Lab 1 Report

Name：劉宏德
Student ID：108598004
Date：2020/03/22

1 Test Plan

1.1 Test requirements

The Lab 1 requires to (1) select **38 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	15	3/14~3/22
2	Learn JUnit	3	3/14
3	Design test cases for the selected methods	5	3/15~3/22
4	Implement test cases	4	3/15~3/22
5	Perform test	2	3/15~3/22
6	Complete Lab1 report	1	3/22

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	測試輸入負數時十進制轉 32 進制是否正確	-2, 2	-02
2	Base32	encodeBase32	測試輸入正數時十進制轉 32 進制是否正確	75324, 4	29jw
3	Base32	encodeBase32(long i)	測試輸入正數時十進制轉 32 進制是否正確	75324	0000000029jw
4	Base32	decodeBase32	測試輸入負數時 32 進制轉十進制是否正確	-29jw	-75324
5	Base32	decodeBase32	測試輸入正數時 32 進制轉十進制是否正確	29jw	75324
6	Base32	getCharIndex	測試輸入不在轉換陣列中的字元是否會有 exception	a	not a base32 character: a
7	Base32	getCharIndex	測試字元轉換的數字是否正確	j	17
8	Base32	padLeftWithZerosToLength	測試 length 大於 32 進制長度時是否會補 0	29jw, 5	029jw
9	Coverage	Coverage	測試 ratio 是否正確	hash, 4, 1.8	1.8
10	Coverage	Coverage	測試 hash 經過轉換後的 set 是否正確	{3, 5, 6, 2}	00, 000, 00000, 000000
11	Coverage	getHashes	測試 hash 是否與原本輸入之 hash 相同	1.5232, 1.9	1.5232, 1.9
12	Coverage	getRatio	測試 ratio 是否與原本輸入之 ratio 相同	1.2	1.2
13	Coverage	getHashLength	測試輸入空 set 時答案是否為 0	nullSet	0
14	Coverage	getHashLength	測試輸入 set 的第一個字串長度	1.5232, 1.9	6

			為和		
15	Coverage	toString	測試利用 hash 和 ratio 所產生的字串是否正確	{1.5232, 1.9}, 1.2	Coverage [hashes=[1.5232, 1.9], ratio=1.2]
16	CoverageLongs	getHashes	測試 hash 是否與原本輸入之 hash 相同	long[]{5, 9, 1}	Long[]{5, 9, 1}
17	CoverageLongs	getRatio	測試 ratio 是否與原本輸入之 ratio 相同	1.8	1.8
18	CoverageLongs	getHashLength	測試輸入 cou 時答案 nt 為 0 時是否為 0	long[]{5, 9, 1}, 0, 1.8	0
19	CoverageLongs	getHashLength	測試輸入 long[] 的第一個值為和	long[]{5, 9, 1}, 3, 1.8	5
20	CoverageLongs	getCount	測試 count 是否與原本輸入之 count 相同	long[]{5, 9, 1}, 3, 1.8	3
21	Info	id	測試 id 是否與原本輸入之 id 相同	88, 12, 20200317, 12, Optional.of(1)	Optional.of(1)
22	Info	lat	測試 lat 是否與原本輸入之 lat 相同	88, 12, 20200317, 12, Optional.of(1)	88
23	Info	lon	測試 lon 是否與原本輸入之 lon 相同	88, 12, 20200317, 12, Optional.of(1)	12
24	Info	time	測試 time 是否與原本輸入之 time 相同	88, 12, 20200317, 12, Optional.of(1)	20200317
25	Info	value	測試 value 是否與原本輸入之 value 相同	88, 12, 20200317, 12, Optional.of(1)	12
26	Info	toString	測試利用參數所形成的字串是否正確	88, 12, 20200317, 12,	Info [lat=88.0, lon=12.0, time=20200317, value=12,

