**Lab 1 Report**

謝狄烽

110598087

2022/03/01

**Contents**

[**1** **Test Plan** 2](#_Toc98199108)

[**1.1** **Test requirements** 2](#_Toc98199109)

[**1.2** **Strategy** 2](#_Toc98199110)

[**1.3** **Test activities** 2](#_Toc98199111)

[**1.4** **Success criteria** 2](#_Toc98199112)

[**2** **Test Design** 2](#_Toc98199113)

[**3** **Test Implementation** 4](#_Toc98199114)

[**4** **Test Results** 5](#_Toc98199115)

[**4.1** **JUnit test result snapshot** 5](#_Toc98199116)

[**4.2** **Code coverage snapshot** 5](#_Toc98199117)

[**4.3** **CI result snapshot (3 iterations for CI)** 6](#_Toc98199118)

[**5** **Summary** 7](#_Toc98199119)

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

The Lab 1 requires to (1) select 15 methods from 6 classes of the SUT (GeoProject), (2) design Unit test cases based on the experience or intuition for the selected methods, (3) develop test scripts to implement the test cases, (4) execute the test script on the selected methods, and (5) report the test results.

In particular, based on the statement coverage criterion, the test requirements for Lab 1 are to design test cases for each selected method so that “each statement of the method will be covered by at least one test case and the minimum statement coverage is 40%”.

* 1. **Strategy**

To satisfy the test requirements listed in Section 1, a proposed strategy is to

1. select those public methods that may be easy or little difficult to understand and may have or may not have primitive types of input and output parameters.
2. set the objective of the minimum statement coverage to be 50% initially and adjust the objective based on the time available.
3. learn the necessary skills and tools as soon as possible.
4. design the test cases for those selected methods by considering
   1. the possible valid values and combinations of the input parameters.
   2. the boundary values of the input parameters.
   3. **Test activities**

To implement the proposed strategy, the following activities are planned to perform.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Activity Name** | **Plan hours** | **Schedule Date** |
| 1 | Study GeoProject | 3 | 2022/03/01 10:30-12:30  2022/03/01 15:00-16:00 |
| 2 | Learn JUnit | 1 | 2022/03/02 15:00-16:00 |
| 3 | Design test cases for the selected methods | 3 | 2022/03/01 16:00-17:00  2022/03/03 16:00-17:00  2022/03/04 13:00-14:00 |
| 4 | Implement test cases | 7.5 | 2022/03/04 14:00-16:00  2022/03/06 14:00-18:00  2020/03/14 20:00-21:30 |
| 5 | Perform test | 1 | 2020/03/14 21:30-22:30 |
| 6 | Complete Lab1 report | 1 | 2020/03/15 10:30-11:30 |

* 1. **Success criteria**

All test cases designed for the selected methods must pass and the statement coverage should have achieved at least 80%.

