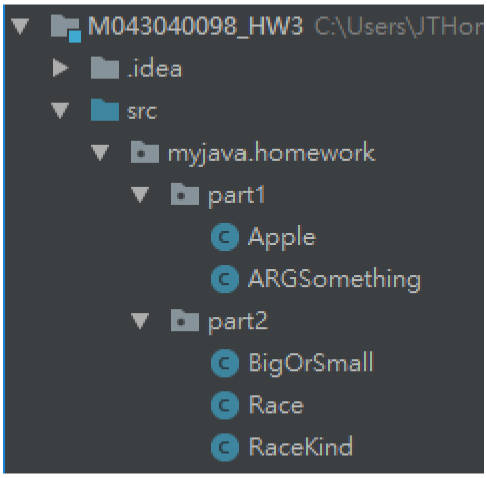
【2018 JAVA 物件導向程式設計 Homework 3】

注意事項

1. 請使用 JAVA 語言，配合 IntelliJ IDEA 寫本次作業並進行測試，並安裝、使用 JAVA SE Development Kit(JDK) 8 函式庫。
2. 請依據作業規定設定 IntelliJ IDEA 專案名稱與 package name，若未依照規定將根據狀況扣分。
3. 嚴禁抄襲其他同學作業，參與者 (抄襲與被抄襲) 皆以零分計算。
4. 請對你的程式碼有深入瞭解，demo 時助教會問。
5. 對題目有問題可以寄信問助教群 (java_ta@net.nsysu.edu.tw)或是到實驗室(EC5018) 詢問，但不幫忙 debug。
6. 逾期以零分計算，不接受補交，有任何因素導致無法如期繳交，請事先告知 Demo 時間會另外通知。

作業規定與上傳

1. IntelliJ IDEA 專案名稱：Student ID_HW3
2. Package path：請參考下圖



```
▼ M043040098_HW3 C:\Users\JTHor
    ▶ .idea
    ▼ src
        ▼ myjava.homework
            ▼ part1
                Apple
                ARGSomething
            ▼ part2
                BigOrSmall
                Race
                RaceKind
```

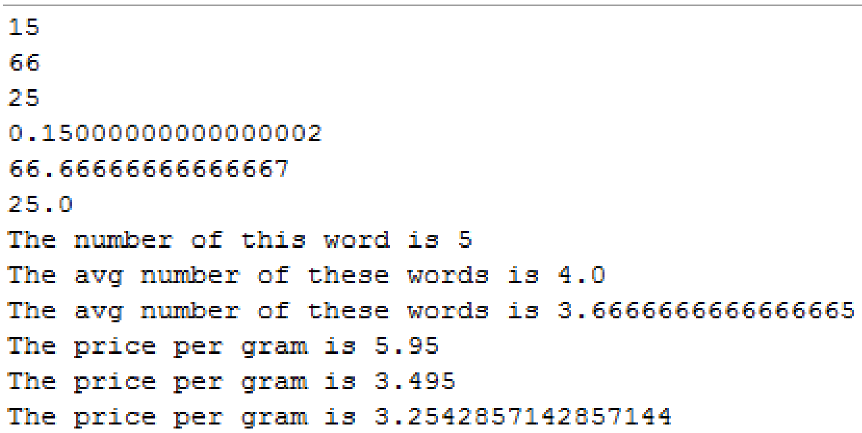
3. 作業請繳交專案之 tar 或 zip archive 並上傳至網路大學。
4. Deadline：2018 年 3 月 26 日(週一) 23:59
5. Example of package explorer(請根據作業規定修改)：

Hint：此次作業需參考到 UML。請注意＋、—、#、底線，都是有意義的。可以參考此 Wiki 之 Members 章節說明：Class Diagram (Wikipedia)

Part1:

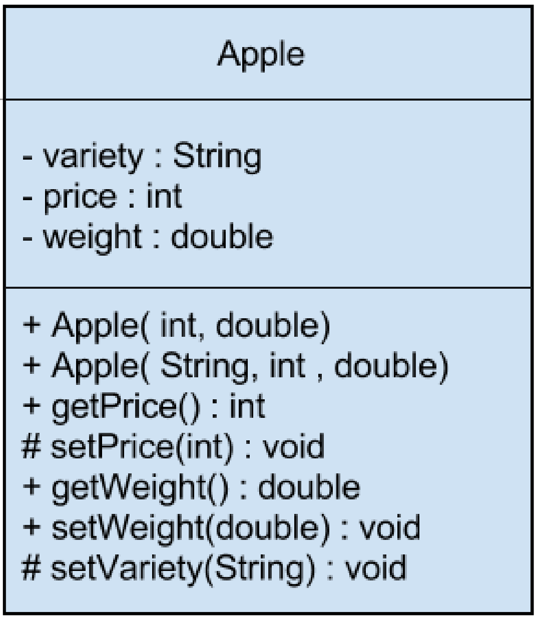
In this part, you have to know what the overloading is. You will get the incomplete code which just include a complete main function. In other words, you cannot modify any character in the main function. What you have to do is make the program run and cause the same result. You have to obey the following rules:

1. You cannot allow any warning to exist in the main function.
2. At least, you must write 4 functions for 4 kinds of input type.
3. These functions must write in the class with the main function
4. For 4 types:
   - i. Int: the output is that the number of the parameters divides into the sum. (Notice: the output is not floating number) (e.g. avg(10,20) = (10+20) / 2 = 15)
   - ii. Double: the output is that the number of the parameters divides into the sum. (You might be curious about the output 0.15000000000000002. If so, you can see this.)
   - iii. String: you have to calculate the average number of characters. (e.g. "Apple" has 5 characters and "Bee" has 3 characters, so (5 + 3) / 2 = 4.0)
   - iv. Apple: you have to calculate the average weight per dollar (e.g. $10, 59.5g and $30, 80.3g → (59.5 + 80.3) / (10 + 30) = 3.495)
5. The UML diagram of class Apple is as below.
6. The output is as below.



```
15
66
25
0.15000000000000002
66.66666666666667
25.0
The number of this word is 5
The avg number of these words is 4.0
The avg number of these words is 3.6666666666666665
The price per gram is 5.95
The price per gram is 3.495
The price per gram is 3.2542857142857144
```

Output



UML diagram of class Apple

Part2:

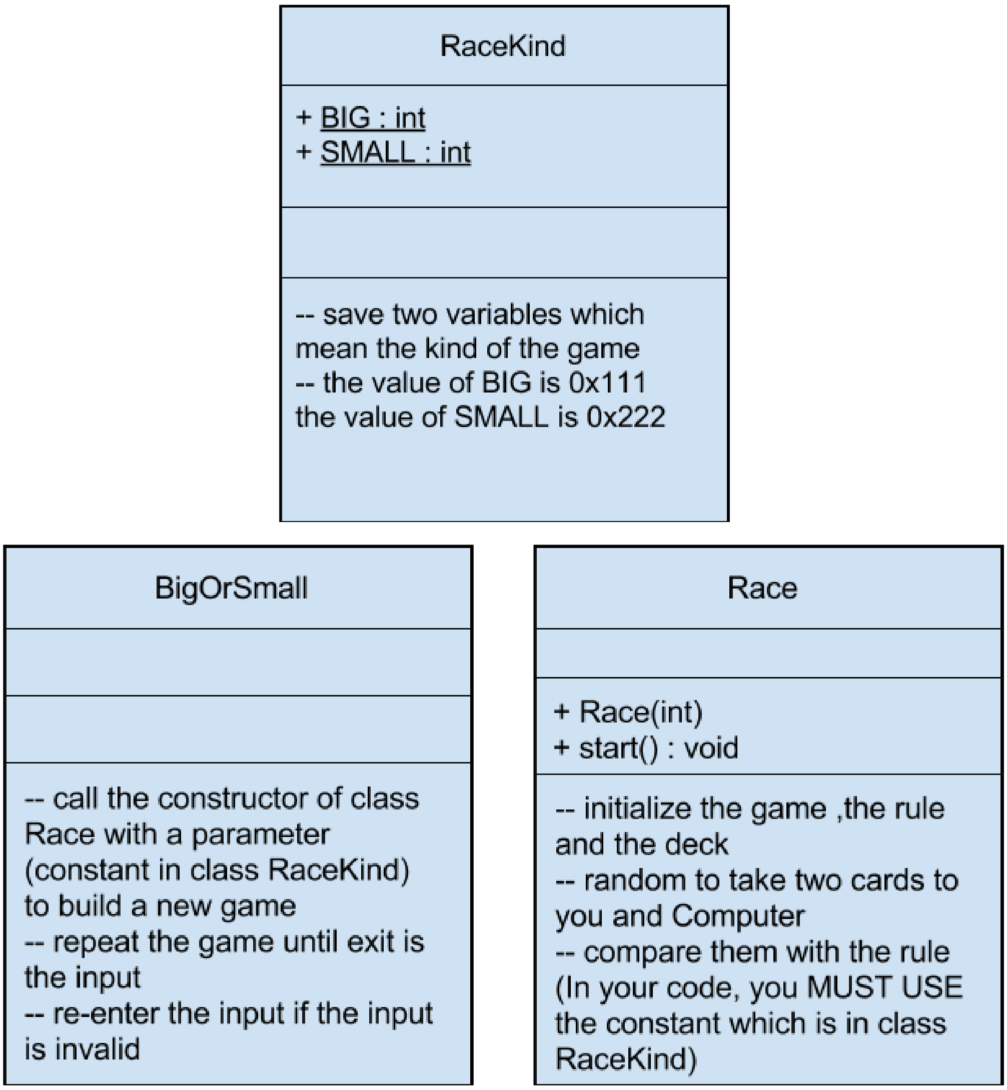
In this part, I want to play a game. You have to play a comparing game with a computer. In Poker, Ace is the biggest, and 2 is the smallest.