				Optional.of(1)	id=Optional.of(1)]
27	GeoHash	right	測試 hash 是 null 時是否有 exception	null	Hash must be non-null
28	GeoHash	right	測試 hash 長度為 0 時是否有 exception	“”	adjacent has no meaning for a zero length hash that covers the whole world
29	GeoHash	right	測試 hash 長度為奇數時的狀況	25845	2584h
30	GeoHash	right	測試 hash 長度為偶數時的狀況	3121	3123
31	GeoHash	right	測試 hash 長度為奇數且最後一個值在邊界點時的狀況	2584z	2586b
32	GeoHash	right	測試 hash 長度為偶數且最後一個值在邊界點時的狀況	232g	2335
33	GeoHash	left	測試 hash 長度為奇數時的狀況	25845	25844
34	GeoHash	left	測試 hash 長度為偶數時的狀況	3122	3120
35	GeoHash	left	測試 hash 長度為奇數且最後一個值在邊界點時的狀況	25840	rgxfp
36	GeoHash	left	測試 hash 長度為偶數且最後一個值在邊界點時的狀況	312j	2crv
37	GeoHash	top	測試 hash 長度為奇數時的狀況	25845	25847
38	GeoHash	top	測試 hash 長度為偶數時的狀況	3121	3124
39	GeoHash	top	測試 hash 長度為奇數且最後一個值在邊界點時的狀況	2584u	2585h
40	GeoHash	top	測試 hash 長度為偶數且最後一個值在邊界點時的狀況	312r	3182
41	GeoHash	bottom	測試 hash 長度為奇數時的狀況	25847	25845
42	GeoHash	bottom	測試 hash 長度為偶數時的狀況	3121	3120
43	GeoHash	bottom	測試 hash 長度為奇數且最後一個值在邊界點時的狀況	2584n	2581y
44	GeoHash	bottom	測試 hash 長度為偶數且最後一個	312b	310z

			值在邊界點時的狀況		
45	GeoHash	adjacentHash	測試 step 為負數時是否會往反方向移動	72892, Direction.RIGHT, -2	7283q
46	GeoHash	adjacentHash	測試是否會移動數格	72892, Direction.RIGHT, 2	72896
47	GeoHash	neighbours	測試九宮格四周的格子是否正確	9372	9370, 9378, 9373, 935r, 9371, 935p, 9379, 935x
48	GeoHash	encodeHash(double latitude, double longitude)	測試 latitude 大於 90 是否有 exception	91, 3	Latitude must be between -90 and 90 inclusive
49	GeoHash	encodeHash(double latitude, double longitude)	測試經緯度轉換出來的 hash 是否正確	2, 3	s065kk0dc540
50	GeoHash	encodeHash(LatLong p, int length)	測試經緯度轉換出來的 hash 是否正確(限定 hash 長度)	LatLong(2, 3), 8	s065kk0d
51	GeoHash	encodeHash(LatLong p)	測試經緯度轉換出來的 hash 是否正確	LatLong(2, 3)	s065kk0dc540
52	GeoHash	fromLongToString	測試 hash 小於零是否有 exception	-1	Invalid long geohash -1
53	GeoHash	fromLongToString	測試 hash 轉換出的 0 數量是否正確	8	00000000
54	GeoHash	hashLengthToCoverBoundingBox	測試此 bounding box 所對應之 hash length	52.4, 4.9, 52.3, 5	3
55	GeoHash	hashContains	測試此 hash 是否為此經緯度轉換出的 hash 之一	S06, 2, 3	true
56	GeoHash	coverBoundingBox	測試此 bounding box 所屬之 hash 的九宮格為何及其 ratio 是否正確	6, 4, 4, 6	{s0d, s0e, s0s, s0f, s0g, s0u, s14, s15, s1h}, 4.449462890625
57	GeoHash	coverBoundingBox	測試此 bounding box 所屬之 hash 的九宮格為何並限制其長度及其 ratio 是否正確	6, 4, 4, 6, 2	Sets{s0, s1}, 31.640625
57	GeoHash	heightDegrees	測試 hash degree 大於 max hash length 的結果	13	4.1909515857 6E-8
58	GeoHash	gridAsString	測試此 hash 周圍 size 大小的格子	“dr”, 2,Set{“	cc f1 f3 f9 fc cb f0 F2 F8 fb

