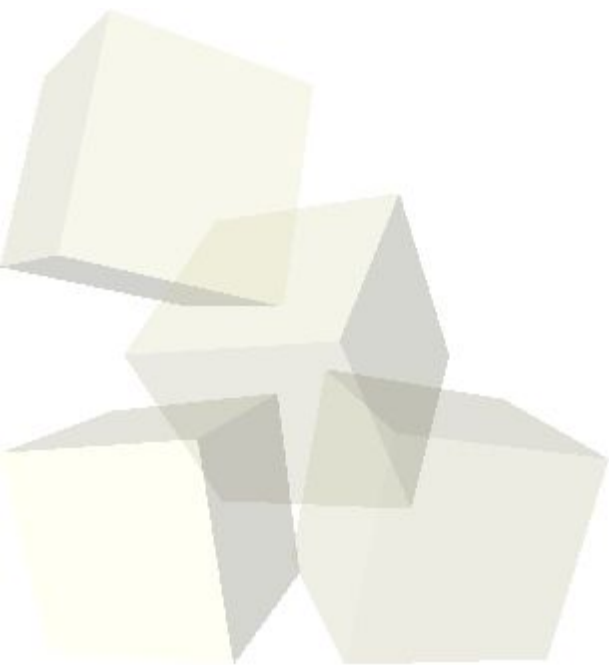




**Использование простого класса на C++  
в программе на Managed C++ с графическим  
интерфейсом пользователя (GUI)  
в виде одной формы Windows Forms  
(Visual Studio 2012+)**





**Managed C++** - версия C++ от компании Microsoft, поддерживающая создание приложений для .NET. Объекты, создаваемые в managed C++ оператором **gcnew**, автоматически высвобождаются сборщиком мусора (Garbage Collector) .NET. Код на managed C++ имеет доступ к библиотекам .NET, в частности, к Windows Forms.



# Программирование

К сожалению, среди стандартных шаблонов проектов C++ начиная с VS2012 **нет** проекта CLR - Windows Forms, поэтому можно предложить 2 варианта :

- 1) Создать пустой проект C++/CLR + потом в него добавить New – UI – Windows Forms и вручную – код функции main() 😞
- 2) Создать обертку для C++ класса на Managed C++, и использовать ее в проекте Windows Forms на C#



# Программирование

Использование простого класса C++ в программе .NET с GUI

## 1 Объявление класса (файл Triangle.h)

```
#include <iostream>

#pragma once

namespace ThreeAngle {

    class Rect3Angle {

        double a, b;

    public :

        Rect3Angle(double _a, double _b);
        const double getA() { return a; };
        const double getB();
        const double getC();

        const double getArea(bool rect);
        const double getPerimeter();

    };
}
```

Поля (закрытые)

Конструктор

Методы доступа  
(const)

Методы



# Программирование

Использование простого класса C++ в программе .NET с GUI

## 2 Определение класса (файл Triangle.cpp)

```
#include <math.h>
#include "triangle.h"
```

Конструктор

```
namespace ThreeAngle {
```

```
    Rect3Angle::Rect3Angle(double _a, double _b) {
        a = _a; b = _b; }

```

```
    double const Rect3Angle::getArea(bool rect) {
        if (rect) return a * b;
        else return a * b / 2;
    }

```

Метод вычисления  
площади

```
    double const Rect3Angle::getC() {
        return sqrt(a * a + b * b);
    }

```

Расчетный  
метод доступа

```
    double const Rect3Angle::getPerimeter() {
        return a + b + getC();
    }

