

# Lab 1 Report

Name 林柔璟

Student ID 108598003

Date

## 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 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%**”*.

### 1.2 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.
- (4) design the test cases for those selected methods by considering
  - i. the possible **valid values** and **combinations** of the input parameters.
  - ii. the **boundary values** of the input parameters.

### 1.3 Test activities

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

| No. | Activity Name                              | Plan hours | Schedule Date            |
|-----|--|------------|--------------------------|
| 1   | Environment Setting                        | 1          | 17 <sup>th</sup> , March |
| 2   | Study GeoProject                           | 3          | 17 <sup>th</sup> , March |
| 3   | Learn JUnit                                | 1          | 17 <sup>th</sup> , March |
| 4   | Design test cases for the selected methods | 2          | 18 <sup>th</sup> , March |
| 5   | Implement <u>Base32</u> test cases         | 1          | 18 <sup>th</sup> , March |
| 6   | Perform <u>Base32</u> test                 | 1          | 18 <sup>th</sup> , March |

|    |  |     |  |
|----|--|-----|--|
| 7  | Implement <u>Coverage</u> test cases     | 1   | 19 <sup>th</sup> , March                   |
| 8  | Perform <u>Coverage</u> test             | 1   | 19 <sup>th</sup> , March                   |
| 9  | Implement <u>CoverageLong</u> test cases | 1   | 21 <sup>th</sup> , March                   |
| 10 | Perform <u>CoverageLong</u> test         | 1   | 21 <sup>th</sup> , March                   |
| 11 | Implement <u>Direction</u> test cases    | 0.5 | 22 <sup>th</sup> , March                   |
| 12 | Perform <u>Direction</u> test            | 0.5 | 22 <sup>th</sup> , March                   |
| 13 | Implement <u>GeoHash</u> test cases      | 1   | 23 <sup>th</sup> , March                   |
| 14 | Perform <u>GeoHash</u> test              | 1   | 23 <sup>th</sup> , March                   |
| 15 | Implement <u>Info</u> test cases         |     | 24 <sup>th</sup> , March                   |
| 16 | Perform <u>Info</u> test                 | 1   | 24 <sup>th</sup> , March                   |
| 17 | Complete Lab1 report                     | 5   | 24 <sup>th</sup> -25 <sup>th</sup> , March |

#### 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 50%.*

## 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   | Test Objective  | Inputs   | Expected Outputs                                  |
|-----|--------|--|---|--|---|
| 1   | Base32 | A. <code>encodeBase32(long i, int length)</code><br>B. <code>encodeBase32(long i)</code> | Returns the base 32 encoding of the given length from a geohash                             | A. <code>Base32.encodeBase32(75324, 4) //positive</code><br><code>Base32.encodeBase32(-122,4) //negative</code><br>B. <code>Base32.encodeBase32((long) 32.0);</code> | A. 29jw<br>-003u<br>B. 000000000<br>010           |
|     |        | C. <code>decodeBase32(String hash)</code>  | Returns the conversion of a base32 geohash to a long  | C. <code>Base32.decodeBase32("w") //positive</code><br><code>Base32.decodeBase32("-j") //negative</code>   | C. 28<br>-17                                      |
|     |        | D. <code>getCharIndex(char ch)</code>  | Throws an <code>IllegalArgumentException</code> if the character is not found in the array. | D. <code>Base32.getCharIndex('-')</code>   | D. Throw message:<br>"not a base32 character : -" |

