

姓名：邱翊碩

系級：航太系 117

學號：F44134057

作業：作業三

主程式檔案名稱：hw3.java

其他檔案：

Data.java

Student.java

Updater.java

Text.java

Score.txt

## 一. 程式設計需求描述：

### A. 基本功能:

1. 新增、刪除與查詢學生資料，無需使用 Excel 或手動整理.txt
2. 即時計算各項作業與期末專案平均成績
3. 即時儲存資料到文字檔
4. 文字介面操作

### B. 設計方面

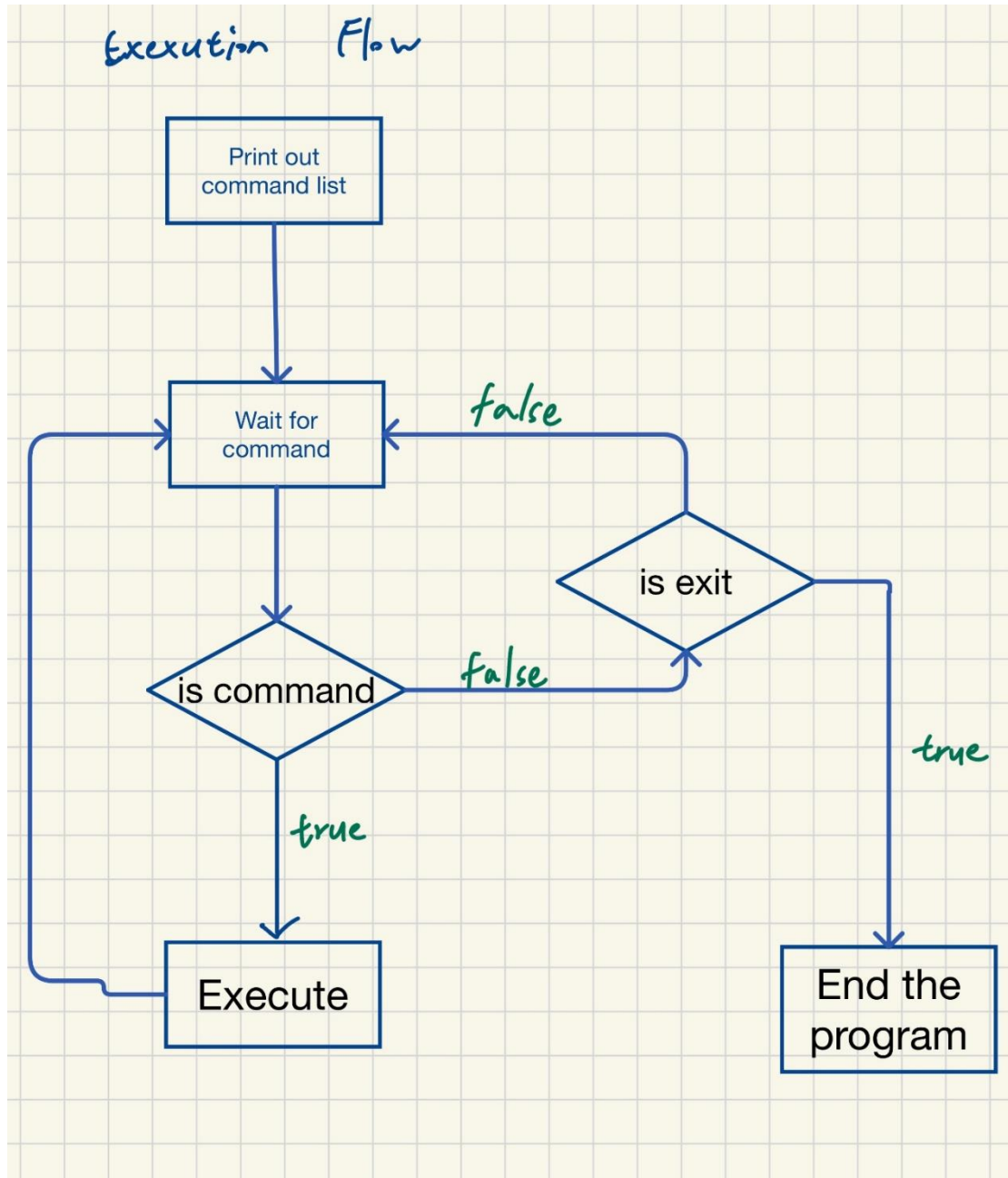
1. 指令有防呆（輸入錯誤時顯示訊息）
2. 使用 try-catch 進行處理 exception，避免 Crash
3. 每次修改資料後都自動更新 Score.txt
4. 使用物件之間的關係進行資料的處理