1. **Test Design**

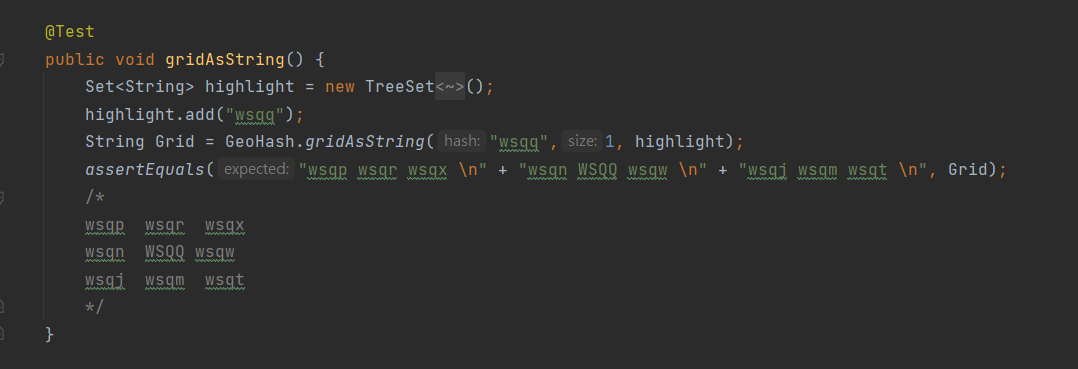
To fulfill the test requirements listed in section 1.1, the following methods are selected and corresponding test cases are designed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Class** | **Method** | **Test Objective** | **Inputs** | **Expected Outputs** |
| **1** | **Base32** | *encodeBase32 (long, int)* | Encode with fixed length | 32769, 7 | 0001001 |
| **2** | *decodeBase32 (Str)* | If I understand how it works | 01001 | 32769 |
| **3** | *getCharIndex (char)* | Throwing exception correctly | ‘a’ | Throw IllegalArgument Exception |
| **4** | **CoverageLongs** | *CoverageLongs (long[], int, double)* | Construct correctly | {30893887395, 30893887396L, 30893887375L},3, 0.5 | “[30893887395, 30893887396, 30893887375], 3, 0.5” |
| **5** | **Coverage** | *Coverage (Set<Str>, double)* | Construct by geohashes | {wsqqmx4, wsqqmx3}, 0.5 | “Coverage [hashes=[ wsqqmx4, wsqqmx3], ratio=0.5]” |
| **6** | **GeoHash** | *adjacentHash (Str, direction, int)* | Move position with step | "wsqqmx4", Direction.TOP, 2 | “wsqqmxd” |
| **7** | *neighbours (Str)* | Get position nearby | "wsqqmx4" | {wsqqmx1, wsqqmx5, wsqqmx6, wsqqmwf, wsqqmx3, wsqqmwc, wsqqmx7, wsqqmwg} |
| **8** | *encodeHash (double, double)* | Generate correct geohash | 25.04386042, 121.53397182 | “wsqqmx474ccw” |
| **9** | *encodeHash (LatLong)* | Handel object correctly | LatLong latlon | wsqqqqqqqebj |
| **10** | *fromLongToString (long)* | Throwing exception correctly | 13089388739600L | Throw IllegalArgument Exception |
| **11** | *hashContains (Str, double, double)* | Check geshash area includes giving lat lon | “wsqqmx4”, 25.04386042, 121.53397182 | *True* |
| **12** | *gridAsString (Str, int, Set<String>)* | Grid nearby geohash correctly | “wsqq”, 1, {wsqq} | wsqp wsqr wsqx  wsqn WSQQ wsqw  wsqj wsqm wsqt |
| **13** | **Geomem** | *add (Info)* | Adding Info object correctly | Info info<String, String> | [Info [lat=25.04, lon=121.53, time=1647265758, value=SID, id=Optional.of(8087)]] |
| **14** | *find(double,double,double,double,long,long)* | Find objects correctly | 26,121,25,122, 1647265750, 1647265760 | Info info<String, String> |
| **15** | **Info** | *Info (double, double, long, T, Optional<R>)* | Construct correctly | 25.0438604, 21.5339717, 1646110492, "value", Optional.of ("1105980") | "Info [lat=25.0438604, lon=21.5339717, time=1646110492, value=value, id=Optional.of (1105980)]" |

