

Programmazione orientata agli oggetti (OOP)

[Home](#)

Esercizi oggetto vettore

Preparazione

Creare una cartella in cui salvare il file di supporto Vettore.java e in cui creare le proprie classi secondo le indicazioni fornite. Aprire il prompt dei comandi posizionandosi sulla cartella appena creata (controllando con il comando `dir` per vedere se la cartella è quella giusta).

Per la compilazione si ricorda di compilare usando il comando `javac *.java` in modo da compilare tutte le classi.

Esercizio 0

Preparare una classe **EsercitazioneVettore** in cui aggiungere i metodi richiesti degli esercizi successivi. Realizzare poi il metodo **main** per effettuare il test del programma.

Nel main creare un oggetto di tipo Vettore, assegnare poi al vettore, mediante il metodo **assegnaVettore(...)** i seguenti valori: {3, 14, 8, 3, 7, 12, 6, 7, 3, 5}. Stampare l'oggetto Vettore così ottenuto.

Esercizio 1

Scrivere il metodo statico **public static int contaOccorrente(Vettore v, int numero)** che restituisce il numero di occorrenze del valore specificato dal parametro **numero**. Le occorrenze di un valore nel vettore sono il numero di volte in cui il valore compare all'interno del vettore. Per esempio nel vettore dell'esercizio 0, ci sono 2 occorrenze del valore 7.

Dopo aver scritto il metodo, modificare il main e provare il metodo utilizzando il vettore dell'esercizio 0 e passando come valore per il parametro **numero** uno per volta i valori 5, 7 e 80.

Esercizio 2

Scrivere il metodo statico **public static int maxOccorrente(Vettore v)** che restituisce il numero contenuto nel vettore che compare con il massimo numero di occorrenze.

Provare il metodo usando il vettore dell'esercizio 0, modificando il main in maniera opportuna.

Esercizio 3

Scrivere il metodo statico **public static boolean uguali(Vettore v1, Vettore v2)** che restituisce true se i due vettori sono uguali (stessa lunghezza e stessi valori), false altrimenti.

Provare il metodo con

- o {1,2,3,4,5} e {1,2,3,5,4}
- o {3, 3, 3} e {3, 3, 3}
- o {1,5,6,7} e {1,5,6,7,8}

Esercizio 4

Scrivere il metodo statico **public static boolean contieneTutto(Vettore v, Vettore v2)** che restituisce true se tutti i valori

di v2 sono contenuti in v.

Provare il metodo con v e v2 pari a:

- o **v**: {1,2,3,5,4}, **v2**: {4,5,1}
- o **v**: {1,2,3,5,4}, **v2**: {4,5,6}
- o **v**: {1,2,3,5,4}, **v2**: {4,5,5}
- o **v**: {4}, **v2**: {4,5,5}

Esercizio 5

Scrivere il metodo statico **public static boolean sottovettore(Vettore v, Vettore v2)** che restituisce true se il vettore v è un sottovettore di v2. Un vettore v è un sottovettore di v2 se esiste un sottoinsieme di celle contigue di v2 che contengono tutti i valori di v, nello stesso ordine con cui sono presenti in v.

Per esempio:

- o il vettore v:{1,2,3} è un sottovettore di v2:{**1,2,3**,4,5};
- o il vettore v:{3,4,5} è un sottovettore di v2:{1,2,**3,4,5**,6,7};
- o il vettore v:{1,2,3} non è un sottovettore di v2:{1,2,2,3,4,5};
- o il vettore v:{4,4,5} è un sottovettore di v2:{1,2,4,5,4,4,5,6}.

Provare il metodo con i vettori forniti come esempio.

Esercizio 6

Scrivere il metodo statico **public static boolean controllaOrdinato(Vettore v)** che restituisce true se il vettore v è ordinato, false altrimenti.

Provare il metodo con i vettori:{1,2,5,86}, {4,5,6,7,6} e {3,5,7,9,9,10}

Esercizio 7

Scrivere il metodo statico **public static void inserisci(Vettore v, int posizione, int valore)** che inserisce il valore del parametro **valore** nella posizione indicata dal parametro **posizione**, spostando gli elementi da posizione a fine vettore di una posizione a destra e rimuovendo l'ultimo valore.

Esempio: dato il vettore {1,2,3,4,5,6,7,8,9} chiamando il metodo sulla posizione 4 con il valore 80 si ottiene il vettore {1,2,3,4,**80**,5,6,7,8} mentre chiamandolo sulla posizione 8 con valore 80 si ottiene {1,2,3,4,5,6,7,8,**80**}

Provare il metodo con i vettori descritti.

Materiale didattico per la lezione

Classe Vettore

(da copiare e salvare nel file Vettore.java)

```
import javax.swing.*;
public class Vettore{
    private int[] vet;
```

```
public Vettore(int dimensione){
    vet=new int[dimensione];
}
public void assegnaVettore(int[] v){
    vet=new int[v.length];
    for(int i=0;i<v.length;i++){
        vet[i]=v[i];
    }
}

public void inizializzaVettore(int n){
    vet=new int[n];
}

public int lunghezza(){
    int l=this.vet.length;
    return l;
}

public void stampa(){
    for(int i=0; i<this.lunghezza(); i++){
        System.out.print(this.vet[i]+" ");
    }
    System.out.println();
}

public int get(int indice){
    return vet[indice];
}

public void set(int indice,int valore){
    vet[indice]=valore;
}

public void leggi(){
    for(int i=0; i<this.lunghezza(); i++){
        this.vet[i]=Integer.parseInt(JOptionPane.showInputDialog("Leggi "+i));
    }
}
}
```