User Manual **使用手冊**

本成績系統(a grade system)讀入全班成績，有 lab1, lab2, lab3, midterm exam, final exam 等成績，內建各成績的配分(weights)為0.1, 0.1, 0.1, 0.3, 0.4，依配分算出總成績(total grade)，供使用者輸入指令(command)、查詢成績(grade)及排名(rank)，並可更新配分(update weights)重算總成績，系統可處理三種異常狀況：1) ID 含字母 2) ID長度不對3) 無這ID

系統主要畫面如下：

UI顯示：輸入ID或 Q (結束使用)？ 使用者輸入ID 或Q

若有異常狀況則UI顯示：這ID含字母 錯了! 或 這ID長度不對 錯了! 或 無這ID 錯了!

否則顯示ID對應的名字。

E.G 輸入李威廷的ID（962001051） UI顯示: Welcome 李威廷

輸入Q (結束使用) UI顯示: 結束了

輸入指令 1) G 顯示成績 (Grade)

2) R 顯示排名 (Rank)

3) W更新配分 (Weight)

4) E 離開選單 (Exit)

如輸入指令 G 則UI顯示：   
李威廷成績：lab1： 81

lab2： 98

lab3： 84

midterm exam : 90

final exam: 93

total grade : 91

如輸入指令 R 則UI顯示：  
 李威廷排名第22

如輸入指令 W 則UI顯示：  
 舊配分

lab1 10%

lab2 10%

lab3 10%

midterm exam 30%

final exam 40%

輸入新配分

lab1 20

lab2 20

lab3 20

midterm exam 20

final exam 20

請確認新配分

lab1 20%

lab2 20%

lab3 20%

midterm exam 20%

final exam 20%

以上正確嗎? Y (Yes) 或 N (No)

使用者接著不斷輸入上述指令，直到輸入E (Exit離開選單)。

UI再度顯示：輸入ID或 Q (結束使用)？ 使用者輸入Q (結束使用) 系統就結束了。

Minimal Input file **極簡輸入檔**

下麵是開發時trace pseudo-code所用的極簡輸入檔:

962001044 淩宗廷 87 86 98 88 87

962001051 李威廷 81 98 84 90 93

上面資料的 欄位名稱 如下:

ID name lab1 lab2 lab3 midtermExam finalExam

真正的輸入檔 ”gradeInput.txt” 要先備齊(見後附)，但要到最後才 run。

Acceptance Test Cases **驗收測試案例** (紅色表示使用者輸入)

1. A user interface (aUI) 顯示：輸入ID或 Q (結束使用)？ Q

aUI顯示：結束了

1. aUI顯示：輸入ID或 Q (結束使用)？ 962001051

aUI顯示：Welcome 李威廷

輸入指令 1) G 顯示成績 (Grade)

2) R 顯示排名 (Rank)

3) W更新配分 (Weight)

4) E 離開選單 (Exit)

使用者輸入： E

aUI顯示：輸入ID或 Q (結束使用)？ Q

aUI顯示：結束了

1. 開始至輸入指令如上 (註：重複部分不寫，使檔減量)

使用者輸入： G

aGradeSystem顯示：   
李威廷成績：lab1： 81

lab2： 98

lab3： 84

midterm exam: 90

final exam： 93

total grade : 91

輸入指令如上

使用者輸入： E

aUI顯示：輸入ID或 Q (結束使用)？ Q

aUI顯示：結束了

1. 開始至輸入指令如上

使用者輸入： R

aGradeSystem顯示：

做單元測試時，因為使用只有兩筆資料的極簡輸入檔，所以顯示：

**李威廷排名第 1**

當使用真正輸入檔時，則顯示：

**李威廷排名第22**

(例子: 李陳張 totalGrade 分別為 86 86 83，則三人的rank為1 1 3)

輸入指令如上

使用者輸入：W

aGradeSystem顯示：

舊配分

lab1 10%

lab2 10%

lab3 10%

midterm exam 30%

final exam 40%

輸入新配分

lab1 20

lab2 20

lab3 20

midterm exam 20

final exam 20

請確認新配分

lab1 20%

lab2 20%

lab3 20%

midterm exam 20%

final exam 20%

以上正確嗎? Y (Yes) 或 N (No)

使用者輸入：Y

輸入指令如上

使用者輸入： E

aUI顯示：輸入ID或 Q (結束使用)？ Q

aUI顯示：結束了

1. aUI顯示：輸入ID或 Q (結束使用)？ 96z001051

aUI顯示：這ID含字母 錯了!

