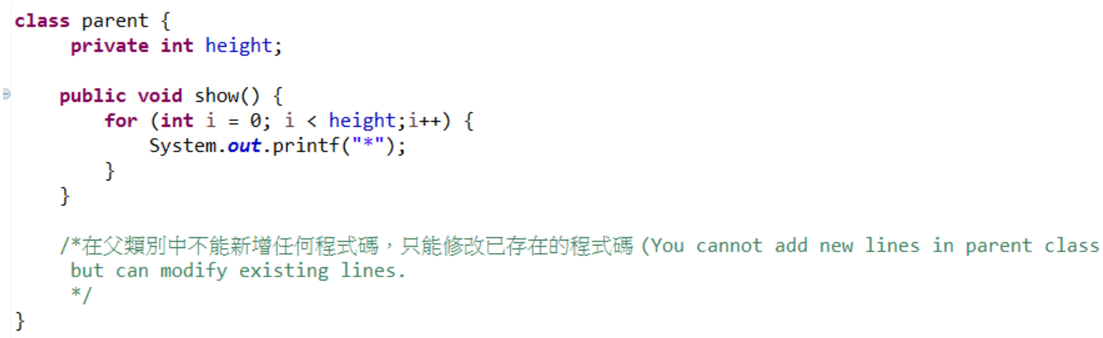
物件導向程式設計 第四次小考

Object-Oriented Programming Quiz 4

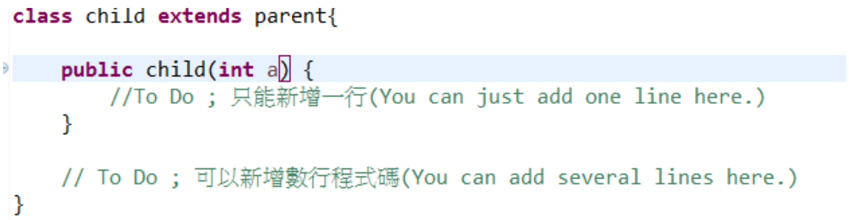
1.

請修改下列程式碼使得使用者可以輸入奇數 *N*，並且用 *show()* function 印出高度為 *N* 的「特殊」三角形 (形狀請見下例)。

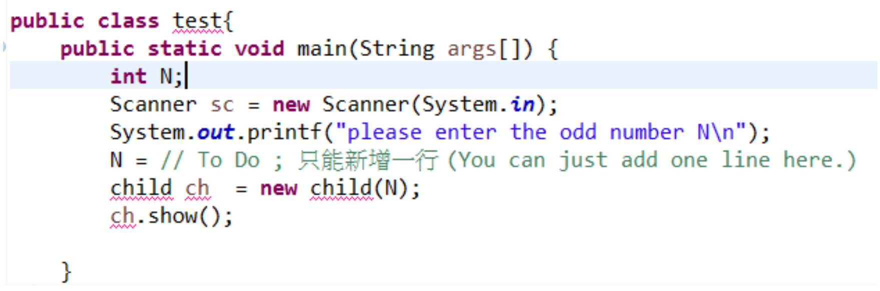
Your program is required to take as input an odd number *N*, and prints a triangle of height *N* by using *show()* function. Your program needs to be completed based on the following framework.



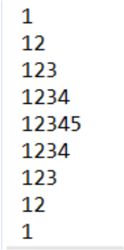
```
class parent {
     private int height;

    public void show() {
        for (int i = 0; i < height;i++) {
            System.out.printf("*");
        }
    }

    /*在父類別中不能新增任何程式碼，只能修改已存在的程式碼 (You cannot add new lines in parent class
     but can modify existing lines.
     */
}
```



```
class child extends parent{

    public child(int a) {
        //To Do ; 只能新增一行(You can just add one line here.)
    }

    // To Do ; 可以新增數行程式碼(You can add several lines here.)
}
```



```
public class test{
    public static void main(String args[]) {
        int N;
        Scanner sc = new Scanner(System.in);
        System.out.printf("please enter the odd number N\n");
        N = // To Do ; 只能新增一行 (You can just add one line here.)
        child ch  = new child(N);
        ch.show();

    }
```

