

Java Programming Computer-Based Test

注意事項：

- 請先創建一個學號的目錄(例如：學號 00666666 的同學要創建一個 00666666 的目錄)。
- 底下再創建 Test1~Test5 五個子目錄，再將你五題程式題解答之 source code 與 binary code (依照套件目錄)分別放置於對應的目錄下。
- 類別都要設定 package，名稱為 ntou.cs.java2021.tx.你的英文名字 (x 為第幾題)。例如：ntou.cs.java2021.t1.ohtani, ntou.cs.java2021.t2.ohtani, ...
- 不可修改的類別或程式範例均可修改套件名稱以符合上述標準。
- 最後一題請創立"學號.txt"的文字檔(如 00666666.txt)，在文字檔內填寫心得與建議。
- 最後請整個壓縮成"學號.zip"，再上傳至 TronClass。
- 最高分為 100 (全部總分為 130)。

1. (25%) Craps 遊戲模擬器(陣列、存檔)

請基於提供之 Craps.java、Status.java 與 CrapsSimulationTest，寫出 CrapsSimulation.java，以可模擬執行 Craps 遊戲 1000 次與分析以下問題：

(a) Craps 遊戲輸的機率(the probability of losing)有多高？

(b) 平均一次 Craps 遊戲需要擲幾次骰子(average dice count)？

預期之執行結果如下：

The chances of losing are 509 / 1000 = 50.90%

The average game length is 3.31 rolls

此外，請在過程中請將每次遊戲結果紀錄於 records.txt，檔案內容應如以下格式(第幾次 輸贏 骰子次數)，總共應有 1000 筆資料。

1 LOST 2

2 LOST 7

3 WON 2

4 WON 5

5 WON 5

....

Hint:

- 可宣告 Craps 陣列，將輸贏結果與擲骰子次數紀錄於 Craps 物件中，之後再分析陣列中每個 Craps 物件的屬性資料。
- 可參考課本之 Fig. 15.3 CreateTextFile.java，運用 Formatter API 將資料寫入至檔案中。

2. (25%) 陣列排序(多型、委託)

(重構 20%)你目前已有既有程式 `SortingData.java` 和 `SortingDataTesting.java` (請先測試看看)，但目前的程式是用 `if ... else` 與魔術數字(Magic Number)來切換排序演算法，但當我們想新增新的排序演算法選項時，就得修改主程式的 `if` 判斷式，對於程式的可維護性是不理想的。

因此，我們導入了策略模式(Strategy Pattern)。策略模式是用策略介面來替換在物件運用的演算法，可以經由替換不同的策略使得物件擁有不同的行為。相關資料可參考此網頁：<https://skyyen999.gitbooks.io/-study-design-pattern-in-java/content/strategy.html>。

基於策略模式的概念，我們設計了 `Sorter` 這個抽象父類別，它的角色就是策略模式的策略介面，但為了提供共用的 `swap` 方法，所以將其建立為抽象父類別，而不是 `interface`。

請嘗試改寫 `SortingData.java`，將其改為 `SortableData`、`BubbleSorter` (`Sorter` 的子類別)、`SelectionSorter` (`Sorter` 的子類別)等三個類別，讓 `SorterTest` 可執行出以下結果：

```
Sorting..  
Bubble Sort is done!  
The array: [7, 23, 43, 68, 87, 92, 100]  
The last number is: 100
```

```
Sorting..  
Selection Sort is done!  
The array: [7, 23, 43, 68, 87, 92, 100]  
The last number is: 100
```

請留意，`SortableData` 類別內**不能有任何排序演算法的程式碼**，而最後的執行結果必須有實際執行 `Bubble Sort` 與 `Selection Sort`。

(轉接 5%)後續你拿到了一支程式 `QuickSort.java`，我們將把它納入變成另一個可替換的演算法，我們需要創立一個新的類別 `QuickSorter`，讓它是 `Sorter` 的子類別，且讓它擁有一個 `QuickSort` 型態的 `field` 來實現 `sort` 功能。接著，讓 `SorterTest2` 可執行出以下結果：

```
Sorting..  
Quick Sort is done!  
The array: [7, 23, 43, 68, 87, 92, 100]  
The last number is: 100
```

