

Подключение GoogleTest к проекту в Visual Studio

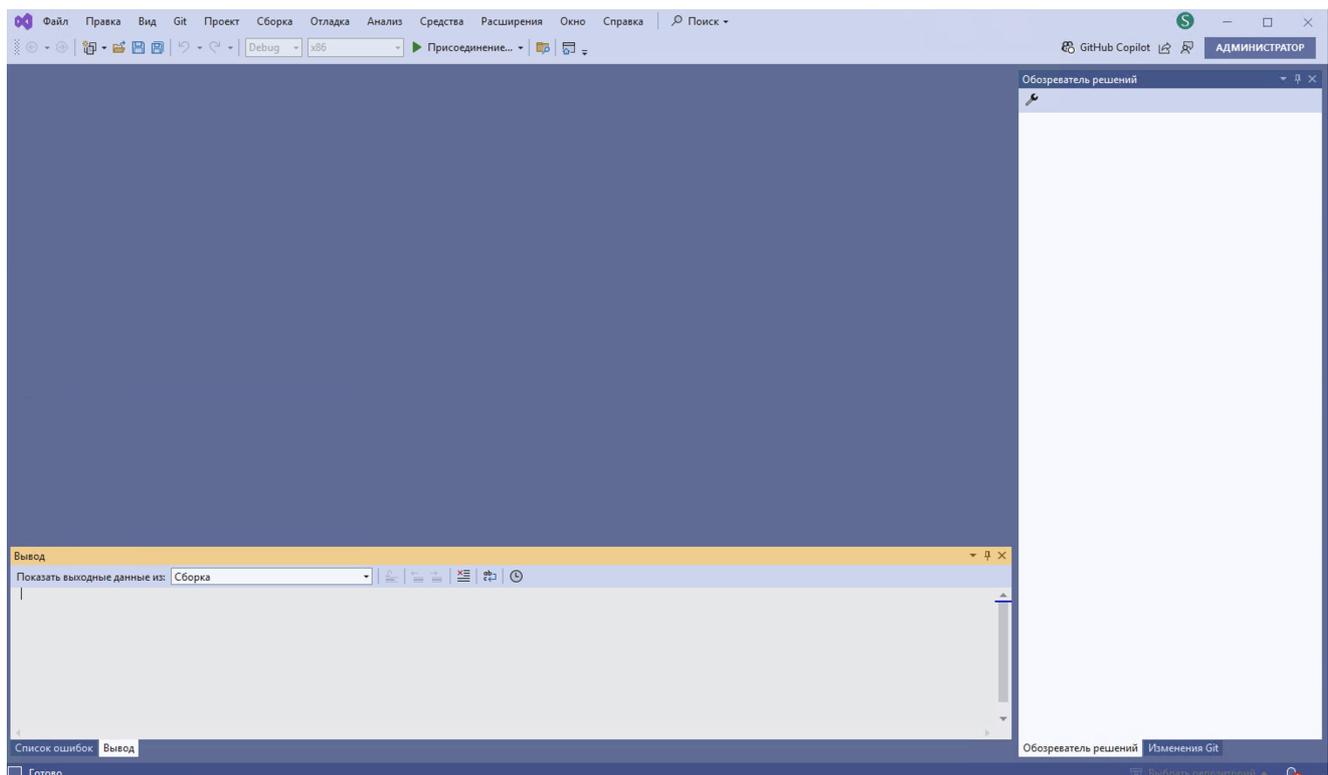
Tags ::

В этом документе рассматривается процесс создания библиотеки статической компоновки в Visual Studio с подключением фреймворка модульного тестирования GoogleTest

