軟體工程-HW2 Document 鄭凱文 104062223 賴明毅 104062201

**<User manual>**

本成績系統透過gradeinput.txt，讀入全班成績，包含lab1, lab2, lab3, mid-term, final exam 等成績，各成績的加權配分預設為0.1, 0.1, 0.1, 0.3, 0.4，系統依配分算出總成績total grade建檔，供使用者輸入ID查詢自己的成績及排名，同時可以更新配分重算總成績。

系統可處理三種異常：1) 輸入錯誤ID 2) 輸入不正確指令 3)資料格式錯誤

* 系統主要畫面如下

螢幕顯示：輸入ID或 Q (結束使用)？

使用者輸入ID Ex:962001051(李威廷的ID)

螢幕顯示：

Welcome 李威廷

輸入指令 1) G 顯示成績 (Grade)

2) R 顯示排名 (Rank)

3) A 顯示平均 (Average)

4) W更新配分 (Weight)

5) E 離開選單 (Exit)

如輸入指令 ”G”，螢幕顯示：

李威廷成績：lab1： 81

lab2： 32\*

lab3： 50\*

mid-term : 90

final exam： 93

total grade : 81

如輸入指令 ”R”，螢幕顯示：

李威廷排名第63

如輸入指令 ”A”，螢幕顯示：

班平均： lab1： 90

lab2： 88

lab3： 89

mid-term： 90

final exam： 90

total grade： 90

如輸入指令 ”W”，螢幕顯示：

舊配分

lab1 10%

lab2 10%

lab3 10%

mid-term 30%

final exam 40%

輸入新配分(提醒：每項需介於0~100之間，五項總和必須等於100)

lab1 15

lab2 15

lab3 15

mid-term 25

final exam 30

請確認新配分

lab1 15%

lab2 15%

lab3 15%

mid-term 25%

final exam 30%

以上正確嗎？Y(Yes)或 N(No)

每次指令結束後，系統會提示使用者繼續輸入，直到使用者輸入E(離開選單)

螢幕再度顯示：輸入ID或 Q (結束使用)？

使用者輸入 ”Q”，系統就結束了

**<Acceptance test cases>** (紅色表示使用者輸入)

* 正常使用情形

1. 螢幕promptID 或 結束使用

-> 使用者選擇結束使用

螢幕 showFinishMsg

2. 螢幕promptID或 結束使用

-> 使用者選擇輸入ID

螢幕showWelcomeMsg

螢幕 promptCommand:

顯示四種指令1) 顯示成績 2) 顯示排名 3) 顯示平均 4) 更新配分

5) 離開選單

-> 使用者選擇離開選單

螢幕promptID或 結束使用

-> 使用者選擇結束使用

螢幕showFinishMsg

3. 螢幕promptID或 結束使用

-> 使用者輸入ID

螢幕showWelcomeMsg

螢幕 promptCommand

-> 使用者選擇顯示成績

螢幕 showGrade(ID)

螢幕 prompCommand

-> 使用者要離開選單

螢幕promptID或 結束使用

-> 使用者要結束使用

螢幕showFinishMsg

4. 螢幕promptID或 結束使用

-> 使用者輸入ID

螢幕showWelcomeMsg

螢幕 promptCommand

-> 使用者選擇顯示排名

螢幕 showRank(ID)

螢幕 prompCommand

-> 使用者要離開選單

螢幕promptID或 結束使用

-> 使用者要結束使用

5. 螢幕promptID或 結束使用

-> 使用者輸入ID

螢幕showWelcomeMsg

螢幕 promptCommand

-> 使用者選擇顯示平均

螢幕 showAverage()

螢幕 prompCommand

-> 使用者要離開選單

螢幕promptID或 結束使用

-> 使用者要結束使用

6. 螢幕promptID或 結束使用

-> 使用者輸入ID

螢幕showWelcomeMsg

螢幕 promptCommand

-> 使用者選擇updateWeights (更新配分)