Input	Output
9	 1 12 123 1234 12345 1234 123 12 1

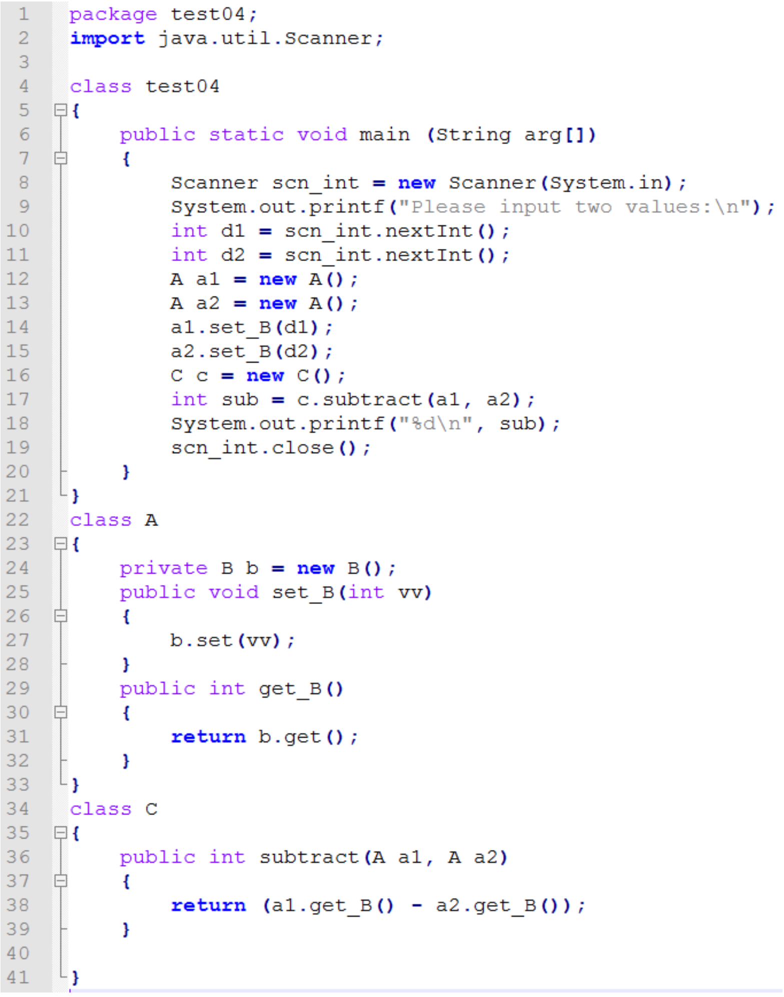
2.

請寫出一個程式讓使用者可以輸入兩個數字 *N*、*M*，你需要在某處新增 private class B，能夠正常計算 *N* 減 *M* 的結果。你的程式將符合以下限制。

- 限制 1：private class B 內部的 data member 皆為 private。
- 限制 2：private class B 不可以設定建構元。
- 限制 3：只能新增 private class B，**不可**刪減現有程式。
- 限制 4：以下圖架構完成程式碼。

Your program is required to take as inputs two numbers *N* and *M*, and outputs *N*-*M*. You need to add private class B somewhere.

- Limit 1：all data members in private class B are private.
- Limit 2：private class B does not have constructors.
- Limit 3：you are allowed to create private class B; however, you are **NOT** allowed to delete the code given in this quiz.
- Limit 4：your program needs to be completed based on the following framework.



```java
package test04;
import java.util.Scanner;

class test04
{
    public static void main (String arg[])
    {
        Scanner scn_int = new Scanner(System.in);
        System.out.printf("Please input two values:\n");
        int d1 = scn_int.nextInt();
        int d2 = scn_int.nextInt();
        A a1 = new A();
        A a2 = new A();
        a1.set_B(d1);
        a2.set_B(d2);
        C c = new C();
        int sub = c.subtract(a1, a2);
        System.out.printf("%d\n", sub);
        scn_int.close();
    }
}
class A
{
    private B b = new B();
    public void set_B(int vv)
    {
        b.set(vv);
    }
    public int get_B()
    {
        return b.get();
    }
}
class C
{
    public int subtract(A a1, A a2)
    {
        return (a1.get_B() - a2.get_B());
    }

}
```