|   |          |   |   |  |  |
|---|----------|---|---|--|--|
| 2 | Coverage | <p>A. Coverage(Set&lt;String&gt; hashes, double ratio)</p> <p>B. Coverage(CoverageLongs coverage)</p> | How well the coverage is covered by the hashes. Will be $\geq 1$ . Closer to 1, the closer the coverage is to the region in question. | <pre>getHashSets.add("22"); getHashSets.add("33"); getHashSets.add("44");</pre> <p>A. Coverage(getHashSets, 2.5) //Coverage Test</p> <pre>----- long[] hashes = new long[]{22,33,44}</pre> <p>B. CoverageLongs coverageLongs = new CoverageLongs(hashes, 3, 2.5); Coverage coverage_long = new Coverage(coverageLongs); //CoverageLongs Test</p> | <p>A. Hashes = getHashSets</p> <p>Ratio = 2.5</p> <p>B. getHashLength = 1</p> <p>Ratio = 2.5</p>   |
|   |          | C. getHashes()  | Returns the hashes which are expected to be all of the same length.   | <pre>getHashSets.add("aaa") getHashSets.add("bbb") Coverage getHashCoverage = new Coverage(getHashSets, 5.5)</pre> <p>C. getHashCoverage.getHashes()</p>   | C. getHashSets   |
|   |          | D. getRatio()   | Returns the measure of how well the hashes cover a region.  | <pre>Coverage(getHashSets, 0.005)</pre> <p>D. getRatioCoverage.getRatio()</p>  | D. Ratio = 0.005   |
|   |          | E. getHashLength()  | Returns the length in characters of the first hash returned by an iterator on the hash set.   | <pre>Coverage getHashLengthCoverage = new Coverage(getHashSets, 2.4)</pre> <p>E. getHashLengthCoverage.getHashLength() //hashes.size() == 0</p> <pre>----- getHashSets.add("something") getHashLengthCoverage.getHashLength() //hashes.size() != 0</pre>   | <p>E. HashLength():</p> <ul style="list-style-type: none"> <li>➤ When hash size = 0: 0</li> <li>➤ When hash size <math>\neq</math> 0: 9</li> </ul> |
|   |          | F. toString()   | Show Coverage [hashes, ratio]   | <pre>Coverage testToStringCoverage = new Coverage(getHashSets, 6.66)</pre>   | F. "Coverage [hashes=[   |

|   |               |                    |   |   |   |
|---|---------------|--------------------|---|---|---|
|   |               |                    |   | F. testToStringCoverage<br>.toString()  | ],<br>ratio=6.6<br>6]"  |
| 3 | CoverageLongs | A. getHashes()     | Returns the hashes which are expected to be all of the same length  | hash = new long[]{33,22};<br>CoverageLongs<br>coverageLongsGetHashes = new<br>CoverageLongs(hash, 2, 4.5);<br>A. coverageLongsGetHashes.getHashes()   | A. hash =<br>long[]{33<br>,22}                                  |
|   |               | B. getRatio()      | Returns the measure of how well the hashes cover a region   | B. coverageLongsGetRatio.getRatio()   | B. 3.0  |
|   |               | C. getHashLength() | Returns the length in characters of the first hash returned by an iterator on the hash set<br>//compare[0] with 0000 1111(0x0f) | long[] hashZero = new long[]{};<br>CoverageLongs<br>coverageLongsGetZeroLength = new<br>CoverageLongs(hashZero,0,3.3);<br>C. coverageLongsGetZeroLength.getHashLength()<br>//length = 0<br>-----<br>hash = new long[]{33,22};<br>CoverageLongs<br>coverageLongsGetHashLength = new<br>CoverageLongs(hash,count,2.2);<br>coverageLongsGetHashLength.getHashLength()<br>// //length !=0 | C. Hash Length:<br>➤ When length = 0: 0<br>➤ When length ≠ 0: 1 |
|   |               | D. getCount()      | Count hash element  | hash = new long[]{33,22};<br>CoverageLongs<br>coverageLongsGetCount = new<br>CoverageLongs(hash, count,3.00);<br>D. coverageLongsGetCount.getCount()  | D. 2  |