螢幕顯示舊配分

-> 使用者輸入新配分

螢幕 promptCommand

-> 使用者要離開選單

螢幕promptID或 結束使用

-> 使用者要結束使用

螢幕showFinishMsg

* Exceptional scenarios異常使用情節

7. 螢幕promptID或 結束使用

-> 使用者輸入錯誤的ID

螢幕顯示錯誤訊息

8. 螢幕promptID或 結束使用

-> 使用者輸入ID

螢幕showWelcomeMsg

螢幕 promptCommand

-> 使用者輸入不正確指令

螢幕顯示錯誤訊息

9. 螢幕promptID或 結束使用

-> 使用者輸入ID

螢幕showWelcomeMsg

螢幕 promptCommand

-> 使用者選擇updateWeights (更新配分)

螢幕顯示舊配分

-> 使用者輸入不合法的新配分配分

螢幕顯示錯誤訊息

* 正常使用情形

1.螢幕顯示：輸入ID或 Q (結束使用)？ Q

螢幕顯示：結束了

2.螢幕顯示：輸入ID或 Q (結束使用)？ 962001051

螢幕顯示：

Welcome 李威廷

輸入指令 1) G 顯示成績 (Grade)

2) R 顯示排名 (Rank)

3) A 顯示平均 (Average)

4) W 更新配分 (Weight)

5) E 離開選單 (Exit)

使用者輸入： E

螢幕顯示：輸入ID或 Q (結束使用)？ Q

螢幕顯示：結束了

3.開始至輸入指令如上

使用者輸入： G

螢幕顯示：

李威廷成績：lab1： 81

lab2： 32\* (不及格會以\*註記)

lab3： 32\*

mid-term： 90

final exam： 93

total grade： 81

使用者輸入： E

螢幕顯示：輸入ID或 Q (結束使用)？ Q

螢幕顯示：結束了

4.開始至輸入指令如上

使用者輸入： R

螢幕顯示：李威廷排名第63

使用者輸入： E

螢幕顯示：輸入ID或 Q (結束使用)？ Q

螢幕顯示：結束了

5.開始至輸入指令如上

使用者輸入： A

螢幕顯示：

班平均： lab1： 90

lab2： 88

lab3： 89

mid-term： 90

final exam： 90

total grade： 90

使用者輸入： E

螢幕顯示：輸入ID或 Q (結束使用)？ Q

螢幕顯示：結束了

6.開始至輸入指令如上

使用者輸入： W

螢幕顯示：

舊配分

lab1 10%

lab2 10%

lab3 10%

mid-term 30%

final exam 40%

輸入新配分(提醒：每項需介於0~100之間，五項總和必須等於100)

lab1 15

lab2 15

lab3 15

mid-term 25

final exam 30

請確認新配分

lab1 15%

lab2 15%

lab3 15%

mid-term 25%

final exam 30%

以上正確嗎？Y(Yes)或 N(No)

使用者輸入： Y

螢幕顯示： 加權已更新!

使用者輸入： E

螢幕顯示：輸入ID或 Q (結束使用)？ Q

螢幕顯示：結束了

* 異常使用情形

7.螢幕顯示：輸入ID或 Q (結束使用)？ 123456789

螢幕顯示：ID錯了!

8.螢幕顯示：輸入ID或 Q (結束使用)？ 962001051

輸入指令如上

使用者輸入： H

螢幕顯示：指令錯了!

9.開始使用指令如6.，至輸入新配分

輸入新配分(提醒：每項需介於0~100之間，五項總和必須等於100)

lab1 20

lab2 20

lab3 20

mid-term -10

final exam 50

請確認新配分

lab1 20%

lab2 20%

lab3 20%

mid-term -10%

final exam 50%

以上正確嗎？Y(Yes)或 N(No)

使用者輸入： Y

螢幕顯示：調整失敗！第4項超出範圍，請重新輸入

更新加權指令已取消!

