

姓名：邱翊碩

系級：航太系 117

學號：F44134057

## 1. 程式設計需求描述：

本程式為電影訂票與查詢系統，提供使用者透過指令完成電影資訊查詢、依類型或時長查詢、訂票、取消訂單以及查看個人訂單等功能。程式使用英文提示說明，並對使用者輸入進行格式檢查，以確保程式運行正確，避免非預期錯誤發生。

## 2. 程式流程：

### 第一步：歡迎與說明

- 呼叫 `user.welcome()` 顯示歡迎訊息與操作說明。
- 宣告必要變數

### 第二步：整理輸入的字串

- 程式進入 `while(condition)` 迴圈，從接下來開始整個操作流程會在此迴圈內重複，直到使用者輸入 `exit` 退出。
- 使用 `Scanner` 接收輸入，使用 `trim()` 移除首尾空格。
- 計算空格數 `count`，並將字串拆分為多個 `token`。

### 第三步：檢查輸入格式

- 若空格數 `count > 4` 或 `< 0`，程式會提示格式錯誤，要求使用者重新輸入。

### 第四步：解析指令與執行

- 利用 `token`，`count` 與 `String.match()` 來辨別指令
- **exit**：退出程式，顯示感謝訊息。
- **help**：顯示使用說明。(額外寫的，怕使用者用到一半忘記指令，然後我

自己也會忘)

- **clear**：清除所有歷史訂單。(也是額外寫的，方便使用者清空資料)
- **add [movie\_number] [ticket\_number] [user\_id]**：新增訂單，呼叫 `guest.add()`。
- **show movie**：顯示所有電影。
- **show movie [number]**：依電影播放時間查詢電影，呼叫 `movie.movie_time_search()`。
- **show movie [type]**：依電影類型查詢，呼叫 `movie.movie_type_search()`。
- **show order [user\_id]**：顯示使用者的歷史訂單，呼叫 `guest.show_order_for_user()`。
- **cancel order [user\_id] [order\_id]**：取消指定訂單，呼叫 `guest.cancel_order()`。

### 第五步：錯誤處理

- 若指令不符合格式或順序錯誤，提示使用者檢查拼寫或順序，返回第二步重新輸入。

## 3. 功能/邏輯說明（每一步使用語法）：

### 第一步：程式啟動

- 宣告變數初始化程式。

在主程式會宣告多種變數，其中有些變數會作為 `parameters` 給 `user` 的 `method` 後進行資料的儲存

`user.java` 會宣告 `movie_id[]`, `user_id[]`, `ticket_number[]`, `delete[]`陣列來儲存使用者後續儲存的資料，其中還有 `int order_number = 0` 來記錄訂單編號。

- 呼叫 `user.welcome()` 顯示操作說明。

`user.welcome()`位於檔案 `user.java` 中，主要內容為字串，文本為介紹程式本身以及指令說明。

- 電影的資訊已經預先儲存到 `movie.java` 裡頭。

## 第二步：輸入與拆分指令

- 使用迴圈、字串 method：`substring()`和 `trim()`處理字串，以空格為分割儲存到 `token[]`陣列，`token` 的每個元素都是單獨一串單字或數字。
- 將空格數量儲存到變數 `count`，方便後續第四步 `if` 的檢測。(p.s.現在想想用 `token.length()`就好了...)

## 第三步：先行檢查輸入格式

- 使用 `if` 判斷空格數是否超過規定範圍，如果是會用 `continue` 回到迴圈第二步，之後在第四步如果沒有相對應的指令也會 `continue` 回到第二步。

## 第四步：指令解析與執行

- 使用多重 `if / else if` 判斷 `token` 內容對應的指令。
- 使用 `Integer.parseInt()`與 `Double.parseDouble()`將 `String` 轉換成 `int` 並儲存到相對應的變數。
- 對應的呼叫、執行功能與基礎邏輯：

