**Lab 1 Report**

Name: 盧家馨

Student ID: 106598005

Date: 2018/03/17

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 result.

In particular, based on the statement coverage criterion, the **test requirements** for Lab 1 are to design test casesfor each selected method so that “*each statement of the method will be covered by at least one test case* and *try to fulfill coverage as more as possible*”.

* 1. **Strategy**

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

1. select those public methods that are easy to understand and have primitive types of input and output parameters (if possible).
2. set the objective of the minimum statement coverage to be 40% initially and (if necessary) 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 the possible valid values and combinations of the input parameters.
   1. **Test activities**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Activity Name** | **Plan hours** | **Schedule Date** |
| 1 | Study GeoProject | 1 | 2018/03/13 |
| 2 | Learn to use JUnit | 1 | 2018/03/13 |
| 3 | Design test cases for the selected methods | 4 | 2018/03/14 |
| 4 | Implement test cases | 6 | 2018/03/15  2018/03/16  2018/03/17 |
| 5 | Modify test case | 4 | 2018/03/15  2018/03/16  2018/03/17 |
| 6 | Complete Lab1 report | 2 | 2018/03/18 |

* 1. **Success criteria**

All test cases designed for the selected methods must pass*.*

1. **Test Design**

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

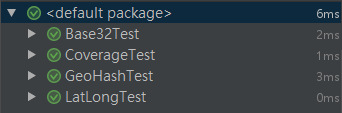
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Class** | **Method** | **Test Objective** | **Inputs** | **Expected Outputs** |
| 1 | Base32 | encodeBase32 | encode | (75324, 4) | 29jw |
| 2 | Base32 | decodeBase32 | decode | 29jw | 75324 |
| 3 | Base32 | getCharIndex | Get Index | 3 | 3 |
| 4 | LatLong | getLat | Get Lat | (2.3, 3.7) | 2.3 |
| 5 | LatLong | getLon | Get Lon | (2.3, 3.7) | 3.7 |
| 6 | LatLong | add | Add Lat Lon | (2.3, 3.7) + (0.5, 0.7) | (2.8, 4.4) |
| 7 | Coverage | getHashes | Get Hash | Set | Same set |
| 8 | Coverage | getRatio | Get Ratio | (set, 10) | 10 |
| 9 | Coverage | getHashLength | Get Hash Length | <12, 34> | 2 |
| 10 | GeoHash | right | Right of GeoHash | 12 | 18 |
| 11 | GeoHash | left | Left of GeoHash | 13 | 11 |
| 12 | GeoHash | Top | Top of GeoHash | 15 | 1h |
| 13 | GeoHash | bottom | Bottom of GeoHash | 20 | 0p |
| 14 | GeoHash | encodeHash | encode | (10, 20) | s3y0zh7w1z0g |
| 15 | GeoHash | decodeHash | Decode | abc | (49.921875, 170.859375) |

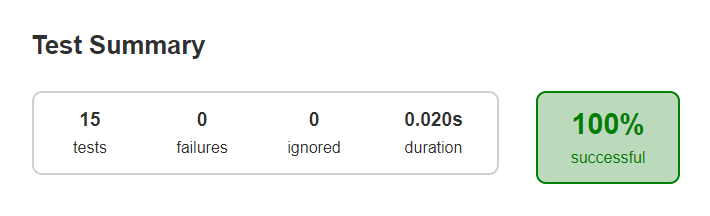
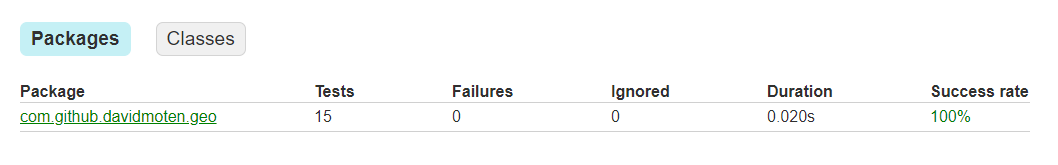
1. **Test Implementation**

The design of test cases specified in Section 2 was implemented using JUnit 4. The test script of 3 selected test cases are given below. The rest of test script implementation can be found in the [link](https://github.com).