10.開始使用指令如6.，至輸入新配分

輸入新配分(提醒：每項需介於0~100之間，五項總和必須等於100)

lab1 20

lab2 20

lab3 20

mid-term 20

final exam 25

請確認新配分

lab1 20%

lab2 20%

lab3 20%

mid-term 20%

final exam 25%

以上正確嗎？Y(Yes)或 N(No)

使用者輸入： Y

螢幕顯示：調整失敗！五項總和不等於100，請重新輸入

更新加權指令已取消!

**<Architecture design-Part1> (Class Interface)**

* class Main

public method:

main()

* class UI (user interface)

主要method:

UI() // 建構子，用來建構aGradeSystem

checkID(ID) // call containID

promptCommand(ID) // 接收指令輸入

promptID() // 接收ID輸入

showFinishMsg() // 顯示結束訊息

showWelcomeMsg(ID) // 顯示歡迎訊息

resetWeights() // 重置weights

* class GradeSystems:

member: aList(學生成績), weights(加權配分)

主要method:

GradeSystems() // 建構子，用來建構aList

containsID(ID) // 看aList內是否含有ID

showGrade(ID) // 顯示使用者各項成績

showRank(ID) // 顯示使用者排名

showAverage() // 顯示班上各項成績平均

updateWeights(ID) // 更新加權配分

* class Grades:

member: ID, name, lab1, lab2, lab3, midTerm, finalExam, totalGrade

主要method:

Grades() // 建構子

calculateTotalGrade(weights) //計算總成績

**<Architecture design-Part2> (Detailed design)**

本Grade system能讓使用者(學生)，透過輸入自己的ID，取得他的總成績total grade及排名rank。Total grade基於加權配分(weights)來計算，而 使用者也可以修改weight。

PS1: Rank表示此ID的total grade在全班學生的分數排序。

PS2: 英文指令輸入不分大小寫。

Input file: gradeinput.txt，位於專案根目錄，去除BOM編碼。

例如： 962001044 凌宗廷 87 86 98 88 87

985002016 王淳 97 96 89 85 80

PS1: 注意data field names如下:

ID name lab1 lab2 lab3 midTerm finalExam

PS2: Name為兩個字者，姓名中間一律不留空白。

* class Main(程式初始進入點)

public class Main extends Object {

public static void main (String args[])

try { call UI() -> 建構 aUI } end try

end class Main

* class UI (user interface)

member: aGradeSystem (type: GradeSystems)

-------------------------------

UI() constructor throws IOExceptions

try {

1.call GradeSystems() -> 建構 aGradeSystem

2.Loop until cmd equal to“Q”(Quit)

1. promptID() -> get user’s ID

2. checkID(ID) -> 看ID是否在aGradeSystem內

if return true -> showWelcomeMsg(ID)

-> promptCommand(ID) 無限迴圈取得指令

3. resetWeights() -> 使用者有可能變動weights，而單一使用 者變更加權不會影響到其他人，因此需恢復原值

} end try

catch (NoSuchIDExceptions e) { print "ID錯了！" }

catch (NumberFormatException e) { print "輸入格式內容錯誤！" }

catch (NoSuchCommandExceptions e) { print "指令錯了！" }

finally { showFinishMsg() }

end UI

-------------------------------

Boolean checkID(ID) throws NoSuchIDExceptions

參數: ID (type: int,使用者的ID)

1. 透過aGradeSystem.containsID(ID)，看ID是否含在 aGradeSystem

2. if not, throw NoSuchIDExceptions

3. 回傳 true

-------------------------------

promptCommand(ID) throws NoSuchCommandExceptions

參數： ID (type: int,使用者的ID)

1. 無限迴圈：要求使用者輸入以下指令: ”G”,”R”,”A”,”W”,”E”

