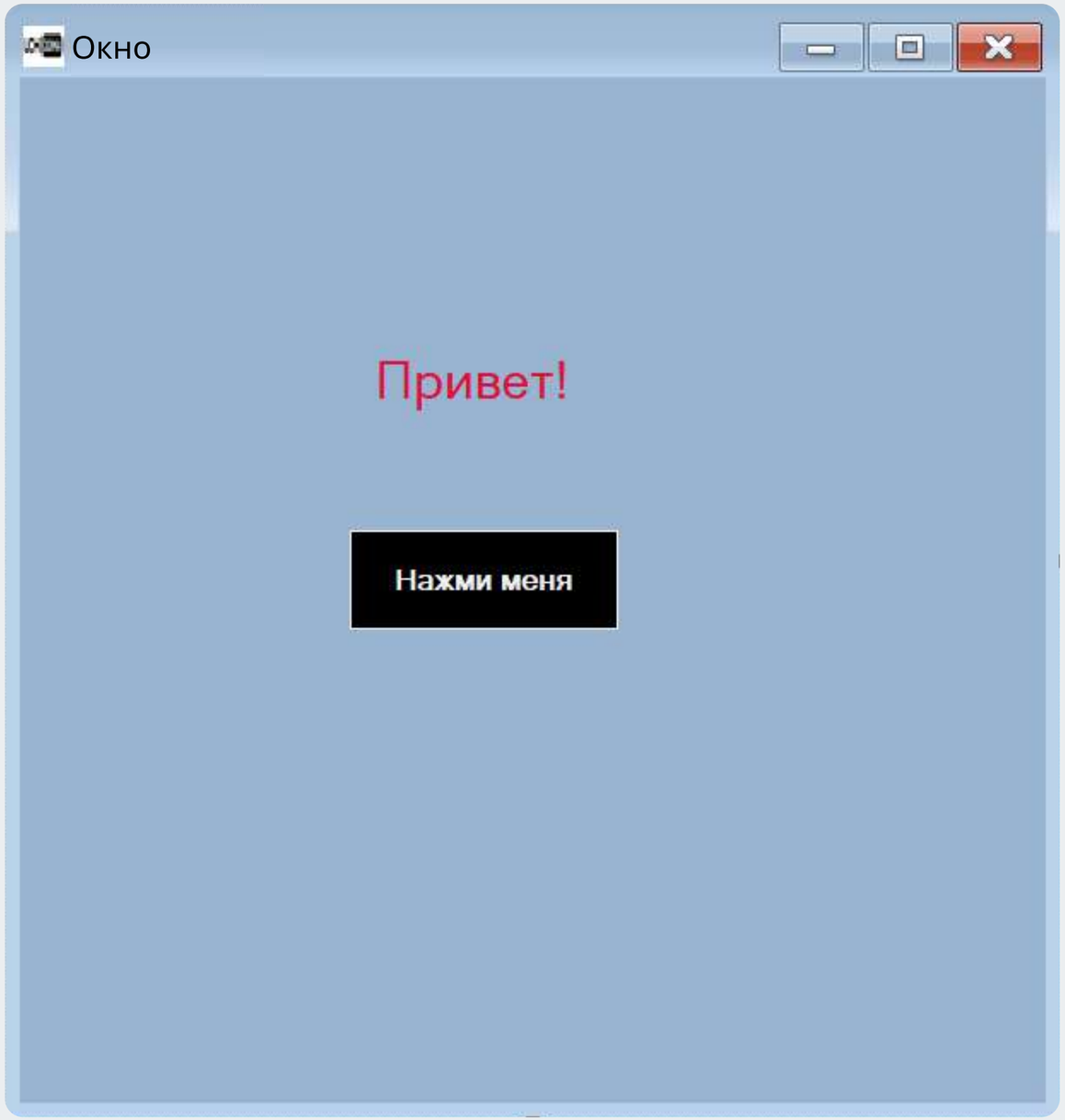
```
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->BackColor = System::Drawing::SystemColors::ActiveCaption;
this->ClientSize = System::Drawing::Size(600, 600);
this->Icon = (cli::safe_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));
this->Name = L"MyForm";
this->Text = L"Оно";
```

Добавим компоненты



Добавьте на форму кнопку (button) и текстовое поле — метку(Label).

