

## Создание модульных тестов

Модульное тестирование (или Unit-тестирование) предназначено для проверки правильности выполнения небольшого блока кода, решающего свою конкретную задачу.

### 1 Создание тестируемой библиотеки DLL

Для примера создается библиотека вычисляющая площадь прямоугольника. Имя проекта: Geometry. Имя решения: Geometry. Тип приложения: Библиотека динамической компоновки DLL. Создаем через мастер классических приложений Windows (рисунок 1). При создании установите галочку «Экспорт символов» (рисунок 2).

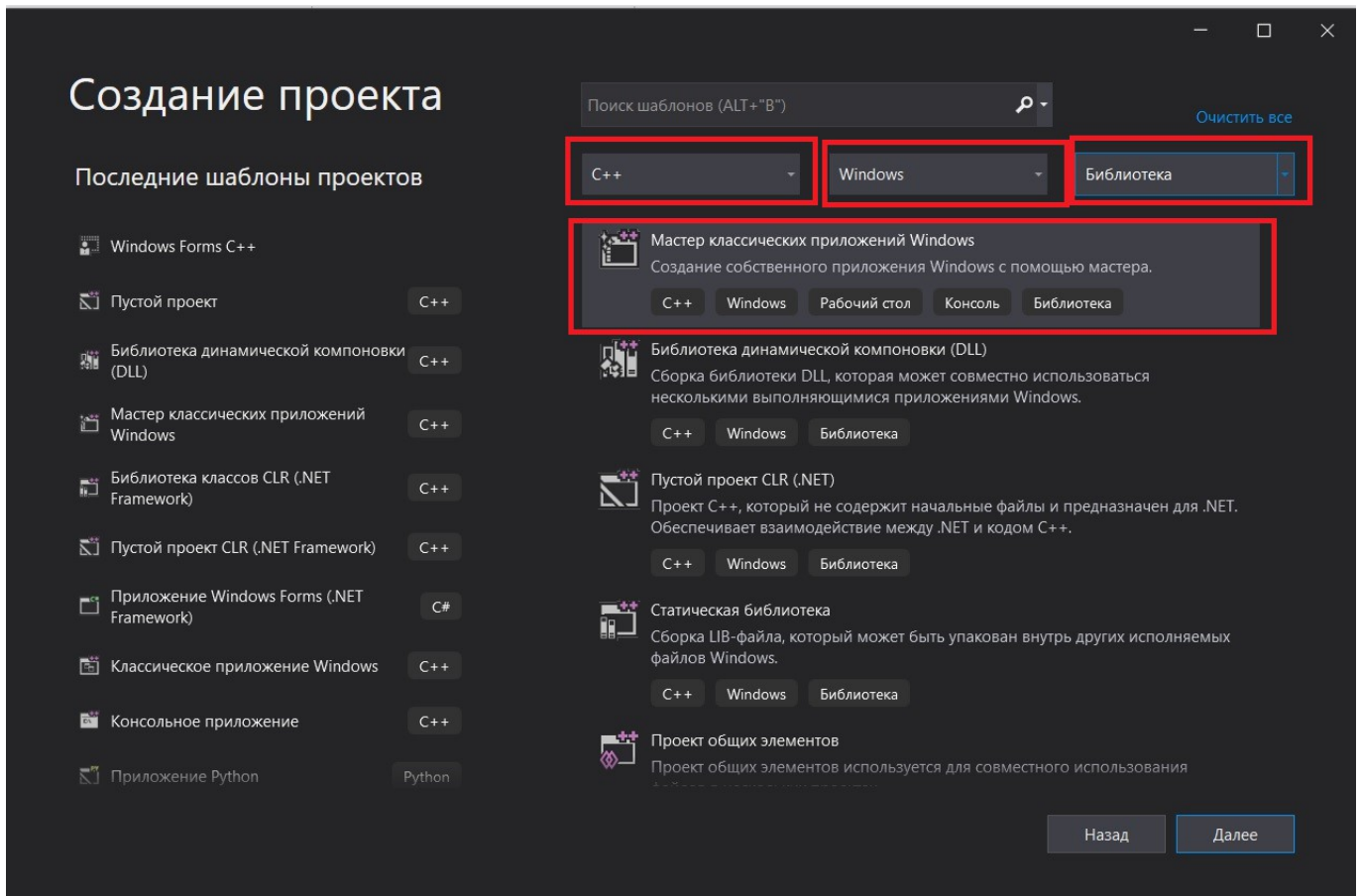


