

Programmazione orientata agli oggetti (OOP)

[Home](#)

Esercizi matrici

Esercizio 1

Scrivere un metodo statico `immettiDaTastiera` che riceve come parametro un numero `n`, chiede all'utente di inserire `n` numeri, e restituisce un array con i numeri immessi nell'ordine in cui sono stati inseriti.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare i metodi `main` e `stampa`:

```
public class ProvaImmettiArray {  
    public static void main(String[] x) {  
        int[] v = immettiDaTastiera(4);  
        stampa(v); // dovrebbe stampare quattro numeri inseriti dall'utente  
    }  
    public static void stampa(int[] x) {  
        for (int i=0; i<x.length; ++i) System.out.print(x[i] + " ");  
        System.out.println();  
    }  
}
```

Esercizio 2

Scrivere un metodo statico `toString` che, dato un array di interi, restituisce una stringa con tutti i numeri nell'array separati da virgole. Ad esempio, dato l'array `{1, 2, 3, 4}`, la stringa restituita dovrebbe essere `"1, 2, 3, 4"`.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare il metodo `main`:

```
public class ProvaToString {  
    public static void main(String[] x) {  
        int[] v = { 1, 2, 3, 4, 3, 2, 1 };  
        String s = toString(v);  
        System.out.println(s); // dovrebbe stampare "1, 2, 3, 4, 3, 2, 1"  
    }  
}
```

- Suggerimento: usare un ciclo definito con accumulatore di tipo `String`.

Esercizio 3

Scrivere un metodo statico `toArray` che, data una stringa, restituisce un array di stringhe contenente tutti i numeri

rappresentati nella stringa. Ad esempio data la stringa `123cielo1è12azzurro44o`, l'array restituito dovrebbe essere `{23,1,12,44}`.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare il metodo `main`:

```
import java.util.Arrays;
```

```
public class ProvaToArray {  
    public static void main(String[] x) {  
        String s = "1-2-3-4-lorem23ip567um1";  
        String[] ris = toArray(s);  
        System.out.println(Arrays.toString(ris)); //dovrebbe stampare [1, 2, 3, 4, 23, 567, 1]  
    }  
}
```

- Suggerimento: incapsulate la funzionalità che vi permette di capire se un carattere della stringa è un numero o una lettera all'interno di un metodo `eUnaCifra(String c)` a cui passare una stringa costituita da un unico carattere e che dovrà restituire un valore boolean.

Esercizio 4

Scrivere un metodo statico `equals` che, dati due array di interi, restituisce `true` se sono uguali (cioè hanno la stessa dimensione e celle con lo stesso indice contengono lo stesso valore) e `false` altrimenti.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare il metodo `main`:

```
public class ProvaEquals {  
    public static void main(String[] x) {  
        int[] a1 = { 1, 2, 3, 4, 3, 2, 1 };  
        int[] b1 = { 1, 2, 3, 4, 3, 2, 1 };  
        boolean r1 = equals(a1, b1);  
        System.out.println(r1); // dovrebbe stampare true  
  
        int[] a2 = { 1, 2, 3, 4, 3, 2, 1 };  
        int[] b2 = { 1, 2, 3, 4, 3, 2 };  
        boolean r2 = equals(a2, b2);  
        System.out.println(r2); // dovrebbe stampare false  
  
        int[] a3 = { 1, 2, 3, 4, 3, 2, 1 };  
        int[] b3 = { 1, 2, 3, 9, 3, 2, 1 };  
        boolean r3 = equals(a3, b3);  
        System.out.println(r3); // dovrebbe stampare false  
    }  
}
```

- Suggerimento: usare un ciclo definito con filtro e interruzione.

Esercizio 5

Un array è palindromo se leggendone gli elementi dal primo all'ultimo o dall'ultimo al primo si ottiene lo stesso risultato.

Scrivere un metodo statico palindromo che, dato un array di interi, restituisce true se è palindromo e false altrimenti.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare il metodo main:

```
public class ProvaPalindromo {
    public static void main(String[] x) {
        int[] v1 = { 1, 2, 3, 4, 3, 2, 1 };
        int[] v2 = { 7, 2, 9, 4, 0, 2 };
        int[] v3 = { 971 };
        int[] v4 = { 5, 5 };
        int[] v5 = { 5, 1 };
        boolean r1 = palindromo(v1);
        boolean r2 = palindromo(v2);
        boolean r3 = palindromo(v3);
        boolean r4 = palindromo(v4);
        boolean r5 = palindromo(v5);
        System.out.println(r1); // dovrebbe stampare true
        System.out.println(r2); // dovrebbe stampare false
        System.out.println(r3); // dovrebbe stampare true
        System.out.println(r4); // dovrebbe stampare true
        System.out.println(r5); // dovrebbe stampare false
    }
}
```

Esercizio 6

Scrivere un metodo rovescia che, dato un array di interi, ne rovescia il contenuto.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare i metodi main e stampa:

```
public class ProvaRovescia {
    public static void main(String[] x) {
        int[] v = { 1, 2, 3, 4, 5 };
        rovescia(v);
        stampa(v); // dovrebbe stampare: "5 4 3 2 1"
    }
    public static void stampa(int[] x) {
        for (int i=0; i<x.length; ++i) System.out.print(x[i] + " ");
        System.out.println();
    }
}
```

- Suggerimento: usare un ciclo definito che scambia il primo con l'ultimo, il secondo con il penultimo, ecc.

