

# Lab 1 Report

Name: 呂聖文

Student ID:108598007

Date:2020/3/19

## 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 **80%**”*.

### 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	Study GeoProject	2	3/11
2	Learn JUnit	1	3/11
3	Design test cases for the selected methods	1	3/11
4	Implement test cases	5 days	3/12~3/16
5	Read Info class		
6	Read Geohash class		
7	Read Base32		
8	Perform test		
9	Complete Lab1 report	3 days	3/17~3/19

## 1.4 Success criteria

All test cases designed for the selected methods 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	Test Objective	Inputs	Expected Outputs
1.	Base32	encodeBase32()	編碼是否正確	(75324,4)	"29jw"
2.	Base32	encodeBase32()	負數是否正確	(-75324,4)	"-29jw"
3.	Base32	encodeBase32()	測試無 length 參數是否為 12 位數	(75324)	"0000000029jw"
4.	Base32	decodeBase32()	解碼是否正確	("29jw")	75324
5.	Base32	decodeBase32()	負數解碼是否正確	("-29jw")	-75324
6.	Base32	getCharIndex()	搜尋'j'為陣列第幾 index	('j')	17
7.	Base32	getCharIndex()	搜尋非這列中字元是否正確拋出例外	('I')	IllegalArgumentException
8.	Base32	padLeftWithZerosToLength()	測試補 0 功能	("jw",6)	"0000jw"
9.	Coverage	Coverage()	使用 CoverageLongs 物件套入 Coverage 中是否正確	<pre>long[] hashes=new long[]{40,36,34}; int count=3; double ratio=3.14;</pre>	<pre>getRatio()==3.14 getHashLength()==2</pre>
10.	Coverage	getHashes()	參數陣列與回傳陣列是否為同一記憶體位置	Coverage(hashes,3.14);	getHashes==hashes(object)
11.	Coverage	getRatio()	傳入參數與回傳參數是否為同一個數字	3.14	getRatio() ==3.14
12.	Coverage	getHashLength()	只取陣列中第一個元素的前四個位元 測試 10001 AND 000001111 結果為 1	17	1
13.	Coverage	getHashLength()	測試陣列沒有元素應回傳 0 字元	hashes={}	getHashLength()==0
14.	CoverageLongs	getHashes()	測試 CoverageLongs 的 getHashes 比對參數陣列與回傳陣列內容是否相同	hashes={40,36,34}	getHashes()==hashes

15.	CoverageLong s	getRatio()	測試參數 ratio 與 回傳 ratio 是否同 一數字	3.14	getRatio()==3.14
16.	CoverageLong s	getCount()	測試 hashes 陣列 的元素共有幾個	hashes={17,36,34 }	getCount()==3
17.	GeoHash	adjacentHash()	km 區域上面是 kq	("km",Direction.TOP)	kq
18.	GeoHash	adjacentHash()	w 往左五格是 9	("w",Direction.LEFT,5)	9
19.	GeoHash	decodeHash()	將 geohash “0000”轉為經緯 度	("0000")	getLat()=-89.91210938  getLon()=-179.82421875
20.	GeoHash	encodeHash()	將經緯度- 38.23242188, - 149.58984375 轉 換為 geohash	(-38.23242188, -149.58984375,4)	“26jw”
21.	GeoHash	Left()	測試 w 區域左邊 (西邊)區域	("w")	“t”
22.	GeoHash	Right()	測試 w 區域左邊 (東邊)區域	("w")	“x”
23.	GeoHash	Top()	測試 w 區域左邊 (北邊)區域	("w")	“y”
24.	GeoHash	Bottom()	測試 w 區域左邊 (南邊)區域	("w")	“q”
25.	GeoHash	neighbours()	測試 w 區域八方 位	("w")	geohash.get(0)=="t" geohash.get(1)=="x" geohash.get(2)=="y" geohash.get(3)=="q" geohash.get(4)=="v" geohash.get(5)=="m" geohash.get(6)=="z" geohash.get(7)=="r"
26.	GeoHash	hashContains()	測試 geohash wz 是否再此經緯度 參數區域內	("wz",42.18750000 , 129.37500000)	isContain==true
27.	GeoHash	fromLongToStr ing()	測試當 hash 小於 1 將拋出例外	-1	Catch IllegalArgumentException
28.	Info	id()	測試回傳 id 是否 與 optional 為同 一記憶體位置	optional = Optional.of("NTUT");	Optional==info.id()
29.	Info	lat()	測試回傳 lat 是 否為當初設定的 3.123	info=new Info(3.123,4.123,10, 5, optional);	info.lat()==3.123

