1. **Architec****tural Design (class interface) 架構設計範例**

* **class Main**

main ( )

* **class UI (user interface)**

checkID(ID)

promptCommand()

promptID()

showFinishMsg()

showWelcomeMsg()

UI() 建構子 建構 aGradeSystem

**class GradeSystems**儲存 a list of student grades, a list of weights a list of string for display

containsID(ID) // 看aGradeSystem有否含此ID

GradeSystems () //建構子

showGrade(ID) //顯示成績

showRank(ID) //顯示排名

updateWeights () //更新配分

showAverages() //顯示全班平均

**class Grades** 儲存 ID, name, gradeList, totalGrade

gradeList的儲存項目分別為lab1, lab2, lab3, midTerm, finalExam

* calculateTotalGrade(weights) //計算加權平均

Grades () { } //建構子

1. **De****tailed Design (design sketch and pseudo-code) 細部設計範例**/\*\* #################################################################

本Grade system 讓使用者(學生)取得他的總成績total grade 及排名 rank.

Total grade 基於配分weights 來算 而 weights可以update.

Rank 表示此 total grade 在全班學生的分數排序

Input file名字必須為inputdata.txt

Input file: 全班學生的分數 例如

962001044 凌宗廷 87 86 98 88 87

962001051 李威廷 81 32 50 90 93

注意 data field names 如下:

ID name lab1 lab2 lab3 midTerm finalExam

#################################################################### \*/

public class Main extends Object {

public static void main (String args[])

try { call UI() 建構 aUI } end try

catch (NoSuchIDExceptions e1) {print msg1} //ex ID錯了!

catch (NoSuchCommandExceptions e2) {print msg2}//ex指令錯了!

end class Main

public class NoSuchIDExceptions extends Exception { }

public class NoSuchCommandExceptions extends Exception { }

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

class UI (user interface)

checkID(ID)

promptCommand()

promptID()

showFinishMsg()

showWelcomeMsg()

UI() 建構子 建構 aGradeSystem

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

public class UI

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

UI() 建構子 throws NoSuchIDExceptions, NoSuchCommandExceptions

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

try

1.call GradeSystems() to建構 aGradeSystem

2.loop1 until Q (Quit)

1. promptID() to get user ID 輸入ID或 Q (結束使用)？
2. checkID (ID) 看 ID 是否在 aGradeSystem內
3. showWelcomeMsg (ID) ex. Welcome李威廷
4. loop2 until E (Exit)

promptCommand() to prompt for inputCommand

end loop2

end loop1

3.showFinishMsg()結束了

end try

finally {}

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

checkID (ID) throws NoSuchIDExceptions return Boolean

parameter: ID a user ID ex: 123456789

time: O(n) n is aGradeSystem 內全班人數

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

1. 要aGradeSystem 做containsID(ID) 看 ID 是否含在 aGradeSystem內
2. if not, throw an object of NoSuchIDExceptions
3. 回傳 true

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

promptCommand () throws NoSuchCommandExceptions

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

1. prompt user for inputCommand

2. if inputCommand is not G (Grade),R (Rank), W (Weights), or E (Exit),

throws an object of NoSuchCommandException

3. if inputCommand is E (Exit) then break

else: G aGradeSystem.showGrade(ID), R showRank(ID), W updateWeights() end if

promptID()

showFinishMsg()

showWelcomeMsg()

end class UI

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

class GradeSystems儲存 a list of student grades, a list of weights, a list of string for display

containsID(ID) //看aGradeSystem有否含此ID

GradeSystems () //建構子

showGrade(ID) //顯示成績

showRank(ID) //顯示排名

updateWeights () //更新配分

showAverages() //顯示全班平均

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

public class GradeSystems

float[5] weights; (初始值 lab1 0.1, lab2 0.1, lab3 0.1, midTerm 0.3, finalExam 0.4)(注意weights 之和須等於1.0)

String[] gradeItemList; 用於在螢幕上顯示成績所需的文字，為了程式簡潔，列出各項成績的部分皆以for loop來完成，搭配gradeItemList的固定文字和aGrade中gradeList的資料結構，並用if判斷式加註”\*”在小於60的成績之後。Total grade不在gradeList中需另外顯示

用Java LinkedList class 的object叫**aList** (它是 doubly-linked list) 來儲存 **aGrade** objects.

見下面的設計草圖 (design sketch):

|  |  |  |
| --- | --- | --- |
|  | null header |  |

|  |  |  |
| --- | --- | --- |
|  | 962001044凌宗廷 87 86 98 88 87 88 |  |

|  |  |  |
| --- | --- | --- |
|  | 962001051李威廷  81 32 50 90 93 91 |  |

public GradeSystems () //建構子

1. 開檔 input file

2. 用**Java** **LinkedList**建構an empty list of grades叫 aList

3. read line

4. while not endOfFile

1.call Grades() 建構aGrade

2.用 **Java BufferedReader** 來 read line 並用split把各欄位存入aGrade

3. aGrade.calculateTotalGrade(weights) 回傳aTotalGrade把它存入aGrade

4. 把 aGrade 存入 aList

5. if endOfLine then read line end if

end while

public showGrade(ID) { show 這 ID 的 grade }

1. loop aGrade in aList
2. if ID match, show the grades

public showRank(ID)

1. 令rank 為 0
2. 找到ID的aGrade

3. 令rank 為 1

4. loop aGrade in aList if aTotalGrade > theTotalGrade then rank加1(退1名) end loop

4. 回傳 rank

public updateWeights ()

1. show old weights

2. get new weights

3. show new weights

4. ask the user if the update is correct

5. set weights and show success messenger if correct

6. loop aGrade in aList to calculateTotalGrade(weights) end loop

public containsID(ID) loop through aList看 ID 是否含在 aGradeSystem內

end class GradeSystems

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

class Grades 儲存 ID, name, gradeList, totalGrade

gradeList的儲存項目分別為lab1, lab2, lab3, midTerm, finalExam

calculateTotalGrade(weights)

Grades () { } //建構子

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

public class Grades

String name, ID; int gtradeList[5];

(注意 input file內無 totalGrade )

public Grades () { } //建構子

產生gradeList的instance

參數依序存入ID,name,gradeList中

public calculateTotalGrade(weights) {totalGrade須四捨五入}

1. gradeList的每個項目分別乘上weight後相加
2. 結果取四捨五入存到totalGrade
3. 回傳totalGrade

end class Grades