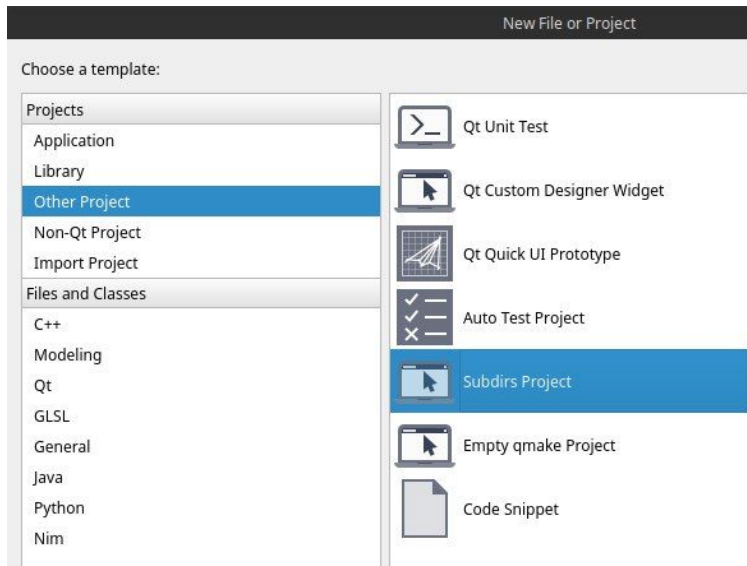
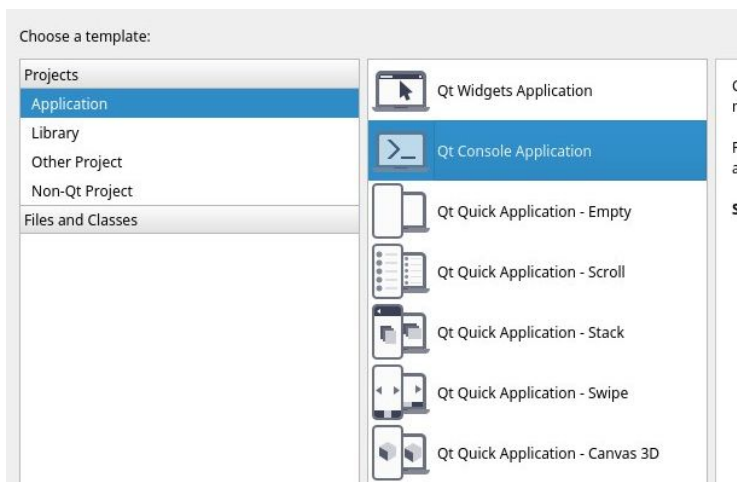


# Юнит тестирование в среде Qt

Создаем новый проект. Обычно в примерах или уроках на Хабрахабре приводят реализацию тестирования, когда тесты и непосредственно сам тестируемый код находятся в одном проекте, но мы пойдем другим путем.

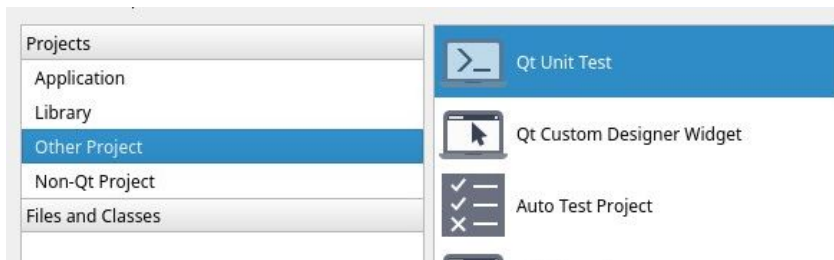


Выбираем Subdirs project, это позволяет хранить проект в древовидной структуре, когда проект с тестируемым кодом и тесты находятся в параллельных проектах (как в Visual Studio). Далее идет стандартная процедура создания проекта.



В завершении Qt предложить создать первый подпроект. Для наглядности каждый из тестируемых классов будет являться отдельным проектом консольного приложения. На стадии Project Location не обязательно создавать поддиректорию, Qt сделает это сам. Создаем тестируемые подпроекты: ArrayProcessor, StringFormatter, Triangle.

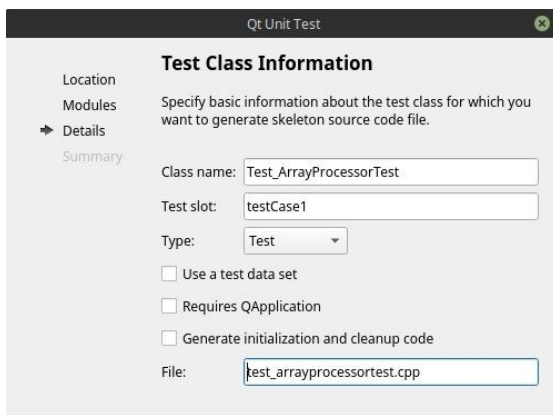
Именно из-за выпадающего списка “Add as subproject” проект создавался по шаблону Subdirs project, если создать просто консольное/GUI приложение, данный элемент недоступен.



