Lab 2 Report

Name:劉宏德

Student ID: 108598004

Date

1 Test Plan

1.1 Test requirements

The Lab 2 requires to (1) select 19 methods from 6 classes of the SUT (GeoProject), (2) design Unit test cases by using **input space partitioning (ISP)** technique for the selected methods, (3) develop test scripts to implement the test cases, (4) execute the test scripts on the selected methods, (5) report the test results, and (6) specify your experiences of designing test cases systematically using the ISP technique.

In particular, based on the statement coverage criterion, the **test requirements** for Lab 2 are to design test cases with **ISP** for each selected method so that "each statement of the method will be covered by <u>at least one test case</u> and the <u>minimum</u> statement coverage is 70% (greater than Lab 1)".

1.2 Test Strategy

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

- (1) select **those 10 methods that were chosen in Lab1** and **5 new methods** that are NOT selected previously. If possible, some of the methods do NOT have primitive types of input or output parameters (if possible).
- (2) set the objective of the minimum statement coverage to be greater than that of Lab 1 and adjust the test objective based on the time available (if necessary).
- (3) design the test cases for those selected methods by using the **input space** partitioning (ISP) technique.

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	5	2020/04/15
2	Learn ISP and JUnit	4	2020/04/15~04/18
3	Design test cases for the selected methods	8	2020/04/16~04/18
4	Implement test cases	8	2020/04/17~04/18

5	Perform tests	4	2020/04/18
6	Complete Lab2 report	2	

1.4 Design Approach

The **ISP** technique will be used to design the test cases. Specifically, the possible <u>partitions</u> and <u>boundary values</u> of input parameters shall be identified first using the **Mine Map** and **domain knowledge** (if applicable). The possible **valid** <u>combinations</u> of the <u>partitions</u> (i.e., **all combination coverage**) as well as the boundary values shall be computed for the input parameters of each selected method. Each of the partition combination can be a possible test case. *Add more test cases by considering the possible values and boundary of the outputs for the methods or by using test experiences.*

1.5 Success criteria

All test cases designed for the selected methods must pass and *the statement* coverage should have achieved at least 70%.

2 Test Design

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

N o.	Class	Method	Test Object ive	Inputs	Expected Outputs	
1	Base32	encodeBase32(long I, int length)	將 g u ses Base3 2 編答長以 th en en en en en en en en en en en en en	(-10, 5)	-0000b	
2	Base32	encodeBase32(long i)	將 long 值以 Base3 2 進碼	-10	- 00000000 000b	
3	Base32	decodeBase32(String hash)	將 hash 值碼為 long 值	-29jw	-75324	

			日ロク日			
4	Base32	getCharIndex(char ch)	取輸字為Base3 Base3 中幾	b	10	
5	Base32	padLeftWithZerosToLe ngth(String s, int length)	將 s 左 邊零指 長度	1234, 6	001234	
6	CoverageL ongs	getHashes()	取得 hashes	CoverageLon gs({5, 9, 1}, count=3, ratio=1.2)	{5, 9, 1}	
7	CoverageL ongs	getRatio()	取指區中hh后比得定域中hb占域例	CoverageLon gs({5, 9, 1}, count=3, ratio=1.2)	1.2	
8	CoverageL ongs	get Hash Length()	取在hast第個元代的度得上的中一字所表長	CoverageLon gs({5, 9, 1}, count=1, ratio=1.2)	5	
9	CoverageL ongs	getCount()	取得 count	CoverageLon gs({5, 9, 1}, count=1, ratio=1.2)	1	
10	GeoHash	adjacentHash(String hash, Direction direction)	取指方的 hash hash 居	dg, right	e5	
11	GeoHash	right(String hash)	取得 hash 的右 鄰居	dg	E5	
12	GeoHash	left(String hash)	取得 hash 的左 鄰居	d4	9f	
13	GeoHash	top(String hash)	取得 hash 的上	dx	f8	