### C. 可能吸引人的點

1. 無需使用 Excel
2. 動態更新 Score.txt
3. 輸入辨識嚴謹，同時不容易 Crash
4. 極簡風且友善的介面
5. 輸入錯誤時會顯示相對應的錯誤訊息

## 二. 程式流程：

程式簡易流程圖.



### a. 文字解說

第一步. 先列出歡迎訊息與指令列

第二步. 等待指令輸入

第三步. 判斷指令是否符合規定格式，是的話則執行相對應的指令，

如果為否則看是否輸入 `exit`，是的話退出程式

## **b. Execute 指令部分的說明**

指令 1. `add [name] [score1,score2,score3,final1,final2]`

新增一位學生與其成績資料(共五項：三個作業 + 兩個期末專案)

ex. `add Alice 100,0,45,98,48`

指令 2. `delete [name]`

刪除指定學生的資料

ex. `delete Alice`

指令 3. `show individual score [name]`

顯示指定學生的所有作業與期末專案成績

ex. `show individual score Alice`

指令 4. `show individual average [name]`

顯示指定學生的平均分數。

ex. `show individual average Alice`

指令 5. `show Homework[number] average`

顯示所有學生在該作業的平均分數

ex. `show Homework1 average`

指令 6. `show Final Project[number] average`

顯示所有學生在該期末專案的平均分數

ex. `show Final Project1 average`

指令 7. `show total`

列出所有學生的完整成績清單

ex. `show total`

指令 8. **clear** (額外新增功能)

清空所有學生的資料

ex. **clear**

指令 9. **Help** (額外新增功能)

