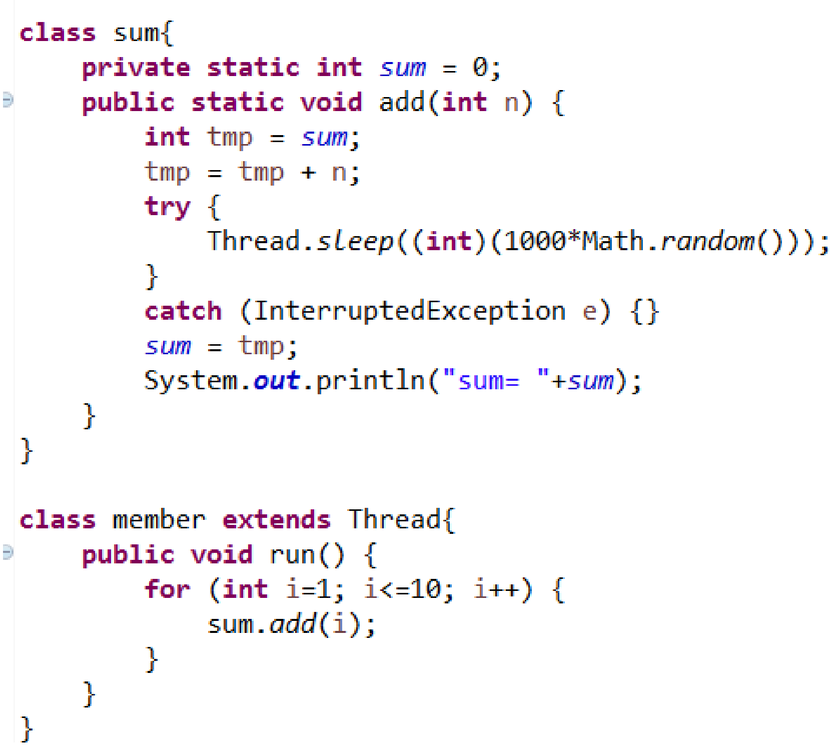
物件導向程式設計 第七次小考

Object-Oriented Programming Quiz 7

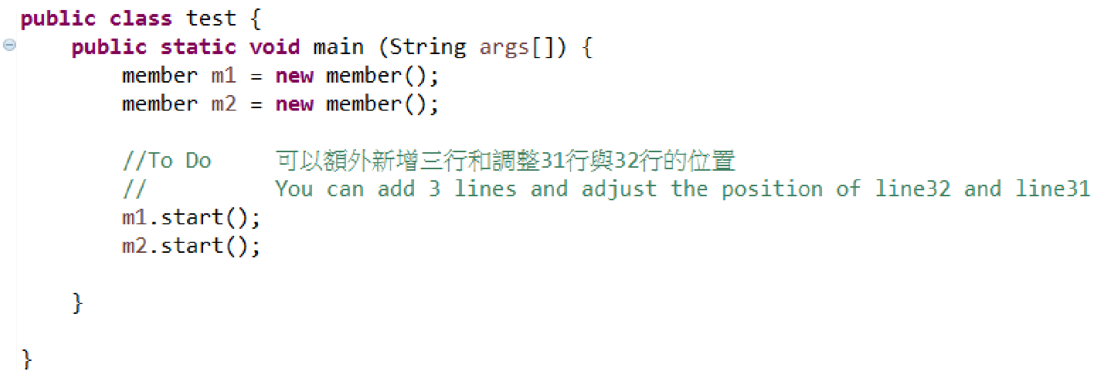
1.

請按照以下的框架，使得程式輸出如下圖。

Your program needs to be written based on the following framework and the output is shown below. (You cannot modify **class *sum*** and **class *member***.)

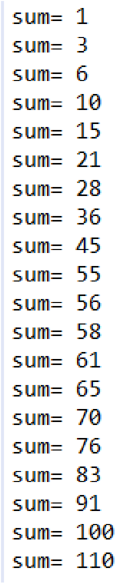


```java
class sum{
    private static int sum = 0;
    public static void add(int n) {
        int tmp = sum;
        tmp = tmp + n;
        try {
            Thread.sleep((int)(1000*Math.random()));
        }
        catch (InterruptedException e) {}
        sum = tmp;
        System.out.println("sum= "+sum);
    }
}

class member extends Thread{
    public void run() {
        for (int i=1; i<=10; i++) {
            sum.add(i);
        }
    }
}
```



```java
public class test {
    public static void main (String args[]) {
        member m1 = new member();
        member m2 = new member();

        //To Do     可以額外新增三行和調整31行與32行的位置
        //          You can add 3 lines and adjust the position of line32 and line31
        m1.start();
        m2.start();

    }

}
```

Output



```
sum= 1
sum= 3
sum= 6
sum= 10
sum= 15
sum= 21
sum= 28
sum= 36
sum= 45
sum= 55
sum= 56
sum= 58
sum= 61
sum= 65
sum= 70
sum= 76
sum= 83
sum= 91
sum= 100
sum= 110
```