Example:

Input:	**Output:**
10 15	-5
or	
55 11	44

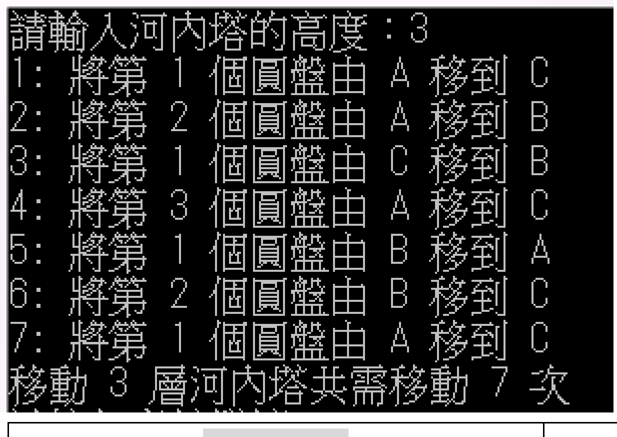
3.

請修改以下程式讓使用者可以輸入一個小於 31 的正整數 *N*，並根據程式碼需求在 main 中使用 class *Teacher* 的函數 *show()*輸出一個字串為**第 *N* 次移動** (*N* 為使用者輸入的正整數參數，須設置給 class *Teacher* 內的 *number*) **的目的為 ”A/B/C 哪一個” 杆子**。

class Student 中的函數 hanoi(n, A, B, C) 此 n 初始值為 5 且並非為使用者輸入的正整數參數 *N*，其定義如下：
有三根杆子 A、B、C。A 杆上有 n 個穿孔圓盤，盤的尺寸由下到上依次變小。要求按下列兩條規則將所有圓盤移至 C 杆。請將**第 *N* 次移動 (*N* 為使用者輸入的正整數參數) 的目的地杆子**傳給 class *Teacher* 的字元成員。移動的規則如下：

- 每次只能移動一個圓盤。
- 大盤不能疊在小盤上面。

以下為 n=3 的 hanoi(n, A, B, C)範例。



```
請輸入河內塔的高度：3
1: 將第 1 個圓盤由 A 移到 C
2: 將第 2 個圓盤由 A 移到 B
3: 將第 1 個圓盤由 C 移到 B
4: 將第 3 個圓盤由 A 移到 C
5: 將第 1 個圓盤由 B 移到 A
6: 將第 2 個圓盤由 B 移到 C
7: 將第 1 個圓盤由 A 移到 C
移動 3 層河內塔共需移動 7 次
```

Input (N)	**Output**
5	第 5 次移動的目的地為 A 杆子
6	第 6 次移動的目的地為 C 杆子

Your program is required to take as input a positive integer *N* which is less than 31, and use the *show*() function to output a string **“the destination of the N^{th} move is**

<u>“A or B or C”-rod</u>. *N* must be set to the *number* of the class *Teacher*.