顯示所有指令的使用說明

ex. **help**

指令 10. **Exit**

離開程式

ex. **exit**

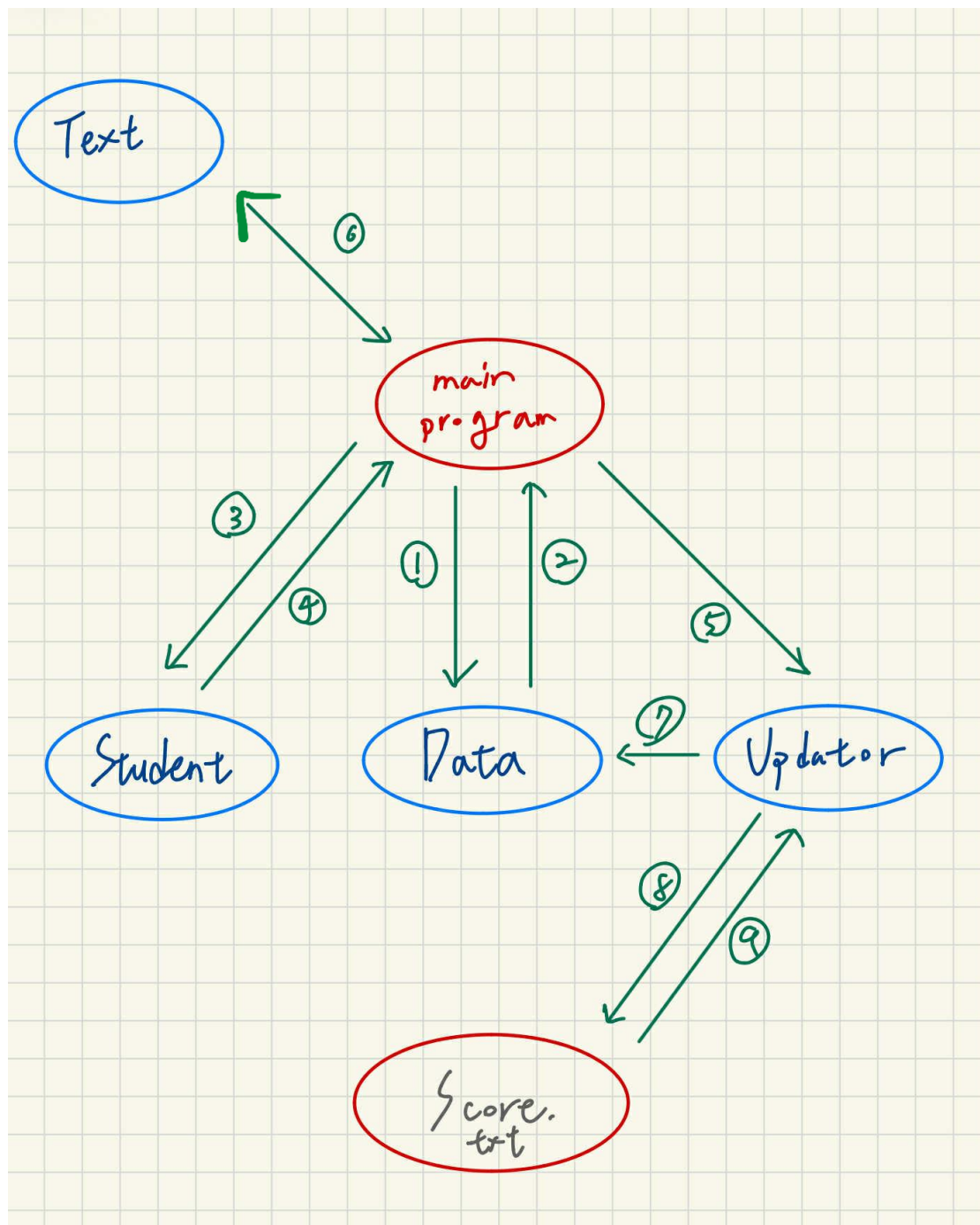
指令 11. **sorting** (額外新增功能)

依照每位學生的平均值從上到下由高到低做排序並印出結果(注意  
**Score.txt** 的資料排序會跟著改變)

ex. **sorting**

### 三. 功能/邏輯說明：

整個程式之間的簡易關係圖。



#### a. 程式之間的簡易關係圖數字編號說明

編號 1. Main program 將收到的文字串處理後整理成資料，然後給 Data 的 object 辨識。或者用處理後的資料生成 Student object 給 Data 的 object 做資料儲存

編號 2. 收到從 Main program 傳來的資料後執行相對應的指令，之後再回到主程式

編號 3. 新增 Student object 並用 Constructor 設定初始值。或者使用 Class Student 的 method

編號 4. 將 Student 裡頭的 method 處理後的值回傳給 main program，或者將設好值的 Student object 回傳

編號 5. 將 Data 和 PrintWriter 的 Object 給 Updator 後，Updator 對 Score.txt 做更新的動作

編號 6. 由主程式呼叫 Text 的 method 並回傳字串

編號 7. 將初始的 Score.txt 的資料處理完後輸給 Data object 做儲存

編號 8. 更新 Score.txt

編號 9. 讀取 Score.txt 的初始資料

#### b. 程式中的字串辨識邏輯

1. 讀取整行輸入字串並試著找出字串第一個空格位置
  2. 由第一點得到的空格位置得到第一個單字並命名 keyword
  3. 用 keyword 區分每一個指令，並依照相對應的指令條件做出相對應比對
  4. 比對的條件包括 keyword 後面的字串本身，逗號、空格的數量等等，依照相對應的條件用 if else 進入相對應的指令
  5. 指令失敗時程式會盡可能依照辨識出的結果印出相對應的錯誤原因
- 註. 沒有空格時 keyword 即為 input 本身

