

Lab 1 Report

張嗣岱

107598069

3/9

1 Test Plan

1.1 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 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%**”*.

1.2 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.

1.3 Test activities

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

No.	Activity Name	Plan hours	Schedule Date
1	Study GeoProject	1	3/4
2	Learn Junit	1	3/4
3	Design test cases for the selected methods	0.5	3/4
4	Implement test cases	3	3/4
5	Perform test	1	3/4
6	Complete Lab1 report	2	3/8

1.4 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%**.

2 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	Inputs	Expected Outputs
1	Base32	encodeBase32()	233,2	79
2	Base32	encodeBase32()	258	000000000082
3	Base32	decodeBase32()	"123"	1091
4	Base32	getCharIndex()	'1'	1
5	Base32	padLeftWithZerosToLength()	"123",4	0123
6	Coverage	getHashes()	Null	[abc]
7	Coverage	getRatio()	Null	0
8	Coverage	getHashLength()	Null	3
9	Coverage	toString()	Null	Coverage [hashes=[abc], ratio=0.0]
10	CoverageLongs	getHashes()	Null	1
11	CoverageLongs	getRatio()	Null	1.1
12	CoverageLongs	getHashLength()	Null	1
13	CoverageLongs	getCount()	Null	1
14	GeoHash	adjacentHash()	"1",TOP	"3"
15	GeoHash	right()	"1"	"4"
16	GeoHash	left()	"1"	"0"
17	GeoHash	top()	"1"	"3"
18	GeoHash	bottom()	"1"	"j"

3 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](#).**

No.	Test method	Source code
1	Base32.encodeBase32()	<pre>@Test public void encodeBase32() { assertEquals("000000000082",Base32.encodeBase32(258)); }</pre>
2	GeoHash.adjacentHash()	<pre>@Test public void adjacentHash() { assertEquals("3",GeoHash.adjacentHash(hash,Direction.TOP)); }</pre>
3	GeoHash.right()	<pre>@Test public void right() {</pre>

```
    assertEquals("4", GeoHash.right(hash));
}
```

4 Test Results

4.1 JUnit test result snapshot

▼ ✓ geo (com.github.davidmoten)	60 ms
▶ ✓ Base32Test	12 ms
▶ ✓ CoverageLongsTest	0 ms
▶ ✓ CoverageTest	0 ms
▶ ✓ GeoHashTest	48 ms

Test Summary

18 tests	0 failures	0 ignored	0.141s duration	100% successful
-------------	---------------	--------------	--------------------	--------------------

Packages

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
com.github.davidmoten.geo.Base32Test	5	0	0	0.016s	100%
com.github.davidmoten.geo.CoverageLongsTest	4	0	0	0s	100%
com.github.davidmoten.geo.CoverageTest	4	0	0	0s	100%
com.github.davidmoten.geo.GeoHashTest	5	0	0	0.125s	100%

4.2 Code coverage snapshot

- Coverage of each selected method

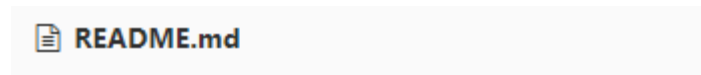
▼ java	60% classes, 47% lines covered
▼ com.github.davidmoten.geo	60% classes, 47% lines covered
▼ mem	0% classes, 0% lines covered
C Geomem	0% methods, 0% lines covered
C Info	0% methods, 0% lines covered
▶ util	100% classes, 66% lines covered
C Base32	100% methods, 93% lines covered
C Coverage	83% methods, 56% lines covered
C CoverageLongs	83% methods, 85% lines covered
E Direction	66% methods, 22% lines covered
C GeoHash	44% methods, 48% lines covered
C LatLong	60% methods, 42% lines covered
package-info.java	
E Parity	100% methods, 100% lines covered

- Total coverage

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
com.github.davidmoten.geo	<div><div></div></div>	58%	<div><div></div></div>	40%	85 149	154 348	25 68	2 10
com.github.davidmoten.geo.mem	<div><div></div></div>	0%	<div><div></div></div>	0%	30 30	61 61	20 20	3 3
com.github.davidmoten.geo.util	<div><div></div></div>	36%	<div><div></div></div>	50%	2 4	2 6	0 2	0 1
Total	1,157 of 2,326	50%	119 of 186	36%	117 183	217 415	45 90	5 14

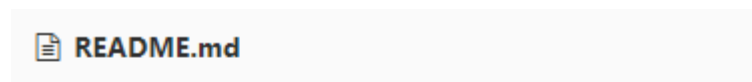
4.3 CI result snapshot (3 iterations for CI)

- CI#1



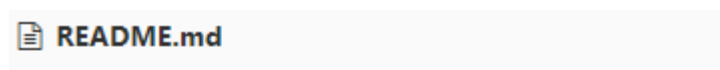
pipeline passed coverage 42%

- CI#2



pipeline passed coverage 49%

- CI#3



pipeline passed coverage 50%

- CI Pipeline

Overview	All 5	Pending 0	Running 0	Finished 5	Branches	Tags
Repository	Status	Pipeline	Commit	Stages		
Issues 0	passed	#951 by latest	master -> b8fd21a1 3/9	✓ ✓	00:05:27	about a minute ago
Merge Requests 0	passed	#950 by	master -> 7995413e 3/9	✓ ✓	00:02:57	12 minutes ago
CI / CD	passed	#886 by	master -> 76403932 lab1	✓ ✓	00:03:02	3 days ago
Pipelines	passed	#883 by	master -> 61f0922e lab1	✓ ✓	00:03:55	30 minutes ago
Jobs	passed	#858 by	master -> d666df2d Update README.md	✓ ✓	00:04:47	5 days ago
Schedules						
Environments						
Charts						
Cluster						
Wiki						

5 Summary

In Lab 1, **18 test cases** have been designed and implemented using JUnit. The test is conducted in **3 CI** and **the execution results of the 18 test methods are all passed**. **The total statement coverage of the test is 50%**. Thus, the test requirements described in Section 1 are satisfied. Some lessons learned in this Lab are ...