			為何並將 highlight 部分轉為大寫	f2", "f8"}	9z dp dr dx dz 9y dn dq dw dy 9v dj dm dt dv
59	GeoHash	gridAsString(String hash, int fromRight, int fromBottom, int toRight, int toBottom)	測試此 hash 周圍的格子為何	"dr", -1, -1, 1, 1	f0, f2, f8 dp, dr, dx dn, dq, dw
60	Geomem	find	測試此 bounding box 轉出的經緯度是否正確(需先將經緯度登記在 map 中)	6, 4, 4, 6, 0, 10	lat=4.921875, lon=4.921875, time=3, value=18, id=Optional,of(18)

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	testEncodeBase32	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java
2	testEncodeBase32_2	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java
3	testDecodeBase32	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java
4	testGetCharIndex	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java
5	testPadLeftWithZeroToLength	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/Base32Test.java
6	testCoverage	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java

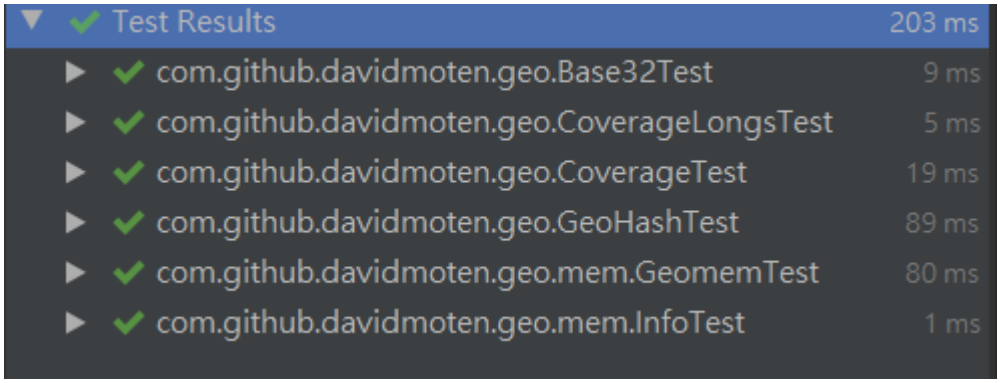
7	testGetHashes	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java
8	testGetRatio	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java
9	testGetHashLength	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java
10	testToString	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageTest.java
11	testGetHashes	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageLongsTest.java
12	testGetRatio	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageLongsTest.java
13	testGetHashLength	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageLongsTest.java
14	testGetCount	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/CoverageLongsTest.java
15	testId	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java
16	testLat	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java
17	testLon	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java
18	testTime	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java
19	testValue	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/

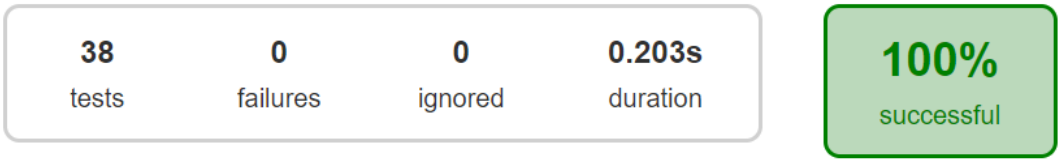
		geo/mem/InfoTest.java
20	testToString	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/InfoTest.java
21	testRight	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
22	testLeft	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
23	testTop	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
24	testBottom	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
25	testAdjacentHash	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
26	testNeighbours	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
27	testEncodeHashWithMaxHashLength	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
28	testEncodeHashWithLatAndLon	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
29	testEncodeHashWithLatLonAndMaxLength	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
30	testFromLongToString	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
31	testHashLengthToCoverBoundingBox	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
3	testHashContains	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/

2		blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
33	testCoverBoundingBox	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
34	testCoverBoundingBoxWithLength	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
35	testHeightDegrees	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
36	testGridAsString	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
37	testGridAsStringWithLine	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/GeoHashTest.java
38	testFind	https://stv.csie.ntut.edu.tw/liuhongde/GeoProject/blob/master/src/test/java/com/github/davidmoten/geo/mem/GeomemTest.java

