**Lab 1 Report**

Name 吳承軒

Student ID 106598068

Date 03/19/2018

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 *the minimum statement coverage is 50%*”.

* 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 50% 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
   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 | 2hr | 3/7 |
| 2 | Learn JUnit | 1hr | 3/8 |
| 3 | Design test cases for the selected methods | 3hr | 3/9 |
| 4 | Implement test cases |  |  |
|  | --InfoTest test case | 0.5hr | 3/10 |
|  | --CoverageLongs test case | 1hr | 3/10 |
|  | --Coverage test case | 1hr | 3/10 |
|  | --GeoHash test case | 1hr | 3/11 |
|  | --Direction test case | 0.5hr | 3/11 |
| … | --Base32 test case | 1hr | 3/12 |
|  | --LatLong test cast | 1hr | 3/14 |
| … | Complete Lab1 report |  | 3/18 |

* 1. **Success criteria**

All test cases designed for the selected methods must pass (or "90% of all test cases 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.

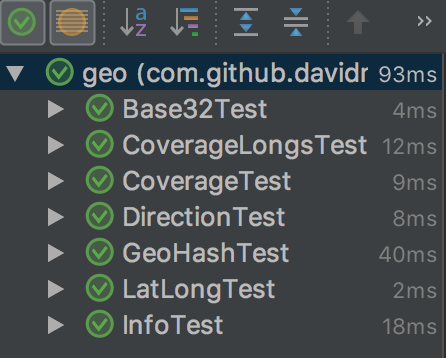
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Class** | **Method** | **Test Objective** | **Inputs** | **Expected Outputs** |
| 1 | Base32 | encodeBase32 | Test encoding | long | string |
| 2 | Base32 | encodeBase32MaxLength | Test encoding | long | string |
| 3 | Base32 | getCharIndex | Test get index | string | int |
| 4 | Coverage | getRatio | Test get hash | set | string |
| 5 | Coverage | getHashLenth | Test get hashlength | set | Int |
| 6 | Coverage | tostring | Test toString | set | string |
| 7 | CoverageLong | getHashes | Test get hashlength | long | Int |
| 8 | CoverageLongs | getRatio | Test getradio | long | string |
| 9 | CoverageLongs | getCount | Test  getCount | long | int |
| 10 | InfoTest | createInfoWithToString | Test create Info | long | string |
| 11 | Direction | create | Test create Direction | null | int |
| 12 | GeoHash | adjacentHash | Test create Border | string | string |
| 13 | GeoHash | encodeHash | Test encodeHash | double | string |
| 14 | GeoHash | decodeHash | Test decodeHash | string | string |
| 15 | LatLong | getLat | Test getLat | double | string |
| 16 | LatLong | getLon | Test getLong | double | string |
| 17 | LatLong | add | Test add | double | string |
| 18 | LatLong | tostring | Test toString | double | String |

