

# Lab 1 Report

Name: 陳日揚

Student ID: 106598014

Date: 2018/3/14

## 1 Test Plan

### 1.1 Test requirements

Lab1 要求從 GeoProject 中(共 6 個 Classes·50 個 Methods)挑選出 15 個 methods 來進行測試。首先，必須先理解及分析待測程式的功能及目的為何？分析完之後便可根據程式碼的資訊來設計相對應的 Test cases。接著，將設計的 Test cases 轉換為可實際執行測試的程式碼。最後將測試的結果記錄在本測試報告中。

根據題目要求，15 個待測的 Methods 皆必須被我們所設計出來的 Test cases 包含。同時，整體程式的 statement coverage 要達到 35%。

### 1.2 Strategy

為了達成 Section1.1 所描述的需求，將使用以下的策略：

- (1) 挑選參數值與回傳值只包含 primitive type 的 Methods 以方便測試。
- (2) 在設計測試案例前先分析與理解待測程式執行的內容，同時也必須瞭解該領域的 domain knowledge 才能正確地定義出測試案例的輸出是否正確。
- (3) 學習 Junit 單元測試工具及查看 Coverage 與專案自動建置的 Bundle 工具。
- (4) 利用邊界測試(Boundary Testing)來確保 Method 的輸入參數在極限值的情況下還是能夠正確地執行。

### 1.3 Test activities

下列為本次測試過程中所包含的活動。

No.	Activity Name	Plan hours	Schedule Date
1	Study GeoProject		2018/3/13
2	Learn JUnit		2018/3/13
3	Design test cases for the selected methods		2018/3/14
4	Implement test cases		2018/3/14 - 2018/3/15
5	Perform test		2018/3/14-2018/3/15
6	Complete Lab1 report		2018/3/14-2018/3/15

## 1.4 Success criteria

因為待測程式是一 Open Source 的 Library，理論上正確率應該非常高，因此本測試所設計之 Test cases 的通過率至少需達成 95%以上，並且單一 Method 的 Statement coverage 必須至少達成 80%。

## 2 Test Design

下表為設計的 Test cases 詳細資訊：

No	Class	Method	Test Objective	Inputs	Expected Outputs
1	Base32	String encodeBase32 (long i, int length)	正值座標值編碼	i=75324, length=4	29jw
2			負值座標值編碼	i=-75324, length=4	-29jw
3			最大值座標編碼	i=Long.MAX_VALUE, length=4	7zzzzzzzzzzzz
4			最小值座標編碼	i=Long.MIN_VALUE, length=4	-8000000000000
5		String encodeBase32 (long i)	最大 Hash 長度正 值座標值編碼	i=75324	000000029jw
6			最大 Hash 長度負 值座標值編碼	i=-75324	-000000029jw
7		long decodeBase32 (String hash)	正值座標值編碼	hash=29jw	75324
8			負值座標值編碼	hash=-29jw	-75324
9			最大值座標編碼	hash=7zzzzzzzzzzzz	Long.MAX_VALUE
10			最小值座標編碼	hash=-8000000000000	Long.MIN_VALUE
11		int getCharIndex (char ch)	取得編碼字元的 index	char=b	10
12			取得不在編碼字元 表內的例外情況	char=a	Throw IllegalArgumentException
13		String padLeftWithZeros ToLength (String s, int length)	s length > length 情況	s=4bca, length=8	00004bca
14			s length < length 情況	s=4bca, length=2	4bca

No	Class	Method	Test Objective	Inputs	Expected Outputs
15	GeoHash	String right(String hash)	取得右方相鄰的 Hash 值	hash=29jw	29jy
16			取得右方相鄰的 Hash 值但輸入為 null 的例外情況	hash=null	Trow IllegalArgument Exception
17		String left(String hash)	取得左方相鄰的 Hash 值	hash=29jw	29jq
18			取得左方相鄰的 Hash 值但輸入為 null 的例外情況	hash=null	Trow IllegalArgument Exception
19		String top(String hash)	取得上方相鄰的 Hash 值	hash=29jw	29jx
20			取得上方相鄰的 Hash 值但輸入為 null 的例外情況	hash=null	Trow IllegalArgument Exception
21		String bottom (String hash)	取得下方相鄰的 Hash 值	hash=29jw	29jx
22			取得下方相鄰的 Hash 值但輸入為 null 的例外情況	hash=null	Trow IllegalArgument Exception
23	Info	T id()	在 id 有輸入的情況下，取得 id 正確值	T=Integer, id=30	30
24			在 id 為 Null 的情況下，id().isPresent()應該為 False	T=Integer, id=null	info.id().isPresent() == false
25		double lat()	取得緯度正確值	lat=23.25	23.25
26		double lon()	取得經度正確值	lon=120.55	120.55
27		long time()	取得時間正確值	time=123	123
28		R value()	取得 value 正確值	R=String, value=2	2
29		String toString()	取得正確格式的回傳字串	Info<String, Integer> (23.25, 120.55, 123, "2", Optional.of(30))	Info [lat=23.25, lon=120.55, time=123, value=2, id=Optional.of(30)]