4 Test Results

4.1 JUnit test result snapshot



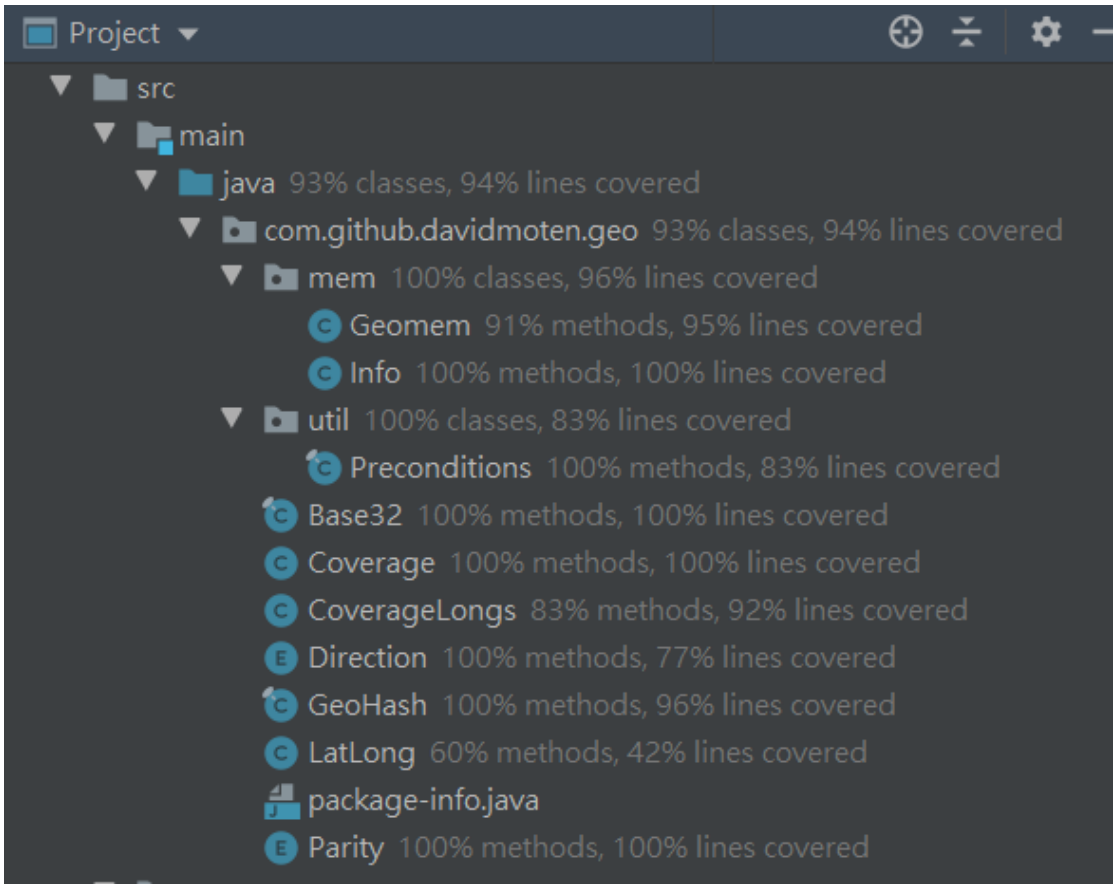


Packages | Classes

Package	Tests	Failures	Ignored	Duration	Success rate
com.github.davidmoten.geo	31	0	0	0.122s	100%
com.github.davidmoten.geo.mem	7	0	0	0.081s	100%

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%		87%	23	149	20	348	3	68	0	10
com.github.davidmoten.geo.mem		96%		65%	8	30	2	61	1	20	0	3
com.github.davidmoten.geo.util		68%		75%	1	4	1	6	0	2	0	1
Total	132 of 2,326	94%	28 of 186	84%	32	183	23	415	4	90	0	14

