**Lab 1 Report**

劉宸宗

106598047

2018/3/18

1. **Test Plan**
   1. **Test requirements**

Lab 1 的需求是 (1) 從GeoProject的6個class中挑出15個methods,(2) 根據經驗，對所選的methods來設計單元測試案例, (3) 開發測試腳本,實作測試案例,(4) 執行測試腳本, 並且(5) 呈現測試結果.

根據 the statement coverage criterion, the **Lab1的測試需求**是去針對每個所選的method，設計測試案例。每個*method statement*會被至少一個測試案例涵蓋到*。*此外 *最小statement coverage為 80%*.

* 1. **Strategy**

為了達到Section 1的測試需求, 在此提出的strategy 是

1. 選擇容易理解並且輸入輸出含有primitive type的public methods (如果有的話).
2. 最初設定最小的statement coverage 為 50% ,此外如果時間足夠的話，會將statement coverage往上提升.
3. 盡可能快速的學習必要的技術及工具.
4. 為了要設計測試案例，需要考慮到
   1. 輸入參數要盡可能是**valid values** 及**其他可想得到的組合**.
   2. 輸入參數的**boundary values**.
   3. **Test activities**

為了實現以上提到的 strategy, 以下的 activities 會被執行.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Activity Name** | **Plan hours** | **Schedule Date** |
| 1 | 研究 GeoProject | 2 | 2018/3/16 18:00-20:00 |
| 2 | 學習JUnit | 1 | 2018/3/16 20:00-21:00 |
| 3 | 設計,、實作、和執行 test cases for the selected methods | 12 | 2018/3/17 10:00-12:00  2018/3/17 14:00-17:00  2018/3/17 18:30-20:30  2018/3/18 10:00-13:00  2018/3/18 15:00-17:00 |
| 4 | 完成Lab1 report | 2 | 2018/3/18 18:00-20:00 |

* 1. **Success criteria**

所有的測試案例必須被通過，還有*the statement coverage 至少到達80%.*

1. **Test Design**

為了達到section 1.1所提的測試需求, 以下的methods 會被拿來做測試 以及產生相對應的測試案例.

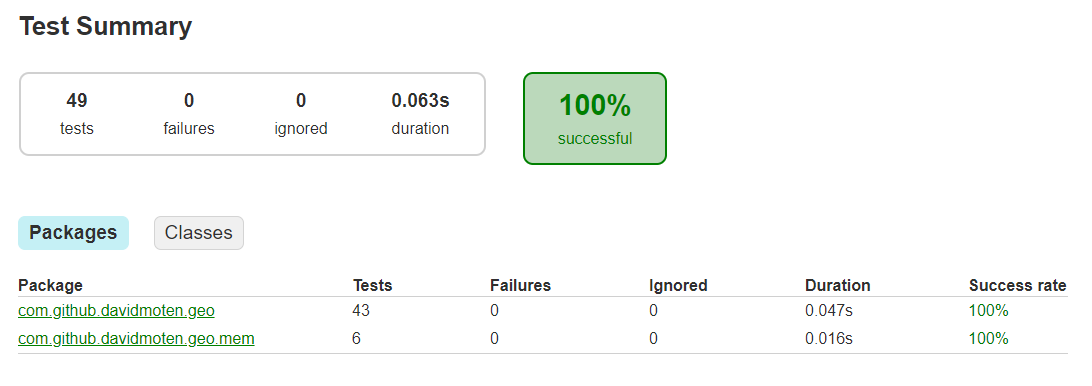
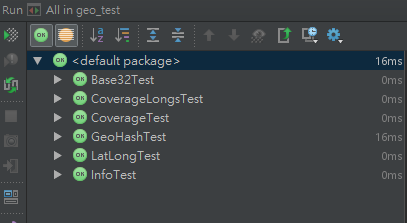
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Class** | **Method** | | **Test Objective** | **Inputs** | **Expected Outputs** |
| 1 | Base32 | | encodeBase32(i,length) | 功能測試 | i:75324, length:4 | 29jw |
| 2 | Base32 | | encodeBase32(i) | 功能測試 | i:-75324 | -29jw |
| 3 | Base32 | | decodeBase32(hash) | 功能測試 | hash:29jw | 75324 |
| 4 | Base32 | | getCharIndex(ch) | 功能測試 | ch: s | 24 |
| 5 | Base32 | | padLeftWithZerosToLength(s,length) | 功能測試 | s:12,  length:3 | 012 |
| 6 | Coverage | | getHashes() | 功能測試 | hashes: [world, hello] | [world,hello] |
| 7 | Coverage | | getRatio() | 功能測試 | Ratio:1.5 | 1.5 |
| 8 | Coverage | | getHashLength() | 功能測試 | hashes: [world, hello] | 5 |
| 9 | Coverage | | toString() | 功能測試 | hashes: [world, hello]  Ratio:1.5 | Coverage [hashes=[world, hello], ratio=1.5] |
| 10 | Coverage | | Coverage(hashes,ratio) | 功能測試 | hashes: [world, hello]  Ratio:1.5 | Coverage [hashes=[world, hello], ratio=1.5] |
| 11 | CoverageLongs | | getHashes() | 功能測試 | hashes: [75324,75325] | [75324,75325] |
| 12 | CoverageLongs | | getRatio() | 功能測試 | Ratio:1.5 | 1.5 |
| 13 | CoverageLongs | | getHashLength() | 功能測試 | hashes: [75324,75325]  count:2 | 12 |
| 14 | CoverageLongs | | getCount() | 功能測試 | count:2 | 2 |
| 15 | Info | | value() | 功能測試 | value:  "Info Value" | Info Value |

