**Lab 2 Report**

Name:朱峻平

Student ID:107598058

Date:2019/3/28

1. **Test Plan**
   1. **Test requirements**

The Lab 2 requires to (1) select 15 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 at least one test case* and *the minimum statement coverage is 70% (greater than Lab 1)*”.

* 1. **Test Strategy**

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

1. 選較有變化或程式較複雜的method去設計Test case盡量達到最大statement覆蓋率。
2. 使用ISP 設計Test case 完成所有可能性的測試盡量達到該method最大statement覆蓋率。
3. Getter or Setter不特別進行全數完整ISP設計只做基本測試。
   1. **Test activities**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Activity Name** | **Plan hours** | **Schedule Date** |
| 1 | Study GeoProject | 3 | 3/25 |
| 2 | Learn **ISP** and JUnit | 2 | 3/25 |
| 3 | Design test cases for the selected methods | 12 | 3/26 |
| 4 | Implement test cases | 5 | 3/27 |
| 5 | Perform tests | 1 | 3/28 |
| *n* | Complete Lab2 report | 3 | 3/28 |

* 1. **Design Approach**

使用ISP設計Test Case需先辨別輸入參數可能的partitions 和邊界值，針對input設計條件，並設計可能組合以及邊界值，最終根據這些組合，去設想測試資料並實作出來，最終將所有可能組合cover住，確保method測試完整性。

我在這次的lab都會先設想什麼樣情況此method會丟例外，去找參數可設計值，再設想測試條件，並經過組合，最終實作並列出每種組合的設計參數以及期待值。

* 1. **Success criteria**

所以用ISP設計的Test case要全數通過,statement覆蓋率達70%以上。

1. **Test Design**

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

**詳細Design放在master/LabReport/Lab2與lab2報告在同層資料夾下。**

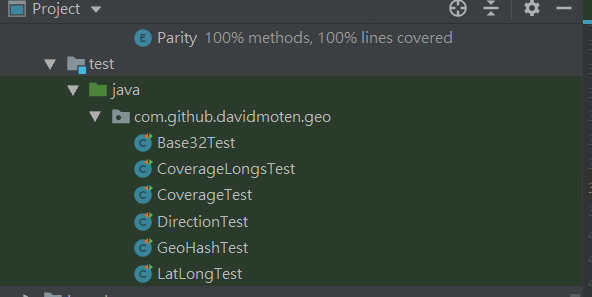
The Excel file of test cases… [link](https://stv.csie.ntut.edu.tw/107598058/GeoProject/tree/master/LabReport/Lab2)

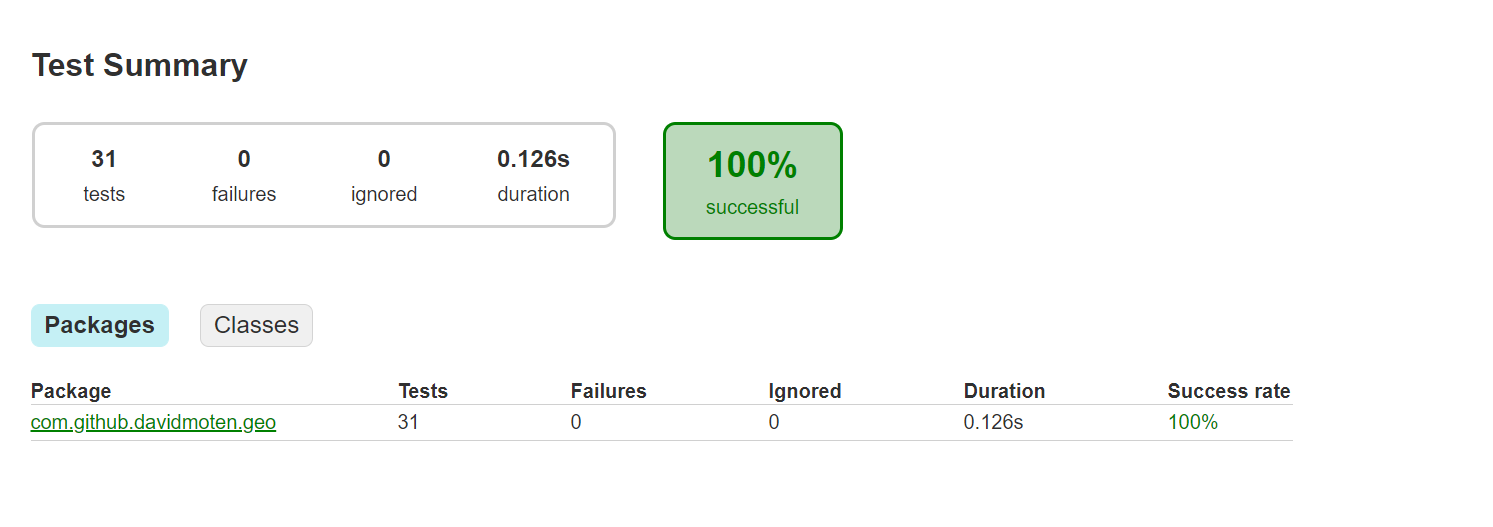
1. **Test Implementation**

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 [link](https://stv.csie.ntut.edu.tw/107598058/GeoProject.git) (or JUnit files).

