

**Corso di Laurea in Informatica**  
*Programmazione in Rete e Laboratorio*  
Prova scritta del 1-7-2004

**Esercizio 1.**

Sia data la classe **MessageBox**, con i metodi *writeMessage(String s)*, *String readMessage()*, *boolean isEmpty()* e *boolean isFull()*.

Definire una classe **GestoreMessaggi** costituita da un array di N **MessageBox** e da due metodi *scrivi(int i, String s)* e *String leggi(int i)*, che rispettivamente scrivono e leggono un messaggio s nell'i-esimo **MessageBox**. Il metodo *scrivi* può scrivere un messaggio solo se il **MessageBox** relativo non è pieno mentre *leggi* può leggere un messaggio solo se il **MessageBox** non è vuoto.

**Esercizio 2.**

Si illustri la metodologia *observer-observable*, spiegando quale è il suo obiettivo, di quali classi o interfacce fa uso e come queste sono utilizzate.

**Esercizio 3.**

Dato il seguente programma

```
class A {
    metodo1() {
        System.out.println("Metodo 1 in A");
    }
}

class B extends A {
    metodo1() {
        System.out.println("Metodo 1 in B");
    }
    metodo2() {
        System.out.println("Metodo 2 in B");
    }
}

class C extends B {
    metodo2() {
        System.out.println("Metodo 2 in B");
    }
}
```

```

class Prova {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new B();
        B a3 = new A();
        B a4 = new C();
        a1.metodo1();
        a2.metodo2();
        ((B)a2).metodo2();
        a3.metodo1();
        a4.metodo2();
        ((C)a4).metodo1();
        ((C)a1).metodo1();
    }
}

```

Spiegare quali istruzioni del main danno errore in compilazione e quali durante l'esecuzione ma non in compilazione. Dopo aver eliminato queste istruzioni, spiegare che cosa produce in output il programma, giustificando le risposte.

#### ***Esercizio 4.***

Si supponga che il codice di un metodo `metodo` sia composto dalle quattro istruzioni:

```

Istruzione-1;
Istruzione-2;
Istruzione-3;
Istruzione-4;

```

dove ogni Istruzione-X può sollevare una eccezione di tipo Eccezione-X. Scrivere il codice del metodo `metodo` in modo che:

1. le istruzioni Istruzione-1, Istruzione-2 e Istruzione-3 vengano eseguite nella sequenza *se e solo se* nessuna di queste solleva la rispettiva eccezione;
2. se durante l'esecuzione viene sollevata l'eccezione di tipo Eccezione-1 o Eccezione-2 venga eseguita l'istruzione di correzione `Recovery-1` o `Recovery-2`, rispettivamente;
3. dopo l'esecuzione delle istruzioni di correzione `Recovery-1` o `Recovery-2`, l'esecuzione riprende con l'esecuzione dell'istruzione Istruzione-4;
4. se durante l'esecuzione viene sollevata l'eccezione di tipo Eccezione-3 o Eccezione-4 questa deve essere passata al metodo chiamante del metodo `metodo`.