```

Метод расчета  
периметра

```
    double const Rect3Angle::getB() {
        return b;
    }
}

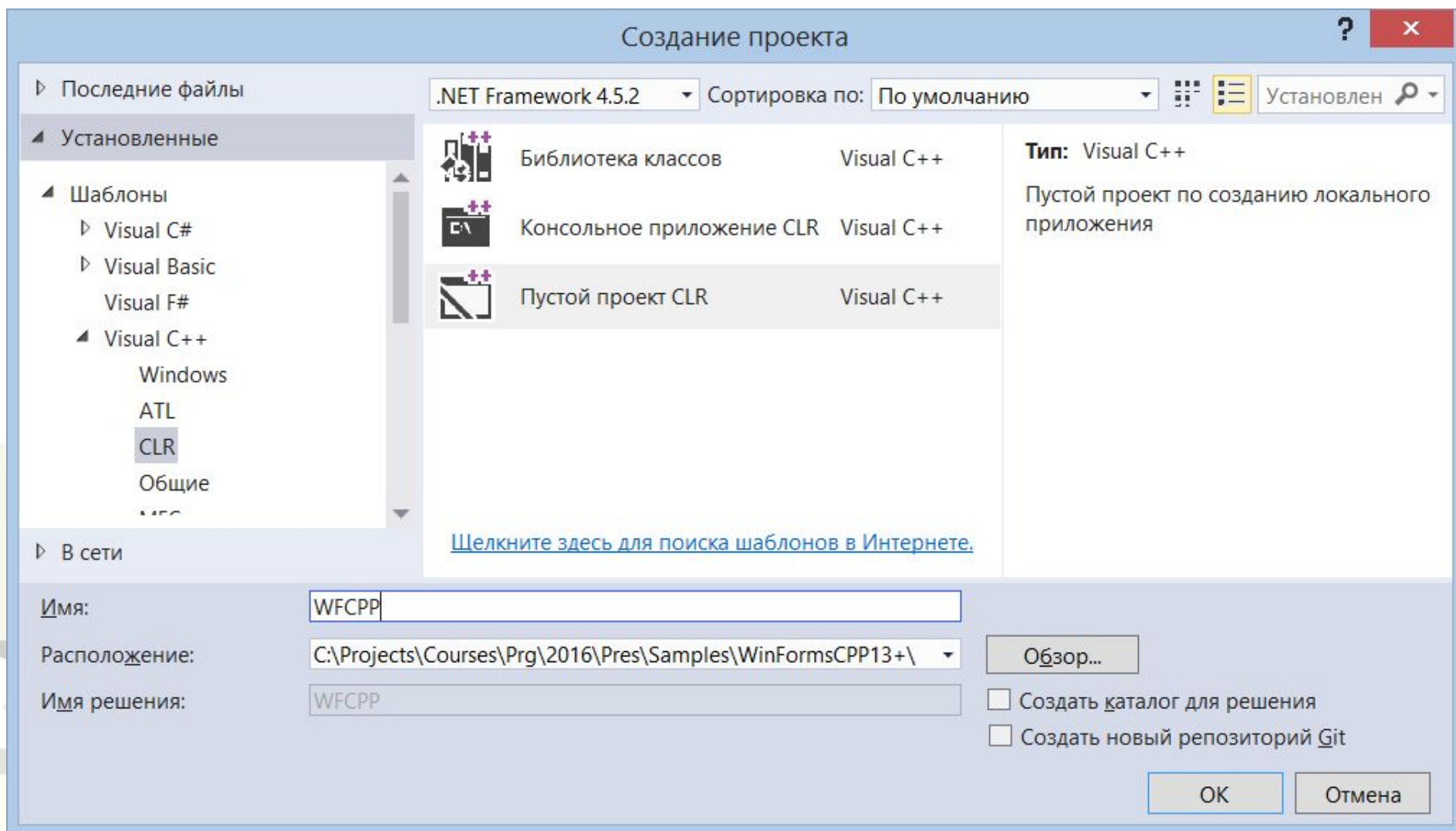
```