1. **Test Implementation**

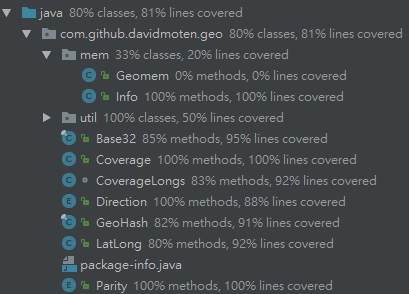
採用JUnit 4來實現Section 2所寫到的測試案例設計. 以下為其中3個測試案例. 其餘的測試案例可以在 [link](https://stv.csie.ntut.edu.tw/106598047/GeoProject)中找到.

|  |  |  |
| --- | --- | --- |
| **No.** | **Test method** | **Source code** |
| 1 | encodeBase32() | String encode = Base32.encodeBase32(75324, 4);  assertEquals("29jw", encode); |
| 2 | encodeBase32\_withoutParamLength() | String encode = Base32.encodeBase32(-75324);  assertEquals("-0000000029jw", encode); |
| 3 | decodeBase32() | long decode = Base32.decodeBase32("29jw");  assertEquals(75324, decode); |

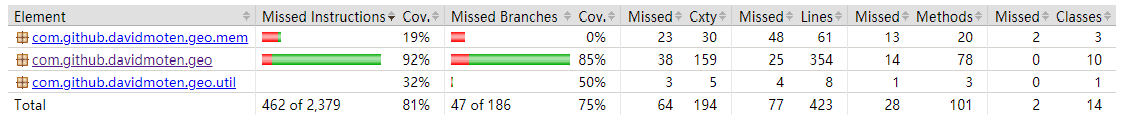
1. **Test Results**
   1. **JUnit test result snapshot**



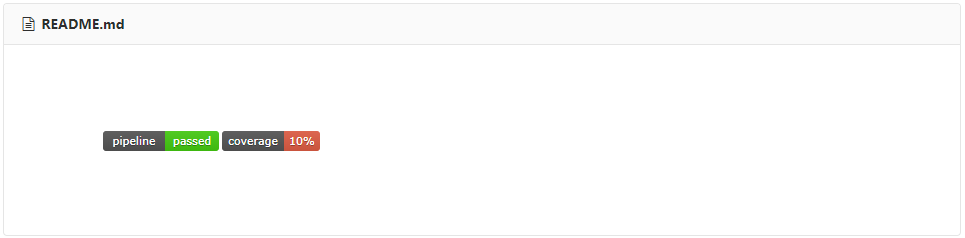
* 1. **Code coverage snapshot**
* Coverage of each selected method