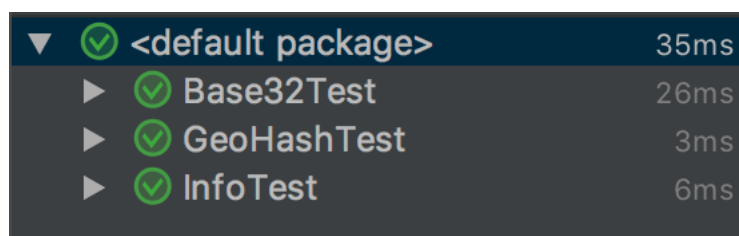
### 3 Test Implementation

本次 Lab 使用的測試工具為 Junit 4，下列挑選 Section 2 中設計三個 Test cases 實作內容，未列出的測試內容可在 GitLab 上查看。

No.	Test method	Source code
1	encodeBase32_positive ()	<pre>@Test public void encodeBase32_positive() throws Exception {     String encode = Base32.encodeBase32(75324, 4);     assertEquals("29jw", encode); }</pre>
2	testMethod2()	<pre>@Test public void right() {     String adjacentHash = GeoHash.right("29jw");     assertEquals("29jy", adjacentHash); }</pre>
3	testMethod3()	<pre>@Test public void idTest_null() {     Integer i = null;     Info&lt;String, Integer&gt; info = new Info&lt;String,     Integer&gt;(23.25, 120.55, 123, "2",     Optional.fromNullable(i));     assertTrue(!info.id().isPresent()); }</pre>

### 4 Test Results

#### 4.1 JUnit test result snapshot



▼	✓ <default package>	35ms
▶	✓ Base32Test	26ms
▶	✓ GeoHashTest	3ms
▶	✓ InfoTest	6ms

## Test Summary

29  
tests

0  
failures

0  
ignored

0.162s  
duration

100%  
successful

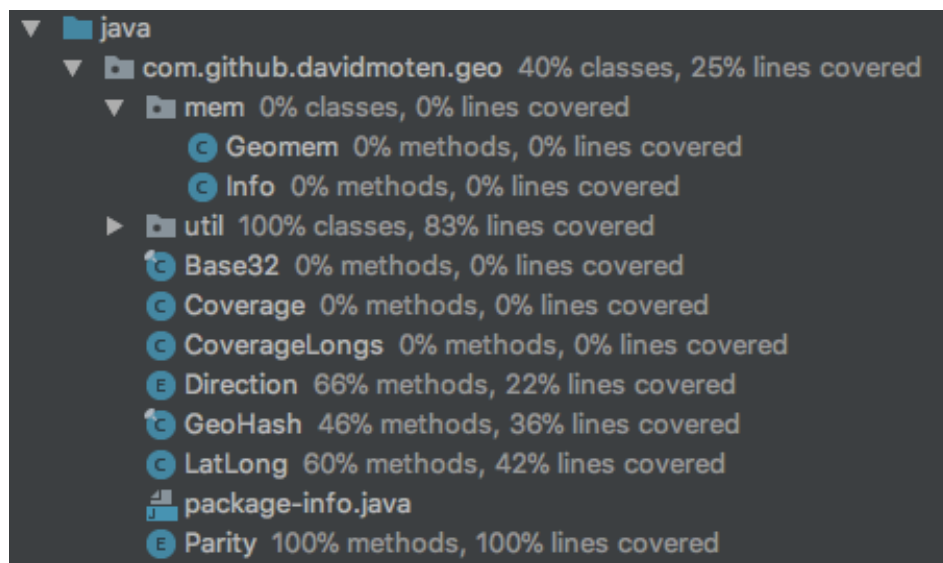
Packages

Classes

Package	Tests	Failures	Ignored	Duration	Success rate
<a href="#">com.github.davidmoten.geo</a>	22	0	0	0.153s	100%
<a href="#">com.github.davidmoten.geo.mem</a>	7	0	0	0.009s	100%