			鄰居		
14	GeoHash	bottom(String hash)	取得 hash 下鄰 居	d2	6r
15	GeoHash	adjacentHash(String hash, Direction direction, int steps)	取 hash定向 steps 程的定向 steps 基鄰居	d2, bottom, 2	6q
16	GeoHash	neighbours(String hash)	取得 hash 附的 個 居	d2	<"d0", "d8", "d3", "6r", "d1", "6p", "d9", "6x">
17	GeoHash	encodeHash(double latitude, double longitude)	利經度換長為的 GeoHa sh	50, 50	v0gs3y0zh 7w1
18	GeoHash	encodeHash(LatLong p, int length)	利經度換指長的 GeoHa sh	50, 50, 12	v0gs3y0zh 7w1
19	GeoHash	encodeHash(LatLong p)	利經度換長為的 GeoHa sh	LatLong(50, 50)	v0gs3y0zh 7w1

The details of the design are given below:

The Excel file of test cases...

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 the test script

implementations can be found in the <u>link</u> (or JUnit files).

```
N
        Test method
                                                         Source code
0.
                                      test encodeBase32(long i) with ISP
     testEncodeBase3
        2WithISP 2()
                                    public void testEncodeBase32WithISP_2() {
                                        String encodeHash = Base32.encodeBase32( i: -33);
1
     testGetCharIndex
          WithISP()
                                     int ch_Index = Base32.getCharIndex('b');
assertEquals( expected: 10, ch_Index);
2
     testPadLeftWithZ
     erosToLengthWit
            hISP()
                                   public void testPadLeftWithZerosToLengthWithISP() {
3
```

4 Test Results

4.1 JUnit test result snapshot

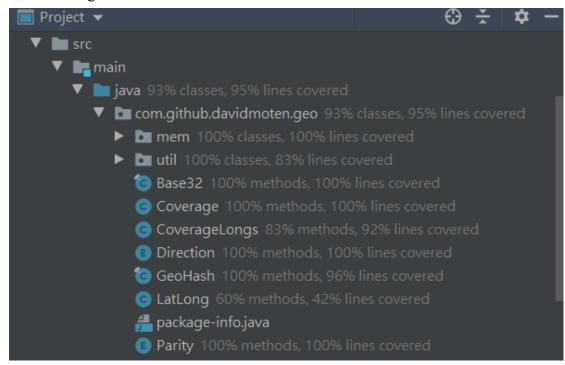
▼	4	Test Results	399 ms
	▶	✓ com.github.davidmoten.geo.Base32Test	36 ms
	▶	✓ com.github.davidmoten.geo.CoverageLongsTest	7 ms
	▶	✓ com.github.davidmoten.geo.CoverageTest	26 ms
	▶	✓ com.github.davidmoten.geo.GeoHashTest	197 ms
	▶	✓ com.github.davidmoten.geo.mem.GeomemTest	132 ms
	▶	✓ com.github.davidmoten.geo.mem.InfoTest	1 ms

Test Summary



4.2 Code coverage snapshot

Coverage of each selected method

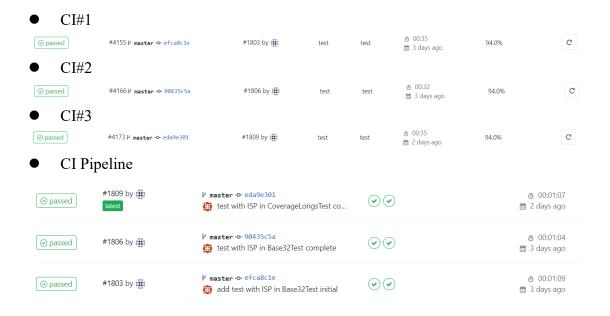


Total coverage

geo

Element	Missed Instructions +	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods *	Missed *	Classes
com.github.davidmoten.geo		94%		90%	19	149	18	348	3	68	0	10
com.github.davidmoten.geo.util		68%	1	75%	1	4	1	6	0	2	0	1
# com.github.davidmoten.geo.mem		99%	_	75%	5	30	0	61	0	20	0	3
Total	118 of 2,326	94%	22 of 186	88%	25	183	19	415	3	90	0	14

4.3 CI result snapshot (3 iterations for CI)



5 Summary

In Lab 2, 19test cases have been designed and implemented using JUnit and the ISP technique. 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 70%. Thus, the test requirements described in Section 1 are satisfied. Some lessons learned in this Lab are ...