

Programmazione

Appello del 22/05/2012

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 = {10,9,8,7,6,5,4,3,2,1};
    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[]){
    for (int i=A.length -2; i>=0; i--){
        {
            A[i] += A[i+1];
        }
    }
}
}
```

b. (4 punti) Riscrivere il metodo enigma in modo equivalente (ossia in modo che QUALSIASI sia l'array A lo modifichi nello stesso modo) facendo uso esclusivamente del costrutto iterativo **do while**.

Esercizio 2 (8 punti)

Scrivere un metodo **public static boolean monotono (int [] A)** che preso in input un array A di numeri interi, restituisce **true** se e solo se A è ordinato in modo non decrescente oppure non crescente.

Ad esempio se A=[1,2,2,4,5] o se A=[5,4,3,3,2] deve restituire true, mentre se A=[1,2,1,5,6] deve restituire false.

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**

SECONDA PARTE

Esercizio 3 (8 punti)

Si implementino in Java le classi **Prodotto** e **Carrello**.

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

- **codice** (una **stringa**)
- **descrizione** (una **stringa**)
- **prezzoUnitario** (un **intero** che esprima il prezzo in centesimi di euro)

ed i seguenti metodi di istanza:

- **costruttore** che crea un oggetto della classe **Prodotto** assegnando descrizione, codice e prezzo unitario (si assuma che il prezzo unitario sia fornito al costruttore in euro, come un numero in virgola mobile).
- metodi **"get"** per tutti gli attributi, cioè metodi che restituiscono i valori di ciascun attributo (anche in questo caso, il prezzo deve essere restituito in euro, come un numero in virgola mobile) ;
- metodo **public boolean equals (Object o)** che restituisce true se e solo se due prodotti hanno lo stesso codice e lo stesso prezzo unitario (indipendentemente dalla descrizione).

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

- **nome** (una **Stringa**)
- **prodotti** (un **arrayList** di **Prodotto**)
- **quantita** (un **arrayList** di **Integer**)

*Gli **arrayList** **prodotti** e **quantita** sono pensati per essere **arrayList** "paralleli", tali che l'**arrayList** **quantita** in posizione **i** contiene il numero di prodotti presenti nel carrello corrispondenti alla posizione **i** dell'**arrayList** **prodotti**.*

La classe **Carrello** ha i seguenti metodi di istanza:

- **costruttore** che crea un oggetto della classe **Carrello** prendendo in input il nome del carrello
- metodo **public void addProdotto (Prodotto p, int n)**, che (sfruttando il metodo **equals** della classe **prodotto**)
 - aggiunge il prodotto **p** all'**arrayList** **prodotti** (e corrispondentemente la quantità **n** all'**arrayList** **quantita**), se nel carrello non è già presente il prodotto in questione.
 - altrimenti aggiorna la quantità corrispondente al prodotto **p** incrementandola di **n**.
- metodo **public String getNome ()** che restituisce il nome del carrello.
- metodo **public float getTotal ()** che restituisce il prezzo totale in euro dei prodotti presenti nel carrello.

Esercizio 4 (8 punti)

- Si scriva una classe astratta **Funzione** per rappresentare funzioni sul piano cartesiano
- La classe deve avere un metodo astratto **public abstract double calcola (double x)**; che sarà implementato dalle classi che estendono **Funzione** con lo scopo di calcolare il valore della funzione nel punto **x**.
- La classe deve inoltre implementare un metodo **public boolean LEq (Funzione f, double x)** che ritorna **true** se e solo se il valore della funzione su cui è invocato calcolata nel punto **x** è minore o uguale al valore della funzione **f** calcolata sempre in **x** [si sfrutti il metodo **calcola**]
*Ad esempio, se **f** e **g** sono due funzioni, **f.LEq(g,3)** deve ritornare true se e solo se **f(3) ≤ g(3)***
- Si implementino infine le seguenti classi **concrete** che estendono **Funzione**, ognuna con le opportune variabili di istanza, ed in ognuna delle quali bisogna implementare un **costruttore** ed il metodo **calcola**:
 - La classe **IperboleEquilatera**, il cui costruttore prende quattro **double** (**a,b,c,d**) tali che la funzione rappresentata è **f(x)=(ax+b)/(cx+d)**
 - La classe **FunzionePolinomiale**, il cui costruttore prende un **array** **A** di **d+1** **double** (dove **d** è il grado del polinomio) tali che la funzione rappresentata è
f(x)=A[0] + A[1] x+ A[2] x²+ ... + A[i] xⁱ + ... + A[d] x^d