# Программирование

Использование простого класса C++ в программе .NET с GUI

## 3 Создаем новый проект C++ - CLR - Empty

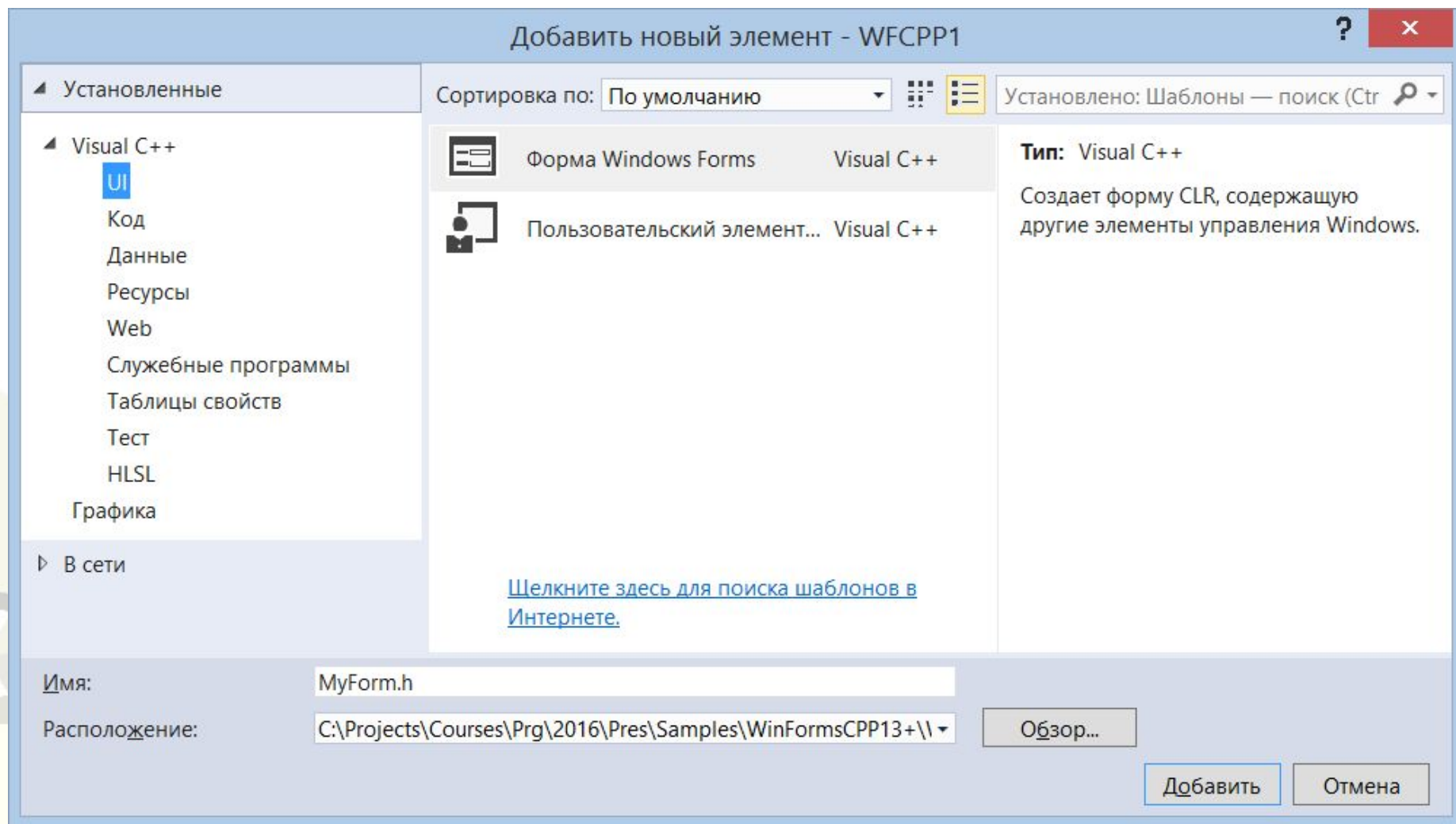




# Программирование

Использование простого класса C++ в программе .NET с GUI

## 3 Добавляем форму – Add New – Visual C++ – UI – Windows Form





# Программирование

Использование простого класса C++ в программе .NET с GUI

## 3 Настраиваем свойства проекта

Страницы свойств WFCPP1

Конфигурация: Активная (Debug) Платформа: Активная (Win32) Диспетчер конфигураций...

- Свойства конфигурации
  - Общие
  - Отладка
  - Каталоги VC++
  - C/C++
  - Компоновщик
    - Общие
    - Ввод
    - Файл манифеста
    - Отладка
    - Система**
    - Оптимизация
    - Внедренный IDL
    - Метаданные Windows
    - Дополнительно
    - Все параметры
    - Командная строка
  - Инструмент манифеста
  - Генератор XML-документов
  - Информация об исходном
  - События сборки
  - Настраиваемый этап сборки
  - Анализ кода

Подсистема	Windows (/SUBSYSTEM:WINDOWS)
Минимальная требуемая версия	
Резервируемый размер кучи	
Фиксируемый размер кучи	
Резервируемый размер стека	
Фиксируемый размер стека	
Включить большие адреса	
Сервер терминалов	
Запускать с компакт-диска с помощью файла пс	Нет
Запускать из сети с помощью файла пс	Нет
Драйвер	Не задано

**Подсистема**  
Параметр /SUBSYSTEM предписывает операционной системе порядок выполнения EXE-файла. Выбор ...





# Программирование

Использование простого класса C++ в программе .NET с GUI

## 3 Настраиваем свойства проекта - 2

Страницы свойств WFCPP1

Конфигурация: **Активная (Debug)** Платформа: **Активная (Win32)** Диспетчер конфигураций...

- Свойства конфигурации
  - Общие
  - Отладка
  - Каталоги VC++
  - C/C++
  - Компоновщик
    - Общие
    - Ввод
    - Файл манифеста
    - Отладка
    - Система
    - Оптимизация
    - Внедренный IDL
    - Метаданные Windows
    - Дополнительно**
    - Все параметры
    - Командная строка
  - Инструмент манифеста
  - Генератор XML-документов
  - Информация об исходном
  - События сборки
  - Настраиваемый этап сбор
  - Анализ кода

Точка входа	main
Без точки входа	Нет
Установить контрольную сумму	Нет
Базовый адрес	
Внесение случайности в базовый адре	Да (/DYNAMICBASE)
Фиксированный базовый адрес	Нет (/FIXED:NO)
Предотвращение исполнения данных	Да (/NXCOMPAT)
Отключить создание сборки	Нет
Выгрузить отложено загружаемые DLL	
Не включать отложено загружаемые D	
Библиотека импорта	
Объединить разделы	
Конечный компьютер	MachineX86 (/MACHINE:X86)
Профиль	Нет
Атрибут потока CLR	
Тип CLR-образа	Тип образа по умолчанию
Файл ключа	
Контейнер ключей	
Отложенная подпись	
Проверка CLR на неуправляемый код	
Отчет об ошибке	Запросить немедленно (/ERRORREPORT:PROMPT)

