

# 海大資工 Java程式設計課程



# Homework 3-1<sub>1</sub>

2

- Create a class called *Complex* for performing arithmetic with complex numbers. Complex numbers have the form

$$realPart + imaginaryPart * i$$

- ▣ Use floating-point variables *real* and *imaginary* to represent the private data of the class.
- ▣ Provide a constructor that enables an object of this class to be initialized when it is declared. (可設定實部與虛部)
- ▣ Provide a **no-argument** constructor with default values in case no initializers are provided.

# Homework 3-1<sub>2</sub>

3

- Provide public methods that perform the following operations:
  - Return the **conjugates** (共軛) of a Complex number
  - Return the **absolute number** (絕對值) of a Complex number
  - Add two Complex numbers
  - Subtract two Complex numbers
  - Multiply two Complex numbers
  - Divide two Complex numbers
  - Print Complex numbers in the form  $a + bi$ , where  $a$  is the real part and  $b$  is the imaginary part. (顯示到小數點後兩位)
- References:
  - [https://en.wikipedia.org/wiki/Complex\\_number](https://en.wikipedia.org/wiki/Complex_number)
  - <https://zh.wikibooks.org/zh-tw/%E8%A4%87%E6%95%B8>

# Sample Output

4

$a = 1.10 + 2.20i$

Conjugates of  $a = 1.10 - 2.20i$

Absolute value of  $a = 2.46$

$b = 3.30 + 4.40i$

Conjugates of  $b = 3.30 - 4.40i$

Absolute value of  $b = 5.50$

$a + b = 4.40 + 6.60i$

$a - b = -2.20 - 2.20i$

$a \times b = -6.05 + 12.10i$

$a \div b = 0.44 + 0.08i$

(不需使用者輸入，只要在Test程式中new兩個 Complex物件，並呼叫其相關的方法，再印出即可)

# Hint

5

- For the *conjugates*, *add*, *subtract*, *multiply*, and *divide* methods of class *Complex*, **return a new Complex object** with the results of the calculations.
- For the *absoluteValue* method of class *Complex*, return a **double** value.

# Homework 3-2<sub>1</sub>

6

- Employee 是一個 abstract class，有一個 abstract method: `int getEarnings()`，並有 4 個 subclass：
  - ▣ (1) SalariedEmployee：領週薪(`weeklySalary`)，薪水為  $\text{weeklySalary} * \text{weeks}$ 。
  - ▣ (2) HourlyEmployee：領時薪(`hourlySalary`)。如果一週工作時數小於或等於 40 小時，薪水為  $\text{hourlySalary} * \text{hours}$ ，如果工作時數多於 40 小時，超過 40 小時部份的薪水再乘 1.5 倍。
  - ▣ (3) CommissionEmployee：薪水為  $\text{commissionRate}$  (佣金比例) \* `grossSales` (銷售金額)
  - ▣ (4) PieceWorker：薪水為  $\text{wage}$  (單件酬勞) \* `piece` (完成作品的件數)

# Homework 3-2<sub>2</sub>

7

- Bonus是一個interface，用來定義額外的獎金，有兩個實作的類別：StaticBonus和DynamicBonus，如附檔：
  - ▣ StaticBonus設定獎金固定為10,000
  - ▣ DynamicBonus設定獎金為薪資的10%
- Employee必須包含一個型態為Bonus的資料成員，並可再設定此資料成員為StaticBonus或DynamicBonus (不在constructor中設定)，以便計算此Employee可獲得之獎金。

# Homework 3-2<sub>2</sub>

8

- 根據上述描述，請設計文字式的互動選單，透過 Scanner API讓使用者選擇要新增上述四種員工的哪一種，並讓使用者輸入選定類型所需的資料（舉例而言，若選Salaried Employee需要輸入weekly salary和working weeks），以及獎金種類（static或dynamic）。
- 當選擇類型時，若使用者輸入-1則輸入程序結束，直接顯示所有使用者輸入的詳細員工資料、計算薪資總和、並印出其獎金。



# Homework 3-2<sub>3</sub>

9

- 處理方法上，所有類型的員工物件資料需儲存至型態為 **Employee之陣列**，並呼叫陣列內Employee物件之
  - toString()方法列印詳細資訊。
  - getEarnings()方法取得各別之薪資以**進行加總**，並輸出總和薪資。
  - 其包含之Bonus資料成員的getBonus()方法，以取得獎金並進行加總與輸出。
- 請注意：程式中不能出現底下六個型態之變數：  
SalariedEmployee, HourlyEmployee, CommissionEmployee,  
PieceWorker, StaticBonus, DynamicBonus

