Architectural Design (class interface) 架構設計

* Class Main

main()

* Class Grades 儲存 ID, name, lab1, lab2, lab3, midTerm, finalExam, and totalGrade

calculateTotalGrade(weights)

Grades () { } //建構子

* Class GradeSystems

GradeSystems () //建構子

readFile(String fileName)

showGrade (int ID)

showRank (int ID)

updateWeights()

containsID(int ID)

getName(int ID)

showOldWeights()

getNewWeights()

enterWeight()

showAverage()

* Class UI

UI() //建構子

checkID(int ID)

promptCommand()

printCommand()

promptID()

showFinishMsg()

showWelcomeMsg()

Detailed Design (design sketch and pseudo-code) 細部設計

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

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: 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[])

call UI()建構UI

end class Main

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

class UI (user interface)

UI() 建構子 建構 aGradeSystem

checkID(int ID)

promptCommand()

printCommand()

promptID()

showFinishMsg()

showWelcomeMsg()

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

public class UI

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

UI() 建構子

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

1.call GradeSystems() to建構 aGradeSystem

2.loop

1.若尚未輸入ID call prompID()

2.若以輸入ID的話checkID()並showWelcomeMsg()、promptCommand()

3.若this.fin == true 的話結束

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

checkID (ID) 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 yes, return true
3. if not, show error msg, set curUser = 0

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

promptCommand ()

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

1. call printCommand()
2. if command == G call this.aGradeSystem.showGrade
3. if command == R call this.aGradeSystem.showRank
4. if command == A call this.aGradeSystem.showAverage
5. if command == W call this.aGradeSystem.getNewWeights
6. if command == E call this.aGradeSystem.showFinishMsg
7. else show error msg.

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

printCommand()

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

1. print msg
2. read command

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

promptID()

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

1. read in command
2. if command = q, set this.fin = true, end system
3. else set this.curUser = command

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

showFinishMsg()

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

1. show leaving msg.

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

showWelcomeMsg()

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

1. show welcome msg.

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

class GradeSystems儲存 a list of student grades.

containsID(ID) //看aGradeSystem有否含此ID

GradeSystems () //建構子

readFile(String fileName)

showGrade (int ID)

showRank (int ID)

updateWeights()

containsID(int ID)

getName(int ID)

showOldWeights()

getNewWeights()

enterWeight()

showAverage()

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

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)

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. this.readFile()讀檔

2. loop aList計算totalGrade

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

public showRank(ID)

1. 取得這 ID 的 theTotalGrade

2. 令rank 為 1

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

4. 印出Rank

private updateWeights ()

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

public containsID(ID) 看 ID 是否含在 aGradeSystem內

private readFile(String fileName)

1. 開檔
2. While not EOF
3. readline
4. split data
5. call aGrade to new a Grades
6. add to aList

end of while

public getName(int ID) 取得ID的name

private showOldWeights() 顯示目前的weights

public getNewWeights()

1. call this.showOldWeights()
2. call this.enterWeights()
3. call this.showOldWeights()
4. call this.updateWeights()

private enterWeight() 讀進新的五個成績的Weights

public showAverage() 計算並顯示目前totalGrade的平均

end class GradeSystems

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

class Grades 儲存 ID, name, lab1, lab2, lab3, midTerm, finalExam, and totalGrade

calculateTotalGrade(weights)

Grades () { } //建構子

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

public class Grades

String name,; int ID, lab1, lab2, lab3, midTerm, finalExam, totalGrade;

(注意 input file內無 totalGrade )

public Grades () { } //建構子

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

end class Grades