Настройте кнопку и метку

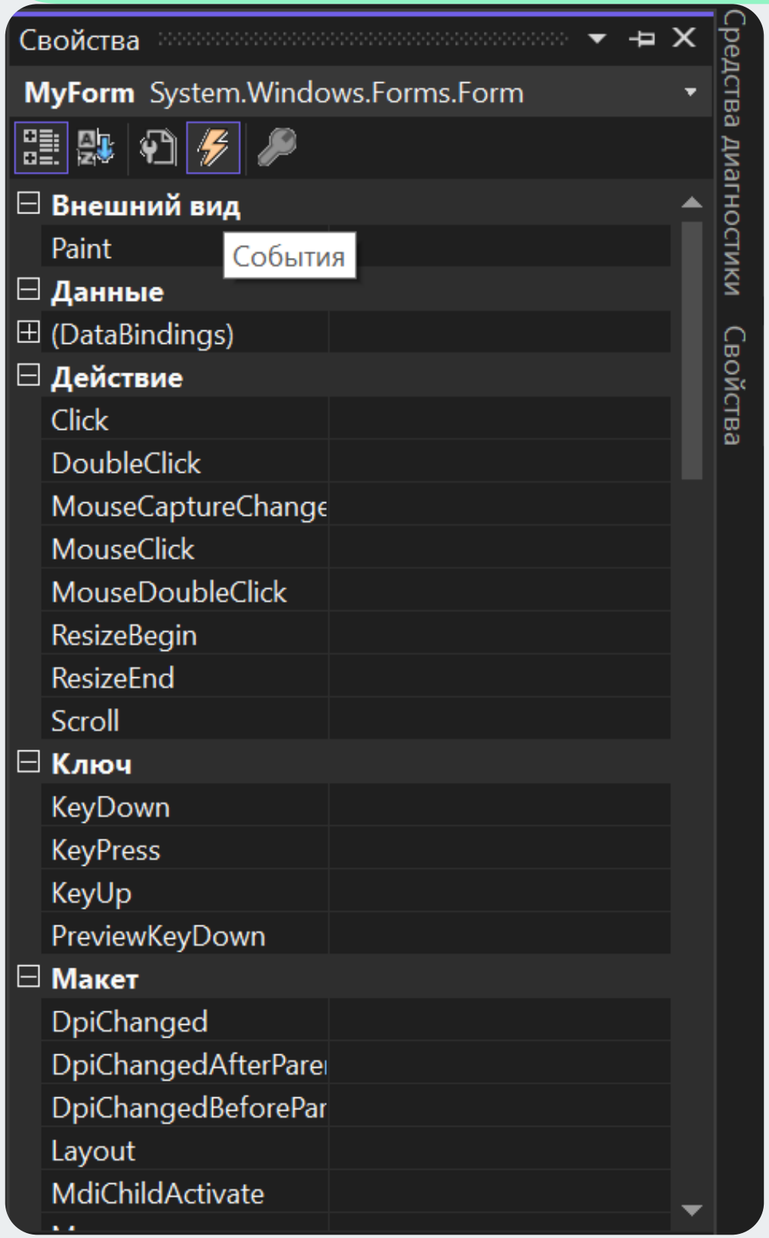


Поменяйте размер,цвет, шрифт, цвет текста.

Добавим событие

Пусть при нажатии на кнопку меняется текст в метке.

Добавим событие



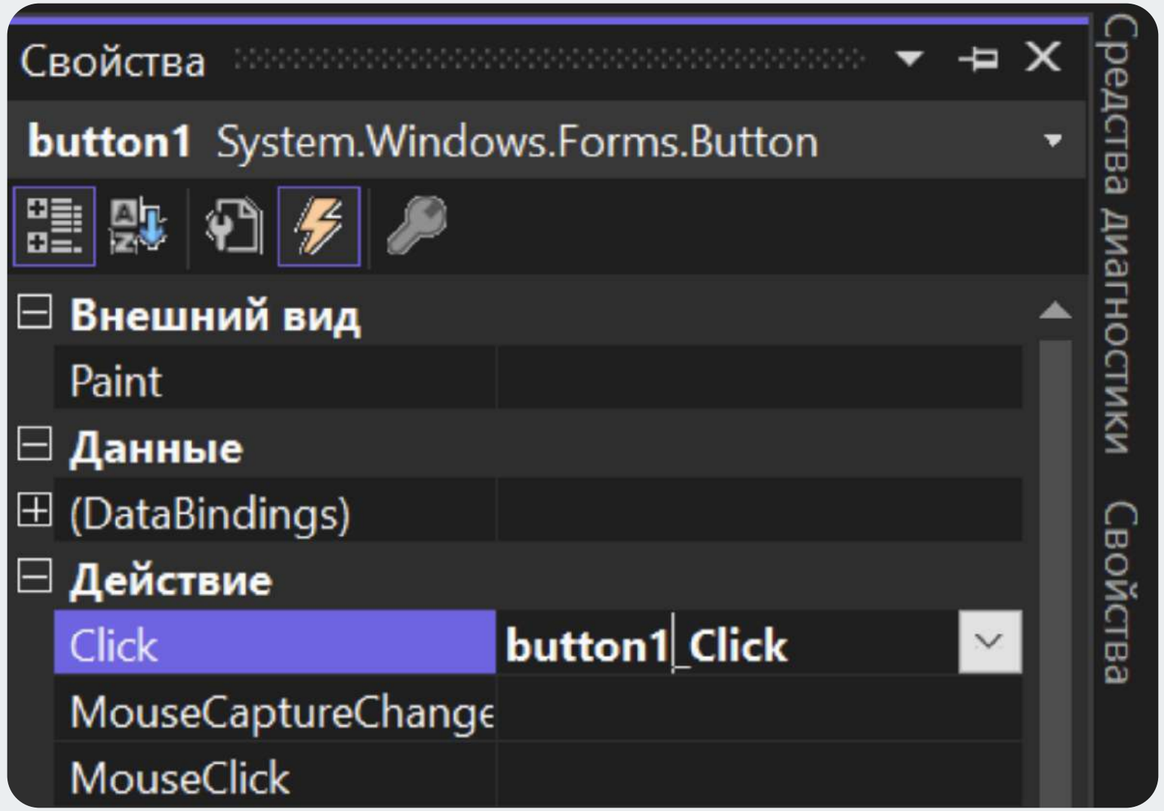
В окне «Свойства» перейдите в раздел «События».

