

Programmazione Concorrente e Distribuita

Esame scritto del 19 Dicembre 2011

Versione B

Esercizio 1 (5 punti)

Si consideri il codice seguente.

```
1 class X {  
2     void do1() {  
3     }  
4 }  
5  
6 class Y extends X {  
7     void do2() {  
8     }  
9 }  
10  
11 class Chrome {  
12     public static void main(String [] args) {  
13         X x1 = new X();  
14         X x2 = new Y();  
15         Y y1 = new Y();  
16         // insert code here  
17     }  
18 }
```

Quali delle seguenti righe di codice producono un codice che compila se sostituite nella riga 16?

A `x2.do2();`

B `(Y)x2.do2();`

C `((Y)x2).do2();`

D Nessuna della precedenti produce un codice che compila

Esercizio 2 (10 punti)

Si consideri il codice seguente.

```
1 class C{
2     private int counter = 0;
3
4     public synchronized void m(){
5         for(int i = 0; i < 1000000; i++)
6             counter++;
7         for(int j = 0; j < 1000000; j++)
8             counter--;
9         counter++;
10        notifyAll();
11    }
12    public synchronized void n() throws InterruptedException{
13        if(counter < 9)
14            wait();
15        System.out.print(counter + " ");
16    }
17 }
18
19 class T1 extends Thread{
20     private C c;
21
22     public T1(C c){
23         this.c = c;
24     }
25
26     public void run(){
27         for(int i = 0; i < 1000; i++)
28             c.m();
29     }
30 }
31
32 class T2 extends Thread{
33     private C c;
34
35     public T2(C c){
36         this.c = c;
37     }
38
39     public void run(){
40         try{
41             c.n();
42         } catch(InterruptedException ie){}
43     }
44 }
45
46 class Test {
47     public static void main(String [] argv){
48         C c = new C();
49         T1 r = new T1(c);
50         T2 s = new T2(c);
51         T2 t = new T2(c);
52
53         t.start();
54         s.start();
55         r.start();
56     }
57 }
```

Ciascuna delle classi definite nel codice precedente compila senza errori. L'esecuzione del metodo `main()` della classe `Test` stampa una coppia di interi n, m , con $n \geq 0$, e $m \geq 0$. Quali delle seguenti affermazioni sono vere?

- A Sicuramente $n \neq m$, $n \geq 9$ e $m \geq 9$
- B Sicuramente $n = m \geq 9$
- C Può accadere che n o m siano 0
- D Sicuramente $n \geq 1$ e $m \geq 1$

Per ciascuna risposta (vera o falsa) si fornisca una motivazione (breve e chiara).

Esercizio 3 (10 punti)

Si consideri il codice seguente.

```
1  class Node{
2      private String info;
3      private Node next;
4
5      public Node(String info){
6          this.info = info;
7      }
8
9      public void setNext(Node next){
10         this.next = next;
11     }
12
13     public static void printInfo(Node n){
14         synchronized(n){
15             if(n != null)
16                 System.out.println(n.info + "_");
17         }
18     }
19
20     public synchronized void printNext(){
21         System.out.print(info + "_");
22         for(int i = 0; i < 9999999; i++){
23             printInfo(next);
24             System.out.print("AFTERPRINTING_");
25         }
26     }
27 }
28 class T extends Thread{
29     private Node n;
30
31     public T(Node n){
32         this.n = n;
33     }
34
35     public void run(){
36         n.printNext();
37     }
38 }
39
40 class Ex{
41     public static void main(String [] argv){
42         Node one = new Node("ONE");
43         Node two = new Node("TWO");
44         one.setNext(two);
45         two.setNext(one);
46         T[] threadArray = {new T(one), new T(two)};
47         int k = (int) (Math.random()*2);
48         threadArray[k].start();
49         threadArray[(k+1)%2].start();
50     }
51 }
```

Le classi nel codice precedente compilano senza errori. Si descriva ogni output possibile causato dall'esecuzione del metodo `main()` della classe `Ex`, considerando le seguenti regole:

- Se una esecuzione del `main()` provoca la scrittura di `SOMETHING` nello standard output e termina normalmente (senza errore), si scriva solo `SOMETHING`;
- Se una esecuzione del `main()` provoca la scrittura di `SOMETHING` nello standard output e quindi provoca una eccezione, scrivere solo `SOMETHING` seguito da `EXCEPTION`;
- Se una esecuzione del `main()` provoca la scrittura di `SOMETHING` nello standard output e quindi provoca un deadlock, scrivere solo `SOMETHING` seguito da `DEADLOCK`.

Inoltre, per ciascun output indicato, si motivi (brevemente e chiaramente) lo scenario in cui l'output è possibile.

Esercizio 4 (5 punti)

Si assuma di avere una classe `C` con due variabili private: `a` e `b`. Quali delle seguenti coppie di metodi possono prevenire problemi di accesso concorrente alla classe `C`?

- A

```
public int read(){return a+b;}
public void set(int a, int b){this.a=a;this.b=b;}
```
- B

```
public synchronized int read(){return a+b;}
public synchronized void set(int a, int b){this.a=a;this.b=b;}
```
- C

```
public int read(){synchronized(a){return a+b;}}
public void set(int a, int b){synchronized(a){this.a=a;this.b=b;}}
```
- D

```
public int read(){synchronized(a){return a+b;}}
public void set(int a, int b){synchronized(b){this.a=a;this.b=b;}}
```
- E

```
public synchronized(this) int read(){return a+b;}
public synchronized(this) void set(int a, int b){this.a=a;this.b=b;}
```
- F

```
public int read(){synchronized(this){return a+b;}}
public void set(int a, int b){synchronized(this){this.a=a;this.b=b;}}
```