Замечание

Используется Visual Studio 2022
Язык разработки - C++

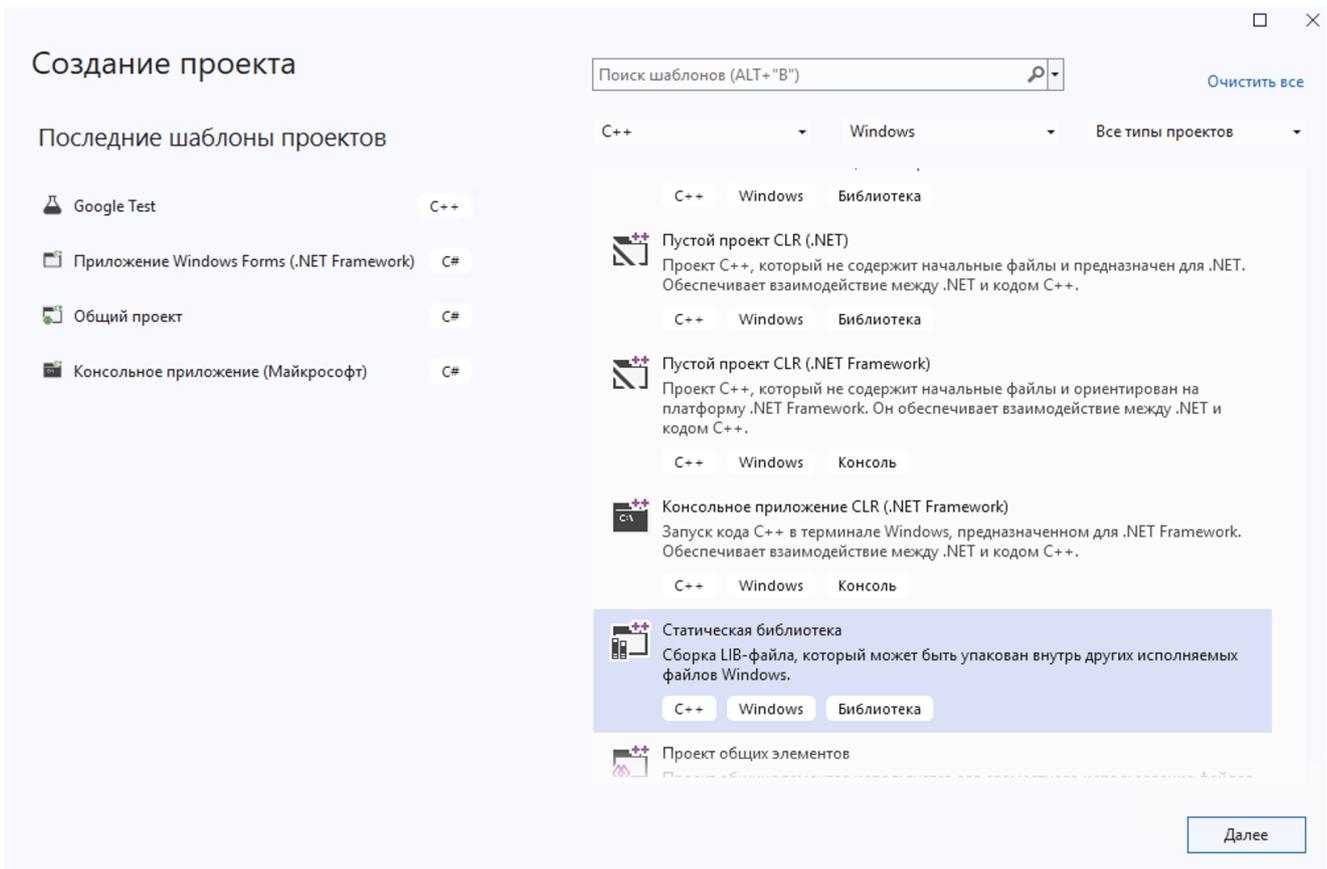
Рассмотрим создание небольшого примера с библиотекой, консольным приложением и тестированием функций из библиотеки.



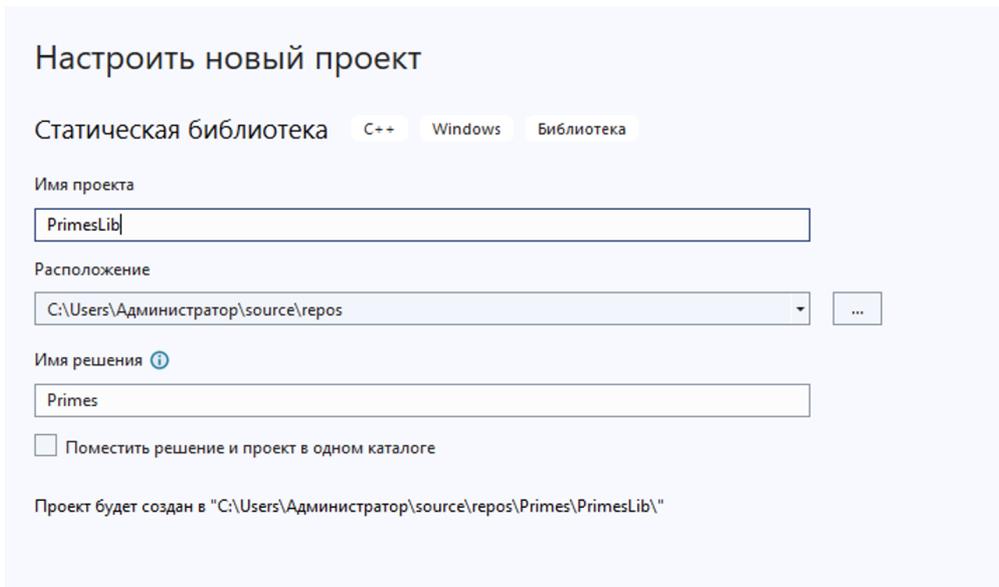
Создание нового решения (solution)

Решение будет автоматически создано при создании проекта. Первым проектом будет библиотека для работы с простыми числами