Рисунок 1 - Создание библиотеки DLL через мастер классических приложений Windows

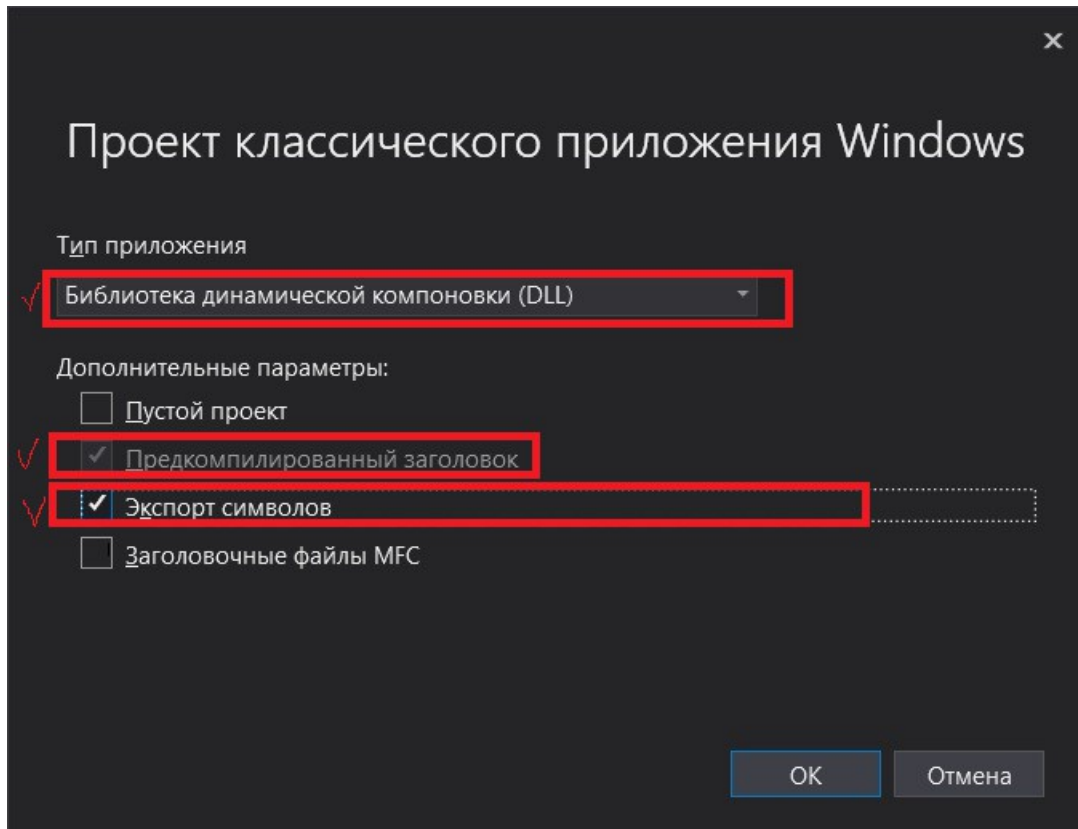


Рисунок 2 - Тип приложения: Библиотека динамической компоновки DLL.

Добавьте в файл Geometry.h метод для вычисления площади прямоугольника.

```
class GEOMETRY_API CGeometry {  
public:  
    CGeometry(void);  
    int RectangleArea(int a, int b);  
};
```

В основной CPP-файл добавьте тело функции:

```
int CGeometry::RectangleArea(int a, int b)  
{  
    return a * b;  
}
```

Выполните сборку решения.