- `exit`：退出程式

在第二步程式進入一個 `true` 迴圈，所以接收到 `exit` 時直接使用 `break`；結束程式。

- `user.help`：重新呼叫指令

直接將 `welcome` 的一部份拿來用然後印出。

- `guest.add()`：新增訂單

使用者會輸入三個資訊，三個資訊皆會儲存到 `user.java` 中的 `array` 儲存，如 `movie_id[order_number]`，並讓訂單編號+1，之後會在下一個訂單儲存並以此類推。

- `movie.show_movie()`：顯示所有電影

基本上這個 `method` 就是把預設的電影資料用 `for` 迴圈從 `movie.java` 全部 `print` 出來，因為電影當初也是按順序輸入給電腦的資訊，所以基本也不用做甚麼處理和特別排序。

- `movie.movie_type_search()`：依類型查詢

把使用者輸入的類型字串給 `method` 作為 `parameter`，然後用 `movie.java` 的電影資料與 `for` 迴圈來一一檢測，如果字串一樣就 `print` 出。

- `movie.movie_time_search()`：依時長查詢

同 `movie.movie_type_search()`，不過會先將數字字串先用 `Int.parseInt()` 處理成 `int`，然後在給 `method` 做檢測並印出。

- `guest.show_order_for_user()`：顯示訂單歷史

也是將使用者輸入的資料給 `method` 後用 `for` 迴圈檢測使用者名稱與編號。

- `guest.clear()`：清除歷史訂單

直接重新宣告在 `user.java` 的變數，同時將 `order_number` 重設為 0。

- `guest.cancel_order()`：取消訂單

用 `boolean delete[]` 儲存被取消的訂單，確認被 `cancel` 的訂單為變成 `true`，如果之後要用 `guest.show_order_for_user()` 顯示歷史訂單時會因為 `delete[]` 為 `true` 而不被印出。

## 第五步：錯誤處理

- 若不符合指令格式或參數錯誤，提示使用者重新輸入。

## 4. 使用說明：

開啟程式時會有說明，基本照著說明使用可以的話就可以正常使用，一開始會印出，請依照指令格式輸入。大致的使用就是依照印出的指令說明與順序輸入你的訊息，然後就可以下單訂票，如果有不知道的訊息像是電影的代號，可以使用 **show movie**。之後輸入時盡量避免冗餘字元或錯誤順序，避免程式崩潰或異常，然後輸入的指令皆為英文，包括：**exit**、**help**、**add**、**show**、**cancel** 等等。如果中途忘記 **command**，可以使用 **help** 指令可隨時查看操作說明，最後輸入 **exit** 即可退出程式並結束服務。

還有另一的重要的須知，本程式至多只能容納 999 筆資訊，如果今天資訊量超過了 999，程式將出現錯誤，如果使用者真的使用了 999 筆以上的訂單，請善用 **clear() function** 或者重啟程式來清空資料。

以下為指令一覽：

**a.add [電影代號][張數][user\_id]**，可以輸入使用者以及訂票資訊，若想知道電影編號的資訊請輸入 **show movie**

ex. add A 12 Eason

**b.show movie**，列印出現在上映的電影，資訊包含了名子、電影代號、時長以及種類

ex. show movie

...(電影資訊)

**c.show movie [類型]**，該功能會列出相同類型的所有電影以及其相關資訊的清單，如電影名稱、電影時長、電影類型、電影代號。

ex. show movie horror

...(電影資訊)

d.show movie [時長], 顯示出在該時長內的所有電影相關資訊的清單, 同 c.

ex. show movie 120

...(電影資訊)

e.show order [user\_id], 顯示出某位使用者的下單紀錄

ex. show eason

...(使用者資訊)

f.cancel order [user\_id] [order\_id], 刪除某位使用者的某筆訂購紀錄, 之後再使用指令 e.也將不會列出

ex. cancel order eason 1

g.clear, 可以用來清空所有訂單

ex. clear