Lab of Object-Oriented Programming:

Unit Test

黃威、陳岳紘、邱彥翔

2022

Unit Test

- 用來測試**程式碼邏輯**的程式
  - 編譯器找不到的邏輯錯誤
- 通常以**函式**為單位做測試

Google Test

- Google 開發的 C++ 用 Unit Test 工具
- 提供各種測試函式的工具
- 在 UNIX 上建議使用 CMake
    - 用來自動產生 Makefile

CMake

```
cmake_minimum_required(VERSION 3.1.0)
project(sphere)

add_library(sphere sphere.cc)
```

CMake

```
cmake_minimum_required(VERSION 3.1.0)
project(unittest)

include_directories(../src)

add_executable(utest test_test.cc)
target_link_libraries(utest gtest_main)
target_link_libraries(utest sphere)
enable_testing()
```

範例專案

- clone 範例專案
  - https://gitea.cglab.cs.nccu.edu.tw/yhchen/gtest_example
- 依照 Readme 操作
  - 建議在工作站上操作

範例專案

```
Running main() from /home1/student/gradee110/ge11021/gtest/example/build/googletest-src/googletest/src/gtest_main.cc
[==========] Running 1 test from 1 test suite.
[----------] Global test environment set-up.
[----------] 1 test from TestSuiteName
[ RUN      ] TestSuiteName.TestName
[       OK ] TestSuiteName.TestName (0 ms)
[----------] 1 test from TestSuiteName (0 ms total)

[----------] Global test environment tear-down
[==========] 1 test from 1 test suite ran. (0 ms total)
[  PASSED  ] 1 test.
```

範例專案

- unittest/test_test.cc
  - TestSuiteName、TestName 是測試的名字
  - EXCEPT_EQ 檢查裡面的參數是否相同
- 一個檔案中可以有多個 TEST

```
#include "gtest/gtest.h"

TEST(TestSuiteName, TestName) {
    EXPECT_EQ(1, 1);
}
```

檢查語法

- EXPECT 類
  - 檢查結果是否符合預期
- ASSERT 類
  - 如果結果不符合預期就會 **立刻停止** 測試

檢查語法

- 判斷真假
  - `EXPECT_TRUE(condition)`
  - `EXPECT_FALSE(condition)`

檢查語法

- 數字比較
  - `EXPECT_EQ(v1, v2)`
  - `EXPECT_NE(v1, v2)`
  - `EXPECT_LT(v1, v2)`
  - `EXPECT_LE(v1, v2)`
  - `EXPECT_GT(v1, v2)`
  - `EXPECT_GE(v1, v2)`

檢查語法

- 字元、字串比較
  - 分大小寫
    - EXPECT_STREQ(str1,str2)
    - EXPECT_STRNE(str1,str2)
  - 不分大小寫
    - EXPECT_STRCASEEQ(str1,str2)
    - EXPECT_STRCASENE(str1,str2)

程式碼覆蓋率

- 計算測試程式的覆蓋率
  - 行數覆蓋率
  - 函式覆蓋率
- 使用 gcov 工具
  - 內建在 gcc、g++ 中

gcov

- 產生覆蓋率檔案
  - 執行程式時會統計覆蓋率
  - 使用 CMake 時路徑在 `build/資料夾位置/CMakeFiles/專案名稱.dir/檔案名稱.cc.gcno`
  - 執行 `gcov <gcno檔位置>` 查看覆蓋率

gcov

```
set(GCC_COVERAGE_COMPILE_FLAGS "-g -O0 -coverage -fprofile-arcs -ftest-coverage")
set(GCC_COVERAGE_LINK_FLAGS    "-coverage -lgcov")
set(CMAKE_CXX_FLAGS  "${CMAKE_CXX_FLAGS} ${GCC_COVERAGE_COMPILE_FLAGS}" )
set(CMAKE_EXE_LINKER_FLAGS  "${CMAKE_EXE_LINKER_FLAGS} ${GCC_COVERAGE_LINK_FLAGS}" )
```

Exercise 9

- https://gitea.cglab.cs.nccu.edu.tw/yhchen/gtest_exercise
- 繳交 .zip 壓縮檔包含以下檔案至 moodle：
  - test_test.cc
  - sphere.cc
  - 執行 utest 結果的截圖
  - sphere.cc 覆蓋率的截圖
- 有缺少或有多餘的檔案會扣分

Exercise 9

```
class Sphere {
  public:
   Sphere();
   Sphere(float ox, float oy, float oz, float rad);
   float SurfaceArea();
   float Volume();
   void setOrigin(float ox, float oy, float oz);
   void setRadius(float rad);
   float* getOrigin();
   float getRadius();
   bool intersect(Sphere&);
   friend std::ostream& operator<<(std::ostream&, const Sphere&);

  private:
   float orig_x, orig_y, orig_z;
   float radius;
};
```

参考連結

- Google Test
  - https://github.com/google/googletest
  - https://shengyu7697.github.io/googletest/
- gcov
  - https://stackoverflow.com/questions/37957583/how-to-use-gcov-with-cmake

Assign 7

Assign7

項目	配分
有交	20
可編譯	10
< int & char queue > enqueue, dequeue	30
< int & char queue > copy constructor, destructor	30
throw three exceptions	10

Your class should throw exceptions on at least three occasions:

1. when an internal error, such as incorrect head or tail pointers in the queue,
2. when you can not allocate more memory in enqueue operations,
3. when you try to dequeue an empty queue.