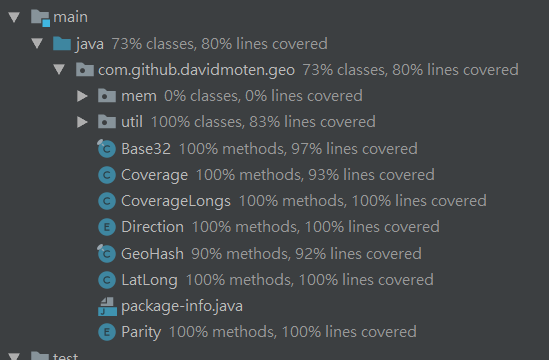
|  |  |  |
| --- | --- | --- |
| **No.** | **Test method** | **Source code** |
| 1 | padLeftWithZerosToLength() | @Test public void padLeftWithZerosToLength() {  *assertEquals*("0000Test1", Base32.*padLeftWithZerosToLength*("Test1",9));  *assertEquals*("Test1", Base32.*padLeftWithZerosToLength*("Test1",-1));  *assertEquals*("000000000", Base32.*padLeftWithZerosToLength*("",9)); } |
| 2 | getCharIndex() | @Test public void getCharIndex() {  *assertEquals*(0, Base32.*getCharIndex*('0'));  *assertEquals*(10, Base32.*getCharIndex*('b'));  try{  *assertEquals*(3, Base32.*getCharIndex*('a'));  }  catch(Exception e)  {  System.*out*.println(e);  } } |
| 3 | decodeBase32() | @Test public void decodeBase32() {  long ans = Base32.*decodeBase32*("bcde");  *assertEquals*(339341, ans);   try{  long ans\_2 = Base32.*decodeBase32*("a");  }  catch (Exception e){  System.*out*.println(e);  }    try{  long ans\_3 = Base32.*decodeBase32*("-a");  }  catch (Exception e){  System.*out*.println(e);  }  } |

1. **Test Results**
   1. **JUnit test result snapshot**

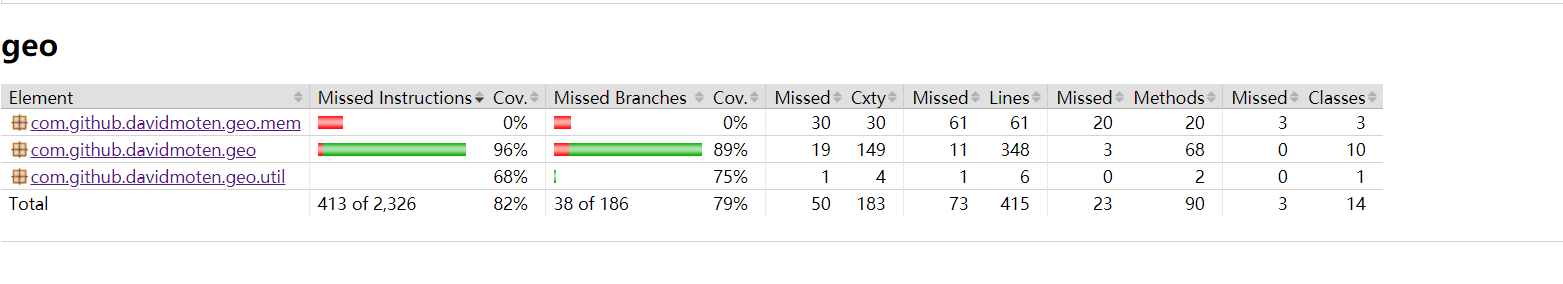
****

****

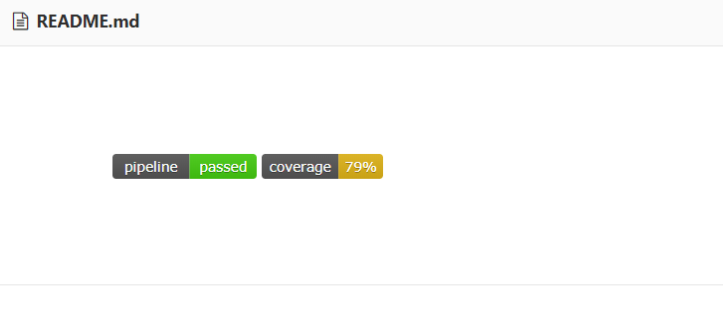
* 1. **Code coverage snapshot**
* Coverage of each selected method