## 2 Создание проекта для модульного тестирования

Выбор типа проекта: «Проект машинного модульного теста» в окне «Создание проекта» представлен на рисунке 3. Проект должен быть создан в том же решении (рисунок 4) Geometry. Имя проекта: MathTest.

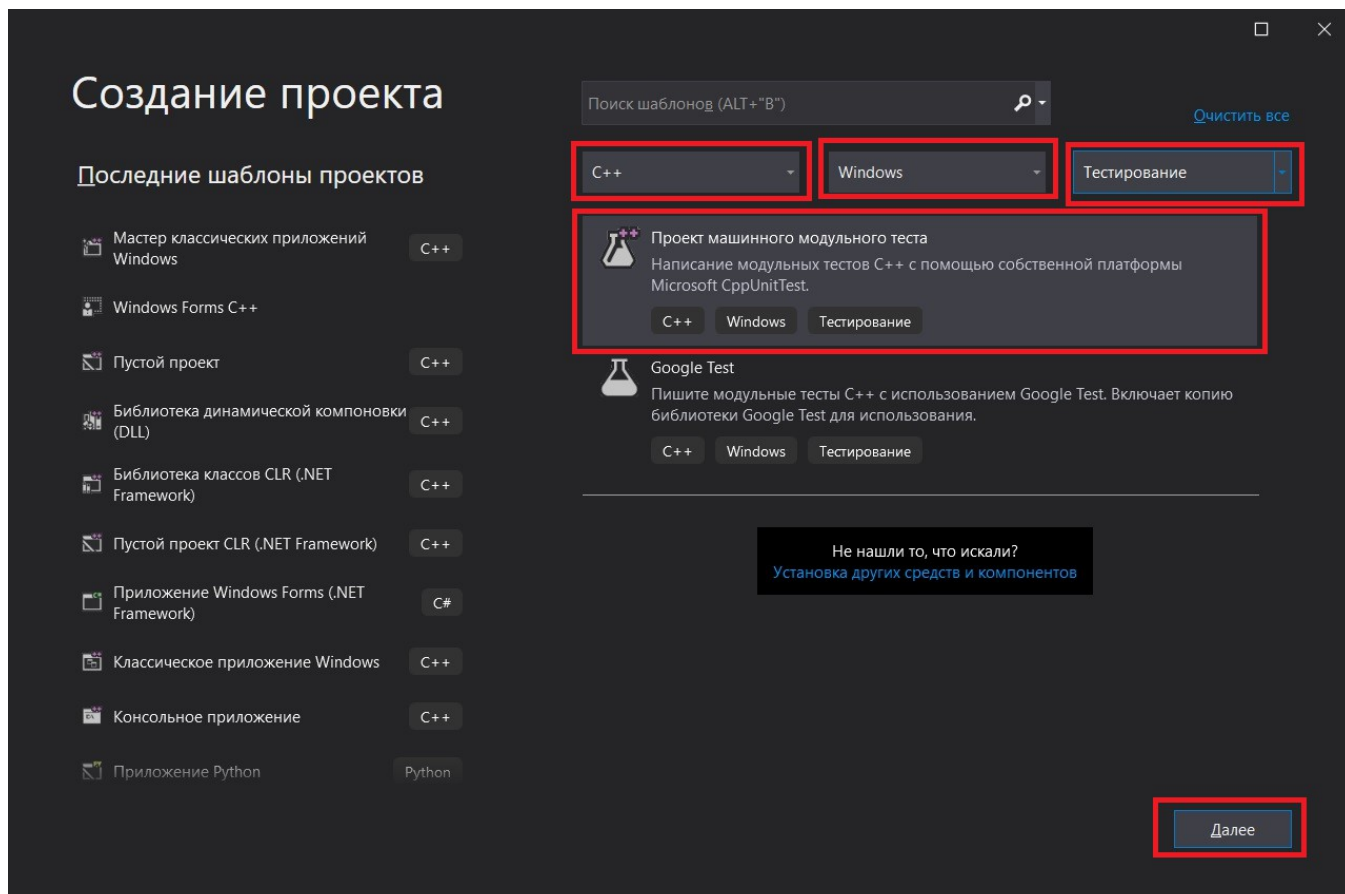