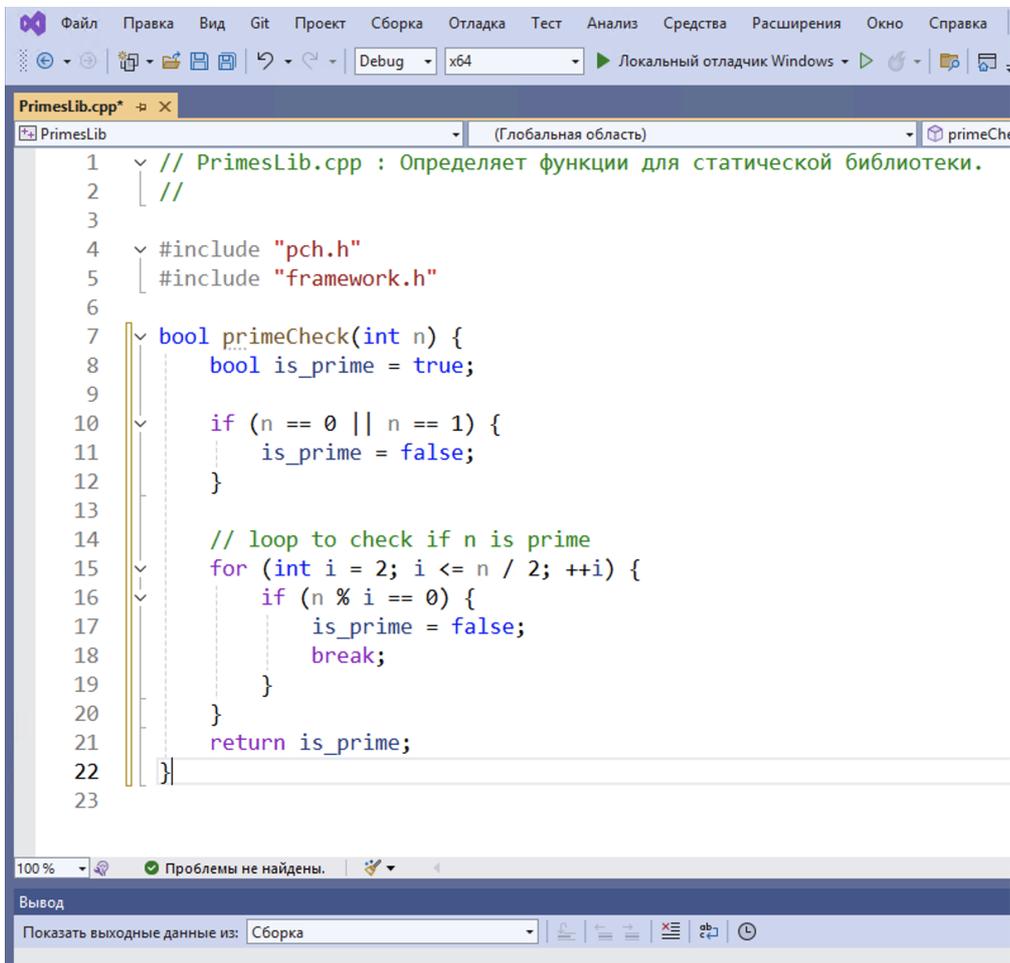
В диалоговом окне для создания проекта указываем тип:



Далее вводим имя проекта и имя всего решения



В окне редактора вводим текст пока единственной функции библиотеки



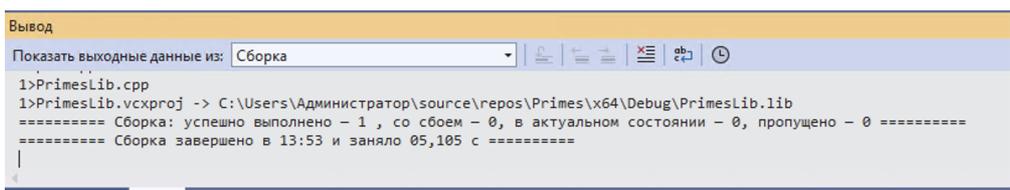
```
1 // PrimesLib.cpp : Определяет функции для статической библиотеки.
2 //
3
4 #include "pch.h"
5 #include "framework.h"
6
7 bool primeCheck(int n) {
8     bool is_prime = true;
9
10    if (n == 0 || n == 1) {
11        is_prime = false;
12    }
13
14    // loop to check if n is prime
15    for (int i = 2; i <= n / 2; ++i) {
16        if (n % i == 0) {
17            is_prime = false;
18            break;
19        }
20    }
21    return is_prime;
22 }
23
```

Также нам необходимо добавить в проект заголовочный файл primes.h



```
1 #pragma once
2
3 bool primeCheck(int n);
4
```

После этого можно осуществить построение проекта. Обязательно выбрать команду Создать, а не Запустить, так как это библиотека а не исполняемое приложение!



```
Вывод
Показать выходные данные из: Сборка
1>PrimesLib.cpp
1>PrimesLib.vcxproj -> C:\Users\Администратор\source\repos\Primes\x64\Debug\PrimesLib.lib
===== Сборка: успешно выполнено - 1, со сбоем - 0, в актуальном состоянии - 0, пропущено - 0 =====
===== Сборка завершено в 13:53 и заняло 05,105 с =====
```

Добавление проекта "console application"

Теперь, когда библиотека подготовлена и скомпилирована, можно создать демонстрационное консольное приложение, например такое:

Добавить новый проект

Поиск шаблонов (ALT+“В”) Очистить

C++ Windows Все типы проектов

Последние шаблоны проектов

- Статическая библиотека C++
- Google Test C++
- Приложение Windows Forms (.NET Framework) C#
- Общий проект C#
- Консольное приложение (Майкрософт) C#

Консольное приложение
Выполнить код в терминале Windows. По умолчанию выводится фраза "Hello World".
C++ Windows Консоль

Проект CMake
Создавайте современные кроссплатформенные приложения C++, не зависящие от файлов SLN или VCXPROJ.
C++ Windows Linux Консоль

Мастер классических приложений Windows
Создание собственного приложения Windows с помощью мастера.
C++ Windows Рабочий стол Консоль Библиотека

Классическое приложение Windows
Проект приложения с графическим интерфейсом в Windows.
C++ Windows Рабочий стол

Библиотека динамической компоновки (DLL)
Сборка библиотеки DLL, которая может совместно использоваться