aUI顯示：輸入ID或 Q (結束使用)？ Q

aUI顯示：結束了

1. aUI顯示：輸入ID或 Q (結束使用)？ 96200105

aUI顯示：這ID長度不對 錯了!

aUI顯示：輸入ID或 Q (結束使用)？ Q

aUI顯示：結束了

1. aUI顯示：輸入ID或 Q (結束使用)？ 123456789

aUI顯示：沒有這ID 錯了!

aUI顯示：輸入ID或 Q (結束使用)？ Q

aUI顯示：結束了

1. 開始至輸入指令如上

使用者輸入： G

aGradeSystem顯示：   
李威廷成績：lab1： 81

lab2： 98

lab3： 84

midterm exam: 90

final exam： 93

total grade : 91

輸入指令如上

使用者輸入： R

**李威廷排名第 1**

輸入指令如上

使用者輸入： E

aUI顯示：輸入ID或 Q (結束使用)？ Q

aUI顯示：結束了

**Test Code 例子**

對每個 public method寫test code。用 parameters 及 calling object的不同組合形成多個 test cases，每個test case寫成一個test method，而這些test methods則組成一個test class。

注意：從前面的架構設計可看出除main()外，共有13個public methods要做單元測試，所以要開發對應的13個 test classes。其中constructor也要測其是否成功建構object。

注意：test classes 中的命名要用已定的名(如ID)，不可更改新名(如studentNumber)。

下麵是method “GradeSystems ” 的Test class：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class TestGradeSystems

參閱下麵細部設計中 GradeSystems class的設計草圖，測試用minimal input file建構的aGradeSystem object。

public testGradeSystrems1() /\*測試aGradeSystem object淩宗廷的lab1。\*/

1. aGradeSystem = new GradeSystems();
2. assert equality of: 1) 87 and 2) aGradeSystem.aTree.get(“962001044”).lab1

public testGradeSystrems2()/\*測aGradeSystem object李威廷的lab1。\*/

1. aGradeSystem = new GradeSystems();
2. assert equality of: 1) 81 and 2) aGradeSystem.aTree.(“962001051”).lab1

public testGradeSystrems3()/\*測aGradeSystem object商揚夏的lab2。\*/

1. aGradeSystem = new GradeSystems();
2. assert equality of: 1) 86 and 2) aGradeSystem.aTree(“965002044“).lab2

public testGradeSystrems3()/\*測aGradeSystem object陳建豪的lab3。\*/

1. aGradeSystem = new GradeSystems();
2. assert equality of: 1) 85 and 2) aGradeSystem.aTree(“966002031“).lab3

下麵是method “showRank (ID) ” 的Test class：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class TestShowRank

aGradeSystem 僅含淩宗廷及李威廷

test case 1: assert 淩宗廷 rank equals 38

test case 2: assert 李威廷 rank equals 22

test case 3: assert 許文馨 rank equals 9

test case 4: assert 蘇亨玠 rank equals 60

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Public testShowRank1() )/\*測aGradeSystem object淩宗廷的rank。\*/

1. aGradeSystem = new GradeSystems();
2. assert equality of: 1) 38 and 2) aGradeSystem.showRank(“962001044 “)

Public testShowRank2()/\*測aGradeSystem object李威廷的rank。\*/

1. aGradeSystem = new GradeSystems();
2. assert equality of: 1) 22 and 2) aGradeSystem.showRank(“962001051“)

Public testShowRank3() /\*測aGradeSystem object許文馨的rank。\*/

1. aGradeSystem = new GradeSystems();
2. assert equality of: 1) 9 and 2) aGradeSystem.showRank(“955002056 “)

Public testShowRank4() /\*測aGradeSystem object蘇亨玠的rank。\*/

1. aGradeSystem = new GradeSystems();
2. assert equality of: 1) 60 and 2) aGradeSystem.showRank(“965002038 “)

下麵是method “calculateTotalGrade (weights) ” 的Test class：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-

class TestCalculateTotalGrade

aGrade: name李威廷 lab1 81 lab2 98 lab3 84 midtermExam 90 finalExam 93

test case 1: lab1 0.1 lab2 0.1 lab3 0.1 midtermExam 0.3 finalExam 0.4

test case 2: lab1 0.2 lab2 0.2 lab3 0.2 midTermExam 0.2 finalExam 0.2

test case 3: lab1 0.1 lab2 0.1 lab3 0.0 midtermExam 0.4 finalExam 0.4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public testCalculateTotalGrade1()