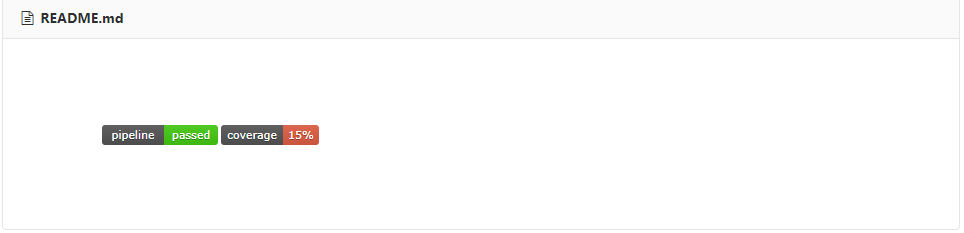
* Total coverage



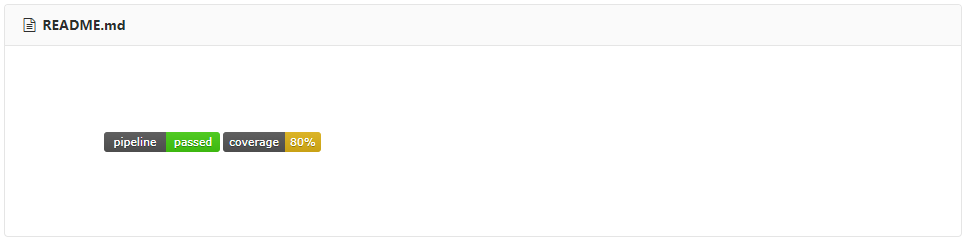
* 1. **CI result snapshot (3 iterations for CI)**
* CI#1



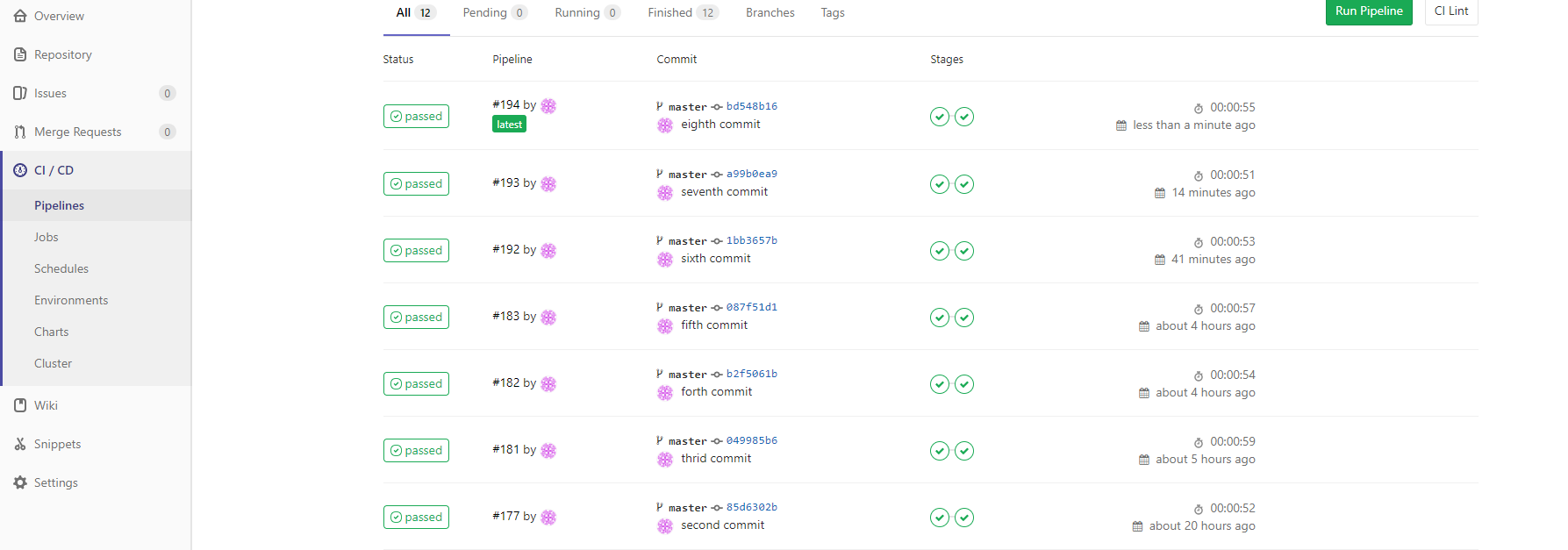
* CI#2



* CI#3



* CI Pipeline



1. **Summary**

In Lab 1, **15 測試method採用 JUnit 來設計與實作**。測試在CI中被執行了3次，而且**這15個測試method的結果都通過**。**測試總覆蓋率為 80%.** 因此, 滿足Section 1提到的測試需求。在這次的Lab中，我學到了如何使用JUnit來做測試，思考如何設計測試案例及其輸入值來達到較高的覆蓋率，還有文件的撰寫格式。