Moreover, ♠ > ♥ > ♦ > ♣ . And it is a game that the computer and you individually draw a card at random. And then, compare with their ranks and suits.
In the beginning, you should choose the kind of this game which winner is the bigger one or smaller one. And class RaceKind is a flag class which exists two constants.
The value of constants is not important, but you cannot use the value directly.
i.e. in the “bigger” race:

Race(0x111) ----wrong

Race(RaceKind.BIG) ---right

In class BigOrSmall, you have to build a new game repeatedly until you enter the “exit”. A class Race means a new game(52 cards), start() means the game starts and the parameter of constructor is the rule(bigger is winner? Or smaller?).



RaceKind

+ BIG : int
+ SMALL : int

-- save two variables which mean the kind of the game
-- the value of BIG is 0x111
the value of SMALL is 0x222

BigOrSmall

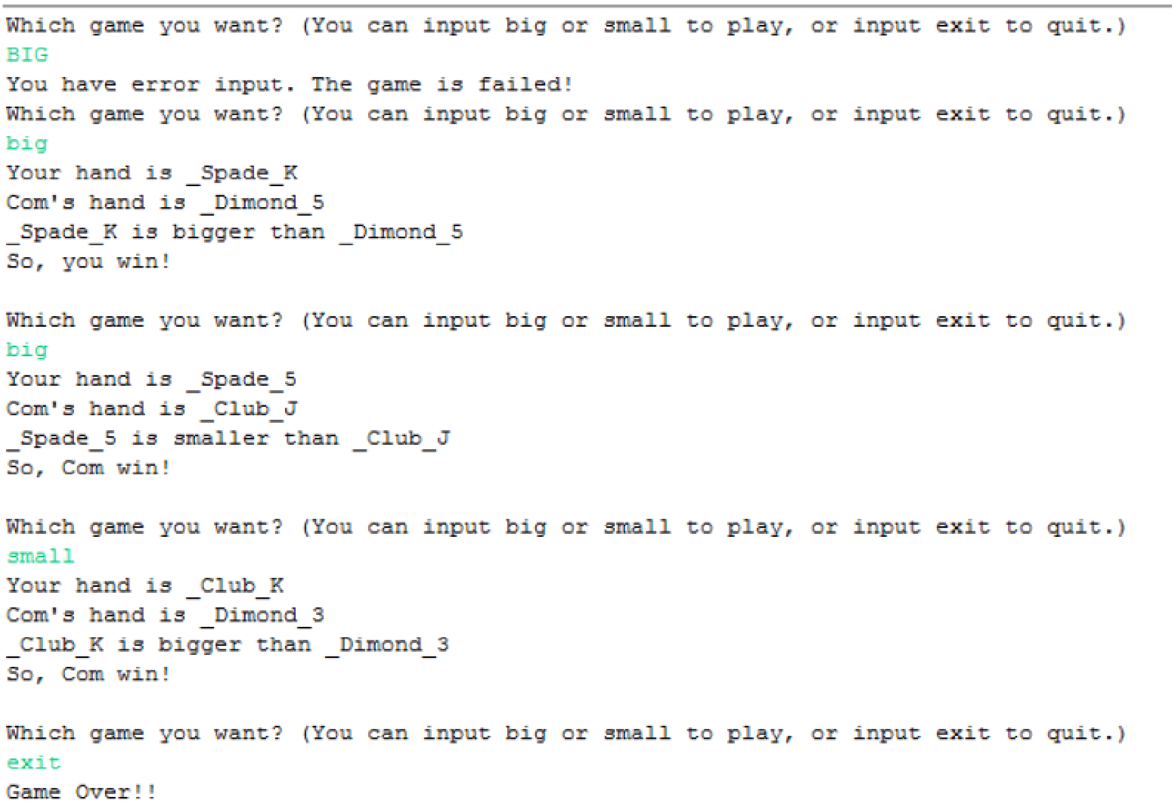
-- call the constructor of class Race with a parameter (constant in class RaceKind) to build a new game
-- repeat the game until exit is the input
-- re-enter the input if the input is invalid

Race

+ Race(int)
+ start() : void

-- initialize the game ,the rule and the deck
-- random to take two cards to you and Computer
-- compare them with the rule (In your code, you MUST USE the constant which is in class RaceKind)

The output example is as below:



```
Which game you want? (You can input big or small to play, or input exit to quit.)
BIG
You have error input. The game is failed!
Which game you want? (You can input big or small to play, or input exit to quit.)
big
Your hand is _Spade_K
Com's hand is _Dimond_5
_Spade_K is bigger than _Dimond_5
So, you win!

Which game you want? (You can input big or small to play, or input exit to quit.)
big
Your hand is _Spade_5
Com's hand is _Club_J
_Spade_5 is smaller than _Club_J
So, Com win!

Which game you want? (You can input big or small to play, or input exit to quit.)
small
Your hand is _Club_K
Com's hand is _Dimond_3
_Club_K is bigger than _Dimond_3
So, Com win!

Which game you want? (You can input big or small to play, or input exit to quit.)
exit
Game Over!!
```