|   |         |  |   |  |   |
|---|---------|--|---|--|---|
| 4 | GeoHash | <p>A. adjacentHash (String hash, Direction direction, int steps)<br/>Method: adjacentHash()</p> <p>B. adjacentHash (String hash, Direction direction)<br/>Method: testAdjacentHash()</p> | <p>Returns the adjacent hash N steps in the given Direction. A negative N will use the opposite Direction</p> <hr/> <p>Returns the adjacent hash in given Direction</p> | <pre>hash = "11w"; direction = Direction.TOP; A. adjacentHash =   geoHash.adjacentHash     (hash, direction,-     1); //steps&lt;0 adjacentHash =   geoHash.adjacentHash     (hash, direction,     5); //steps&gt;0 B. adjacentHash =   geoHash.adjacentHash     (hash, direction); //normal ----- ➤ adjacentHash =   geoHash.adjacentHash("zzz",     Direction.TOP); //out of border(top) ➤ adjacentHash =   geoHash.adjacentHash("145",Direction     .BOTTOM); //out of border(bottom) ➤ adjacentHash =   geoHash.adjacentHash("000",Direction     .LEFT); //out of border(left) ➤ adjacentHash =   geoHash.adjacentHash("ppp",Direction     .RIGHT); //out of border(right)</pre> | <p>A. geoHash.adjacentHash("11w", Direction.BOTTOM, 1)<br/>"14y"</p> <p>B. normal:<br/>"11y"<br/>-----<br/>out of<br/>border:<br/>➤ top:<br/>"gzz"<br/>➤ bottom:<br/>"11g"<br/>➤ left:<br/>"pbp"<br/>➤ right:<br/>"pr0"</p> |
|   |         | C. right (String hash)   | Returns the adjacent hash to the right (east)   | <pre>hash = "11w"; C. geoHash.right(hash)</pre>  | C. "11x"  |

|   |           |   |   |   |  |
|---|-----------|---|---|---|--|
|   |           | D. left (String hash)                                 | Returns the adjacent hash to the left (west)  | hash = "11w";<br>D. geoHash.left(hash)  | D. "11t"   |
|   |           | E. top (String hash)                                  | Returns the adjacent hash to the top (north)  | hash = "11w";<br>E. geoHash.top(hash)   | E. "11y"   |
|   |           | F. bottom (String hash)                               | Returns the adjacent hash to the bottom (south)   | hash = "11w";<br>F. geoHash.bottom(hash)  | F. "11q"   |
|   |           | G. neighbours()<br>Method:<br>neighbours(String hash) | Returns a list of the 8 surrounding hashes for a given hash in order:<br>left,right,top,bottom,left-top,left-bottom,right-top,right-bottom. | <pre>List&lt;String&gt; neighbours = new ArrayList&lt;String&gt;();  neighbours = geoHash.neighbours(hash);  List&lt;String&gt; compare = new ArrayList&lt;String&gt;();  compare.add("11t"); //left compare.add("11x"); //right compare.add("11y"); //top compare.add("11q"); //bottom compare.add("11v"); //left-top compare.add("11m"); //left-bottom compare.add("11z"); //right-top compare.add("11r"); //right-bottom</pre> G. neighbours | G. compare   |
| 5 | Direction | A. opposite()   | Returns the opposite direction  | <pre>Direction bottom = Direction.BOTTOM; Direction top = Direction.TOP; Direction left = Direction.LEFT; Direction right = Direction.RIGHT;</pre> A. bottom.opposite()<br>top.opposite()<br>left.opposite()<br>right.opposite()  | A. Opposite of<br>➤ BOTTOM: TOP<br>➤ TOP: BOTTOM<br>➤ LEFT: RIGHT<br>➤ RIGHT: LEFT |
| 6 | Info      | A. id()   | id of Info  | <pre>String a = "a";  Optional&lt;String&gt; id = Optional.of(a);  Info info = new Info(25.5, 30.0, 10000, 555, id);</pre> A. info.id()   | A. Optional. of("a")   |