#### c. 程式的資料處理邏輯

1. 創建 `token[]` 字串陣列，並以空格為界取得每一個單字，期間字串會做過 `.trim()` 處理，所以空格可為多個空格。(後來得知 `split("\\s")` 的方法處理字串並儲存，在程式後面中某些部分有使用)
2. 其中 `add` 的指令包含逗號，所以另創一個 `Scanner object` 並使用 `.useDelimiter` 後再使用 `.next()` 來分割字串並儲存在 `token[]`，之後再用最初創建的整數陣列 `score[]` 將 `token` 中的字串數字變成整數型態
3. 處理好的資料都會在 `Data object`，`list`，之中做儲存，`add` 出的學生資料都會編成 `Student` 的 `object` 並塞在 `list` 裡面，`delete` 時則會把 `list` 的指令資料刪除，`show` 則會整理 `list` 的資料並印出

#### **d. Score.txt 的更新邏輯**

1. `Score.txt` 的更新將由 `Updater` 這個 `file` 做處理，只要主程式呼叫 `update(Data, PrintWriter)` 後就會直接對 `Score.txt` 做更新，`Updater` 也負責處理程式開始時從 `Score.txt` 讀取資料直接存給 `list` 的工作

#### **e. exception 的邏輯**

1. 本程式盡可能包含了所有會遇到的 `exception`，像是找不到檔案、陣列超出索引值、`method` 中參數的資料型態不符等等
2. 程式本身會盡可能依照 `exception` 的原因列出相對應的錯誤訊息

#### **f. 針對各項指令的邏輯**

1. `add`: 以 `Scanner` 讀入整行文字後，用空白與逗號分隔出關鍵字與數字資料，搭配剛剛在 c. 的第 2 點提過的邏輯取得字串 `token`，辨別該資料是否符合規定，如果是便將資料存在 `Student object` 後再儲存在 `list`，最後呼叫 `Updater` 做 `Score.txt` 的更新
2. `delete`: 透過 `split(" ")` 分割出得到 `token`，不用 `//s` 是因為剛好空格導致 `crash` 的狀況會被排除，如果輸入的指令沒問題則會呼叫 `list` 刪除學生，最後呼叫 `Updater` 做 `Score.txt` 的更新
3. `show individual score`: 直接抓取相對應的學生名後記住 `index`，再將相對應 `index` 的學生印出每項成績
4. `show individual average`: 直接抓取相對應的學生名後記住 `index`，再

將相對應 index 的學生成績計算平均後印出

5. show Homework[number] average: 因為要抓取字串中的數字，所以過程假定 token[1]會是 Homework，長度為 8，所以另宣告字串變數存入 token[1] index(8)之後的資料，並使用 Integer 的 method .parseInt 轉換成整數，如果有 Exception 會另外處理。之後將得到的資料傳給 list 做處理後印出資料

6. show Final Project[number] average: 基本和上頭的 Homework average 指令大同小異，只要把 index 的部分稍作修改，並抓取 token[2]做處理就並傳入資料，list 就會把資料印出

7.show total: 這一條指令不需要什麼複雜的邏輯，直接比對 input 後直接列出在 list 中所有學生的資訊即可

8.exit: 同 7.

9.help: 同 7.