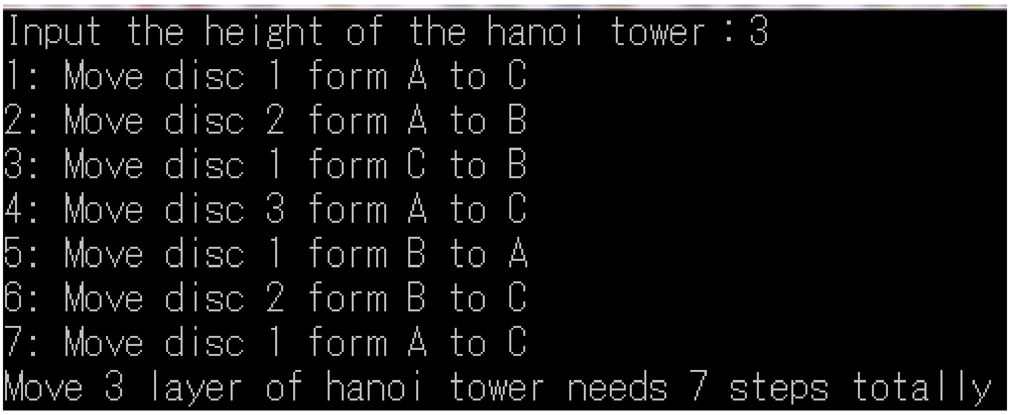
The definition of the hanoi(n,A,B,C) is shown below **<u>(The initial value of n is set to 5 and it’s not equal to the positive integer *N* entered by the user)</u>**：

There are three rods A, B, and C. There are n perforated discs on the A-rod, and the size of the disc becomes smaller from the bottom to top. It is required to move all the discs from the A-rod to the C-rod according to the following rules.

- Only one disc can be moved at a time.
- A larger disc may not be placed on top of a smaller disc.

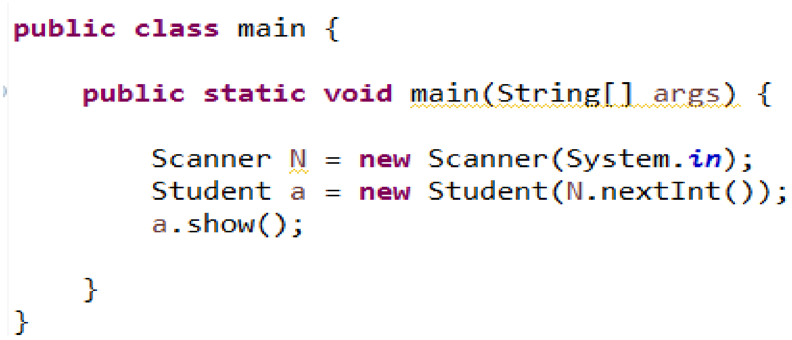
Please assign the destination rod of N^{th} move (*N* is the positive integer entered by the user) to the character member of the class *Teacher.*

The following is an example of hanoi(n,A,B,C) with n=3 .