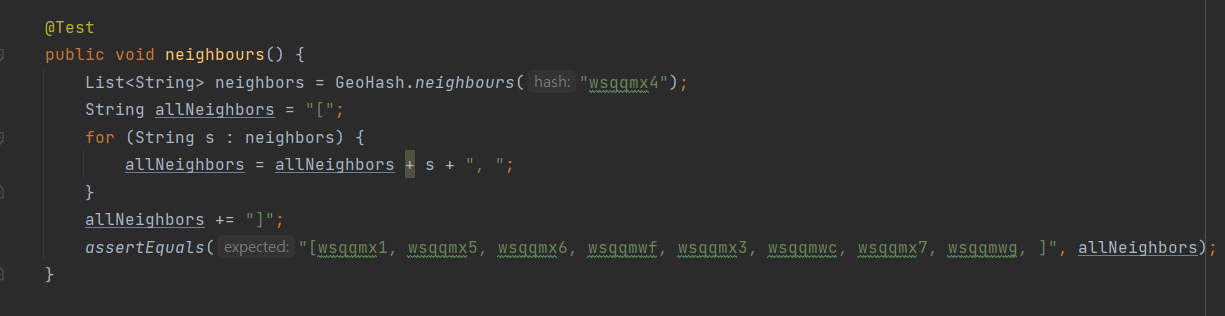
1. **Test Implementation**

The design of test cases specified in Section 2 was implemented using JUnit 4. The test scripts of 3 selected test cases are given below. The rest of test script implementations can be found in JUnit files.