|  |  |               |                              |   |   |
|--|--|---------------|------------------------------|---|---|
|  |  | B. lat()      | Latitude of Info             | String a = "a";<br><br>Optional<String> id =<br><br>Optional.of(a);<br><br>Info info = new Info(25.5, 30.0,<br><br>10000, 555, id);<br><br>B. info.lat()      | B. 25.5   |
|  |  | C. lon()      | Longitude of Info            | String a = "a";<br><br>Optional<String> id =<br><br>Optional.of(a);<br><br>Info info = new Info(25.5, 30.0,<br><br>10000, 555, id);<br><br>C. info.lon()      | C. 30.0   |
|  |  | D. time()     | Time of Info                 | String a = "a";<br><br>Optional<String> id =<br><br>Optional.of(a);<br><br>Info info = new Info(25.5, 30.0,<br><br>10000, 555, id);<br><br>D. info.time()     | D. 10000  |
|  |  | E. value()    | Value of Info                | String a = "a";<br><br>Optional<String> id =<br><br>Optional.of(a);<br><br>Info info = new Info(25.5, 30.0,<br><br>10000, 555, id);<br><br>E. info.value()    | E. 555  |
|  |  | F. toString() | Show all information of Info | String a = "a";<br><br>Optional<String> id =<br><br>Optional.of(a);<br><br>Info info = new Info(25.5, 30.0,<br><br>10000, 555, id);<br><br>F. info.toString() | F. "Info<br>[lat=25.5,<br>lon=30.0,<br>time=10000,<br>value=555,<br>id=Optional<br>.of(a)]" |

### 3 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 the [link](#) (or JUnit files).**

| No. | Test method             | Source code   |
|-----|-------------------------|---|
| 1   | encodeBase32()          | <a href="https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java">https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java</a>               |
| 2   | decodeBase32()          |   |
| 3   | getCharIndexException() |   |
| 4   | testCoverage()          | <a href="https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java">https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java</a>           |
| 5   | getHashes()             |   |
| 6   | getRatio()              |   |
| 7   | getHashLength()         |   |
| 8   | testToString()          |   |
| 9   | getHashes()             | <a href="https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageLongsTest.java">https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageLongsTest.java</a> |
| 10  | getRatio()              |   |
| 11  | getHashLength()         |   |
| 12  | testToString()          |   |
| 13  | getCount()              |   |
| 14  | opposite()              | <a href="https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/DirectionTest.java">https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/DirectionTest.java</a>         |
| 15  | adjacentHash()          | <a href="https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java">https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java</a>             |
| 16  | right()                 |   |
| 17  | left()                  |   |
| 18  | top()                   |   |
| 19  | bottom()                |   |
| 20  | testAdjacentHash()      |   |
| 21  | neighbours()            |   |
| 22  | id()                    | <a href="https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java">https://stv.csie.ntut.edu.tw/rojeanlin/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java</a>           |
| 23  | lat()                   |   |
| 24  | lon()                   |   |
| 25  | time()                  |   |
| 26  | value()                 |   |
| 27  | testToString()          |   |



## 4 Test Results

### 4.1 JUnit test result snapshot

| ▼ ✓ Test Results                                | 225 ms |
|---|--------|
| ▶ ✓ com.github.davidmoten.geo.CoverageLongsTest | 14 ms  |
| ▶ ✓ com.github.davidmoten.geo.Base32Test        | 22 ms  |
| ▶ ✓ com.github.davidmoten.geo.CoverageTest      | 60 ms  |
| ▶ ✓ com.github.davidmoten.geo.GeoHashTest       | 72 ms  |
| ▶ ✓ com.github.davidmoten.geo.DirectionTest     | 0 ms   |
| ▶ ✓ com.github.davidmoten.geo.mem.GeomemTest    | 16 ms  |
| ▶ ✓ com.github.davidmoten.geo.mem.InfoTest      | 41 ms  |

#### Test Summary

|       |          |         |          |
|-------|----------|---------|----------|
| 51    | 0        | 0       | 0.225s   |
| tests | failures | ignored | duration |

**100%**  
successful

Packages Classes

| Package                                       | Tests | Failures | Ignored | Duration | Success rate |
|---|-------|----------|---------|----------|--------------|
| <a href="#">com.github.davidmoten.geo</a>     | 39    | 0        | 0       | 0.168s   | 100%         |
| <a href="#">com.github.davidmoten.geo.mem</a> | 12    | 0        | 0       | 0.057s   | 100%         |

[geo/build/reports/tests/test/index.html](#)