Настроить новый проект

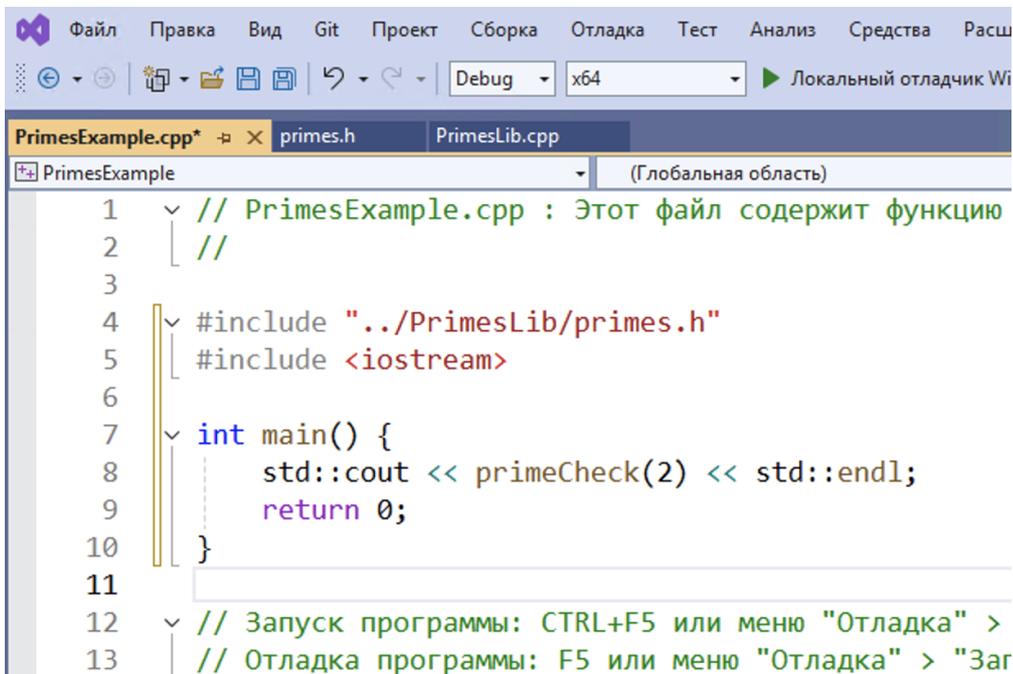
Консольное приложение C++ Windows Консоль

Имя проекта

Расположение
 ...

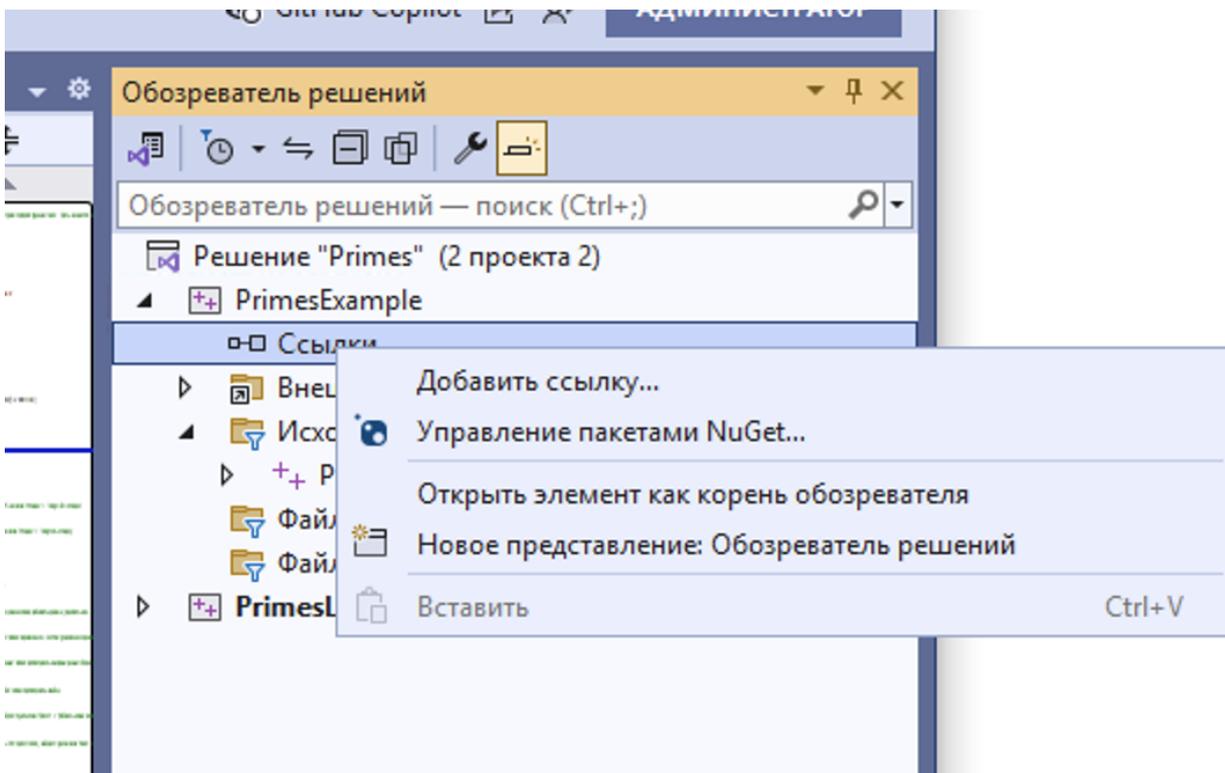
Проект будет создан в "C:\Users\Администратор\source\repos\Primes\PrimesExample\"

