

# Задание на уд

## TL;DR

Есть имеющийся проект на CMake, который даже компилируется. Необходимо покрыть его код тестами

## Предыстория

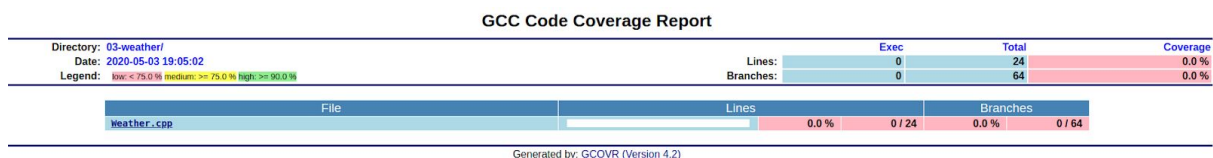
У нас есть работающий проект, теперь мы хотим, чтобы он тестировался!

Вашей целью является использовать готовый репозиторий, в котором необходимо реализовать тесты.

## Алгоритм действия

1. Клонировать репозиторий: <https://github.com/akhtyamovpavel/TechProgTemplates> (он собирается под Linux и Mac).
2. Переключиться на ветку task5
3. Выполните установку пакетов boost:
  - a. Ubuntu: `sudo apt-get install libboost-all-dev`
  - b. MacOS: `brew install boost`
4. Выполните установку openssl:
  - a. Ubuntu: `sudo apt-get install openssl libssl-dev`
  - b. MacOS: `brew install openssl@1.1`
5. Создайте папку build, перейдите в нее
6. Выполните `cmake -DCMAKE_BUILD_TYPE=Debug ..`
7. После этого найдите версию интерпретатора python в выводе, выполните:
  - a. `./install_deps.sh python<your python version> (3.6, 3.8, ...)`
8. Далее запустите сборку проекта:
  - a. `make -j<num cores + 1>`
  - b. `make coverage_report` - запуск покрытия тестами
9. После этого у вас должно появиться несколько html-файлов и xml-файлов, нас интересуют файлы 01-coverage.html, 02-coverage.html и 03-coverage.html, это файлы покрытия кода тестами.
10. Видно, что код покрытия составляет около 0 по линиям. Необходимо сделать так, чтобы процент покрытия был 100% в первом случае, 100% во втором случае, 80% в третьем случае. Если это получится сделать, то задание сдано!

## Скриншоты code coverage



## Что тестируется

1. В первом примере тестируется пример сложения чисел, а также определение, является ли год високосным или нет.
2. Во втором примере тестируется аналог консольной утилиты tree.
3. В третьем примере тестируется adapter для сервисов погоды, который умеет возвращать текущее значение температуры и текстовое сообщение о разнице температур.

## Где писать код

1. Первая часть: AddTestCase, LeapTestCase
2. Вторая часть: TreeTestCase
3. Третья часть: WeatherTestCase, WeatherMock

## Задание на хор

Возьмите проект <https://github.com/akhtyamovpavel/Balda> - в нем необходимо будет протестировать код с точки зрения системного тестирования.

Необходимо будет написать следующего рода тесты:

- Установлено виртуальное окружение, в нем стоят необходимые пакеты
- Все миграции проведены успешно
- Проект успешно запускается

Далее необходимо будет запустить протестировать основные сценарии:

- Регистрация пользователя
- Логин пользователя
- Проверка того, что при проведении некоторых сценариев изменяется поле: был некоторое время назад в сети
- Проверка того, что один пользователь не может стартовать игру
- Проверка того, что один пользователь может начать игру с ботом

## Задание на отл

В том же проекте необходимо будет провести интерактивное тестирование на нескольких пользователей. Для этого необходимо будет проэмулировать среду исполнения для нескольких пользователей, зарегистрировать их в системе, залогинить их (используя код с предыдущих сценариев) и написать простейших ботов, которые позволяют играть в игру (в версии на QT (qt-src) есть примитивные сценарии игр, слова лежат в заранее определенном словаре)

Важно протестировать следующие сценарии и написать сценарии использования для:

- два пользователя решили поиграть в игру
- три пользователя решили поиграть в игру - нельзя, чтобы они одновременно пошли играть

Важно отыграть и продемонстрировать хотя бы одну игру, чтобы удостовериться, что рейтинг пользователей изменился.

Дополнительно необходимо оттестировать таймер - при прохождении отсечки ход должен перейти к другому пользователю.

На отл(10) стоит реализовать поведение ботов разных уровней сложности и проверить, что более крутой пользователь будет почти всегда выигрывать менее крутого пользователя (хотя бы в 90% случаев).

## Примечания

За нахождение багов - плюшки! Просьба делать Issue к проекту с указанием того, как можно воспроизвести баг!

## Литература

<https://www.istqb.org/downloads/send/2-foundation-level-documents/281-istqb-ctfl-syllabus-2018-v3-1.html> - стандарт тестирования по istqb.org

- Harry Percival - Obey The Testing Goat - книга про TDD
- <https://github.com/hjwp/book-example> - примеры из этой книги

<https://github.com/akhtyamovpavel/TestingRepo> - репозиторий, где можно найти простые примеры по тестированию

Фреймворк для тестирования на C++:

- <https://github.com/google/googletest>

Тестирование на Python:

- <https://docs.python.org/3/library/unittest.html> - unittest
- <https://docs.djangoproject.com/en/3.1/topics/testing/tools/> - Django Testing Tools
- <https://docs.python.org/3/library/unittest.mock.html> - mock библиотека для Python

## Полезная информация

- <https://github.com/akhtyamovpavel/TestingRepo> - примеры использования gtest
- <https://github.com/google/googletest/blob/master/googlemock/README.md> - google mock
- <https://github.com/google/googletest/blob/master/googletest/docs/primer.md> - google test
- <https://openweathermap.org/forecast5> - прогноз погоды, API
- <https://whoshuu.github.io/cpr/> - документация по отправке запросов
- [https://www.boost.org/doc/libs/1\\_66\\_0/libs/filesystem/doc/reference.html](https://www.boost.org/doc/libs/1_66_0/libs/filesystem/doc/reference.html) - документация по boost filesystem
- <https://github.com/nlohmann/json> - документация по json