請留意，`QuickSort` 類別**不能做任何修改**，要由 `QuickSorter` 當轉接器，而最後的執行結果必須是實際執行 `Quick Sort` 之結果。

3. (25%：基本 10% + GUI 15%) 戰鬥小遊戲 (物件互動、GUI)

基本功能：請創立一個迷你的 Battle Game：

- Role 類別：即遊戲之角色，包含四個屬性-名字(name)、生命值(hp)、攻擊力(offense)、防禦力(defense)，目前大部分程式碼已完成，請繼續完成 attack(Role enemy)之實作，其功能為攻擊敵人(另一個 Role 物件)。攻擊損傷值(damage)之計算公式為自身之 offense 減去敵人之 defense，同時將敵人之目前之 hp 扣掉 damage。舉例而言：自己的 offense：135、敵人的 defense：64，damage 即為 71。
- Battle 類別：可建立兩個角色之對戰，目前須繼續完成 play()之實作，其功能為持續地依序讓角色 1 攻擊角色 2、角色 2 攻擊角色 1，直到某一個角色 hp 低於 0 為止，再顯示由哪一個角色獲勝之訊息。過程中請持續顯示任一個角色被攻擊後的 hp。
- BattleTest 類別：基本功能之程式進入點，此程式請勿修改。可能的遊戲結果如下(請盡量遵循此訊息內容規格)：

```
Captain America causes damage: 144
Iron Man's HP becomes 794
Iron Man causes damage: 212
Captain America's HP becomes 398
Captain America causes damage: 144
Iron Man's HP becomes 650
Iron Man causes damage: 212
Captain America's HP becomes 186
Captain America causes damage: 144
Iron Man's HP becomes 506
Iron Man causes damage: 212
Iron Man defeats Captain America!
```

GUI 功能：請將上述文字介面功能變成 GUI 形式。

- GameGUI.java 已提供相當篇幅的程式碼，請至少再完成//CODE Here 的部分的實作。
- 請修正 Battle4GUI.java，讓戰鬥相關功能可在 GUI 中被執行，初始畫面如下：



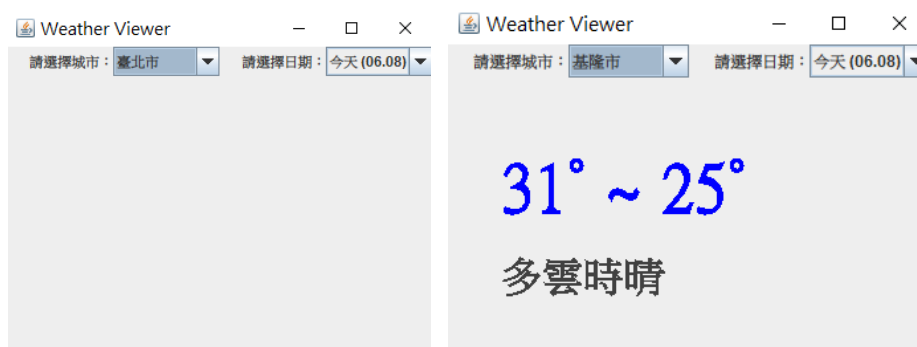
- 設定圖檔的方式：
 - 若在 console 編譯與執行，圖檔要放在套件目錄的最下層。
 - 若用 Eclipse 開發，請在 "package" 底下複製圖檔。
- 先前的文字介面是用 while 讓程式自動執行完畢，此版本則是每按一次 "Attack!!" 按鈕才會進行攻擊一次。執行後此攻擊按鈕要被 disable (setEnabled(false))，同時把另外一方的攻擊按鈕設定為 enable (setEnabled(true))，而生命值也請記得更新。預期按下第一次按鈕後畫面如下：