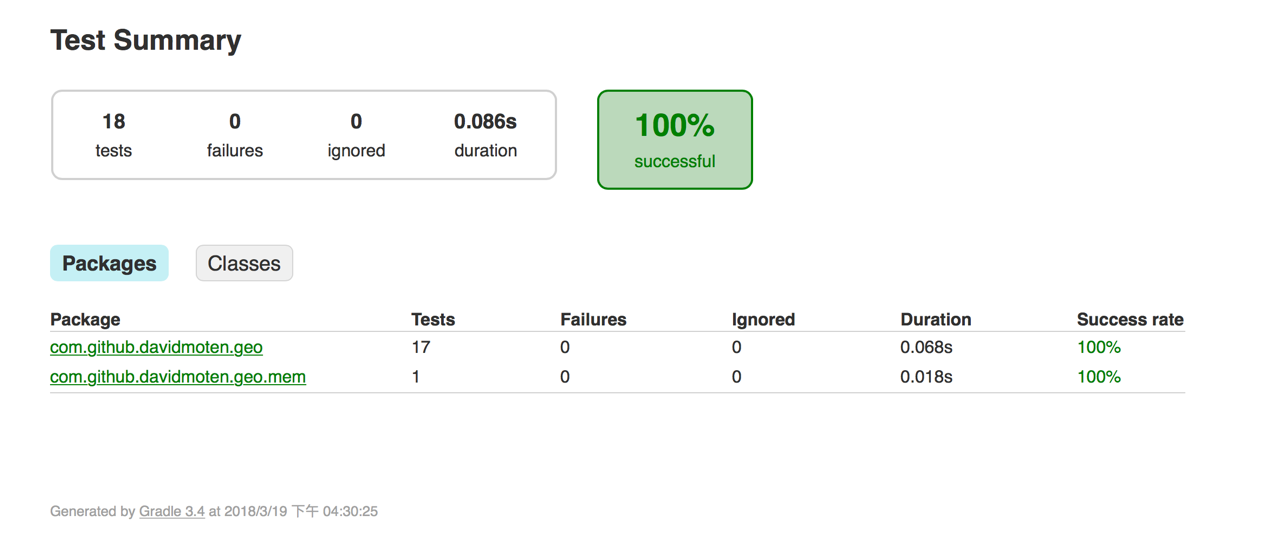
1. **Test Implementation**

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).

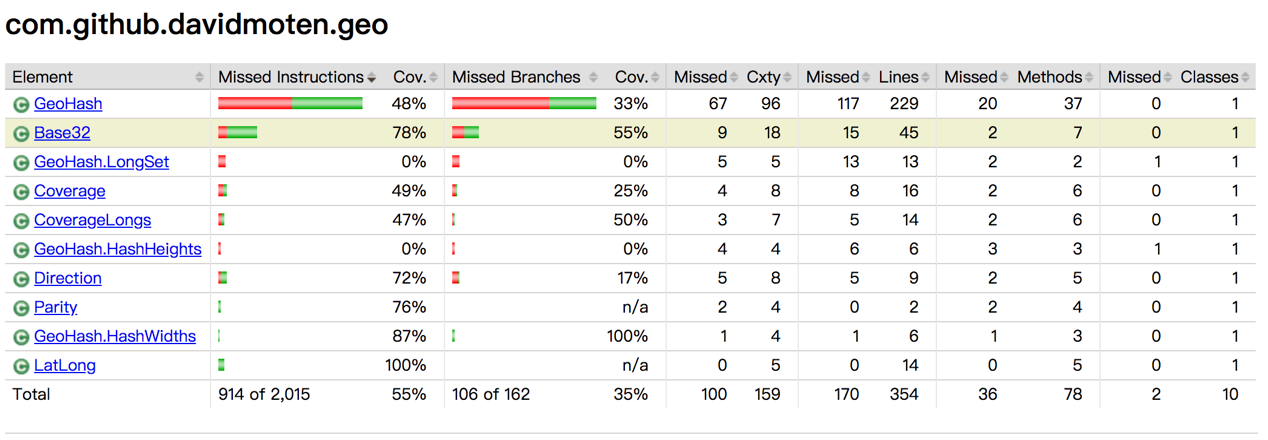
|  |  |  |
| --- | --- | --- |
| **No.** | **Test method** | **Source code** |
| 1 | encodeBase () | @Test  public void encodeBase32() throws Exception {  String encode = Base32.*encodeBase32*(75324, 4);  *assertEquals*("29jw", encode);  } |
| 2 | encodeBase32MaxLength() | @Test  public void encodeBase32MaxLength() throws Exception{  String encode = Base32.*encodeBase32*(75324);  *assertEquals*("0000000029jw",encode);  } |
| 3 | getCharIndex() | @Test  public void getCharIndex() throws Exception{  int result = Base32.*getCharIndex*('e');  *assertEquals*(13,result);  } |

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

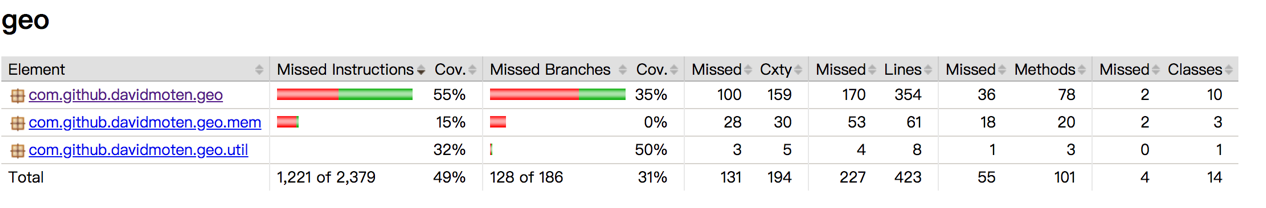
****

****

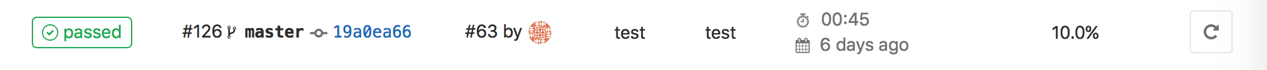
* 1. **Code coverage snapshot**
* Coverage of each selected method