1. 先定義aGrade裡面的分數 aGrade.lab1 = 81, aGrade.lab2 = 98, aGrade.lab3 = 84, aGrade.midtermExam = 90, aGrade.finalExam = 93
2. 手算 expected result (81\*0.1+98\*0.1+84\*0.1+90\*0.3+93\*0.4=90.5四捨五入91)
3. aGrade call calculateTotalGrade (weights) 算actual result
4. assert equal of the two results

public testCalculateTotalGrade2()

1. 先定義aGrade裡面的分數 aGrade.lab1 = 81, aGrade.lab2 = 98, aGrade.lab3 = 84, aGrade.midtermExam = 90, aGrade.finalExam = 93
2. 手算 expected result (81\*0.2+98\*0.2+84\*0.2+90\*0.2+93\*0.2四捨五入89)
3. aGrade call calculateTotalGrade (weights) 算actual result
4. assert equal of the two results

public testCalculateTotalGrade3()

1. 先定義aGrade裡面的分數 aGrade.lab1 = 81, aGrade.lab2 = 98, aGrade.lab3 = 84, aGrade.midtermExam = 90, aGrade.finalExam = 93
2. 手算 expected result (81\*0.2+98\*0.2+84\*0.0+90\*0.4+93\*0.4四捨五入91)
3. aGrade call calculateTotalGrade (weights) 算actual result
4. assert equal of the two results

下麵是method “checkID(ID) ” 的Test class:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-

class TestCheckID

test case 1: ID 96z001044 (2打成z)

test case 2: ID 96200104 (遺漏最後的4)

test case 3: ID 962001044 (正確，淩宗廷的ID)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

@Test (expected = IDExceptions1.class)

public testCheckID1() throws IDExceptions1, IDExceptions2

aUI call checkID(“96z001044”)

@Test (expected = IDExceptions2.class)

public testCheckID2() throws IDExceptions1, IDExceptions2

aUI call checkID(96200104)

注意**：** 上面二cases 丟出異常而無回傳值，故 test code寫法不同。

public testCheckID3() throws IDExceptions1, IDExceptions2

1. 手算 expected result: true
2. aUI call checkID(962001044) 算actual result
3. assert equal of the two results

下面是method “promptID () ” 的部分的test code

**Case 1: testPromptID1()，使用者輸入的字串為”Q”**

/\*用 setIn 設定console input 為inContent，使用者輸入的字串Q \*/

ByteArrayInputStream inContent = new ByteArrayInputStream("Q".getBytes());

System.setIn(inContent);

/\*用 setOut設定console output 為outContent \*/

ByteArrayOutputStream outContent = new ByteArrayOutputStream();

System.setOut (new PrintStream (outContent));

/\* 執行promptID return使用者輸入的字串\*/

UI aUI = new UI();

String userInput = aUI.promptID();

/\*預期promptID會顯示"輸入ID或 Q (結束使用)？" \*/

assertEquals("輸入ID或 Q (結束使用)？", outContent);

/\*預期user input 是 Q \*/ assertEquals ("Q", userInput);

**Case 2: testPromptID2()，使用者輸入的字串為ID “962001044” (淩宗廷的ID)**

/\*用 setIn 設定console input，使用者輸入的字串 962001044 \*/

ByteArrayInputStream inContent = new ByteArrayInputStream("962001044".getBytes());

System.setIn(inContent);

/\*用 setOut設定console output\*/

ByteArrayOutputStream outContent = new ByteArrayOutputStream();

System.setOut (new PrintStream (outContent));

/\* 執行promptID return使用者輸入字串\*/

UI aUI = new UI();

String ID = aUI.promptID();

/\*預期promptID會顯示"輸入ID或 Q (結束使用)？"\*/

assertEquals("輸入ID或 Q (結束使用)？", outContent);

assertEquals ("962001044", ID);

下面是method “showFinishMsg() ” 的部分的test code

public testShowFinishMsg()

1. final ByteArrayOutputStream outContent = new ByteArrayOutputStream();

2. System.setOut (new printStream (outContent));

3. aUI call showFinishMsg();

4. assertEquals(“結束了\r\n”, outContent.toString());

下面是method “updateWeight ” 的test code：

public testUpdateWeight()

1. assert last INCONTENT equals “Y”
2. assert last OUTCONTENT equals “以上正確嗎? Y (Yes) 或 N (No)\r”