Далее добавим в проект сами юнит тесты.

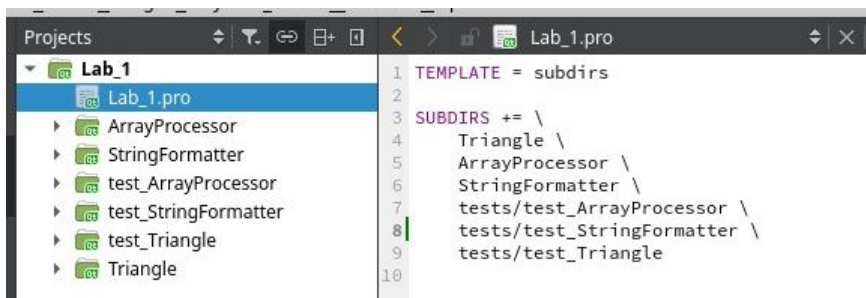
Я решил использовать для тестовых классов префикс "test\_" и создавать их в поддиректории tests. На один класс - один класс-тест.

На этапе выбора необходимых модулей Qt сам добавит необходимые QtCore и QTest.



Тут я предпочел убрать префикс "tst\_" и слово "Test" в конце.

Повторяем процедуру для оставшихся классов.



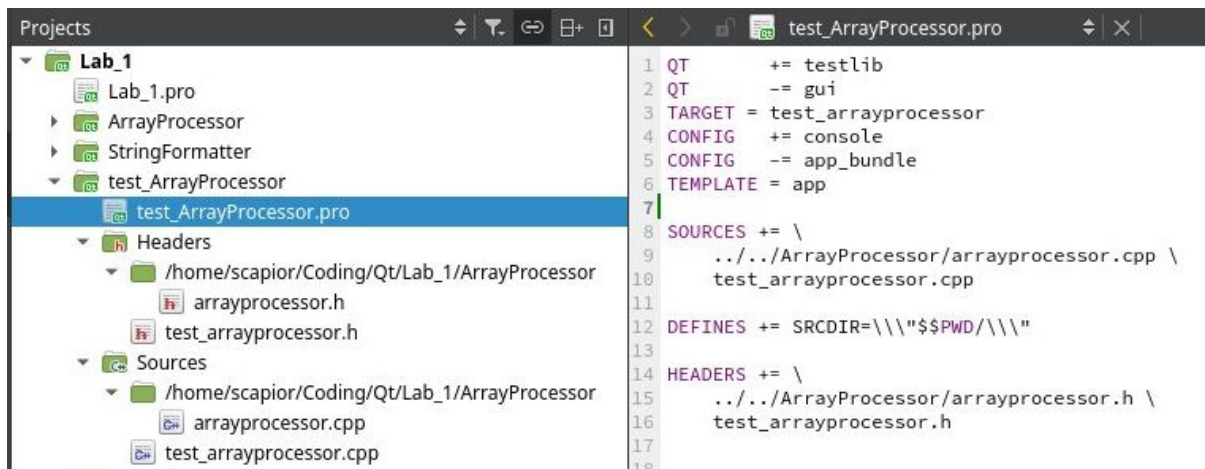
Структура проекта на данный момент должна выглядеть примерно так.

Начинаем реализацию тестов. Один тест - это один приватный слот тестового класса. На каждый метод тестируемого класса свой тест.