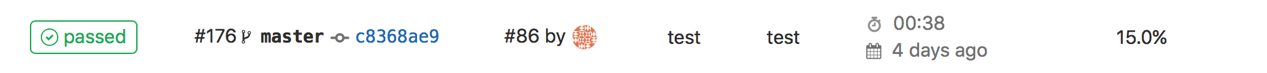
****

* Total coverage

****

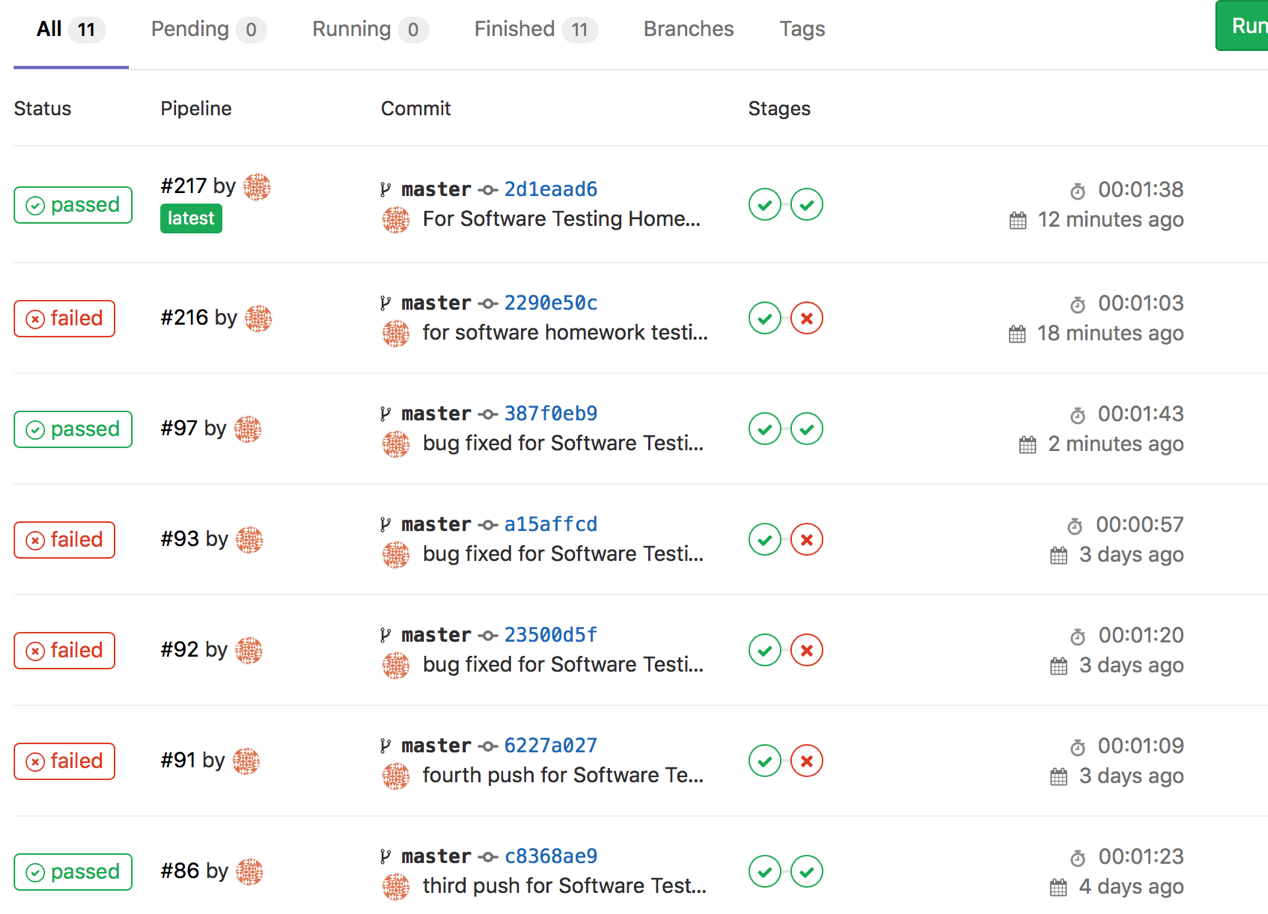
* 1. **CI result snapshot (3 iterations for CI)**
* CI#1/2/3







* 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 80%.** Thus, the test requirements described in Section 1 are satisfied. Some lessons learned in this Lab are …