**Точка входа**  
Параметр /ENTRY задает функцию точки входа в качестве начального адреса для EXE-файла или DLL.



# Программирование

Использование простого класса C++ в программе .NET с GUI

## 3 «Магический код» для функции main()

```
#include "MyForm.h"

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

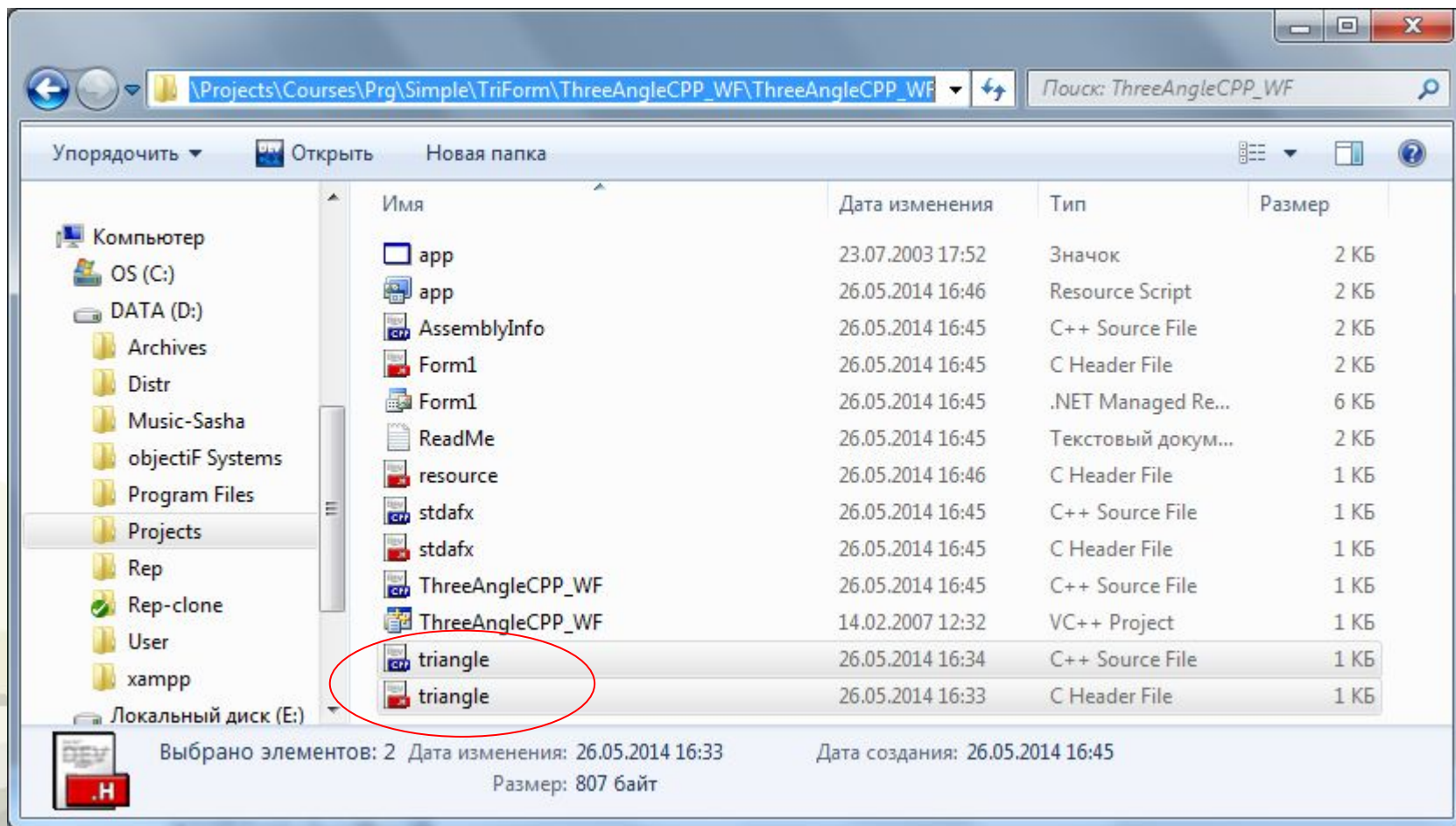
[STAThread]
void main(array<String^>^ arg) {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    WFCPP::MyForm^ form = gcnew WFCPP::MyForm();
    Application::Run(form);
}
```