**Test method:***Geohash.gridAsString()*



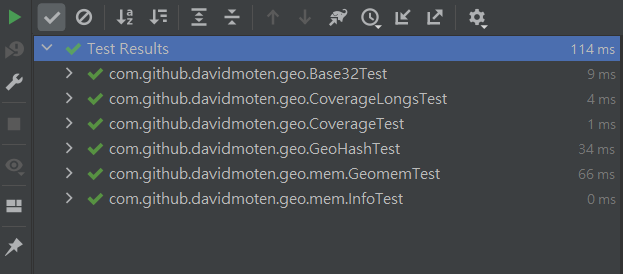
**Test method:***Geohash.neighbours()*

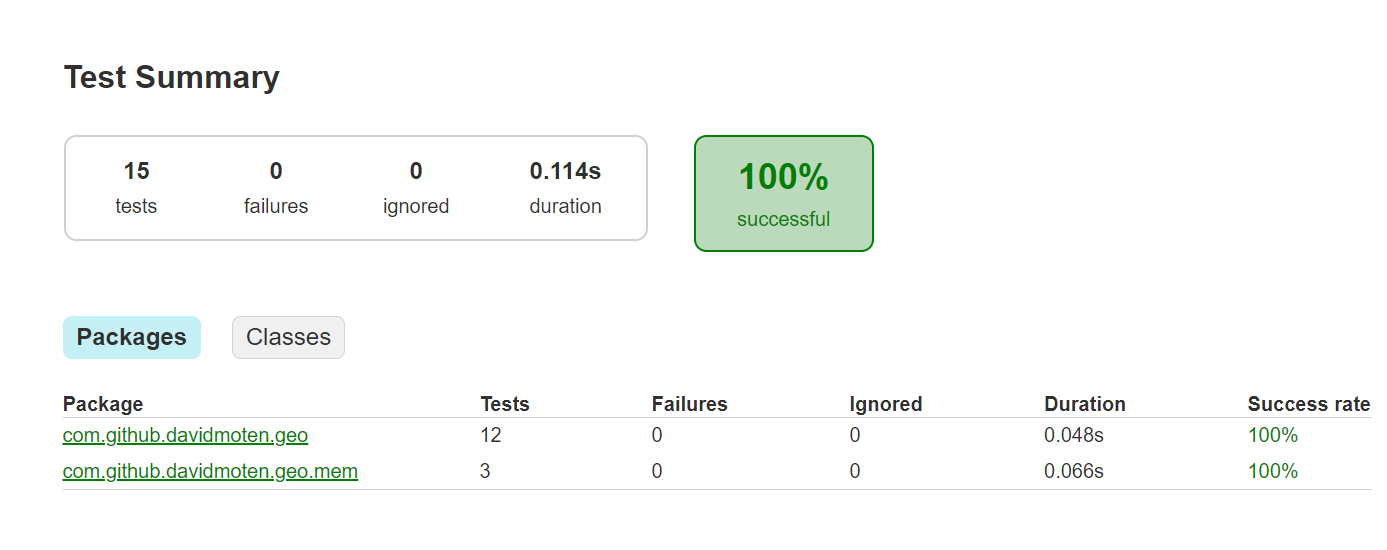


**Test method:***Geomem.find()*

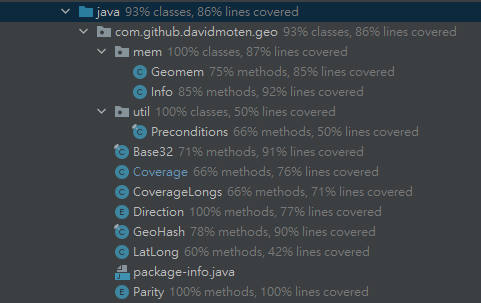


1. **Test Results**
   1. **JUnit test result snapshot**

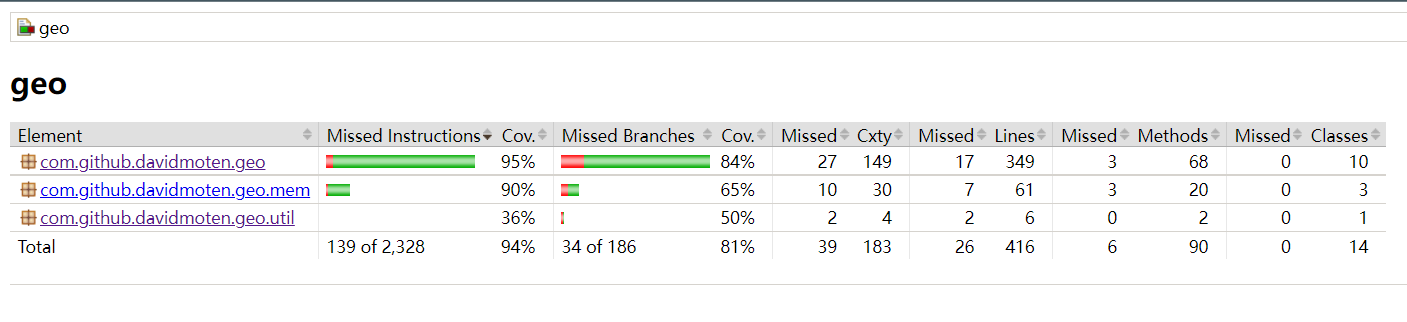




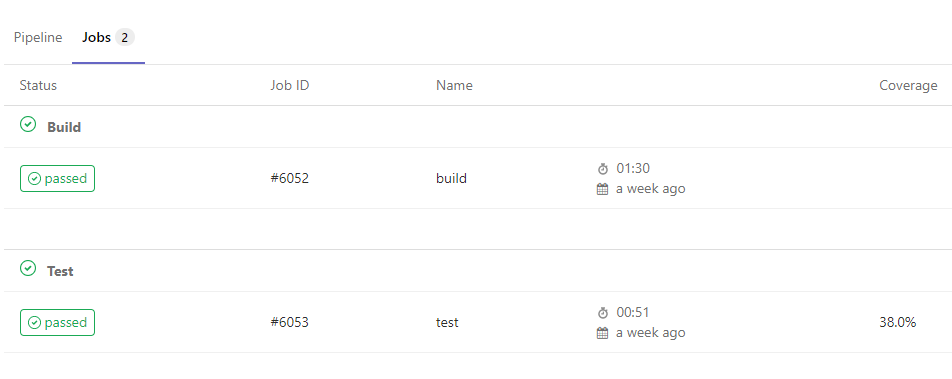
* 1. **Code coverage snapshot**
* **Coverage of each selected method**