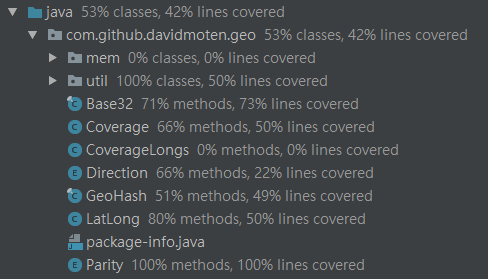
|  |  |  |
| --- | --- | --- |
| **No.** | **Test method** | **Source code** |
| 1 | getHashes () | Set<String> coverage = new HashSet<String>(100); coverage.add("123"); coverage.add("456");  Coverage cov = new Coverage(coverage, 10); *assertNotNull*(cov); *assertEquals*(coverage, cov.getHashes()); |
| 2 | encodeHash () | String encode = GeoHash.*encodeHash*(10,20); *assertNotNull*(encode); *assertEquals*("s3y0zh7w1z0g", encode); |
| 3 | encodeBase32 () | String encode = Base32.*encodeBase32*(75324, 4); *assertNotNull*(encode); *assertEquals*("29jw", encode); |

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

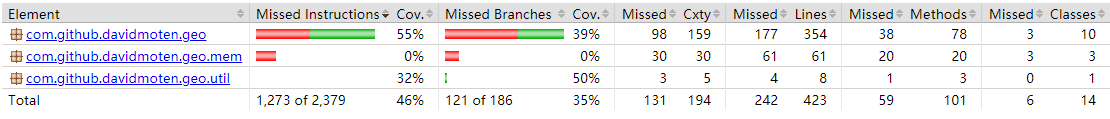


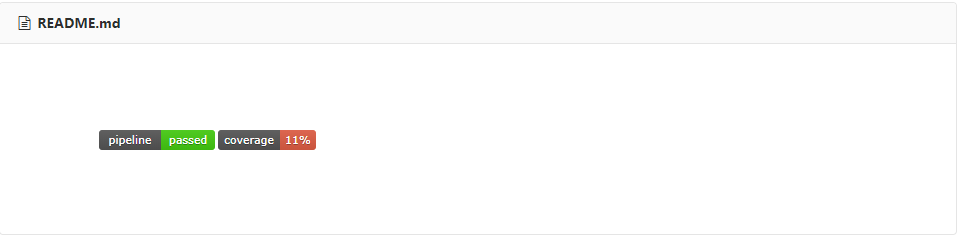
* 1. **Code coverage snapshot**
* Coverage of each selected method



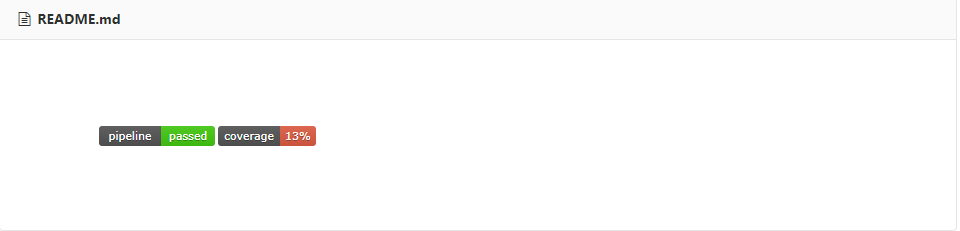
* Total coverage



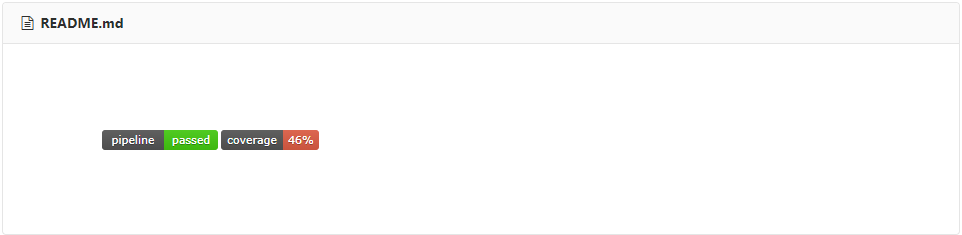
* 1. **CI result snapshot (3 iterations for CI)**
* CI#1