# Программирование

Использование простого класса C++ в программе .NET с GUI

## 4 Копируем файл с классом в папку проекта

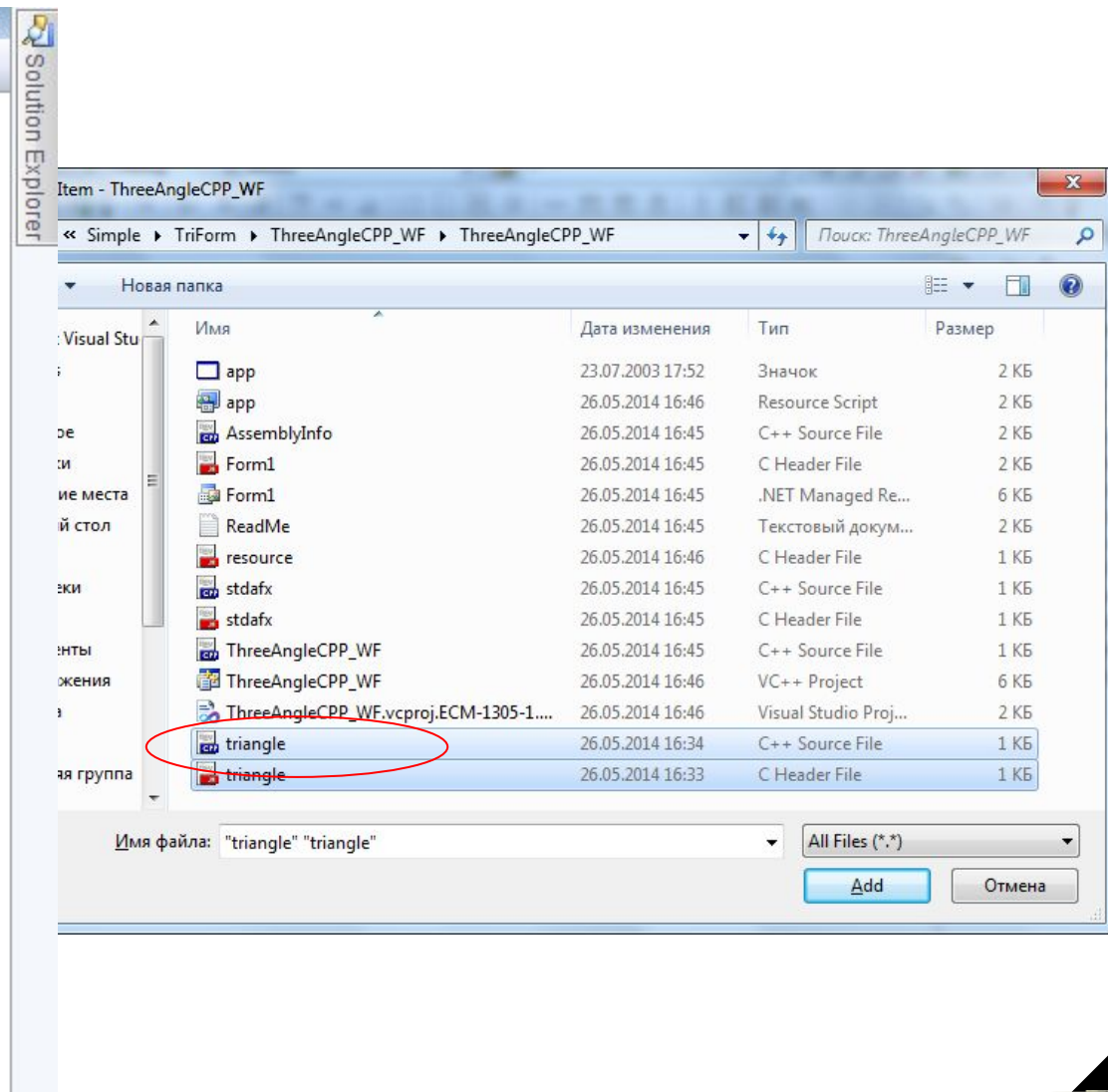
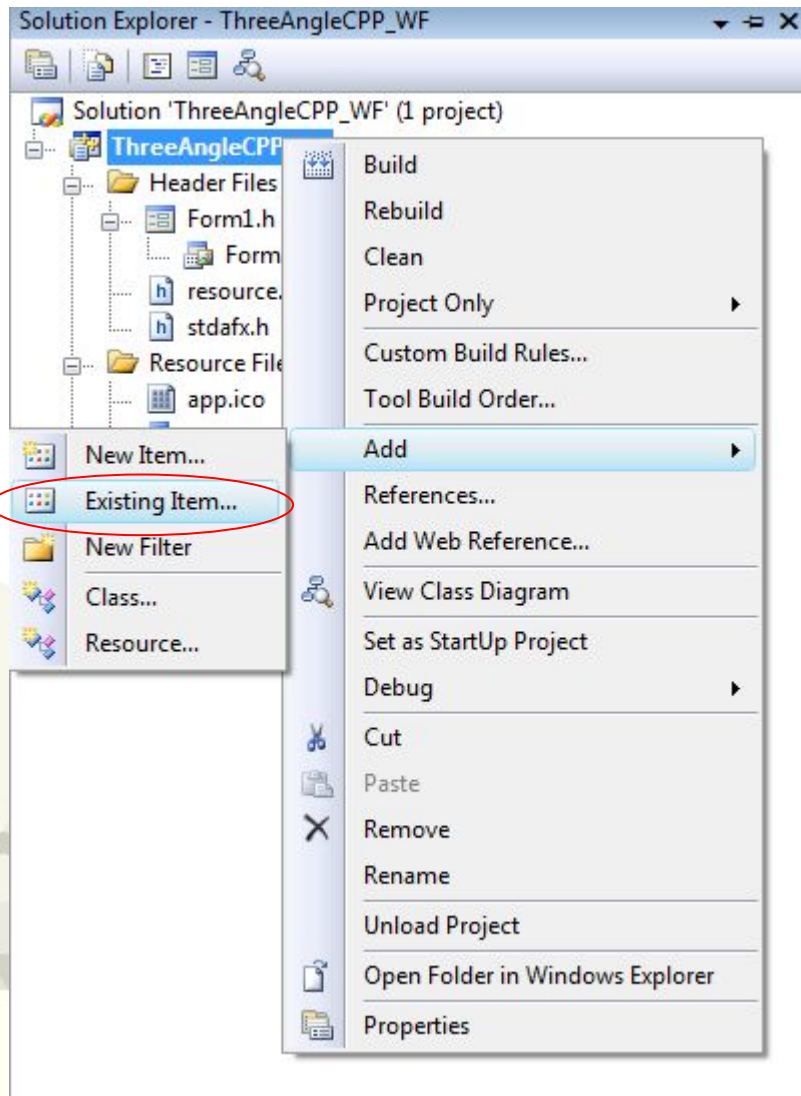