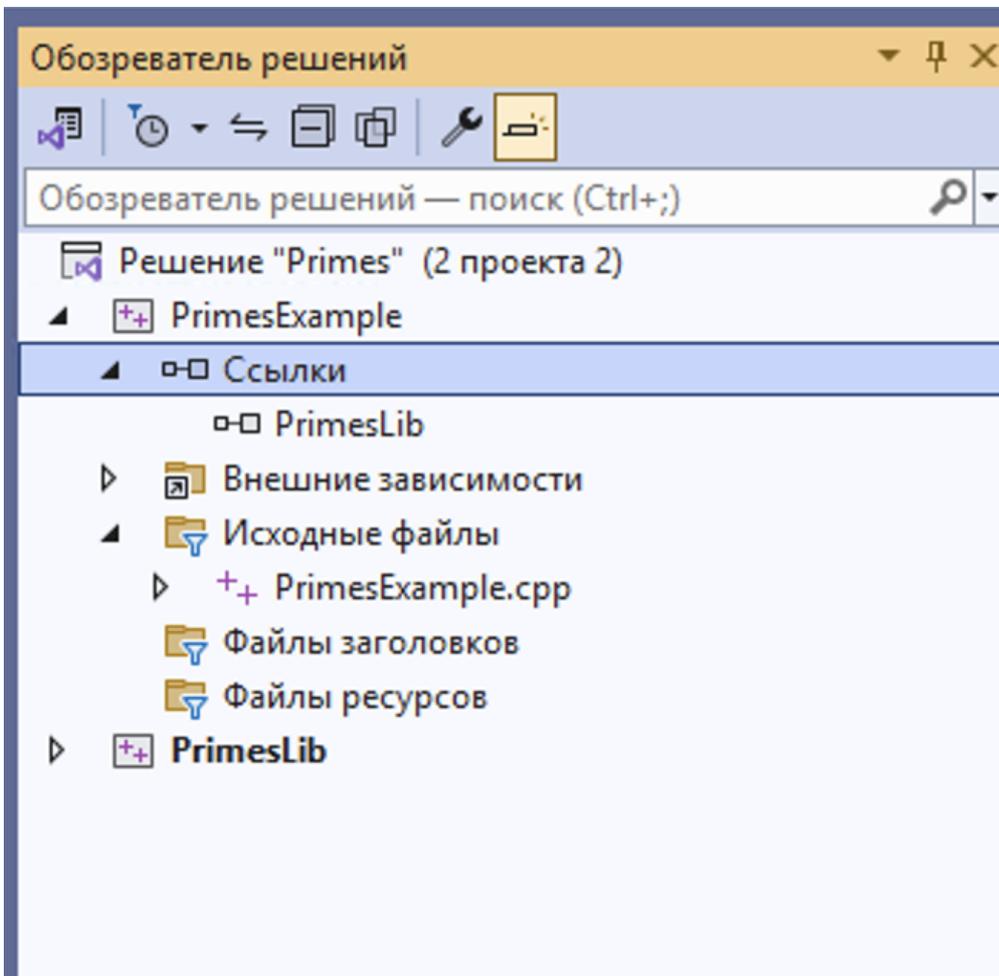
Это приложение будет содержать функцию `main`



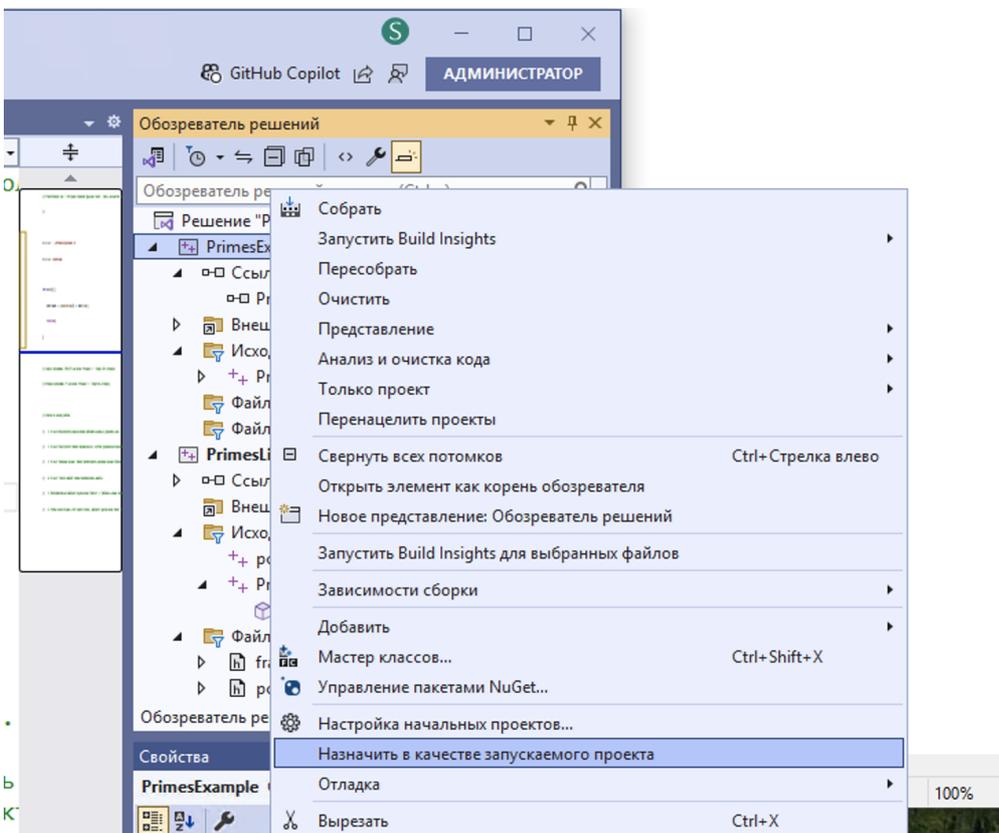
```
1 // PrimesExample.cpp : Этот файл содержит функцию
2 //
3
4 #include "../PrimesLib/primes.h"
5 #include <iostream>
6
7 int main() {
8     std::cout << primeCheck(2) << std::endl;
9     return 0;
10 }
11
12 // Запуск программы: CTRL+F5 или меню "Отладка" >
13 // Отладка программы: F5 или меню "Отладка" > "Заг
```