Здесь отображаются все события для выбранного объекта.

Добавим событие «Клик по кнопке»

Дважды кликните по событию «Click»

Так же событие «Click» можно добавить дважды кликнув на компонент в конструкторе.



В файл MyForm.h добавлена функция

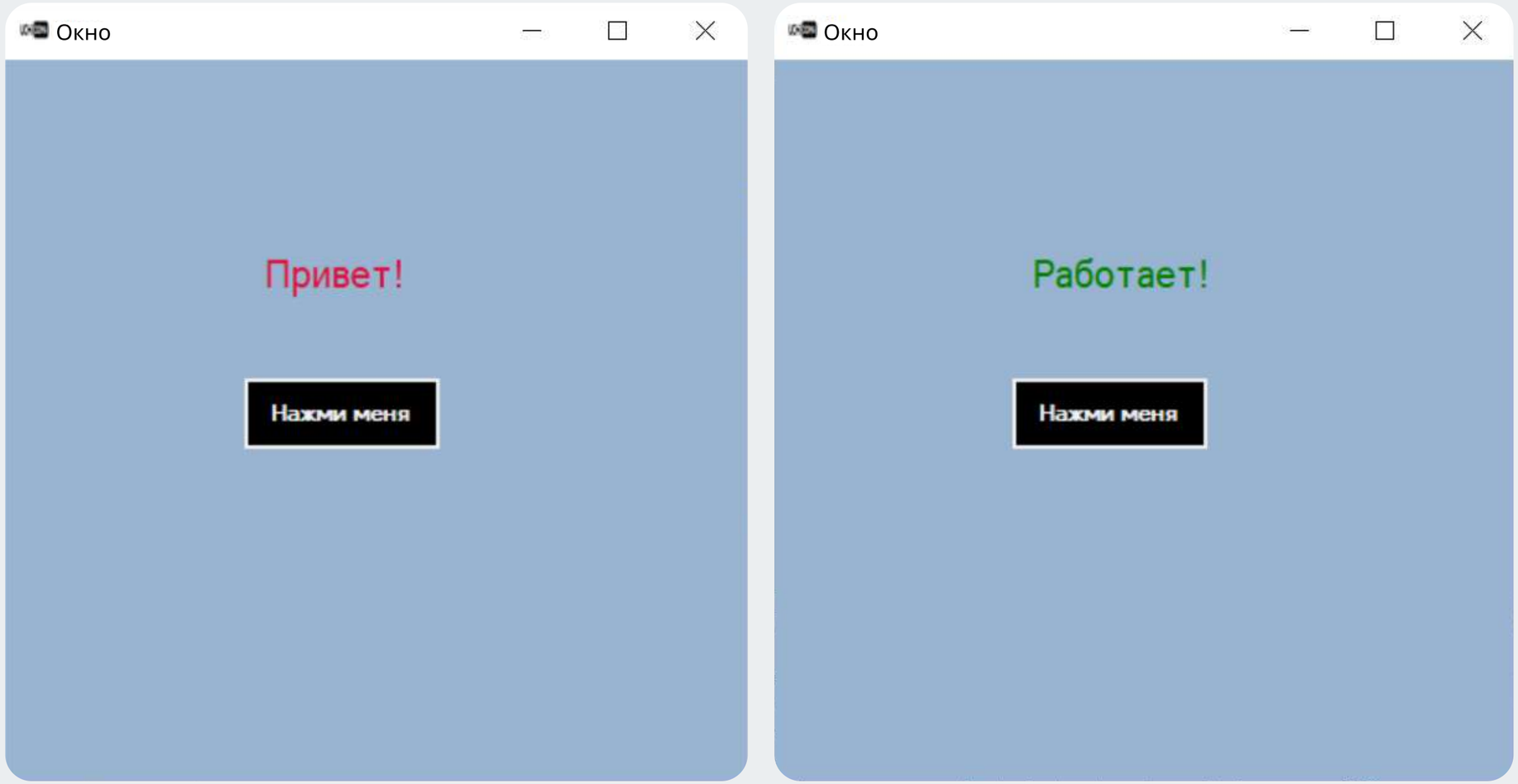
```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
 }
 };
```