****

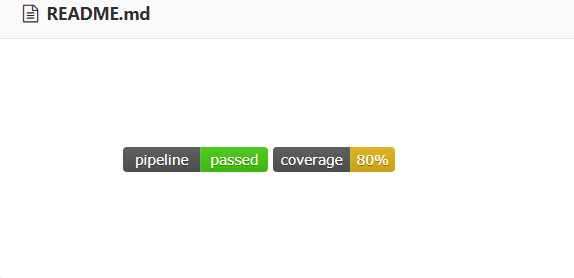
* Total coverage

****

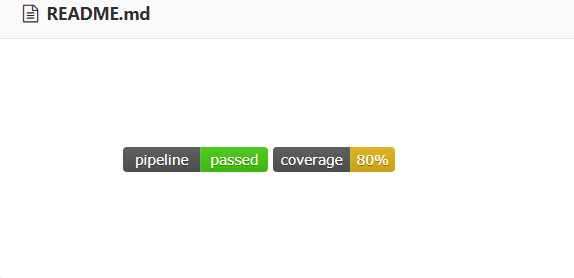
* 1. **CI result snapshot (3 iterations for CI)**
* CI#1

****

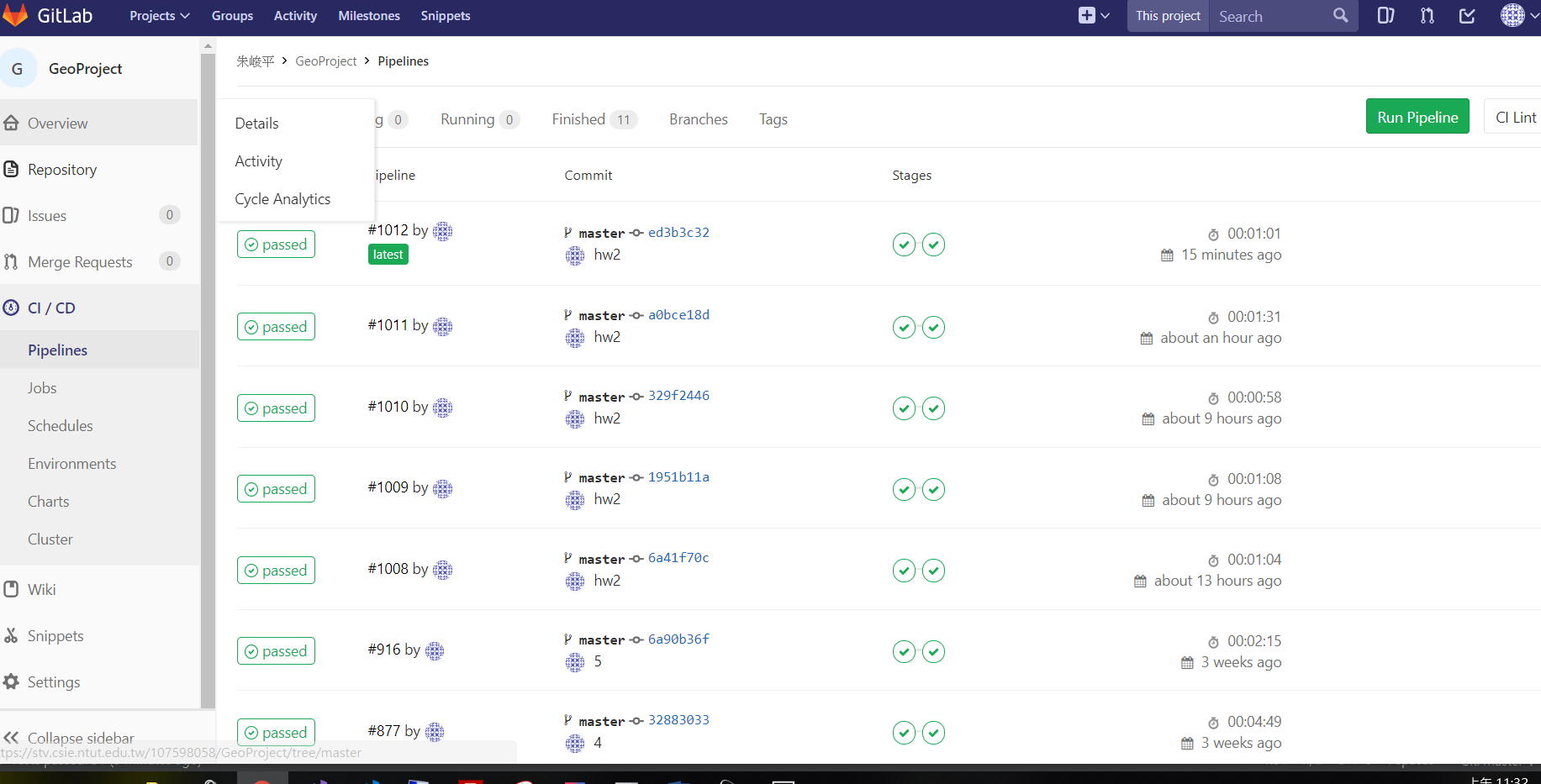
* CI#2



* CI#3



* CI Pipeline

****

1. **Summary**

在Lab2中我列出了15個我使用ISP設計，覺得具代表性的test case，這次作業我已經習慣JUnit以及新的IDE開發工具，這次花了比較多時間在事前設計，列出所有可能，以及設想測試資料該如何設計才能達到最大statement coverage，所以在撰寫測試上遵循自己的策略，沒有遇到太大的問題，最後測試覆蓋率達到80%，之後會再研究程式碼補足我測試沒覆蓋到的部分，以上是我的心得。