

## Добавление в проект фреймворка googletest

- Распаковываем архив с исходниками в папку с проектом, при этом файл "gtest-all.cc" должен оказаться рядом с Вашими файлами "\*.cpp", а файл "gtest.h" — в том же каталоге в подкаталоге "gtest". Если Вы хотите взять файлы из официального источника, то в архиве "gtest-XXX.zip" они находятся в подкаталоге "fused-src".
- В окошке "Solution explorer" правой кнопкой щелкаете на папке "Source files", выбираете "Add — Existing item" и добавляете файл "gtest-all.cc".
- Там же можно "Add — New item | \*.cpp file" и добавить новый файл для тестов. Но можно добавлять тесты в любой \*.cpp-файл.
- Подключаем header `#include "gtest/gtest.h"` в тот файл, где будут тесты
- Создаем пробный тест (в том же файле, вообще тесты принято держать в отдельном файле)

```
TEST(TestTestCase, TestTest) {  
    EXPECT_TRUE(false);  
}
```

- Создаем тестирующий main (вместо основного, основной пока можно закомментировать)
- ```
GTEST_API_ int main(int argc, char **argv) {  
    testing::InitGoogleTest(&argc, argv);  
    return RUN_ALL_TESTS();  
}
```

Запускаем. Наслаждаемся преимуществами модульного тестирования. Обратим внимание, что если тесты не проходят (а у нас так сейчас), то в окне "Output", где отображается результат выполнения программы, появится строчка типа:

```
i:\dropbox\visual studio 2010 projects\unittest\unittest\unittest.cpp(31): error: Value of: false  
Actual: false  
Expected: true
```

Щелкнув на нее мы сразу попадем на сработавший EXPECT.

### Как именуются тесты

Тесты объединяются в наборы (test cases), кроме того, каждый тест имеет уникальное имя. Имя набора — первый аргумент, имя теста — второй при объявлении теста. Каждый тест может содержать произвольное количество проверок и любые другие операторы.

### Какие существуют проверки

Во-первых, проверки бывают EXPECT\_\* и ASSERT\_\*. Первые при провале не прерывают тестирование, а позволяют продолжить (так за один запуск можно выловить несколько ошибок. Вторые же при провале сразу завершают тестирование (нужно, если провал данного теста означает, что все тесты ниже бессмысленны).

Во-вторых, можно проверять:

- 1) ASSERT\_TRUE() / ASSERT\_FALSE — на истинность и ложность аргумента
- 2) ASSERT\_EQ(,) / ASSERT\_NE(,) / ASSERT\_LT(,) / ASSERT\_LE(,) / ASSERT\_GT(,) / ASSERT\_GE(,) — равенство, неравенство, больше, больше или равно, меньше, меньше или равно для пары аргументов
- 3) ASSERT\_STREQ(,) / ASSERT\_STRNE(,) — проверка на равенство/неравенство C-строк.

(те же проверки возможны и с EXPECT\_)

Подробнее можно прочитать здесь: [http://code.google.com/p/googletest/wiki/V1\\_6\\_Primer](http://code.google.com/p/googletest/wiki/V1_6_Primer)