Рисунок 3 - Создание проекта модульного тестирования

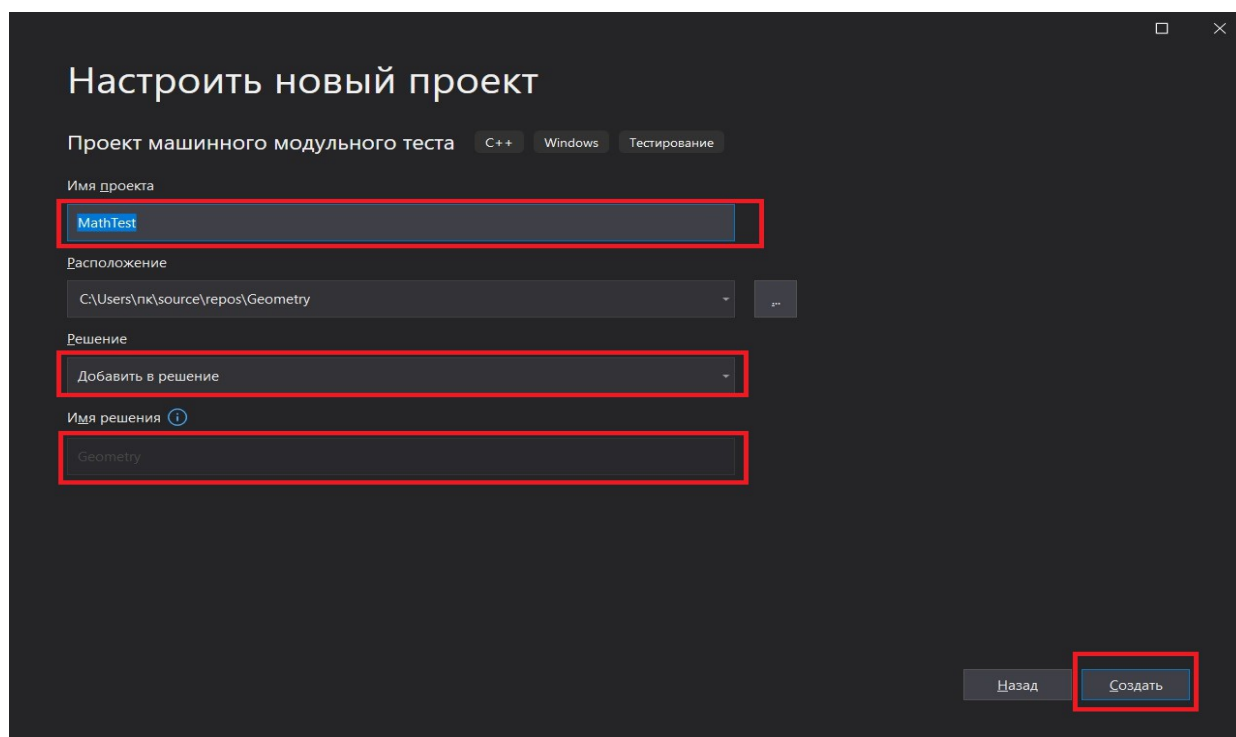


Рисунок 4 - Создание проекта модульного тестирования

Переименуем класс `UnitTest1` в `GeometryTests`.  
Добавьте ссылку на тестируемый проект (рисунок 4).