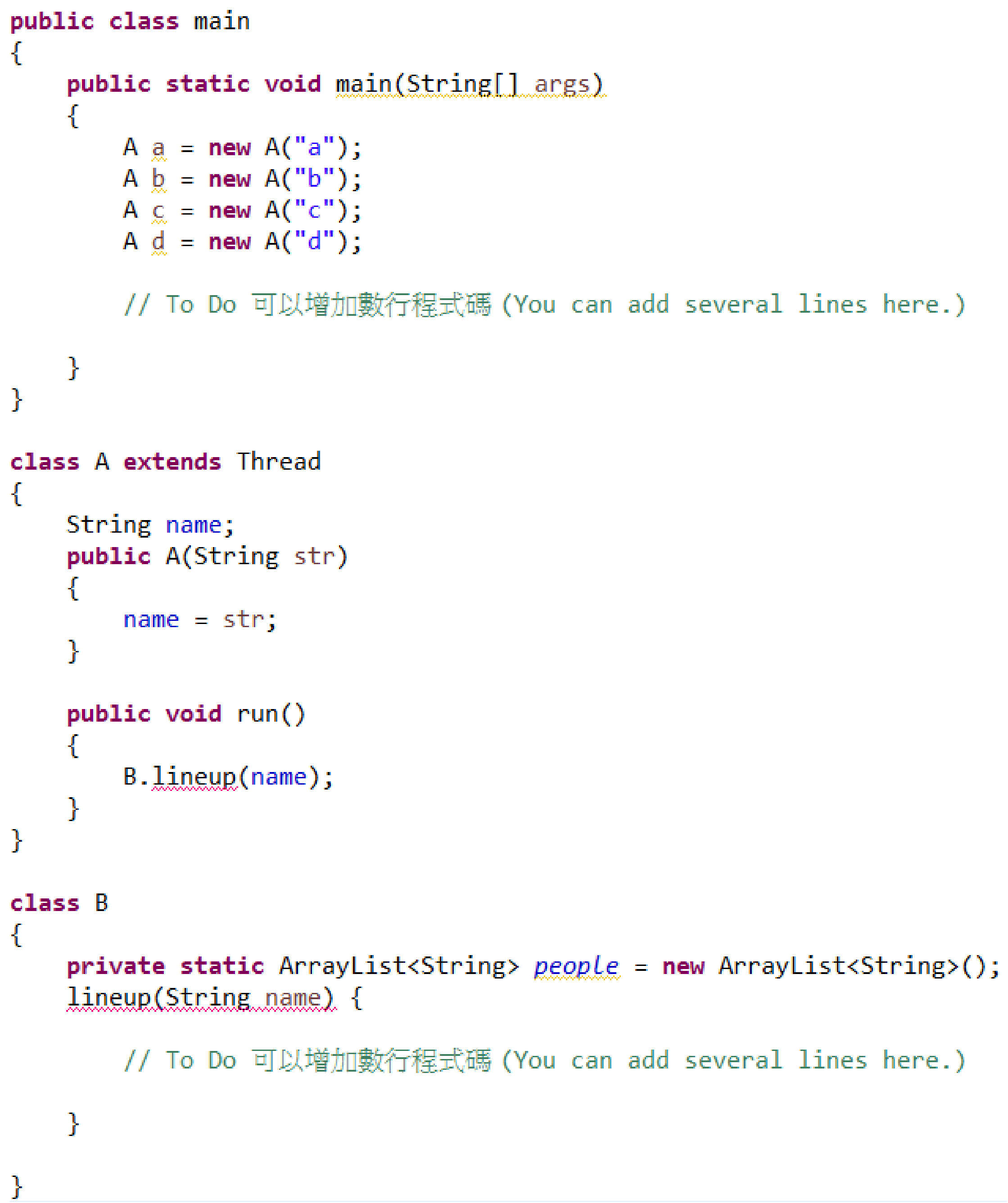
2.

請修改下列程式碼並在 main() 啟動 *a*、*b*、*c*、*d* 執行緒使用 class B 的函數 *lineup( )* 並**輸出每輪添加後的結果。**你的程式將符合以下限制。

- 限制：class B 的函數 *lineup( )* 一次只允許一個執行緒進入。

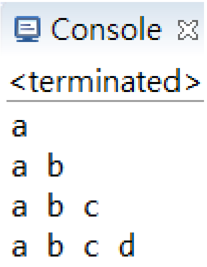
Please revise the code and start threads *a, b, c, d* in main() to use the *lineup( )* function and **output the results after each round of addition.**

- Limit：The function *lineup( )* allows **only one thread to enter at a time**.



```
public class main
{
    public static void main(String[] args)
    {
        A a = new A("a");
        A b = new A("b");
        A c = new A("c");
        A d = new A("d");

        // To Do 可以增加數行程式碼(You can add several lines here.)

    }
}

class A extends Thread
{
    String name;
    public A(String str)
    {
        name = str;
    }

    public void run()
    {
        B.lineup(name);
    }
}

class B
{
    private static ArrayList<String> people = new ArrayList<String>();
    lineup(String name) {

        // To Do 可以增加數行程式碼(You can add several lines here.)

    }

}
```

