

## Corso di Programmazione in Rete e Laboratorio a.a. 2003/2004 - Esonero del 31 ottobre 2003

1. Dato il seguente programma

```
interface Interf1
{
    void m();
    void m1();
}

interface Interf2
{
    void m2();
}

class C implements Interf1
{
    public void m() {m1();}
    public void m1() {System.out.println("C-m1");}
}

class NC extends C implements Interf2
{
    public void m1() {System.out.println("NC-m1");}
    public void m2() {System.out.println("NC-m2");}
}

class ProvaInterfacce
{
    public static void main(String[] args)
    {
        NC nc = new C();
        NC c = new NC();
        Interf1 i1 = new Interf1();
        Interf1 i11 = c;
        Interf2 i2 = c;
        i11.m();
        c.m();
        i11.m2();
        c.m2();
    }
}
```

spiegare quali istruzioni del main danno errore in compilazione. Dopo aver eliminato queste istruzioni, spiegare che cosa produce in output il programma, giustificando le risposte.

2. Si definisca la classe *Studente* contenente i campi privati *matricola* di tipo intero, *nome* di tipo String, ed i relativi metodi *get* e *set*. Il metodo *setMatricola* deve lanciare una eccezione di tipo *IllegalArgumentException* se gli viene passato un valore negativo. Inoltre la classe *Studente* deve implementare l'interfaccia *java.util.Comparable* e quindi contenere il metodo *public int compareTo(Object ob)* che restituisce un intero negativo, zero o un intero positivo a seconda che il nome dello studente corrente preceda, sia uguale o segua quello dello studente *ob* nell'ordine alfabetico.

3. Dato il seguente programma:

```
public class A {
    public static void p() {
        istruzione-0;
        try {
            istruzione-1;
            istruzione-2;
            istruzione-3;
        }
        catch (EccezioneTipo-2 err2) {
            istruzione-4;
        }
        istruzione-5;
    }

    public static void main(String[] args) {
        istruzione-6;
        try {
            p();
        }
        catch (EccezioneTipo-1 err1) {
            istruzione-7;
        }
        catch (EccezioneTipo-2 err2) {
            istruzione-8;
        }
        istruzione-9;
    }
}
```

sapendo che *istruzione-1*, *istruzione-2* e *istruzione-3* possono sollevare una eccezione di tipo *EccezioneTipo-1*, *EccezioneTipo-2* e *EccezioneTipo-3* rispettivamente, si suppongano le seguenti quattro situazioni che possono occorrere durante l'esecuzione del programma:

- i. non viene sollevata nessuna eccezione;
- ii. viene sollevata una eccezione di tipo *EccezioneTipo-1*;
- iii. viene sollevata una eccezione di tipo *EccezioneTipo-2*;
- iv. viene sollevata una eccezione di tipo *EccezioneTipo-3*.

Si risponda alle seguenti domande:

- a) In quali delle sopra elencate situazioni il programma termina regolarmente?
- b) In quali delle sopra elencate situazioni il programma non termina regolarmente?

Motivare le risposte ed elencare per ogni situazione le istruzioni eseguite.

4. Si supponga che il file “*esempio*” contenga un array di oggetti di tipo *Studente*. Scrivere un main che legge l'array dal file e stampa l'indice del primo studente secondo l'ordine alfabetico.