# Программирование

Использование простого класса C++ в программе .NET с GUI

## 5 Добавляем файлы с классом в проект

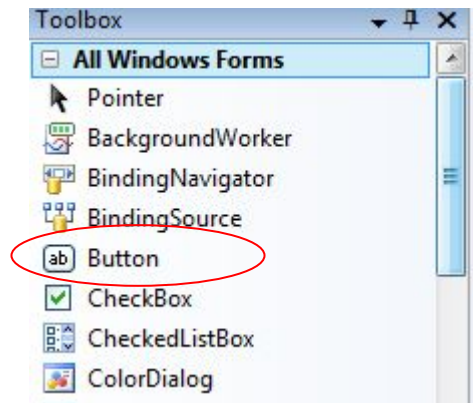
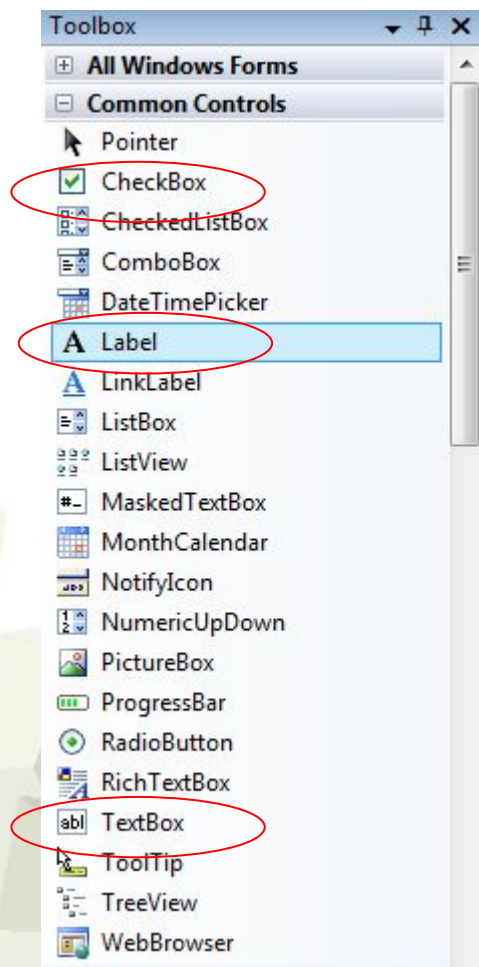




# Программирование

Использование простого класса C++ в программе .NET с GUI

## 6 Переносим на форму элементы управления с панели инструментов (Toolbox)



Можно скопировать из проекта для CS !