Если редактор будет подсвечивать красным директиву `include` или имя функции, то необходимо создать ссылку на внешний проект (на построенную ранее библиотеку)





После этого новый проект должен скомпилироваться. Чтобы быть запущенным, он должен быть назначен запускаемым

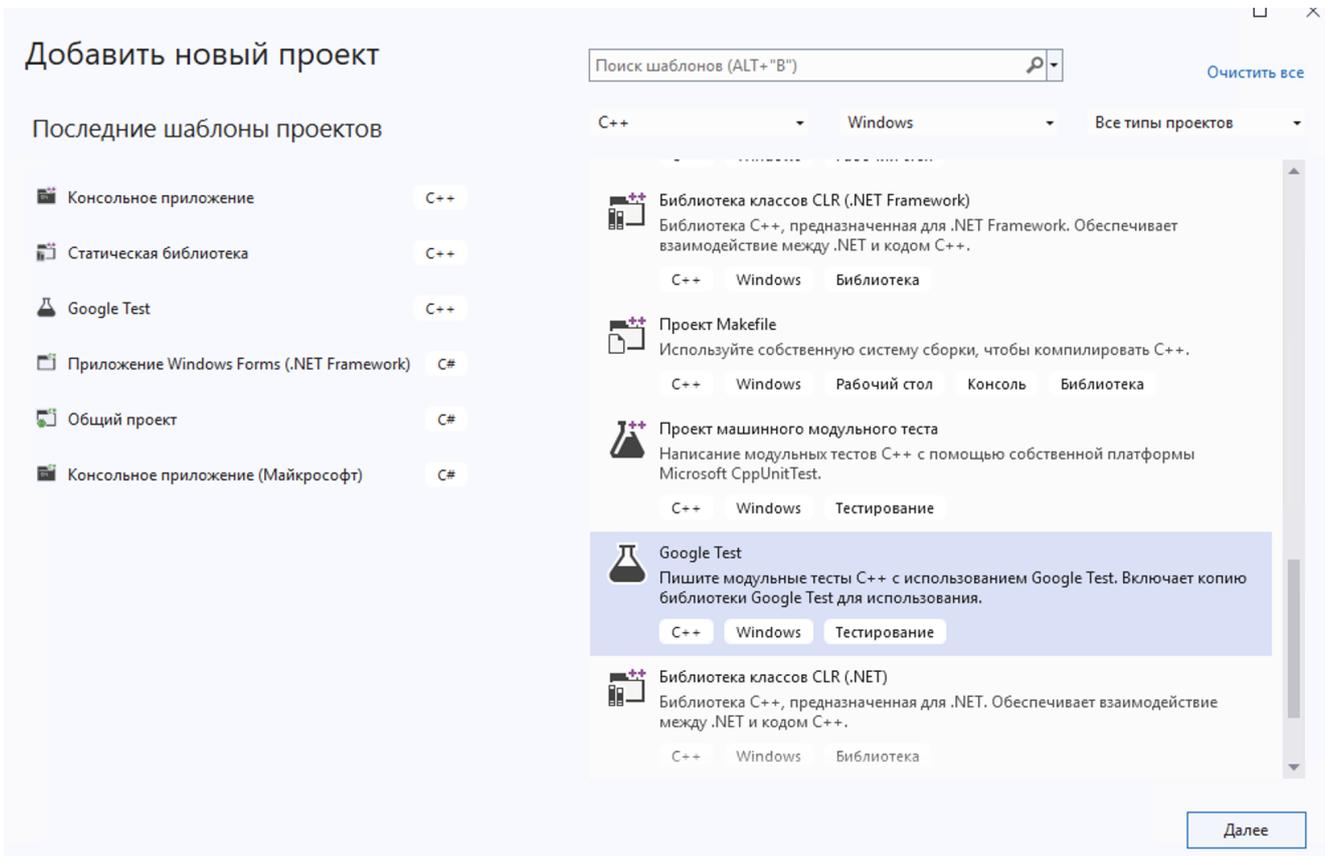


При запуске в режиме без отладки

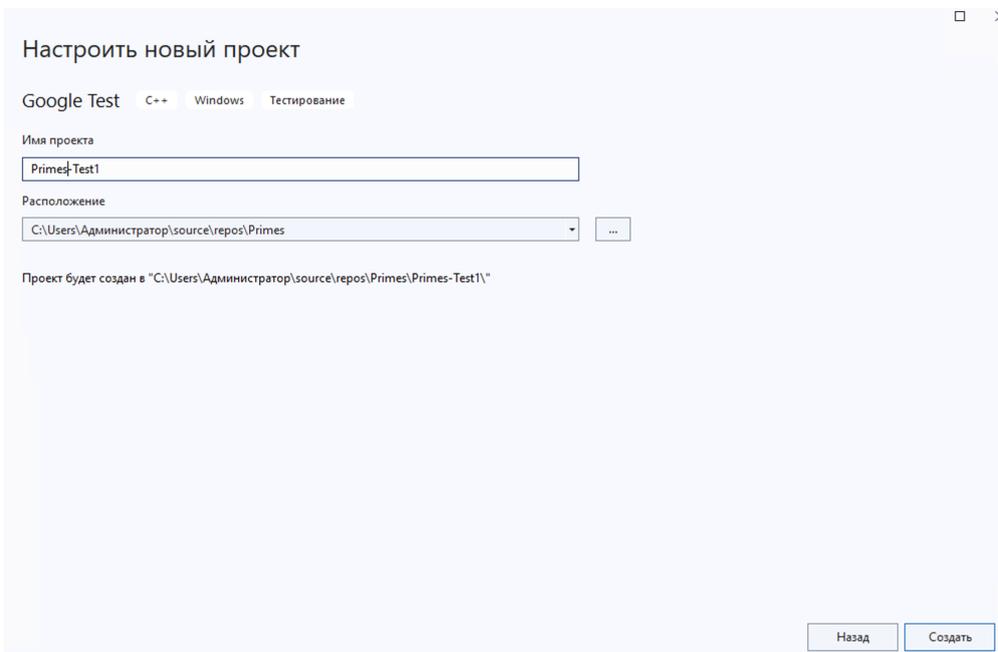
```
Консоль отладки Microsoft Visual Studio
1
C:\Users\Администратор\source\repos\Primes\x64\Debug\PrimesExample.exe (процесс 13840) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Добавление проекта GTest (dynamic linked library)