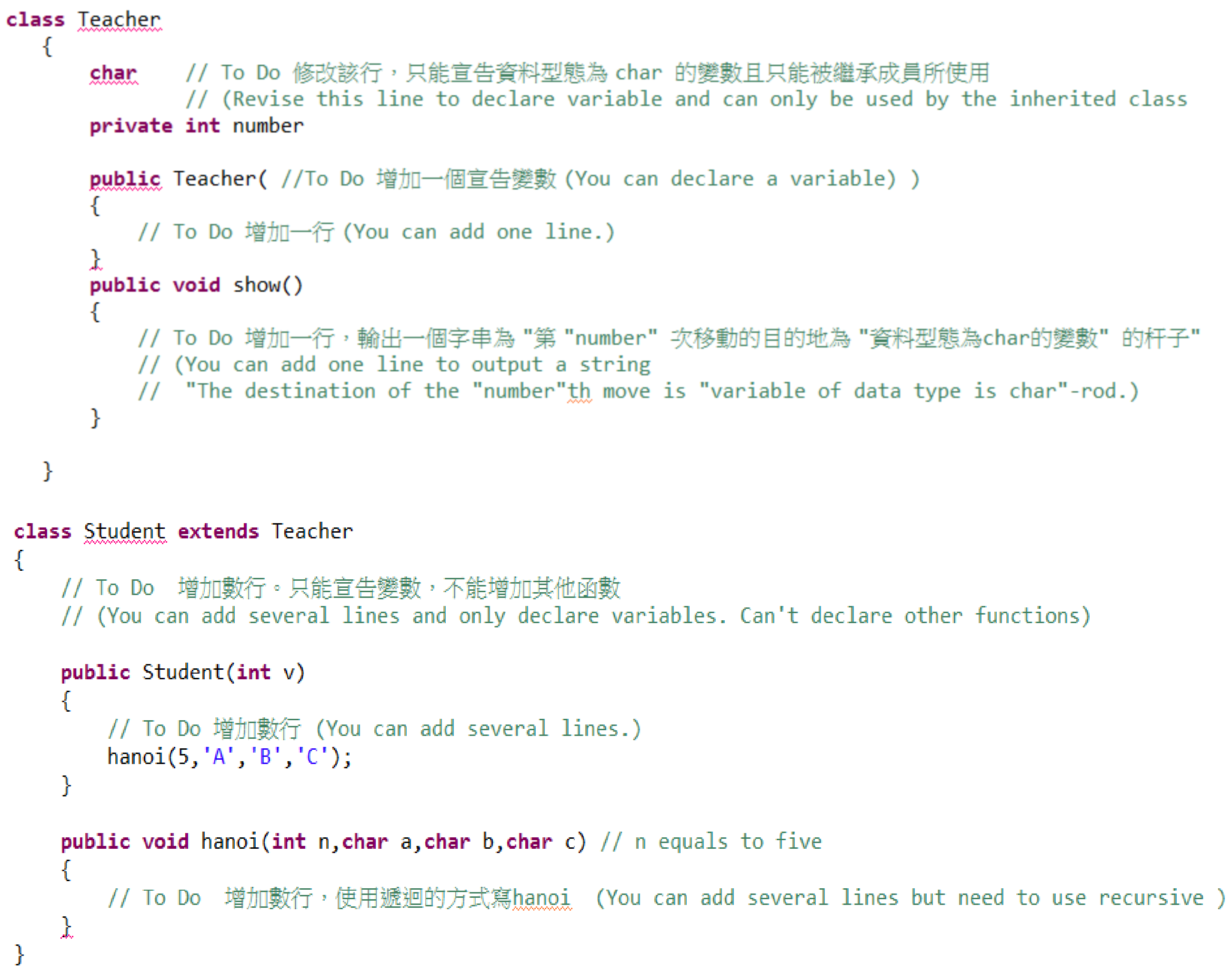


```
Input the height of the hanoi tower : 3
1: Move disc 1 form A to C
2: Move disc 2 form A to B
3: Move disc 1 form C to B
4: Move disc 3 form A to C
5: Move disc 1 form B to A
6: Move disc 2 form B to C
7: Move disc 1 form A to C
Move 3 layer of hanoi tower needs 7 steps totally
```

Input (N)	Output
5	The destination of the 5th move is A-rod.
6	The destination of the 6th move is C-rod.



```
public class main {

    public static void main(String[] args) {

        Scanner N = new Scanner(System.in);
        Student a = new Student(N.nextInt());
        a.show();

    }
}
```



```
class Teacher
    {
        char    // To Do 修改該行，只能宣告資料型態為 char 的變數且只能被繼承成員所使用
                // (Revise this line to declare variable and can only be used by the inherited class
        private int number

        public Teacher( //To Do 增加一個宣告變數 (You can declare a variable) )
        {
            // To Do 增加一行 (You can add one line.)
        }
        public void show()
        {
            // To Do 增加一行，輸出一個字串為 "第 "number" 次移動的目的地為 "資料型態為char的變數" 的杆子"
            // (You can add one line to output a string
            //  "The destination of the "number"th move is "variable of data type is char"-rod.)
        }

    }

 class Student extends Teacher
 {
     // To Do  增加數行。只能宣告變數，不能增加其他函數
     // (You can add several lines and only declare variables. Can't declare other functions)

     public Student(int v)
     {
         // To Do 增加數行 (You can add several lines.)
         hanoi(5,'A','B','C');
     }

     public void hanoi(int n,char a,char b,char c) // n equals to five
     {
         // To Do  增加數行，使用遞迴的方式寫hanoi  (You can add several lines but need to use recursive )
     }
 }
```