Esercizio 7

Scrivere un metodo creaRovesciato che, dato un array, **restituisce** un nuovo array di interi ottenuto da quello di partenza rovesciandolo. L'array di partenza non deve essere modificato.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare i metodi main e stampa:

```
public class ProvaRovescia2 {  
    public static void main(String[] x) {  
        int[] v = { 1, 2, 3, 4, 5 };  
        int[] r = creaRovesciato(v);  
        stampa(v); // dovrebbe stampare: "1 2 3 4 5"  
        stampa(r); // dovrebbe stampare: "5 4 3 2 1"  
    }  
    public static void stampa(int[] x) {  
        for (int i=0; i<x.length; ++i) System.out.print(x[i] + " ");  
        System.out.println();  
    }  
}
```

Esercizio 8

Scrivere un metodo minimo che, dato un array di interi, restituisce il minimo dell'array. Assumere che l'array abbia dimensione almeno 1.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare il metodo main:

```
public class ProvaMinimo {  
    public static void main(String[] x) {  
        int[] v = { 1, -1, 2, 3, -4, 5, 0 };  
        int min = minimo(v);  
        System.out.println(min); // dovrebbe stampare: -4  
    }  
}
```

Esercizio 9

Scrivere un metodo ampiezza, che dato un array di interi, restituisce la differenza tra il massimo e il minimo dell'array. Assumere che l'array abbia dimensione almeno 1.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare il metodo main:

```
public class ProvaAmpiezza {  
    public static void main(String[] x) {  
        int[] v = { 1, -1, 2, 3, -4, 5, 0 };  
        int a = ampiezza(v);  
        System.out.println(a); // dovrebbe stampare: 9  
    }  
}
```

Esercizio 10

Scrivere un metodo minmax che, dato un array di interi, restituisce il minimo ed il massimo dell'array. Assumere che l'array abbia dimensione almeno 1.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare il metodo main:

```
public class ProvaMinimo {  
    public static void main(String[] x) {  
        int[] v = { 1, -1, 2, 3, -4, 5, 0 };  
        int[] mm = minmax(v);  
        System.out.println("Valore minimo: " + mm[0]); // dovrebbe stampare: -4  
        System.out.println("Valore massimo: " + mm[1]); // dovrebbe stampare: 5  
    }  
}
```

- Nota: minmax(v) restituisce **due** valori dello **stesso tipo**.

Esercizio 11

Scrivere un metodo filtraPositivi che, dato un array di interi, restituisce un nuovo array ottenuto da quello di partenza rimuovendo tutti i numeri negativi o zero. Ad esempio, dato in ingresso l'array { -7, 2, 4, -6, 9, -3, 2, -3, 5, -3, -2, 0 } di dimensione 12, il risultato deve essere l'array { 2, 4, 9, 2, 5 } di dimensione 5.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare i metodi main e stampa:

```
public class ProvaFiltro {  
    public static void main(String[] x) {  
        int[] v1 = { -7, 2, 4, -6, 9, -3, 2, -3, 5, -3, -2, 0 };  
        int[] r1 = filtraPositivi(v1);  
        stampa(v1); // dovrebbe stampare: "-7 2 4 -6 9 -3 2 -3 5 -3 -2 0"  
        stampa(r1); // dovrebbe stampare: "2 4 9 2 5"  
  
        int[] v2 = { -7, -6, 0, -2 };  
        int[] r2 = filtraPositivi(v2);  
        stampa(v2); // dovrebbe stampare: "-7 -6 0 -2"  
        stampa(r2); // dovrebbe stampare: ""  
    }  
    public static void stampa(int[] x) {  
        for (int i=0; i<x.length; ++i) System.out.print(x[i] + " ");  
        System.out.println();  
    }  
}
```

Esercizio 12

Scrivere un metodo statico `sottoArray` che, dato un array di interi `v` e due indici `i` e `j` con $i < j$, restituisce un nuovo array di dimensione `j-i` che contiene tutti gli elementi dell'array `v` compresi tra le celle `i` e `j-1` (incluse).

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare i metodi `main` e `stampa`:

```
public class ProvaSottoArray {  
    public static void main(String[] x) {  
        int[] v = { 4, 2, 3, 4, 5, 5, 7, 8, 7 };  
        int[] r = sottoArray(v, 2, 6);  
        stampa(v); // dovrebbe stampare: "4 2 3 4 5 5 7 8 7"  
        stampa(r); // dovrebbe stampare: "3 4 5 5"  
    }  
    public static void stampa(int[] x) {  
        for (int i=0; i<x.length; ++i) System.out.print(x[i] + " ");  
        System.out.println();  
    }  
}
```

- Nota: concettualmente analogo a `substring`, ma su array invece che su stringhe.

Esercizio 13

Scrivere un metodo `contaParole` che, data una stringa, restituisce il numero di parole separate da spazi contenute nella stringa.

Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma senza dover modificare il metodo `main`:

```
public class ProvaContaParole {  
    public static void main(String[] x) {  
        int r1 = contaParole("un cavallo blu");  
        System.out.println(r1); // dovrebbe stampare 3  
        int r2 = contaParole("to be or not to be, that is the question ");  
        System.out.println(r2); // dovrebbe stampare 10  
        int r3 = contaParole(" Ipse autem uni elephanto, qui solus superfuerat, supersedens, vix difficultatem itineris evasit");  
        System.out.println(r3); // dovrebbe stampare 12  
    }  
}
```