## 4.2 Code coverage snapshot




- Coverage of each selected method



- Total coverage

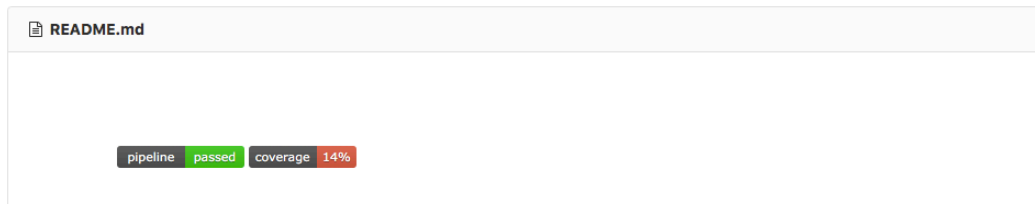
geo

geo

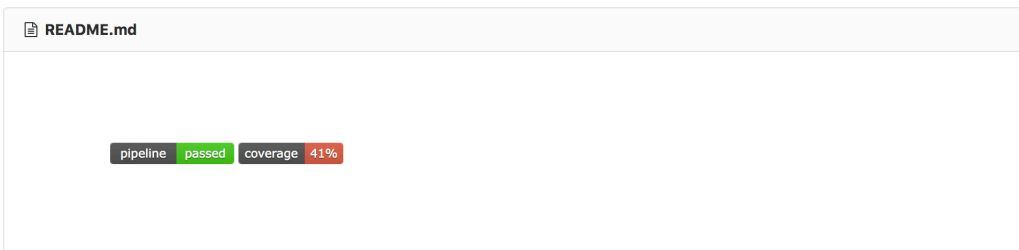
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
 com.github.davidmoten.geo	<div><div></div><div></div></div>	48%	<div><div></div><div></div></div>	33%	105	159	211	354	46	78	4	10
 com.github.davidmoten.geo.mem	<div><div></div><div></div></div>	19%	<div><div></div><div></div></div>	0%	23	30	48	61	13	20	2	3
 com.github.davidmoten.geo.util	<div><div></div><div></div></div>	59%	<div><div></div><div></div></div>	75%	2	5	3	8	1	3	0	1
Total	1,338 of 2,379	44%	130 of 186	30%	130	194	262	423	60	101	6	14

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

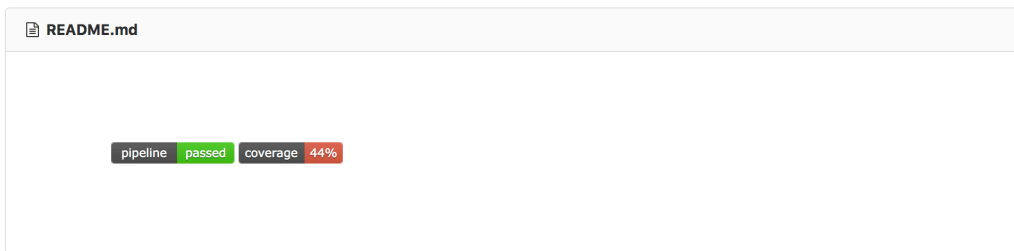
- CI#1



- CI#2



- CI#3



- CI Pipeline

106598014 > GeoProject > Pipelines

All 4					Pending 0	Running 0	Finished 4	Branches	Tags
Status	Pipeline	Commit	Stages						
	#98 by  latest	P master -> 4dc51b8d Finish GeoHashTest.		00:01:35 about 11 hours ago					
	#95 by	P master -> 5c18daed Finish InfoTest.		00:01:25 about 13 hours ago					
	#82 by	P master -> 8f130408 Finish Base32Test.		00:01:53 a day ago					
	#71 by	P master -> 5ede9534 Update README.md		00:01:04 2 days ago					

## 5 Summary

在此次的 Lab1 中，總共設計並利用 JUnit 執行了 15 個單元測試，測試的過程中共執行了三次的 CI。測試結果全數通過，且整體的 Statement coverage 為 44%，有達成 Section 1 所要求的 35% 覆蓋率。同時，在這次 Lab 的執行過程中也學習到了 Junit 測試工具的使用以及 GeoHashing 的基本概念。