10.clear: 同 7.，不過結束這條指令後須更新 Score.txt

11.sorting: 這條指令會以 list 的每位學生的平均分數做資料的整理，在排序的部分使用的是最簡單的 bubble sort，並印出

#### **g. 四捨五入的邏輯**

1.在 Data 的 file 裡面有 print\_rounded\_score 的 method 可以讓相對應的每個平均值印出對應的精度，底層邏輯是運用了 printf("%.f")以及 %1 是否為 0 作為判斷標準並選擇將對應的精度印出



#### 四. 使用說明：

##### a. 程式用途簡介：

本系統為一個簡易的學生成績管理系統，可進行學生資料的新增、刪除、查詢與平均計算，並自動將變更同步更新至 **Score.txt**。

##### b. 使用前確保所有檔案 (HW3.java, Data.java, Student.java, Updator.java, Text.java) 都放在同一個資料夾底下

##### c. 該資料夾中存在一個名為 **Score.txt** 的檔案(用來儲存成績資料)。

##### d. 使用順序，在終端機或 **cmd** 中依序輸入以下指令進行編譯與執行

1. `javac HW3.java`(編譯)

2. `java HW3 Score.txt`(執行)

3. 程式開始執行(列出程式指令清單，在 `readme.txt` 下方也附著指令的使用方法與效果)

4. 輸入指令

5. 執行指令

##### e. 使用上的須知

1. 所有操作皆以指令的形式輸入，請依照畫面提示的格式輸入

2. 程式會根據輸入內容進行分析，若格式錯誤、空白或逗號位置不正確，系統會提示錯誤

3. 程式啟動時會自動讀取 **Score.txt** 的資料，並於每次修改後自動更新該檔案，還請使用者使用該程式修改 **Score.txt**，手動修改是可以做

到的，但很不推薦

4. 指令的空格與大小寫、逗號會影響指令的判讀，還請多加注意
5. 若 `Score.txt` 為空或尚未建立資料，某些指令會顯示「student list is empty」
6. 若要清空所有學生資料，可使用 `clear` 指令重置資料表

**f. 以下為程式指令**

**1. add [name] [score1,score2,score3,final1,final2]**

新增一位學生與其成績資料(共五項：三個作業 + 兩個期末專案)

ex. add Alice 100,0,45,98,48

**2. delete [name]**

刪除指定學生的資料

ex. delete Alice

**3. show individual score [name]**

顯示指定學生的所有作業與期末專案成績

ex. show individual score Alice

**4. show individual average [name]**

顯示指定學生的平均分數

ex. show individual average Alice

**5. show Homework[number] average**

顯示所有學生在該作業的平均分數

ex. show Homework1 average

**6. show Final Project[number] average**

顯示所有學生在該期末專案的平均分數

ex. show Final Project1 average

## 7. show total

列出所有學生的完整成績清單

ex. show total

## 8. clear

清空所有學生的資料

ex. clear

## 9. help

顯示所有指令的使用說明

ex. help

## 10. exit

離開程式

ex. exit

## 11.sorting

依照每位學生的平均值從上到下由高到低做排序並印出結果(注意 Score.txt 的資料排序會跟著改變)

ex. sorting

## g. 其他補充說明

1. 程式在執行期間會不斷等待指令輸入，直到使用者輸入 `exit` 才會結束
2. 若格式正確，會顯示「`command success`」；若格式錯誤，則會顯示相對應錯誤說明
3. 程式設有例外處理，避免因輸入錯誤導致崩潰

註 1. 若出現 `Score.txt didn't be found` 的訊息，請先建立該檔案，或確認檔案路徑與名稱是否正確。

若程式成功啟動，將會印出指令說明與歡迎畫面。

註 2. 如欲在 **e** 退出，請直接輸入 **exit**，否則程式會在 **d** 和 **e** 持續輪迴進行。

註 3. 非常不建議自行更改 **Score.txt** 檔案，非常有可能會出現不預期的結果，還請盡量使用本程式進行修改，如果真的需要自行更改請依照以下格式：

**[Number] [number] [number] [number] [number]**

ex. ➡ Tom 100 54 68 98 78

註 4. 程式本身是用英文表示，如果對於某項功能有疑問可以再參考本文件