- 每次攻擊均需判斷戰鬥是否結束(生命值是否小於等於 0)，並以 JOptionPane 顯示戰鬥結果，並將戰敗方生命值設定為 0。



4. (25%) 天象資訊簡易圖形化介面程式(GUI、物件陣列、外部函式庫)



(初始畫面)

(取選城市或日期後的畫面)

- OneDayTemperature、TemperatureBundle、TemperatureFetcher 等三個類別是取得天氣資訊之核心邏輯程式，背後資料來源是擷取 PCHome 的氣象頁面，各位可先執行 TemperatureFetcher 進行測試，並了解 main 程式碼的用法。(請記得將 jsoup-1.13.1.jar 納入 classpath)
- 可先了解 Map 類別的用法，主要會用到 *get(Object key)*，功能是取得某個 key 值的物件，請記得抓回來的物件要轉型回它真正的型態(以這個案例就是根據城市名稱取回 TemperatureBundle 物件)。

可參考 <https://ithelp.ithome.com.tw/articles/10234424>

- WeatherViewer 類別是 GUI 程式，已經提供部分程式碼：
 - JFrame 畫面之構成為兩個 Panel，上方 Panel 作為 menu bar，下方 Panel 作為顯示的畫布，畫布上可根據指定之字型、位置畫出字串。
 - 上方 Panel 之 Layout 方式是採 GridLayout，將畫面切成 4x1 的格狀，並依序放入 JLabel ("請選擇城市："字串)、JComboBox (城市選單)、JLabel ("請選擇日期："字串)、JComboBox (日期選單)
 - 只要點選任何一個 ComboBox，即會在下方 JPanel 上顯示選取之城市與日期之氣溫與天氣描述。
 - 兩個 ComboBox 之選項分別取自於 TemperatureFetcher 之 getAllCities()，以及 TemperatureBundle 之 getWeekdays()。
 - 每一天的氣溫與天氣描述都在 OneDayTemperature 中。
 - 下方畫布之 JPanel 應為繼承後之子類別，並 override paintComponent()。此子類別應為 inner class(已提供外殼)，以能較方便取得目前要顯示之溫度與天氣描述。在 paintComponent() 中，可透過 drawString 方法繪製字串到 JPanel 上(跟 drawOval 類似)：

drawString(String str, int x, int y)

5. (25%) 美國 2020 大選記票(多執行緒、讀檔)

此程式模擬多個地方記票中心持續提供不同州的票數給新聞媒體，新聞媒體持續同時讀取多個來源的票數，直到彙整出最後投票結果。

請參考 I/O 範例 15.6 以及 Thread 範例 23.8，並根據提供之 Vote.java (應盡量不要修改)、VoteArray.java (需略加修改，不能刪減程式)、ReadVotes.java (需填入必要的實作)、ReadVotesThread (需填入必要的實作)、ReadVotesThreadApplication (需填入必要的實作)，使其透過 thread 機制同時讀取多個檔案，並寫入至共享的 VoteArray 物件。詳細程式需求如下：

- 修改 ReadVotes.java，為使其可指定讀取之檔案，以及欲寫入之共享 VoteArray 物件，constructor 應設定為：

public ReadVotes (String filename, VoteArray sharedArray)

- 修改 ReadVotesThread.java，為 thread 之 task，可進行課本範例 ReadTextFile.java 中 main() 所進行之工作(包含開關讀檔串流與實際讀檔)
- 新增 ReadVotesThreadApplication.java (包含 main)，透過三個 ReadVotesThread 分別讀取 vote-1.txt、vote-2.txt 與 vote-3.txt 三個文字檔(文字檔請放置於最外層目錄)，請透過 thread pool 機制管理與執行 thread，讀取完後請印出 Biden 與 Trump 的最後得票結果，預期之執行結果如 t5-results.txt (應於 10 秒左右執行完畢)。

6. (5%) 請簡要說明你(妳)的期末專案目前的進度，並再提供此門課程的學習心得與建議。