

Programmazione

Appello del 20/12/2011 – Compito n° 1

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 = {55,83,21,16,43,10,56};
    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[]){
    int i=0;
    while (i < A.length - 1){
        if (A[i] > A[i+1] && A[i]%2==1) A[i]=A[i+1];
        else A[i+1]=A[i]/2;
        i++;
    }
}
}
```

b. (4 punti) Riscrivere la procedura enigma in modo equivalente senza far uso né del costrutto while né del costrutto for

Esercizio 2 (7 punti)

Scrivere un metodo **public static ArrayList<Integer> estrai (int [] A, int k)** che preso in input un array di numeri interi ed un intero **k**, crea e restituisce un ArrayList di Integer in cui sono presenti tutti e soli gli interi presenti in **A** esattamente **k** volte, nello stesso ordine della loro prima occorrenza in **A**.

Ad esempio, se **A=[3,1,5,3,9,1]** e **k=2**, il metodo deve restituire un arrayList contenente gli **Integer** 3 ed 1.

SECONDA PARTE

Esercizio 3 (5 punti)

Si implementino in Java le classi **Autore** e **Libro**.

La *classe* **Autore** ha i seguenti attributi:

- **nome** (una stringa)
- **cognome** (una stringa)

ed i seguenti metodi:

- *costruttore* che crea un oggetto della *classe* **Autore** assegnando nome e cognome.
- metodi *"get"* per tutti gli attributi, cioè metodi che restituiscono i valori di ciascun attributo;
- metodo *"toString"*

La classe **Libro** ha i seguente attributi:

- **titolo** (una Stringa)
- **autori** (un ArrayList di Autore)

ed i seguenti metodi:

- *costruttore* che crea un oggetto della classe **Libro** senza autori (ma creando opportunamente un ArrayList vuoto)
- metodo **public addAutore (Autore a)**, che aggiunge l'autore a agli autori del libro
- metodo **public String getTitolo ()** che restituisce il titolo del libro.
- metodo **public int getNAutori ()** che restituisce il numero di autori del libro.
- metodo **public Autore getAutore (int k)** che restituisce il k-esimo autore del libro (k varia tra 1 e getNAutori()). Se k non è un intero valido viene restituito null.
- metodo "toString", che sfrutta il metodo toString della classe Libro

Scrivere infine nel main il codice che crea un libro a vostro piacimento e vi aggiunge almeno due autori.

Esercizio 4 (5 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;
    }
}
```

```
class C extends B{
    int method(A a){
        return 5;
    }
    int method(B a){
        return 6;
    }
}
```

Si dica cosa viene stampato a video dal seguente codice, segnalando eventualmente eventuali errori a tempo di compilazione e **giustificando adeguatamente la risposta** indicando anche per ogni invocazione di metodo il metodo selezionato a tempo di compilazione (guardando solo i tipi statici):

```
A a = new A();
B b = new B();
C c = new C();
A d = new B();
A e = new C();
B f = new C();
System.out.println(a.method(f));
System.out.println(d.method(a));
System.out.println(e.method(b));
System.out.println(e.method(c));
System.out.println(f.method(d));
System.out.println(f.method(e));
```

Esercizio 5 (5 punti)

Data la seguente interfaccia:

```
interface Figura{
    double getArea();
    double getPerimetro();
}
```

scrivere le classi **Quadrato** e **TriangoloRettangolo** che implementano l'interfaccia **Figura**.

Ciascuna classe deve prevedere le opportune variabili di istanza, il costruttore e deve naturalmente implementare i metodi dell'interfaccia.

In particolare, la classe **Quadrato** deve avere un costruttore che prende in input la misura del lato, mentre la classe **TriangoloRettangolo** un costruttore che prende in input la misura dei 2 cateti.

[può essere utile utilizzare il metodo `Math.sqrt(double x)` che restituisce la radice quadrata di `x`]

ESERCIZIO FACOLTATIVO:

Esercizio 6 (3 punti)

Aggiungere alla classe **Pila** (implementata qui sotto), un metodo ricorsivo **public int contaPari ()** che restituisce il numero di interi pari presenti nella pila.

```
public class Nodo {
    private int elemento;
    private Nodo next;

    public Nodo
(int elemento, Nodo next) {
        this.elemento=elemento;
        this.next=next;
    }

    public int getElemento() {
        return elemento;
    }

    public Nodo getNext() {
        return next;
    }

    public void setNext(Nodo n) {
        next = n;
    }
}

public class Pila {
    private Nodo topElement;

    public Pila() {
        topElement=null;
    }

    public boolean isEmpty() {
        return topElement==null;
    }

    public void push(int elemento) {
        topElement = new
Nodo(elemento,topElement);
    }

    public int pop() {
        if(isEmpty())return(-1);

        int top = topElement.getElemento();
        topElement = topElement.getNext();
        return top;
    }
}
```

Attenzione:

- Per svolgere il compito si hanno a disposizione **90** 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**