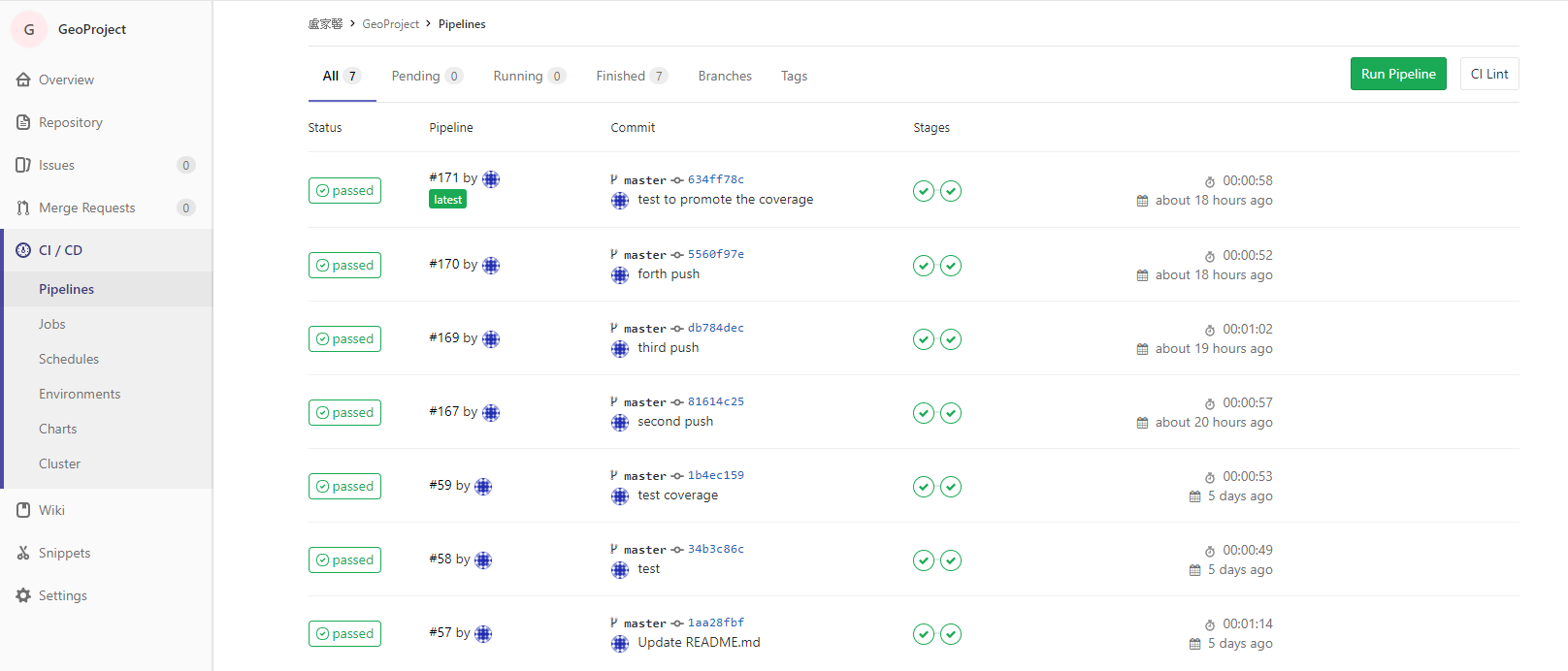
* CI#2



* 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 46%. Thus, the test requirements described in Section 1 are satisfied. Some lessons learned in this Lab are conception of test case and how to design the better test scripts in project.