Нам требуется несколько Label, TextBox, по одному CheckBox и Button



# Программирование

Использование простого класса C++ в программе .NET с GUI

## 7 Переносим на форму элементы управления с панели инструментов (Toolbox)

Label

Треугольник

A

B

C 0

Площадь 0  прямоуг

Периметр 0

Расчет

TextBox

Label

CheckBox

Button



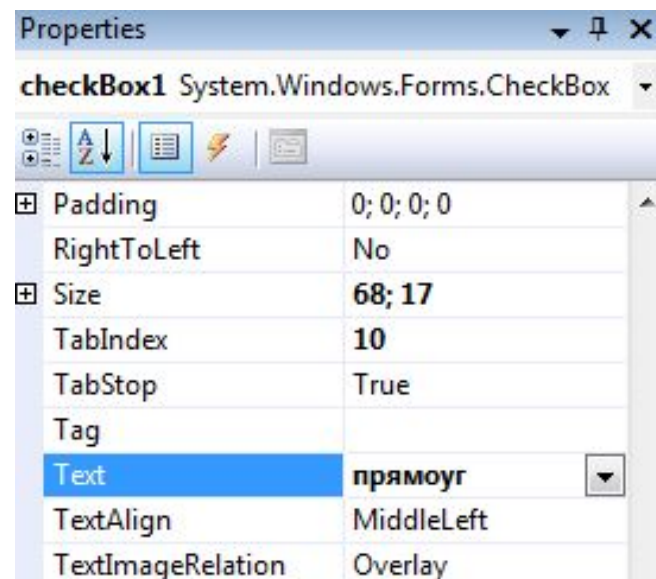
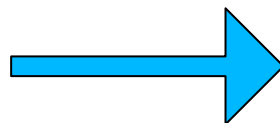
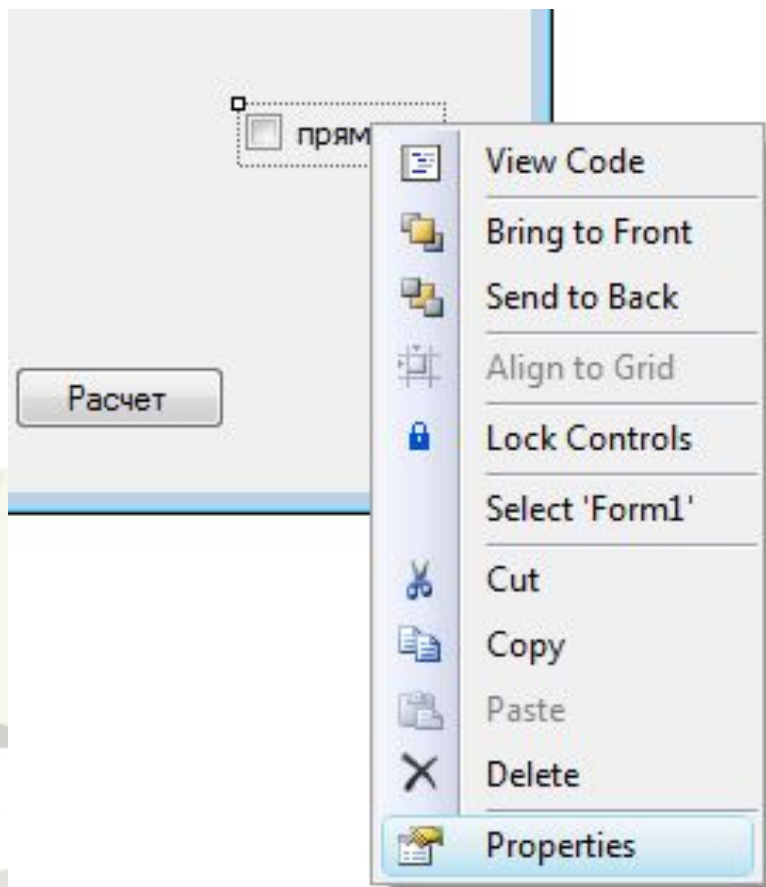


# Программирование

Использование простого класса C++ в программе .NET с GUI

## 8 Задание свойств элементов управления

Задаем свойства  
Text элементов управления  
в окнах свойств Properties,  
вызываемых из  
контекстного  
меню при выбранном  
элементе управления





# Программирование

Использование простого класса C++ в программе .NET с GUI

## 9 Двойным кликом на кнопке создаем обработчик нажатия кнопки `button1_Click`

Треугольник

A

B

C 0

Площадь 0  прямоуг

Периметр 0

**Расчет**

```
Form1.h* Form1 [Design]*
ThreeAngleCPP_WF::Form1
button1_Click(System::Object ^ sender, System::EventArgs ^ e)

this->Controls->Add(this->label2);
this->Controls->Add(this->label1);
this->Controls->Add(this->button1);
this->Name = L"Form1";
this->Text = L"Треугольник";
this->ResumeLayout(false);
this->PerformLayout();

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





# Программирование

Использование простого класса C++ в программе .NET с GUI

## 10 Подключаем заголовочный файл с объявлением класса

```
triangle.cpp Form1.h* Form1.h [Design]*
(Global Scope)
#pragma once
#include "triangle.h"

namespace ThreeAngleCPP_WF {

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

    /// <summary>
    /// Summary for Form1
    ...
```



# Программирование

Использование простого класса C++ в программе .NET с GUI

## 11 Создаем и используем объект `tri` класса `Rect3Angle` в обработчике нажатия кнопки

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

Имя класса с пространством  
ИМЕН

Конструктор объекта

```
    ThreeAngle::Rect3Angle tri(Double::Parse(textBox1->Text),  
        Double::Parse(textBox2->Text));
```