# Sample Output

10

- 請參考employeeSampleOutput.txt

# Hint

11

- 請參考課本範例10.4~10.9
- 可將Scanner物件設定成Employee類別之field，用來輸入資料。
- 可在Employee類別中設計一個abstract的inputData method，在子類別中實作不同的輸入程序。
- 可將Bonus物件設定成Employee類別之field，並設計相對應的setBonus與getBonus方法，來設定與取得Bonus物件。
- 請留意，各種Employee物件不一定包含StaticBonus或DynamicBonus哪一種Bonus物件。

# Homework 3-3<sub>1</sub>

12

- 請設計一個簡易的RPG文字遊戲，包含以下幾個類別：
  - ▣ WarTest.java (不能刪減功能)：程式進入點。
  - ▣ ATK.java (不能刪減功能)：為「可攻擊」物件之interface，attack()方法即為攻擊，會回傳可造成敵方生命值(hp)之損傷大小。
  - ▣ Weapon.java (不能刪減功能)：為武器之abstract class，實作ATK介面，包含offense(攻擊力)和defense(防禦力)兩個屬性。
  - ▣ Pet.java (需擴增)：具攻擊力的寵物，實作ATK介面，其攻擊力為1~設定之最大攻擊力(maxAttack)的亂數。

# Homework 3-3<sub>2</sub>

13

- Character.java (需擴增)：人物，包含兩個屬性：生命值 (hp)與裝配(equipment)。裝備代表武器或Pet，可同時裝備多個武器或Pet (不能固定個數，必須用ArrayList)，會同時造成敵方損傷。
  - 請新增constructor：public Character (double hp)，設定給定之hp，並將equipment設定為僅包含一個攻擊力為DEFAULT\_ATK、防禦力為0之武器。
  - 請新增constructor：public Character ()，設定hp為DEFAULT\_HP，equipment則與前一個constructor相同。

# Homework 3-3<sub>3</sub>

14

- 請額外新增三個Weapon之子類別
  - ▣ NormalWeapon.java：一般武器，是concrete class。攻擊後，敵方之hp損傷值即為此武器之攻擊力。
  - ▣ DoubleWeapon.java：雙次攻擊武器，其有1/5之機率可達成兩次攻擊，亦即敵方損傷有4/5的機會為武器攻擊力，1/5是兩倍攻擊力。
  - ▣ PowerWeapon.java：強力武器，每次均可增強20%之攻擊力，亦即敵方損傷為武器攻擊力乘1.2倍。

# Homework 3-3<sub>4</sub>

15

- 請將War.java程式繼續設計完成，負責戰鬥相關功能，預設之戰鬥雙方為玩家(player)與NPC
  - ▣ 請完成兩個方法實作：
    - init ():
      - 將player之HP設定為150~199的亂數。
      - 將player裝備一個NormalWeapon (攻擊力為20~59之亂數)、一個DoubleWeapon (攻擊力為25~59之亂數)、一個PowerWeapon (攻擊力為30~59之亂數)、一支Pet (攻擊力為50~99之亂數)。
      - 將npc之hp設定為1000減去player之hp。
      - 將npc之equipment設定為僅包含一個攻擊力為DEFAULT\_ATK、防禦力為0之武器。

# Homework 3-3<sub>5</sub>

16

- battle (): 實際的戰鬥系統，玩家每次均可選擇：1.攻擊 2.防禦 3.結束。
  - 若選擇攻擊，則NPC之損傷即為player所有裝備之attack()回傳值總和 (需印出個別武器或Pet的損傷值)。NPC會接續攻擊，player之損傷為NPC武器之attack()回傳值。
  - 若選擇防禦，則NPC不會受到任何損傷，接著NPC會攻擊player，player會有1/2的機會遭受減半損傷(NPC武器之attack()回傳值的一半)，1/2的機會產生「自我療癒」事件，hp反往上增加原本預期之損傷值(NPC武器之attack()回傳值的一半)。
  - 若player或NPC其中任一方之hp已低於1 (即 $\leq 0$ )，則顯示「玩家被擊倒了！」或「擊倒了對手！」之訊息，遊戲結束。



# Homework 3-3<sub>6</sub>

17

- 程式中不可出現底下型態之變數：
  - ▣ NormalWeapon, DoubleWeapon, PowerWeapon, Pet

# Sample Output

18

- 請參考rpgSampleOutput-1.txt與rpgSampleOutput-2.txt

# Hint

19

- 此作業需大量運用亂數，請複習。
- 可運用ArrayList之運用方式，瞭解放入與取得item之作法。
- 與前一題相同，需熟悉多型概念，攻擊時需一次呼叫所有ATK型態物件之attack()方法。

# 作業要求

20

- 命名都要符合Camel Case
- 類別都要設定package，名稱為ntou.cs.java2017.你的英文名字.hw3
- 類別內要有註解，至少要簡述此類別與每個方法
- 每題都至少要有兩個以上的類別(可以更多)
- 請繳交電子檔，電子檔包含.java檔與.class檔(上傳至TronClass)
- 屍體(無法compile或執行)不計分