

Esame di Programmazione III, Programmazione in Rete e Laboratorio, Ist. di Programmazione in Rete e Laboratorio, a.a. 2022/23 - Appello del 24 gennaio 2023

Esercizio 1 (10 punti)

Si consideri il codice riportato sotto. Si valuti se i thread del programma possono interferire o meno l'uno con l'altro nell'accesso alle variabili condivise e si spieghi il perché. In caso positivo, modificare il codice dell'applicazione per risolvere i problemi, preservando il massimo parallelismo possibile nell'esecuzione delle operazioni.

```
class MyThread extends Thread {
    private Contatore pari;
    private Contatore dispari;
    Random r;

    public MyThread(Contatore o1, Contatore o2) {
        pari = o1; dispari = o2;
        r = new Random();
        start();
    }

    public void run() {
        while (true) {
            int numero = r.nextInt(4);
            if (numero%2==0) {
                pari.add(numero); pari.hello(); }
            else {
                dispari.add(numero); dispari.hello(); }
            System.out.println(Thread.currentThread().getName() +
                               ": " + pari.getVal() + " - " + dispari.getVal());
        }
    }
}

class Contatore {
    private int count = 0;
    public void add(int val) { count = count+val; }
    public int getVal() { return count; }
    public void hello() { System.out.println("hello"); }
}

public class Es1 {
    public static void main(String[] args) {
        Contatore c1Pari = new Contatore();
        Contatore c2Dispari = new Contatore();
        MyThread t1 = new MyThread(c1Pari, c2Dispari);
        MyThread t2 = new MyThread(c1Pari, c2Dispari);
    }
}
```

Esercizio 2 (10 punti)

Si sviluppino i seguenti punti:

- Descrivere in dettaglio il modello di programmazione guidata dagli eventi, spiegando su quali concetti (ed elementi architetturali) si basa e in cosa differisce dalla programmazione sequenziale.
- Descrivere brevemente come tale modello viene implementato nelle GUI Java, facendo un semplice esempio che mostri come si gestiscono gli eventi generati da un bottone (JButton SWING).

Esercizio 3 (10 punti)

Si consideri il codice riportato sotto:

1. Il codice compila? In caso negativo spiegare perché.
2. Scrivere l'output a video del programma originale o, nel caso siano state apportate modifiche al codice, di quello modificato.

```
interface I {
    public void m(A a);
    public void m(B b);
}
class A implements I {
    String tipo = "A";

    public void m(A a) {
        System.out.println("m_AA: " + tipo);
    }
    public void m(B b) {
        System.out.println("m_AB: " + tipo);
    }
}
class B extends A {
    String tipo = "B";

    public void m(A a) {
        super.m(a); System.out.println("m_BA: " + tipo);
    }
}

public class Esercizio3 {
    public static void main(String[] args) {
        A a = new A();
        B b = new B();
        a.m(a); System.out.println();
        a.m(b); System.out.println();
        b.m(a); System.out.println();
        b.m(b); System.out.println();
        I aa = new A();
        aa.m(b); System.out.println();
        A ab = new B();
        ab.m(a); System.out.println();
        ab.m(b); System.out.println(); }
}
```

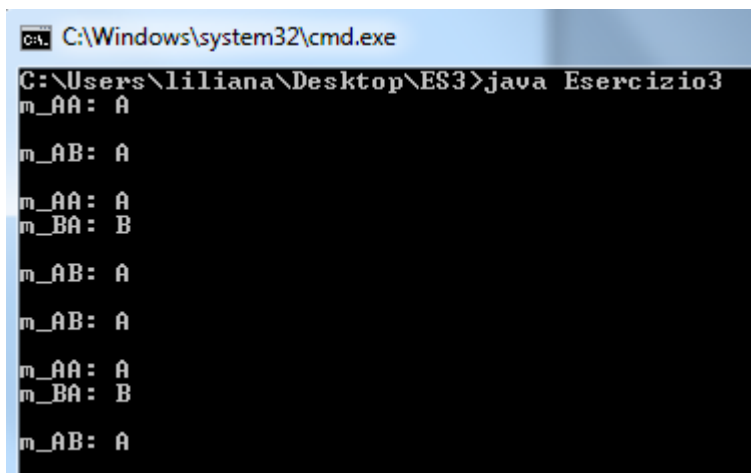
POSSIBILI SOLUZIONI

Esercizio 1

```
class Contatore {  
    private int count = 0;  
    public synchronized void add(int val) {  
        count = count+val;  
    }  
    public synchronized int getVal() {  
        return count;  
    }  
}
```

Il resto del codice dell'applicazione resta invariato. NB: non serve sincronizzare il metodo hello() in quanto non accede alle variabili condivise tra i Thread.

Esercizio 3



```
C:\Windows\system32\cmd.exe  
C:\Users\liliana\Desktop\ES3>java Esercizio3  
m_AA: A  
m_AB: A  
m_AA: A  
m_BA: B  
m_AB: A  
m_AB: A  
m_AA: A  
m_BA: B  
m_AB: A
```