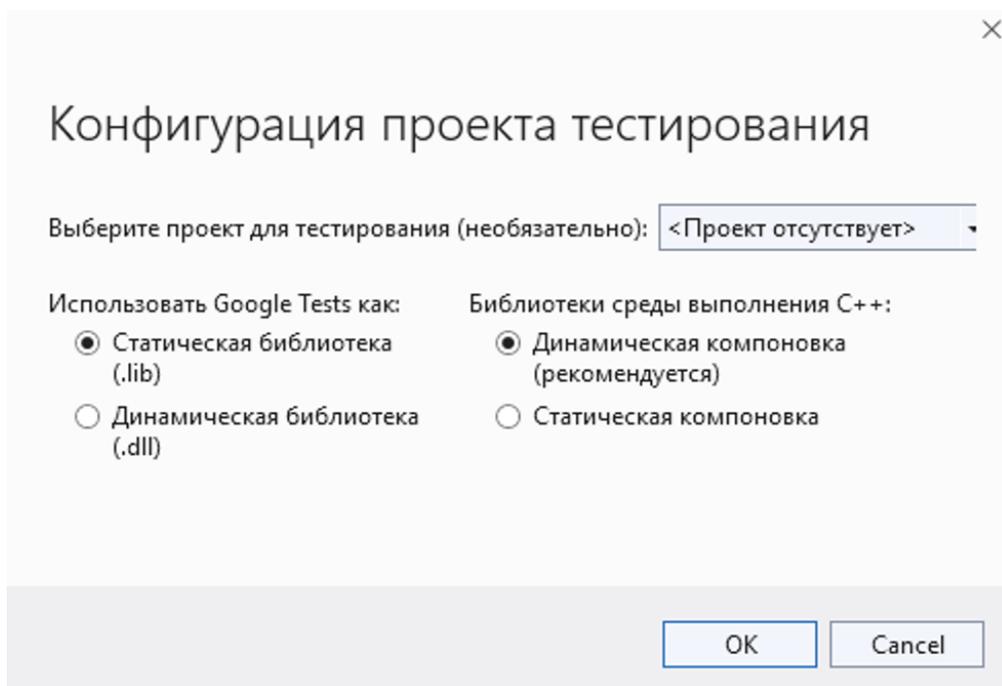
Снова добавляем в решение новый проект



и определяем его имя



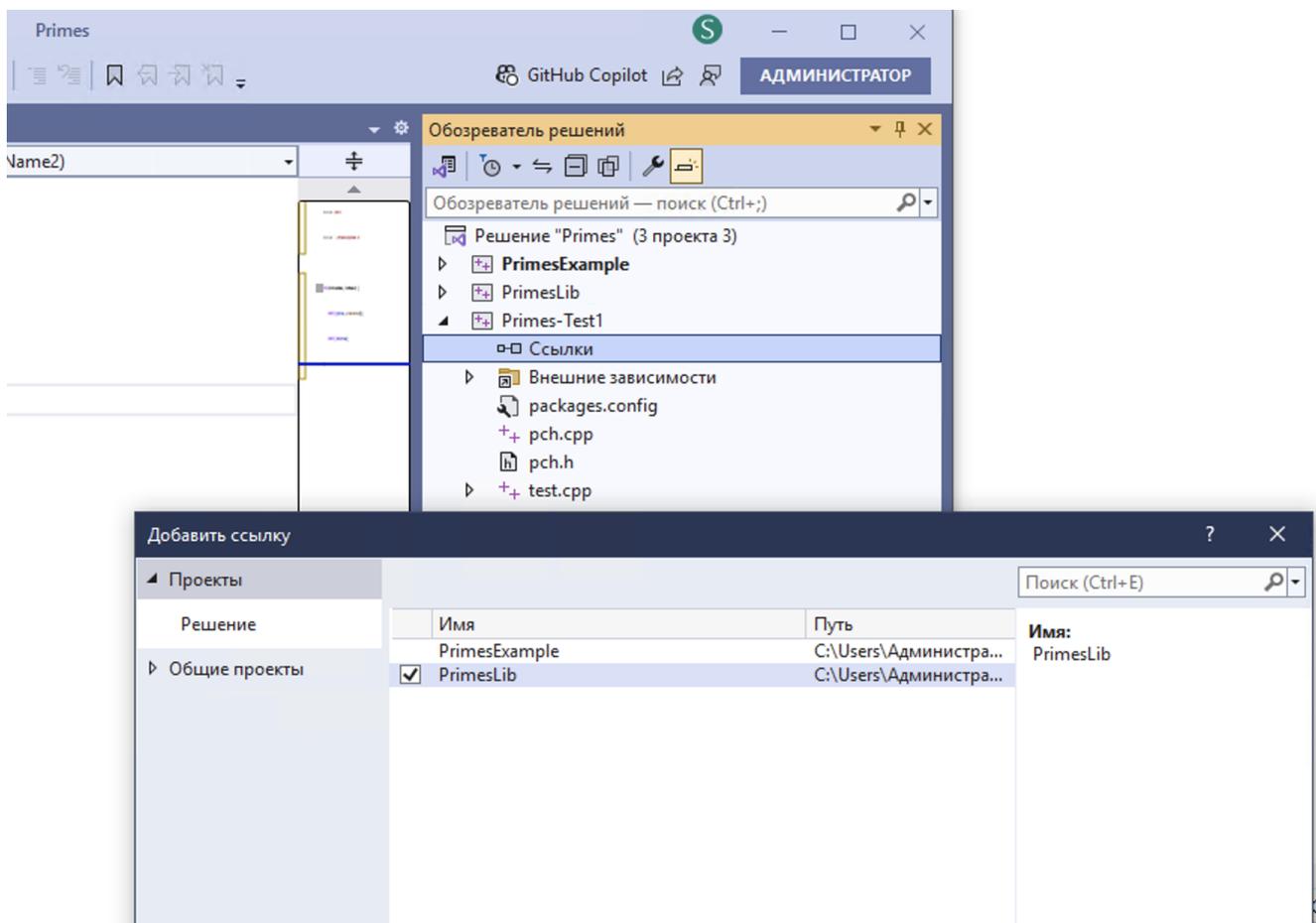
Важный момент: рекомендуется выставить в диалоговом окне следующие настройки:



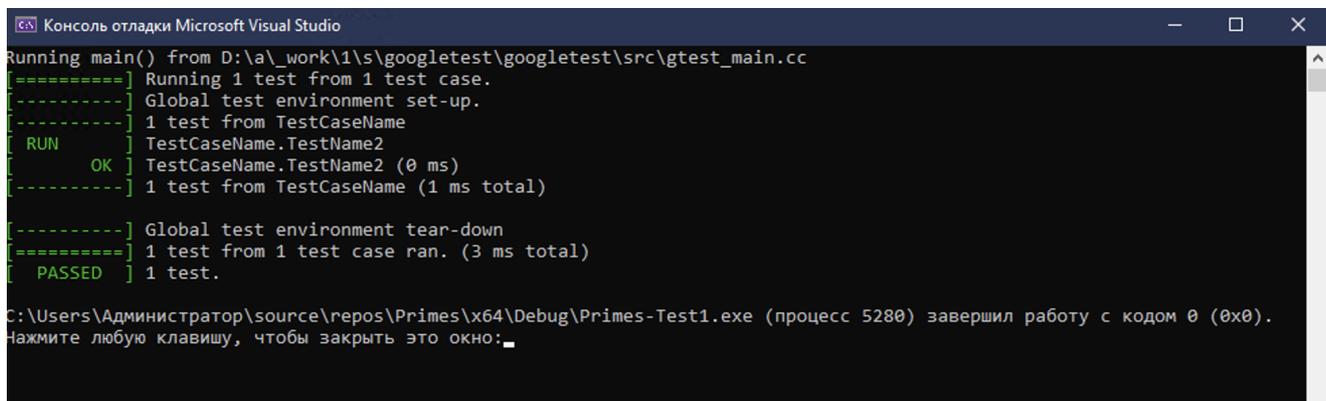
Далее, помещаем текст теста:

```
1 #include "pch.h"
2 #include "../PrimesLib/primes.h"
3
4 TEST(TestCaseName, TestName2) {
5     EXPECT_EQ(true, primeCheck(2));
6     EXPECT_TRUE(true);
7 }
```

Также, как и в случае с консольным приложением, добавленным ранее, нужно создать ссылку на проект:



Делаем новый проект запускаемым и запускаем его



Проект с тестированием, по сути, является консольным приложением, но в качестве кода выполняются тесты и их результаты отображаются в консоли

Приложение: текст функции проверки чисел на простоту

```

bool primeCheck(int n) {
    bool is_prime = true;

    if (n == 0 || n == 1) {
        is_prime = false;
    }

    // loop to check if n is prime
    for (int i = 2; i <= n / 2; ++i) {
        if (n % i == 0) {

```

```
        is_prime = false;
        break;
    }
}
return is_prime;
}
```