### 4.2 Code coverage snapshot

- Coverage of each selected method

|  |
|--|
| ▼ java 66% classes, 61% lines covered                      |
| ▼ com.github.davidmoten.geo 66% classes, 61% lines covered |
| ▶ mem 33% classes, 22% lines covered                       |
| ▶ util 100% classes, 66% lines covered                     |
| C Base32 100% methods, 100% lines covered                  |
| C Coverage 100% methods, 100% lines covered                |
| C CoverageLongs 83% methods, 92% lines covered             |
| E Direction 100% methods, 100% lines covered               |
| C GeoHash 58% methods, 58% lines covered                   |
| C LatLong 60% methods, 42% lines covered                   |
| package-info.java  |
| E Parity 100% methods, 100% lines covered                  |

- Total coverage

geo

| Element                        | Missed Instructions | Cov. | Missed Branches | Cov. | Missed Cxty | Missed Lines | Missed Methods | Missed Classes |
|--------------------------------|---------------------|------|-----------------|------|-------------|--------------|----------------|----------------|
| com.github.davidmoten.geo      |                     | 68%  |                 | 55%  | 62 149      | 114 348      | 21 68          | 2 10           |
| com.github.davidmoten.geo.mem  |                     | 19%  |                 | 0%   | 23 30       | 48 61        | 13 20          | 2 3            |
| com.github.davidmoten.geo.util |                     | 36%  |                 | 50%  | 2 4         | 2 6          | 0 2            | 0 1            |
| Total                          | 903 of 2,326        | 61%  | 94 of 186       | 49%  | 87 183      | 164 415      | 34 90          | 4 14           |

### 4.3 CI result snapshot (3 iterations for CI)

- CI#1

|  |       |                    |          |       |       |                     |      |  |
|--|-------|--------------------|----------|-------|-------|---------------------|------|--|
|  | #3291 | master -> 2236b6e1 | #1486 by | test  | test  | 00:38<br>a week ago | 9.0% |  |
|  | #3290 | master -> 2236b6e1 | #1486 by | build | build | 00:31<br>a week ago |      |  |

- CI#2

|  |       |                    |          |       |       |                    |       |  |
|--|-------|--------------------|----------|-------|-------|--------------------|-------|--|
|  | #3827 | master -> b7e9208c | #1683 by | test  | test  | 00:34<br>a day ago | 33.0% |  |
|  | #3826 | master -> b7e9208c | #1683 by | build | build | 00:37<br>a day ago |       |  |

- CI#3

|  |       |                    |          |       |       |                             |       |  |
|--|-------|--------------------|----------|-------|-------|-----------------------------|-------|--|
|  | #3967 | master -> fe52a053 | #1737 by | test  | test  | 00:33<br>about 21 hours ago | 61.0% |  |
|  | #3966 | master -> fe52a053 | #1737 by | build | build | 00:32<br>about 21 hours ago |       |  |

- CI Pipeline

|  |          |        |   |  |  |                                |
|--|----------|--------|---|--|--|--------------------------------|
|  | #1737 by | latest | master -> fe52a053<br>add InfoTest                          |  |  | 00:01:11<br>about 21 hours ago |
|  | #1695 by |        | master -> 504ec091<br>add GitHashTest(partly)               |  |  | 00:01:11<br>a day ago          |
|  | #1683 by |        | master -> b7e9208c<br>add DirectionTest                     |  |  | 00:01:12<br>a day ago          |
|  | #1631 by |        | master -> 623d876d<br>add CoverageLongsTest                 |  |  | 00:01:10<br>3 days ago         |
|  | #1486 by |        | master -> 2236b6e1<br>build.gradle add mavenCentral for ... |  |  | 00:01:09<br>a week ago         |
|  | #1412 by |        | master -> e8c71014<br>Update README.md                      |  |  | 00:00:13<br>a week ago         |

## 5 Summary

In Lab 1, **27** test cases have been designed and implemented using JUnit. The test is conducted in **4** CI and the execution results of the **15** test methods are **all passed**. The total statement coverage of the test is **61%**. Thus, the test requirements described in Section 1 are satisfied. **Some lessons learned in this Lab are ...**I learnt Junit and GeoHash algorithm in this lab.