Output：

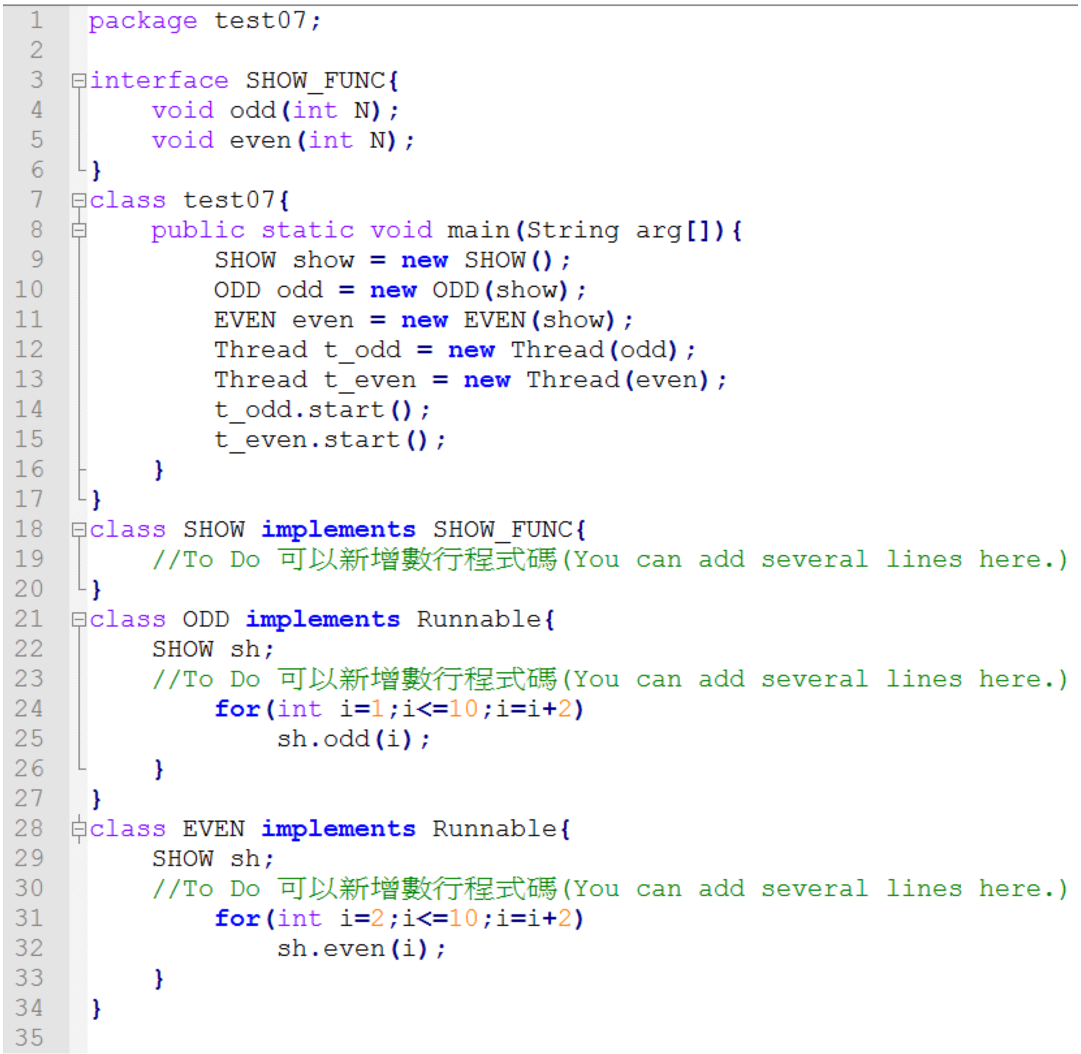


```
<terminated>
a
a b
a b c
a b c d
```

3.

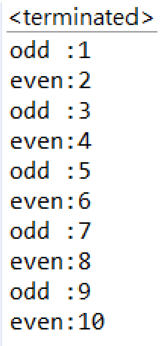
請按照以下的框架，使得程式輸出如下圖。

Your program needs to be written based on the following framework and the output is shown below.



```java
package test07;

interface SHOW_FUNC{
    void odd(int N);
    void even(int N);
}
class test07{
    public static void main(String arg[]){
        SHOW show = new SHOW();
        ODD odd = new ODD(show);
        EVEN even = new EVEN(show);
        Thread t_odd = new Thread(odd);
        Thread t_even = new Thread(even);
        t_odd.start();
        t_even.start();
    }
}
class SHOW implements SHOW_FUNC{
    //To Do 可以新增數行程式碼(You can add several lines here.)
}
class ODD implements Runnable{
    SHOW sh;
    //To Do 可以新增數行程式碼(You can add several lines here.)
        for(int i=1;i<=10;i=i+2)
            sh.odd(i);
    }
}
class EVEN implements Runnable{
    SHOW sh;
    //To Do 可以新增數行程式碼(You can add several lines here.)
        for(int i=2;i<=10;i=i+2)
            sh.even(i);
    }
}

```

Output:



```
<terminated>
odd :1
even:2
odd :3
even:4
odd :5
even:6
odd :7
even:8
odd :9
even:10
```