Обработаем событие click для button1

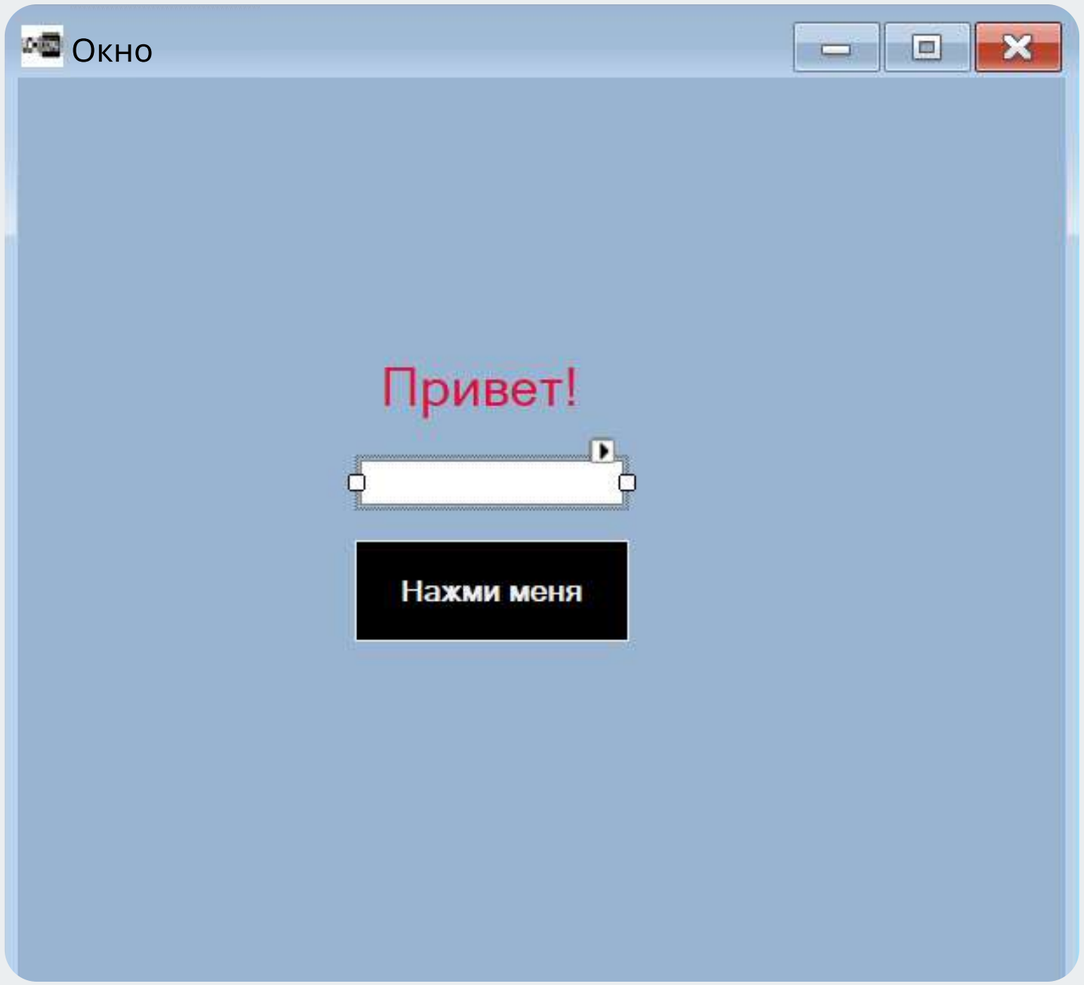
При клике на кнопку (button1) будем менять свойство text у объекта label1 и цвет текста.

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
 {
  this->label1->Text = "Работает!";
  this->label1->ForeColor = Color::Green;
 }
```

Запустите проект, нажмите на кнопку



Добавим поле для ввода текста (textbox)



Изменим метод button1_click

При клике на кнопку (button1) будем менять свойство text у объекта label1 и цвет текста.

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
 {
  this->label1->Text = this->textBox1->Text;
  this->label1->ForeColor = Color::Green;
 }
```

Запустите проект, введите текст в textbox и нажмите кнопку.

Введенный текст должен отобразится в метке (label1).