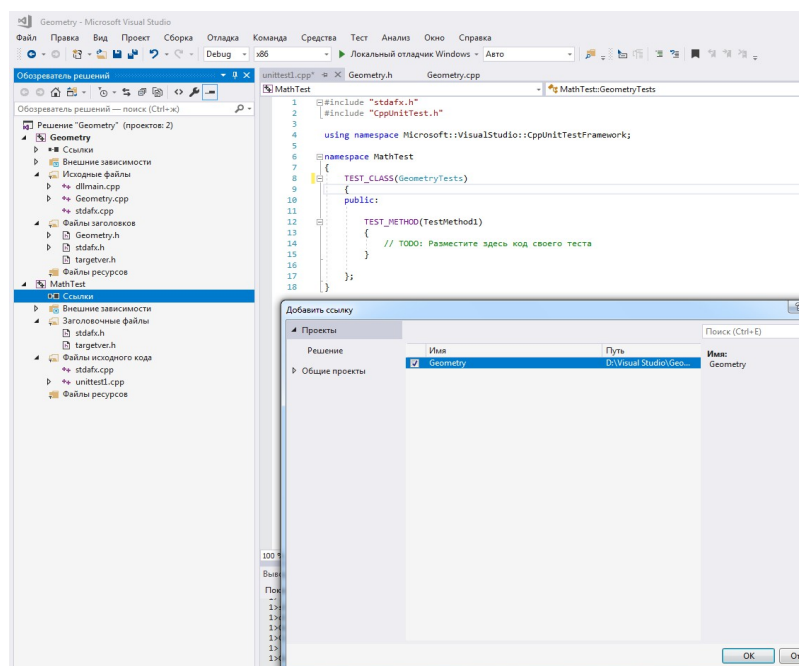


Рисунок 4 - Добавление ссылки на тестируемый проект

В основном CPP-файле модульного теста включите H-файл кода библиотеки DLL.

Напишите тест, который проверяет, правильно ли вычисляется площадь прямоугольника со сторонами 3 и 5. Ожидаемый результат при этом 15.

```
#include "stdafx.h"
#include "CppUnitTest.h"
#include "../Geometry/Geometry.h"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace MathTest
{
    TEST_CLASS(GeometryTests)
    {
    public:

        TEST_METHOD(RectangleS3_5_15)
        {
            // исходные данные
            int a = 3;
            int b = 5;
            int expected = 15;

            // получение значения с помощью тестируемого метода
            CGeometry g;
            int actual = g.RectangleArea(a, b);

            // сравнение ожидаемого результата с полученным
            Assert::AreEqual(expected, actual);
        }
    };
}
```

Выполните сборку решения.

Чтобы посмотреть доступные тесты откройте обозреватель тестов через меню Тест->Обозреватель тестов (рисунок 5).