4.3 CI result snapshot (3 iterations for CI)

- CI#1

- CI#2

- CI#3

- CI#4

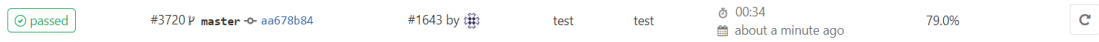
- CI#5

- CI#6

- CI#7

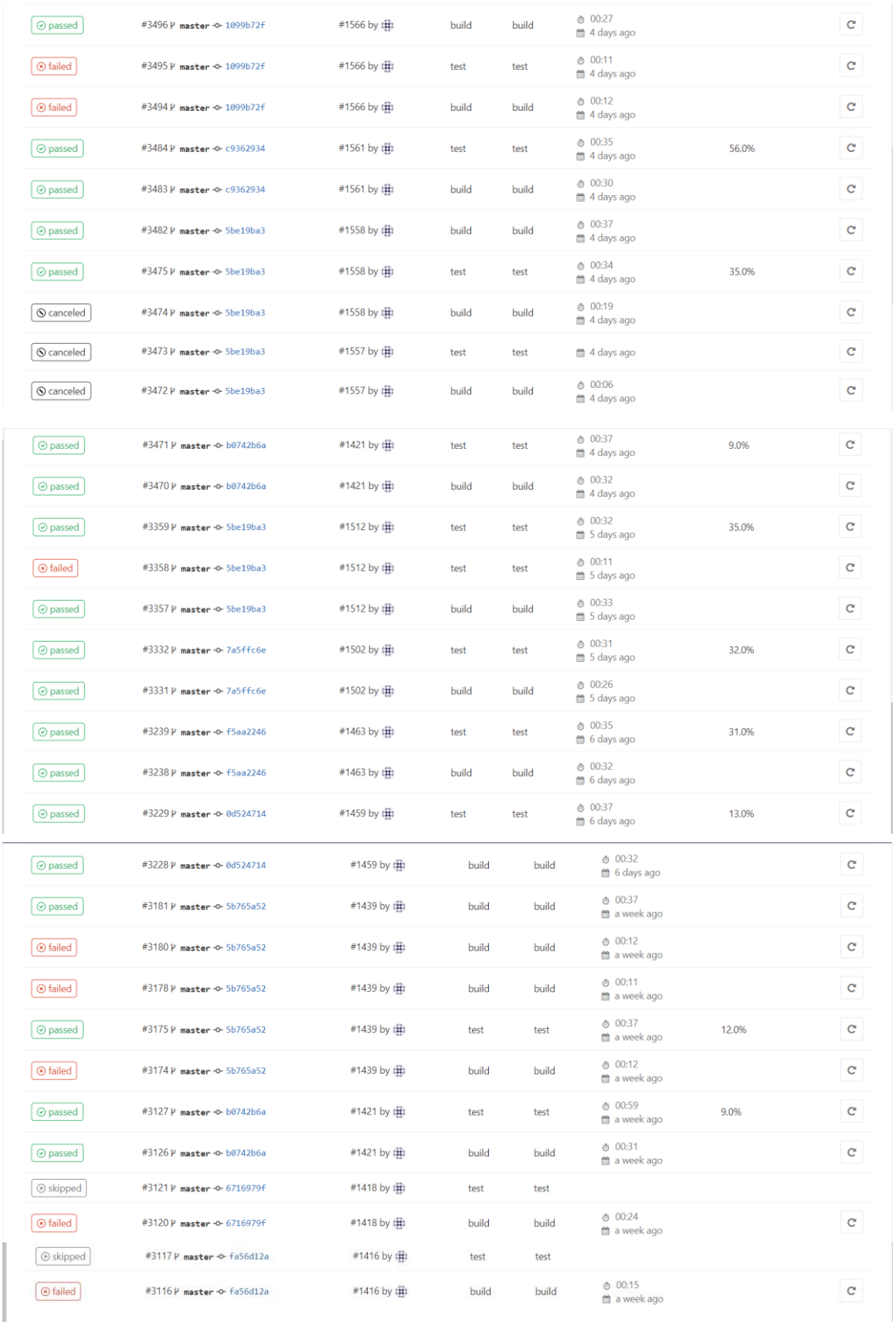
- CI#8

- CI#9



- CI#10

- CI Pipeline



5 Summary

In Lab 1, **38 test cases have been designed and implemented using JUnit**. The test is conducted in 10 CI and **the execution results of the 38 test methods**

are all passed. **The total statement coverage of the test is 80%.** Thus, the test requirements described in Section 1 are satisfied.