下面是測試 constructor的test code：

public testUI1() {assertNotNull ( new UI() );}

注意：UI class 沒宣告data 無法依上例用assertEquals 則測是否確有建構出此object

public testGrades1() {assertNotNull (new Grades() ); }

注意：因testGradeSystems1已測aGrade內值 故只測是否建構此object。

下面是測試showWelcomeMsg 的test code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-

class showWelcomeMsg

test case 1: ID 962001051 (李威廷)

test case 2: ID 962001044 (凌宗廷)

test case 3: ID 955002056 (許文馨)

test case 4: ID 955002056 (許琇筑)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

@Test

public void TestShowWelcomeMsg1() throws IOException{

UI aUi = new UI();

AssertEquals(true, aUi.showWelcomeMsg(“962001051”)) // 李威廷

}

@Test

public void TestShowWelcomeMsg2() throws IOException{

UI aUi = new UI();

AssertEquals(true, aUi.showWelcomeMsg(“962001044”)) ; // 凌宗廷

}

@Test

public void TestShowWelcomeMsg3() throws IOException{

UI aUi = new UI();

AssertEquals(true, aUi.showWelcomeMsg(“955002056”)) ; // 許文馨

}

@Test

public void TestShowWelcomeMsg4() throws IOException{

UI aUi = new UI();

AssertEquals(true, aUi.showWelcomeMsg(“955002056”)) ; // 許琇筑

}

下面是測試showUpdateWeights 的test code:

@Test

public void TestUpdateWeights1() throws IOException {

GradeSystems aGS = new GradeSystems();

assertEquals(true, aGS.updateWeights());

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-

class PromptCommand

test case 1: G (show grade)

test case 2: R (show rank)

test case 3: W (weight)

test case 4: E (Exit)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

下面是測試 PromptCommand 的test code:

@Test

public void TestPromptCommand1() {

UI aUI = new UI(); // user 输入G/R/W/E

assertEquals("G", aUI.promptCommand());

}

@Test

public void TestPromptCommand1() {

UI aUI = new UI(); // user 输入G/R/W/E

assertEquals("R", aUI.promptCommand());

}

@Test

public void TestPromptCommand1() {

UI aUI = new UI(); // user 输入G/R/W/E

assertEquals("W", aUI.promptCommand());

}

@Test

public void TestPromptCommand1() {

UI aUI = new UI(); // user 输入G/R/W/E

assertEquals("E", aUI.promptCommand());

}

下面是測試 ShowGrade 的test code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-

class ShowGrade

test case 1: 962001044 (凌宗廷) total grade: 446

test case 2: 962001051(李威廷) total grade: 446

test case 3: 965002038 (蘇亨玠) total grade: 441

test case 4: 965002044 (商揚夏) total grade: 420

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

@Test

public void TestShowGrade1() throws IOException {

GradeSystems aGS = new GradeSystems();

assertEquals(446, aGS.showGrade("962001044"));} //凌宗廷

@Test

public void TestShowGrade2() throws IOException {

GradeSystems aGS = new GradeSystems();

assertEquals(446, aGS.showGrade("962001051"));} //李威廷

@Test

public void TestShowGrade3() throws IOException {

GradeSystems aGS = new GradeSystems();

assertEquals(441, aGS.showGrade("965002038"));} //蘇亨玠

@Test

public void TestShowGrade4() throws IOException {

GradeSystems aGS = new GradeSystems();

assertEquals(420, aGS.showGrade("965002044"));} //商揚夏

下面是測試 ContainID 的test code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-

class ContainID

test case 1: 962001044 (凌宗廷)

test case 2: 955002056(許文馨)

test case 3: 965002044 (商揚夏)

test case 4: 975002021 (楊祺賢)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

@Test

public void testContainID1() throws IDExceptions3, IOException{

GradeSystems aGS = new GradeSystems();

boolean flag = aGS.containID("962001051");

assertEquals(flag, true);}

@Test

public void testContainID2() throws IDExceptions3, IOException{

GradeSystems aGS = new GradeSystems();

boolean flag = aGS.containID("955002056");

assertEquals(flag, true);}

@Test

public void testContainID3() throws IDExceptions3, IOException{

GradeSystems aGS = new GradeSystems();

boolean flag = aGS.containID("965002044");

assertEquals(flag, true);}

@Test

public void testContainID4() throws IDExceptions3, IOException{

GradeSystems aGS = new GradeSystems();

boolean flag = aGS.containID("975002021");

assertEquals(flag, true);}