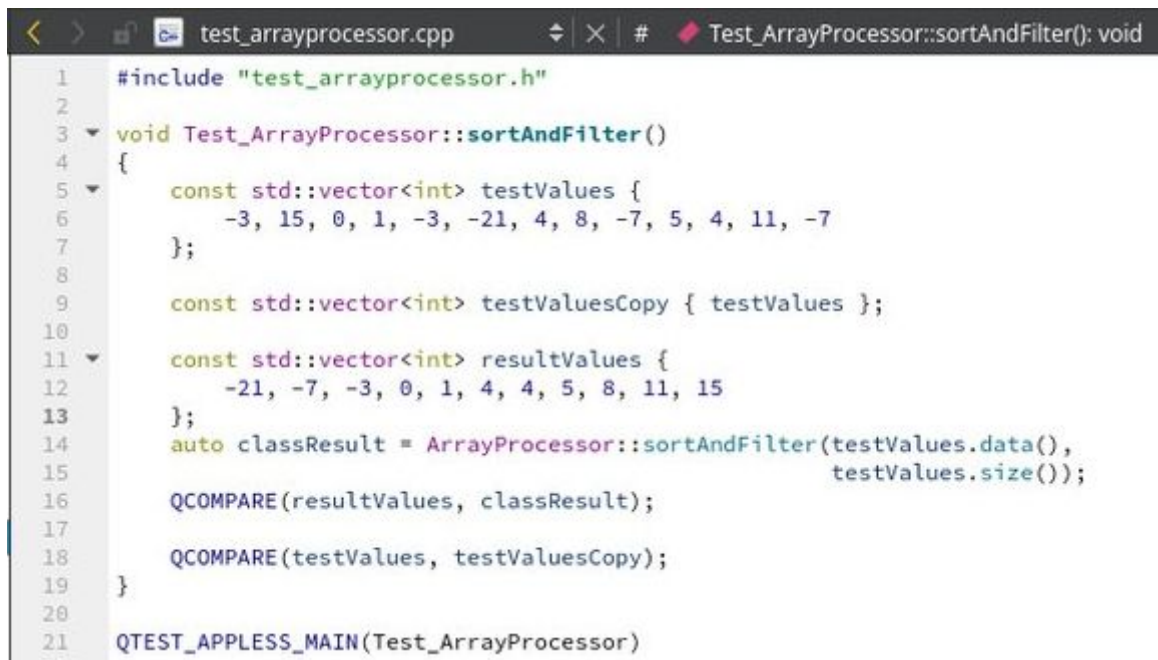


Qt генерирует тестовый класс в .cpp файле, разобьем его на .h и .cpp. Так, как тестируемый класс содержится в другом проекте, необходимо добавить зависимости через контекстное меню "Add Existing Files".

Структура тестового проекта и содержание .pro файла:

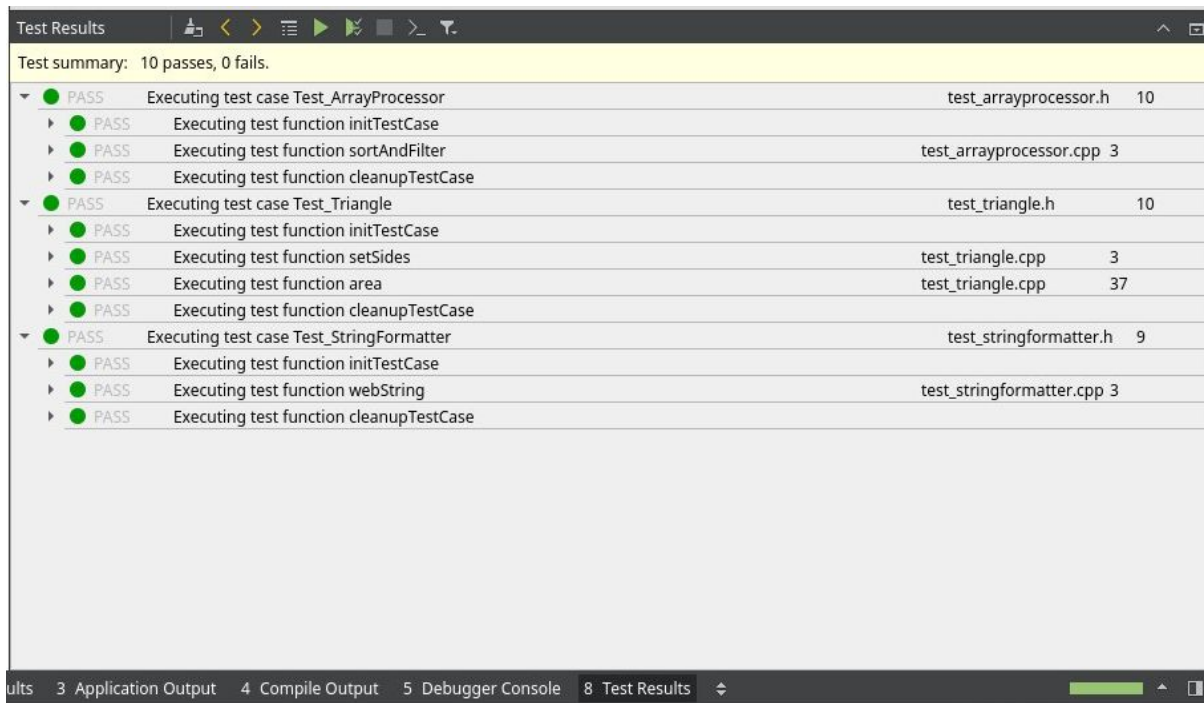


Реализация теста:



Уже на данном этапе, или после того как тесты написаны, можно их запустить. на панели меню: Tools->Tests->Run All Tests.

Если проекты удачно собрались, Qt выведет результаты тестов:



initTestCase и cleanupTestCase это служебные методы, необходимые Qt для инициализации и очистки теста.

Литература для более подробного изучения:

- 1) Макс Шлее Qt 5.3 - страница 658
- 2) Документацию Qt <http://doc.qt.io/qt-5/qttest-index.html>
- 3) Пример реализации тестирования в проекте, разбитого на подпроекты <http://blog.davidecoppola.com/2017/11/cpp-unit-testing-with-qt-test-advanced>



Успешных тестов тебе, сенпай.