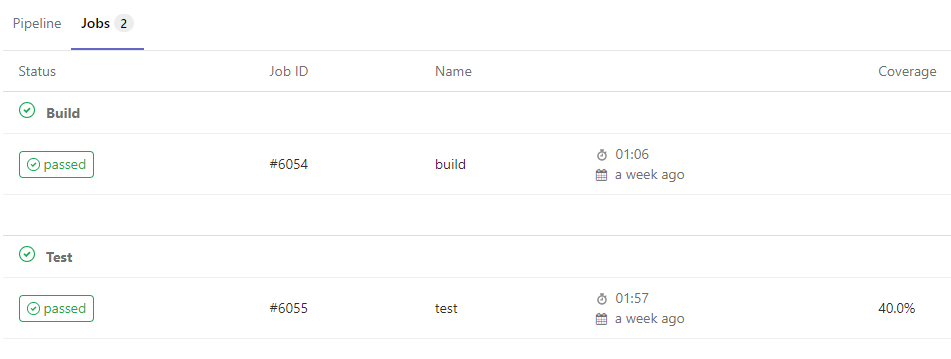
* **Total coverage**



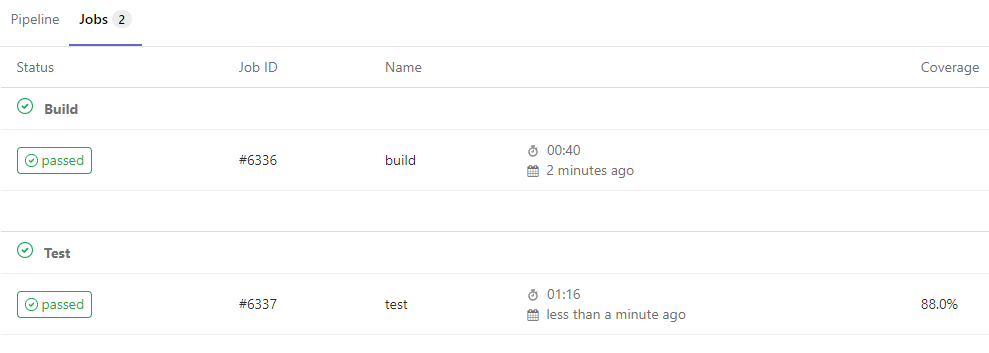
* 1. **CI result snapshot (3 iterations for CI)**
* **CI#1**



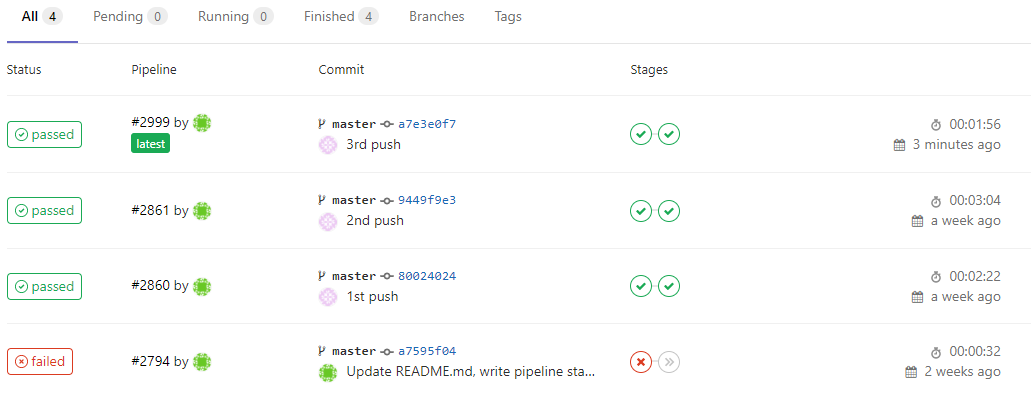
* **CI#2**



* **CI#3**



* **CI Pipeline**



1. **Summary**

In Lab 1, 15 test cases have been designed and implemented using JUnit. The test is conducted in 3 CI and the execution results of the 15 test methods are all passed. The total statement coverage of the test is 81%. Thus, the test requirements described in Section 1 are satisfied. Some lessons learned in this Lab are the experiences of how design the test cases, how to implement the test cases, how to track the data transferring while using different method or passing different object into those ad hoc methods, and how to know whether the test cases were satisfied. And the most important thing is to practice how to test a software step by step.