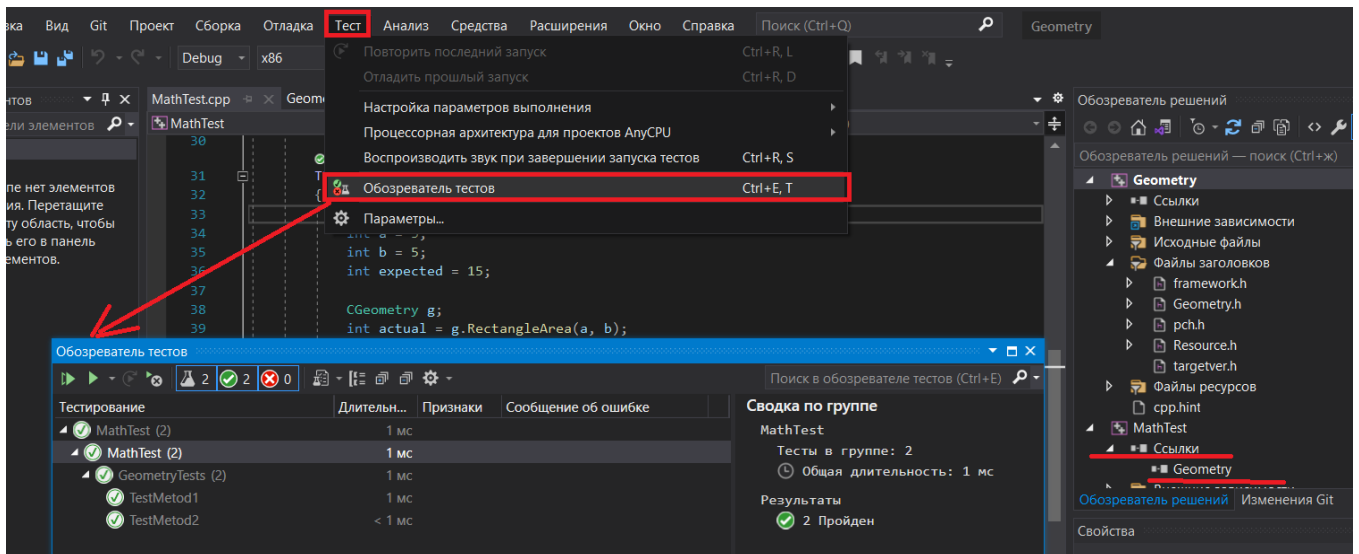


Рисунок 5

Запустите созданный тест, убедитесь, что он выполняется правильно. Если в тестируемой программе есть ошибки, возвращаемое значение не совпадает с ожидаемым, и тест отмечается как не пройденный (рисунок 6).

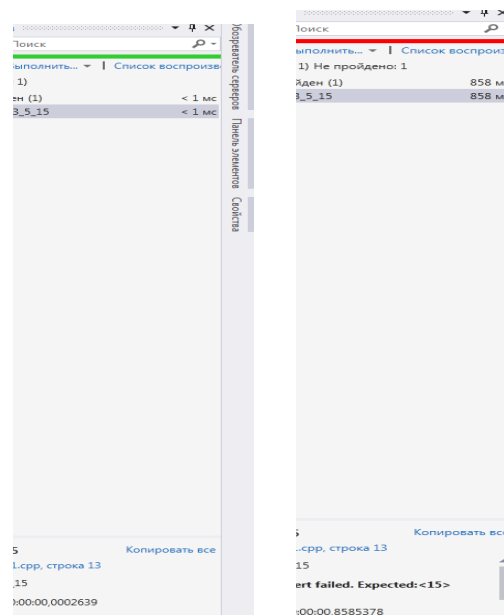


Рисунок 6 - Выполнение теста

### 3 Ссылки

<https://docs.microsoft.com/ru-ru/visualstudio/test/unit-test-your-code?view=vs-2017>

<https://msdn.microsoft.com/ru-ru/library/ms182532.aspx>

#### 4 Индивидуальное задание

Создайте библиотеку DLL, реализующую функцию согласно варианту. Разработайте набор модульных тестов, которые проверяют корректность вычисления функции.

Вариант	Функция
1	Определение площади наибольшего из прямоугольников, заданных координатами двух вершин.
2	Определение взаимного расположения (пересекаются, не пересекаются, вложены, касаются) окружности, заданной координатами центра и радиусом, и прямой, заданной координатами двух точек.
3	Определение взаимного расположения (пересекаются, не пересекаются, вложены, касаются) окружности, заданной координатами центра и радиусом, и треугольника, заданного координатами его вершин.
4	Определить количество непересекающихся квадратов. задается количество квадратов, координаты центров и длины сторон.
5	Определение взаимного расположения двух окружностей (пересекаются, не пересекаются, вложены, касаются), заданных координатами центров и радиусами.
6	Определение взаимного расположения двух треугольников (пересекаются, не пересекаются, вложены, касаются), заданных координатами вершин.
7	Определить взаимное расположение двух прямоугольников (пересекаются, не пересекаются, вложены, касаются), заданных координатами двух вершин.
8	Определить взаимное расположение (пересекаются, не пересекаются, вложены, касаются) треугольника, заданного координатами его вершин, и прямой, заданной координатами двух точек.
9	Определение вида треугольника (равносторонний, прямоугольный, остроугольный, тупоугольный), заданного координатами вершин.

Создайте GUI приложение, использующее разработанную библиотеку и визуализирующее найденное решение.