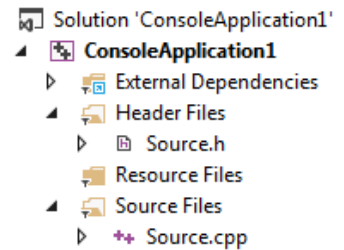


Пусть проект имеет следующую структуру:

Заметим, что наличие .h файла с объявлением всех тестируемых функций обязательно.



**Последовательность действий для добавления тестов:**

0. При создании проекта обязательно снимите галку в поле Precompiled Header (предкомпилированные ? заголовки)

Или:

Делайте пустой проект (Empty project).

Что делать, если в существующем проекте эта опция не была отключена:

Контекстное меню проекта -> Properties (свойства).

В появившемся окне: C\C++->Precompiled Headers -> щёлкнуть в поле Precompiled Headers, нажать на стрелочку, выбрать Not Using Precompiled Headers (Не использовать предкомпилированные заголовки).

1. Вызвать контекстное меню Решения (правой кнопкой мыши по Solution 'ConsoleApplication1' в окне Solution Explorer (справа))->Add->New Project.

В появившемся окне: Visual C++->Test->Native Unit Test Project.

2. Контекстное меню созданного тестового проекта -> Properties (свойства).

В появившемся окне: Linker(Компановщик) -> Input(Ввод) -> Additional Dependencies (Дополнительные зависимости) -> добавить в строку **Source.obj**; (имя .cpp файла + .obj) (не забыть поставить точку с запятой).

3. Необходимо прописать путь к файлу Source.obj. Для этого:

Оставаясь в Input(Ввод), переместитесь в General (Общие) -> Additional Library Directories (Дополнительные директории (папки?) библиотек): щёлкаем по строке с путями, далее -- на стрелку справа -> Edit (Редактировать) -> Добавляем новый путь (жёлтая папка в правом верхнем углу окна), прописываем: **\$(SolutionDir)\ConsoleApplication1\\$(Configuration)** (ConsoleApplication1 - имя Вашего проекта с кодом, которой необходимо тестировать) -> OK.

4. В файле unittest1.cpp проекта тестов необходимо прописать путь к .h файлу: **#include "..\ConsoleApplication1\Source.h"**.

5. Контекстное меню Решения -> Properties (Свойства) -> Startup Project (Запуск проекта) -> выбрать Multiple startup projects (Множественный запуск проектов): сначала должен компилироваться Console Application1, далее - UniTest1 -> OK.

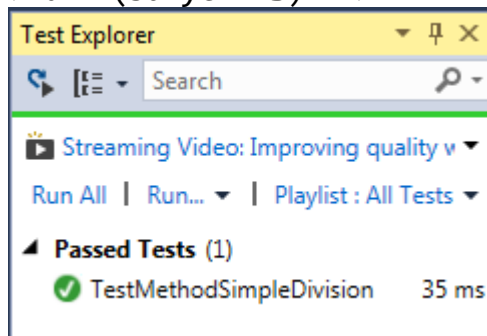
6. Пишем простенький тест в unittest1.cpp:

```
TEST_METHOD(TestMethodFindSquare)
{
    Assert::AreEqual(20, findSquare(10, 2));
}
```

Обязательно даём тесту разумное имя (тут TestMethodFindSquare).

После Assert:: выбираем подходящую функцию.

7. Время запустить тесты: в верхней строке меню Test (Тестирование) -> Run (Запустить) -> All Tests (Все тесты). Появится окошко:



Ура! Все тесты пройдены успешно.