30.	Info	lon()	測試回傳 lon 是否為當初設定的 4.123	<b>info=new</b> Info(3.123,4.123,10, 5, <b>optional</b> );	<b>info.lat()==4.123</b>
31.	Info	time()	測試回傳 time 是否為當初設定的 10	<b>info=new</b> Info(3.123,4.123,10, 5, <b>optional</b> );	<b>info.time()==10</b>
32.	Info	value()	測試回傳 value 是否為當初設定的 5	<b>info=new</b> Info(3.123,4.123,10, 5, <b>optional</b> );	<b>info.value()==5</b>

### 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).**

N o.	Test method	Source code
4	encodeBase32()	<a href="https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java">https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java</a>
5	encodeBase32_negative()	
6	encodeBase32_noLength()	
7	decodeBase32()	
8	decodeBase32_negative()	
9	getCharIndex()	
10	getCharIndex_exception()	
11	getCharIndex_padLeftWithZerosToLength()	<a href="https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java">https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java</a>
12	Coverage()	
13	getHashes()	
14	getRatio()	
15	getHashLength()	
16	getHashLength_sizeZero()	
17	testToString()	
18	getHashes()	<a href="https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageLongsTest.java">https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageLongsTest.java</a>
19	getRatio()	
20	getHashLength()	
21	getHashLength_sizeZero()	
22	getCount()	
23	adjacentHash()	<a href="https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java">https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java</a>
24	adjacentHash_steps()	

25	decodeHash()	
26	encodeHash lat long()	
27	left()	
28	right()	
29	top()	
30	bottom()	
31	neighbours()	
32	hashContains()	
33	fromLongToString()	
34	hashLengthToCoverBound ingBox()	
35	coverBoundingBoxLongs( )	
36	id()	<a href="https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java">https://stv.csie.ntut.edu.tw/108598007/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java</a>
37	lat()	
38	lon()	
39	time()	
40	value()	
41	testToString()	

## 42 Test Results

### 42.1 JUnit test result snapshot

Test Results	43 ms
> ✓ com.github.davidmoten.geo.Base32Test	4 ms
> ✓ com.github.davidmoten.geo.CoverageLongsTest	4 ms
> ✓ com.github.davidmoten.geo.CoverageTest	20 ms
> ✓ com.github.davidmoten.geo.DirectionTest	0 ms
> ✓ com.github.davidmoten.geo.GeoHashTest	10 ms
> ✓ com.github.davidmoten.geo.LatLongTest	0 ms
> ✓ com.github.davidmoten.geo.mem.GeomemTest	4 ms
> ✓ com.github.davidmoten.geo.mem.InfoTest	1 ms

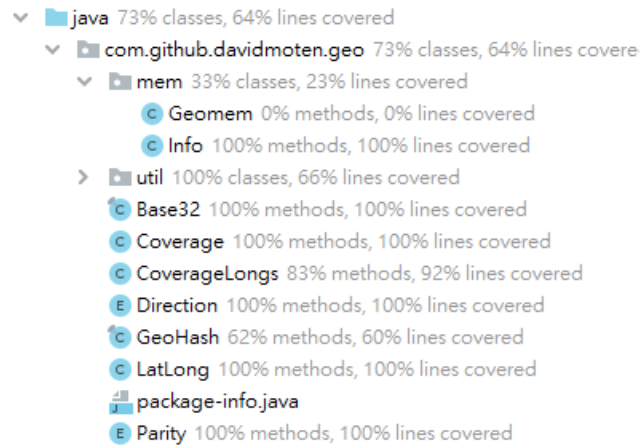
### Test Summary

49 tests	0 failures	0 ignored	0.083s duration	100% successful
-------------	---------------	--------------	--------------------	--------------------

Packages		Classes				
Package	Tests	Failures	Ignored	Duration	Success rate	
<a href="#">com.github.davidmoten.geo</a>	37	0	0	0.076s	100%	
<a href="#">com.github.davidmoten.geo.mem</a>	12	0	0	0.007s	100%	

## 42.2 Code coverage snapshot

- Coverage of each selected method



- Total coverage

geo

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
com.github.davidmoten.geo	<div><div></div></div>	89%	<div><div></div></div>	78%	38 149	34 349	7 68	0 10
com.github.davidmoten.geo.mem	<div><div></div></div>	59%	<div><div></div></div>	25%	16 30	19 62	7 20	1 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	360 of 2,341	84%	52 of 186	72%	56 183	55 417	14 90	1 14

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

- CI#1

<div>passed</div>	#3234 <small>master</small> -> 39d407bc	#1461 by	test	test	<div>00:40</div> <div>a week ago</div>	12.0%
-------------------	---	----------	------	------	--	-------