若不在這些指令範圍 -> throw NoSuchCommandException

2. if inputCommand is E (Exit) -> break

else: G -> aGradeSystem.showGrade(ID), R -> showRank(ID),

A -> showAverage(), W -> updateWeights()

end if

-------------------------------

promptID() print“輸入 ID 或 Q(結束使用)？” 讀取使用者輸入

showFinishMsg() print “結束了”

showWelcomeMsg() print 歡迎訊息 + 各項指令

resetWeights() call aGradeSystem.setWeights(原本weights)

NoSuchIDExceptions extends Exception { }

NoSuchCommandExceptions extends Exception { }

end class UI

* class GradeSystems

member:

float[5] weights：加權分配(預設為0.1, 0.1, 0.1, 0.3, 0.4)，可藉由updateWeights method調整(單項介於0~100，五項總和須為1)。

HashMap<Integer, Grades> aList：利用Java HashMap 資料型態來創建aList，Key為Integer(user’s ID)，對應的Value為Grades(user’s grades)。優點：取得特定使用者之成績時，複雜度為O(1)。

-------------------------------

GradeSystems() constructor throws NumberFormatException, IOException

1. 讀檔(use BufferedReader) input file

2. 用Java HashMap建構aList (Key: ID, Value: Grades)

3. while(read line != null)

1. call Grades() 建構aGrade

2. 將read line藉著空白分開，把各欄位存入aGrade

3. aList.put(aGrade) 存入aList中

end while

end GradeSystems

-------------------------------

booleann containsID(ID) 回傳aList.containKey(ID)

-------------------------------

showGrade(ID)

參數：ID

1. show此ID的各項成績與total grade
2. call private String passOrNot(grade) -> 如果該項成績不及格，標記\*

-------------------------------

showRank(ID)

參數：ID

1. aGrade = aList.get(ID) 取得該ID使用者的Grade

2. 令變數rank 為 1

3. Loop aGrade in aList

if aGrade.totalGrade > others’totalGrade -> rank加1(退1名)

end Loop

4. 回傳 rank

--------------------------------

showAverage()

1. Loop aGrade in aList 加總全班學生的各項成績
2. call method calculateAndShowAverage(grades)

private calculateAndShowAverage(grades): 四捨五入後顯示全班平均

---------------------------------

updateWeights() (變更配分只會修改單一使用者的總成績，其餘人不受影響)

1. showOldWeights() 顯示舊成績

2. float[5] newWeights = getNewWeights() 取得新配分

private float[] getNewWeights(): 讀取使用者輸入新配分後回傳

1. call checkNewWeights(newWeights) 再次確認是否修改配分

private boolean checkNewWeights(newWeights):

1. show 剛輸入的新配分，詢問”以上正確嗎？Y(Yes)或 N(No)”
2. if input =“N”或是其他錯誤指令 -> return false

else -> call inRangeOrNot(newWeights) 檢查是否符合標準

private boolean inRangeOrNot(newWeights):

1. check if newWeights[0]~[4] are all in range(0~100)
2. check if sum of newWeights[0]~[4] is 100

if 滿足以上兩者 -> return true, else -> return false

end inRangeOrNot

end checkNewWeights

if checkNewWeights return true

1. call setWeights(newWeights) 將weights更新
2. call aList.get(ID).calculateTotalGrade(weights) 重新計算該ID的總成績
3. show “加權已更新!”

else show “更新加權指令已取消!”

end updateWeights

end class GradeSystems

* class Grades

member: ID, name, lab1, lab2, lab3, midTerm, finalExam, totalGrade

-------------------------------

public Grades()//建構子

參數：ID, name, lab1, lab2, lab3, midterm, finalExam (不包含totalGrade)

1. 從參數取得五項成績
2. call calculateTotalGrade(weights) 取得totalGrade

-------------------------------

calculateTotalGrade(weights)

參數：weights 成績加權分配

藉由weights算出totalGrade(四捨五入)

end class Grades

**<Picture – Pair Programing Process>**

