

# Programmazione

Appello del 25/05/2011

## Esercizio 1 (8 punti)

- a. (4 punti) Scrivere l'output del seguente programma Java

```
public class Main {
public static void main(String[] args) {
    int [] A = {1,55,83,21,13,43,10,56};
    stampa(A);
    enigma(A);
    stampa(A);
}

public static void stampa (int [] A ){
    for (int x:A) {
        System.out.print (x + " ");
    }
    System.out.println();
}

public static void enigma (int A[]){
    if (A.length < 6 ) return;
    int i=1;
    while (A[i]>A[i-1]){
        A[i-1]=A[i]-1;
        i++;
    }
}
}
```

- b. (4 punti) Dire, giustificando la risposta, se la procedura enigma può generare errori a tempo di esecuzione per qualche array A. In caso di risposta affermativa, fornire un array A per cui la procedura enigma genera un errore a tempo di esecuzione.

## Esercizio 2 (8 punti)

Scrivere un metodo di classe **somma** che prende in input due matrici A e B di numeri interi, e restituisce in output la matrice A+B (che deve essere creata dal metodo). Il metodo deve controllare che le matrici A e B siano rettangolari ed abbiano le stesse dimensioni, altrimenti solleva un'opportuna eccezione.

## Esercizio 3 (6 punti)

Scrivere un metodo di classe **search** che prende in input due array A e B di int, e restituisce **true** se e solo se l'array B ha lunghezza minore o uguale dell'array A, ed è possibile individuare un sottoarray (di elementi consecutivi) di A uguale all'array B.

Per esempio, se A={1, 55, 83, 21, 13, 43, 10, 56}; e B={21, 13, 43}; deve essere ritornato **true**.

#### Esercizio 4 (6 punti)

Si considerino le seguenti classi.

```
class A{
    int method(A a){
        return 1;
    }
    int method(B a){
        return 2;
    }
}
```

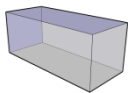
```
class B extends A{
    int method(A a){
        return 3;
    }
    int method(B a){
        return 4;
    }
    int redirect(A a){
        return a.method(this);
    }
}
```

Si dica cosa viene stampato a video dal seguente codice, segnalando eventualmente eventuali errori a tempo di esecuzione:

```
A a = new A();
B b = new B();
A c = new B();
System.out.println(a.method(a));
System.out.println(c.method(a));
System.out.println(c.method(c));
System.out.println(b.redirect(b));
System.out.println(b.redirect(c));
System.out.println(a.redirect(c));
```

#### Esercizio 5 (8 punti)

- Si scriva una classe astratta **Solido** per rappresentare figure solide
- La classe deve avere un metodo **public abstract double volume ();** che sarà implementato dalle classi che estendono **Solido** con lo scopo di calcolare il volume del solido.
- La classe deve avere un campo `pesoSpecifico` che memorizza il peso specifico del solido.
- La classe deve inoltre implementare un metodo **public double peso ();** che ritorna il peso del solido. *[si sfrutti il metodo **volume**]*
- *Si implementino infine le seguenti classi **concrete** che implementano **Solido**, in ognuna delle quali bisogna implementare un **costruttore** e il metodo **volume**:*
  - La classe **Cubo**, il cui costruttore prende un `double (x)` che rappresenta la lunghezza del lato e un `double (ps)` che rappresenta il peso specifico dell'oggetto.



- La classe **Parallelepipedo Rettangolo**, il cui costruttore prende tre `double (x,y,z)` che rappresentano la lunghezza dei lati delle tre dimensioni, e un `double (ps)` che rappresenta il peso specifico dell'oggetto.

#### Attenzione:

- Per svolgere il compito si hanno a disposizione **120** minuti.
- Scrivere **subito** nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- Durante la prova scritta non è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**