

物件導向程式設計 第二次作業

Object-Oriented Programming Assignment II

繳交截止日：2019/04/16 23:59，如果在 2019/4/17-2019/4/24 之間繳交則算遲交，並酌扣 20%成績。這之後的時間均不接受補交。

The submission deadline is 2019/4/16 23:59. You can have a late submission between 2019/4/17-2019/4/24 but can earn 80% credit. We do not accept even later submission.

1.
請寫出一個程式讓使用者可以輸入十個數字。在你的程式碼中，必須含有 `class Sort` 內含一個私有的陣列以及一個建構元，該建構元必須接收一個陣列，且此陣列名稱與私有陣列名稱相同。`class Sort` 內要包含以下函數：

- 一、使用實體函數 `max()` 印出最大值
- 二、使用實體函數 `min()` 印出最小值
- 三、使用實體函數 `sort()` 印出由小到大排序後的結果。

※ 最大值和最小值不可以從排序後的陣列直接輸出。

請使用以下程式碼作為框架,填寫出完整程式碼來符合上述功能。

Your program is required to take as input ten integers. Your code needs to contain a class `Sort` with a private array and a constructor. The constructor needs to take as input an array and the name of the array needs to be the same as the private array.

Class `Sort` needs to contain the following functions :

- A. Need to contain a method `min()` that can print the minimum value.
- B. Need to contain a method `max()` that can print the maximum value.
- C. Need to contain a method `sort()` that can print from smallest to largest.

※ Can't output the maximum and minimum value directly from the sorted array.

Your program needs to be written based on the incomplete code below.

```

public static void main(String[] args)
{
    int array[] = new int[10];
    // To Do
    Sort s = new Sort(array);

    s.max();
    s.min();
    s.sort();
}

```

```

class Sort
{
    private int array[];

    public Sort(int array[])
    {
        //To Do
    }

    public void sort()
    {
        //To Do
    }

    public void max()
    {
        //To Do
    }

    public void min()
    {
        //To Do
    }
}

```

2.

請完成 class *card*，讓 class *hw02* 可以成功執行。

Please finish class *card* so that class *hw02* can be executed successfully.

class *card* 的架構及功能必須符合以下條件：

- I. 含有兩個變數 *number* (使用者 id)、*remain* (餘額，初始為 100)。只能透過 *store* 和 *charge* function 來修改 *remain* 值。
- II. 資料成員 *number* 的值只能用建構元來賦予
- III. 必須含有 *store* function：儲值功能(增加 *remain* 的值)，不能存 0 元或負值，否則印出"error"
- IV. 必須含有 *charge* function：扣款功能(減少 *remain* 的值)，不能扣負值及 0 元，餘額不夠會印出"error"
- V. 必須含有 *getRemain* function：查詢餘額功能(取得 *remain* 值並印出)，且此 function 可以在未建立物件的情況下被呼叫

The function and structure of class *card* must follow the rules below:

- I. There are two variables, *number* (user id) and *remain* (balance, initially 100). The data member *remain* can only be modified by invoking *store* and *charge* function.
- II. The data member *number* can only be assigned by **constructor**.
- III. Must contain *store* function: It can retrieve the correct balance after deposit. If you deposit money equal to or less than 0, it will print “error”.
- IV. Must contain *charge* function: It can get the correct balance after withdraw. You cannot withdraw money if the balance is equal to or less than 0. And if balance is less than the amount of money you want to withdraw, you will print “error”.
- V. Must contain *getRemain* function : You can get *remain* and print it. And it can execute *card.getRemain()* successfully.

```

public class hw02 {
    public static void main(String args[]) {
        Scanner sca = new Scanner(System.in);
        System.out.print("Inpu user id");
        card aa = new card(sca.next());
        System.out.print("Input stored value");
        aa.store(sca.nextInt());
        aa.charge(sca.nextInt());
        card.getRemain();
    }
}

class card{
    remain=100;
    String number;

    card(String ) {
        //To Do
    }

    store(int ) {
        //To Do
    }

    charge(int ) {
        //To Do
    }

    getRemain() {
        //To Do
    }
}

```

3.

輸入三到五個正整數，根據輸入不同數量的數字執行不同的函數功能；輸入三個數字時，計算三個數字的最大公因數與最小公倍數；輸入四個數字時，把四個數字以不重複的方式任意代入 x_1, x_2, x_3, x_4 ，使 $Q = -2x_1^2 - 4x_2 + 5x_3 + \sqrt{x_4^3}$ 這個方程式的 Q 為最大值；輸入五個數字時，計算這五個數字的標準差(standard deviation)。請使用以下程式碼作為框架,填寫出完整程式碼來符合上述功能。

Your program is required to take as inputs three to five positive integers, according to input the number of different and perform different functions. When the user inputs three numbers, your program calculates the **greatest common divisor** and **the least common multiple** of these three numbers. When the user inputs four numbers, substitute the four numbers into x_1, x_2, x_3, x_4 in a non-repeating manner, in such a way that $Q = -2x_1^2 - 4x_2 + 5x_3 + \sqrt{x_4^3}$ is maximized. When the user inputs five

numbers, calculate the **standard deviation** (see below) of these five numbers. Your code must contain the *MultiFunction* class. In *main()*, you are allowed to only invoke *show()* from *MultiFunction* class. Your program needs to be written based on the incomplete code below.

```

3  class hw02
4  {
5      public static void main(String arg[])
6      {
7          //To Do
8          MultiFunction mf = new MultiFunction();
9          if // To Do
10             mf.show(x1, x2, x3); // Calculate gcd and lcm
11         else if // To Do
12             mf.show(x1, x2, x3, x4); // Calculate the max value of the Q-function
13         else if // To Do
14             mf.show(x1, x2, x3, x4, x5); // Calculate standard deviation
15         else // To Do
16             //To Do
17     }
18
19     public class MultiFunction
20     {
21         //To Do
22     }
23 }

```

Example:

(1)

if $in_1 = 10, in_2 = 12, in_3 = 20$

$\Rightarrow \text{gcd} = 2, \text{lcm} = 60$

(2)

if $in_1 = 12, in_2 = 54, in_3 = 9, in_4 = 71$

$$Q = -2x_1^2 - 4x_2 + 5x_3 + \sqrt{x_4^3}$$

$\Rightarrow x_1 = 9, x_2 = 12, x_3 = 54, x_4 = 71$

$\Rightarrow Q = 658.2566$ (Q is maximized)

(3) Standard deviation:

- $N = 5$.
- $\{x_1, x_2, \dots, x_N\}$ are input values.
- u is the average value of these $\{x_1, x_2, \dots, x_N\}$.

$$SD = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - u)^2}$$

if $in_1 = 10, in_2 = 38, in_3 = 23, in_4 = 6, in_5 = 21$

$\Rightarrow SD = 11.2178$