- CI#2

<div>passed</div>	#3252 <small>master</small> -> b0b62caa	#1468 by	test	test	<div>00:49</div> <div>6 days ago</div>	31.0%
-------------------	---	----------	------	------	--	-------

- CI#3

<div>passed</div>	#3306 <small>master</small> -> ca9cc4ea	#1492 by	test	test	<div>00:35</div> <div>6 days ago</div>	34.0%
-------------------	---	----------	------	------	--	-------

- CI#4

<div>passed</div>	#3314 <small>master</small> -> 6a7b11a0	#1495 by	test	test	<div>00:36</div> <div>6 days ago</div>	35.0%
-------------------	---	----------	------	------	--	-------

- CI#5

<div>passed</div>	#3350 <small>master</small> -> 5317bbf1	#1508 by	test	test	<div>00:32</div> <div>6 days ago</div>	38.0%
-------------------	---	----------	------	------	--	-------

- CI#6


<div>passed</div>	#3377 <small>master</small> -> fee0ff27	#1520 by	test	test	<div>00:34</div> <div>a day ago</div>	52.0%
-------------------	---	----------	------	------	---------------------------------------	-------

 passed	#3375 P master -> 2bdb4790	#1519 by 	test	test	⌚ 00:33 📅 6 days ago	51.0%
--	----------------------------	--	------	------	-------------------------	-------

## ● CI#7

 passed	#3388 P master -> 530cd333	#1524 by 	test	test	⌚ 00:37 📅 5 days ago	61.0%
--	----------------------------	--	------	------	-------------------------	-------


## ● CI#8

 README.md

## ●

pipeline **passed** coverage **61%**



































## ● CI#9

 README.md

## ●

pipeline **passed** coverage **84%**

## ● CI Pipeline

<div>GeoProject</div> <ul style="list-style-type: none"> <li>Overview</li> <li>Repository</li> <li>Issues</li> <li>Merge Requests</li> <li>CI / CD           <ul style="list-style-type: none"> <li>Pipelines</li> <li>Jobs</li> <li>Schedules</li> <li>Environments</li> <li>Charts</li> <li>Cluster</li> </ul> </li> <li>Wiki</li> <li>Snippets</li> <li>Settings</li> </ul>	Pipeline				Run Pipeline		CI List
	Status	Pipeline	Commit	Stages			
	 passed	#1573 by 	P master -> 3b9fab89 Lab1-16	 	⌚ 00:01:14 📅 4 days ago		
	 passed	#1569 by 	P master -> d86c8c3b Lab1-15	 	⌚ 00:01:22 📅 4 days ago		
	 passed	#1524 by 	P master -> 530cd333 Lab1-14	 	⌚ 00:01:07 📅 5 days ago		
	 passed	#1521 by 	P master -> 1fc87d7a Lab1-13	 	⌚ 00:01:07 📅 5 days ago		
	 passed	#1520 by 	P master -> fee8ff27 Lab1-12	 	⌚ 00:01:06 📅 5 days ago		
	 passed	#1519 by 	P master -> 2bdb4790 Lab1-11	 	⌚ 00:01:00 📅 6 days ago		
	 passed	#1508 by 	P master -> 5317bbf1 Lab1-12	 	⌚ 00:01:09 📅 6 days ago		
	 failed	#1507 by 	P master -> a991b367 Lab1-11	 	⌚ 00:00:58 📅 6 days ago		
V master -> 54261ha7				 	⌚ no-run		

## 43 Summary

In Lab 1, **41** test cases have been designed and implemented using JUnit. The test is conducted in **9** CI and the execution results of the **15** test methods are **all passed**. The total statement coverage of the test is **84%**. Thus, the test requirements described in Section 1 are satisfied.

過去只有 POSD 有寫過測試經驗，但那時沒有所謂的覆蓋率問題，本次作業是我第一次針對覆蓋率進行撰寫測試，以前我們測的是自己寫的程式，因此馬上就可開始撰寫測試，知道自己要測些甚麼，而這次卻不一樣，我們要測試別人寫的程式，導致需要花許多時間去讀懂程式碼才能知道如何撰寫測試，看懂別人的程式碼也是需要許多精力去完成，且一次又一次覆蓋率增加，心裡也有一種成就感，最終達到自己的覆蓋率目標 80%!

以前我一直疑惑軟體測試到底在做些甚麼，上學期 POSD 也僅測試一點點而已，這學期修了軟體測試這門課程，開始慢慢了解到原來軟體測試分為很多層面，經過第一次作業也慢慢能夠進入狀況，知道該怎麼做會比較好一些。