```
    label6->Text = tri.getC().ToString();
```

Указатель на объект .NET

```
    label7->Text = tri.getArea(checkBox1->Checked).ToString();
```

```
    label8->Text = tri.getPerimeter().ToString();
```

Вызов метода объекта

Вызов метода объекта  
double, автоматически  
приведенного к .NET  
Double



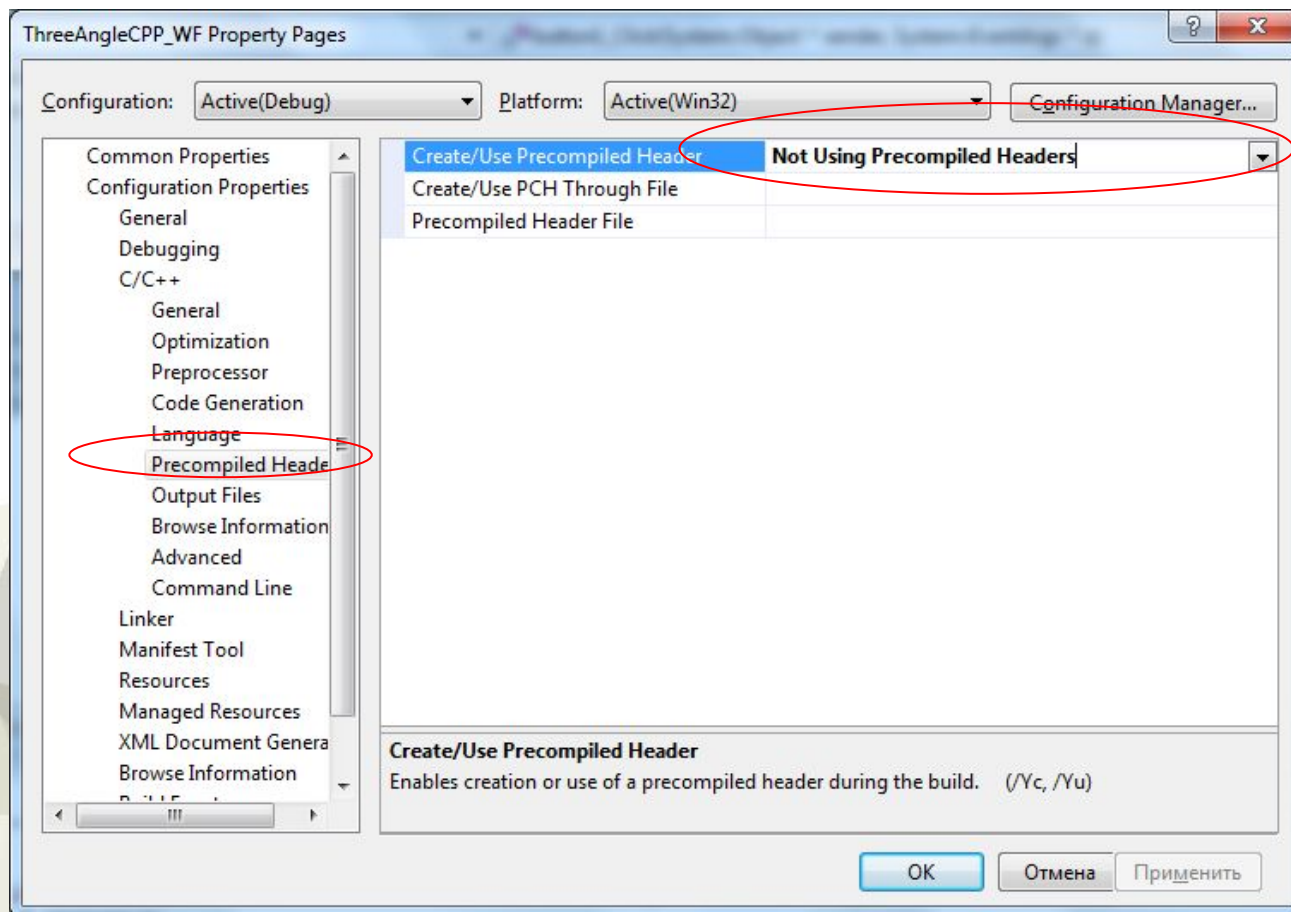
# Программирование

Использование простого класса C++ в программе .NET с GUI

## 12 Отключаем Precompiled Headers

Из-за особенностей среды Visual Studio, чтобы не менять код подключенного заголовочного файла **Triange.h**, проще перед компиляцией отключить предварительно компилируемые заголовки, что несколько увеличит время компиляции.

**Project – Properties – ...**





# Программирование

Использование простого класса C++ в программе .NET с GUI

## 13 GUI приложение в работе

Треугольник

A	3
B	4
C	5
Площадь	6
Периметр	12

прямоуг

Расчет

Треугольник

A	3
B	4
C	5
Площадь	12
Периметр	12

прямоуг

Расчет