

Programmazione orientata agli oggetti (OOP)

[Home](#)

Esercizi oggetti

Esercizio 1

Si consideri la seguente classe:

```
public class Persona {  
    public String nome;  
    public int eta;  
}
```

Scrivere un metodo statico creaDaTastiera che chiede all'utente di inserire nome ed età di una persona, e restituisce un nuovo oggetto con quel nome e quell'età.

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

```
public class ProvaCreaDaTastiera {  
    public static void main(String[] x) {  
        Persona p = creaDaTastiera();  
        System.out.println("La persona inserita è: " + p.nome + " che ha " + p.eta + " anno/i.");  
    }  
}
```

Esercizio 2

Si consideri la seguente classe:

```
public class Persona {  
    public String nome;  
    public int eta;  
}
```

Scrivere un metodo statico equals che dati due (riferimenti a) oggetti Persona, restituisce true se sono uguali (cioè hanno lo stesso nome e la stessa età), 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) {  
  
        Persona p1 = new Persona();  
        p1.nome = "Luca";
```

```
p1.eta = 19;

Persona p2 = new Persona();
p2.nome = "Luca";
p2.eta = 20;

Persona p3 = new Persona();
p3.nome = new String("Luca");
p3.eta = 19;

boolean test1 = equals(p1, p2);
boolean test2 = equals(p1, p3);
boolean test3 = equals(p2, p3);

System.out.println(test1); // deve stampare false
System.out.println(test2); // deve stampare true
System.out.println(test3); // deve stampare false
}
}
```

Esercizio 3

Si consideri la seguente classe:

```
public class Punto {
    public double x, y;
}
```

i cui oggetti rappresentano punti nel piano. Si aggiungano alla classe Punto i seguenti metodi (in questo modo la classe fungerà sia da descrittore della struttura degli oggetti punto che da contenitore di metodi):

- `public static boolean equals(Punto a, Punto b)` che dati due (riferimenti a) punti a e b, restituisce true se i due oggetti sono uguali, e false altrimenti.
- `public static String toString(Punto a)` che, dato un (riferimento a) punto, restituisce una stringa contenente le coordinate di quel punto. Ad esempio, per un oggetto con coordinata x pari a 2.0 e coordinata y pari a -8.3, la stringa prodotta deve essere "(2.0,-8.3)".
- `public static boolean minore(Punto a, Punto b)` che dati due (riferimenti a) punti a e b, restituisce true se entrambe le coordinate di a sono minori o uguali alle corrispondenti coordinate di b, e false altrimenti.
- `public static double distanza(Punto a, Punto b)` che dati due (riferimenti a) punti a e b, calcola la distanza fra i due punti. Si ricordi che la distanza fra due punti (x1,y1) e (x2,y2) si ottiene come la radice quadrata di $(x1-x2)^2 + (y1-y2)^2$. Per calcolare la radice quadrata di un valore double v si usi l'invocazione della funzione `Math.sqrt(v)`.

Scrivere una classe `ProvaPunto` contenente un metodo `main` di prova per verificare il funzionamento dei metodi scritti. Il metodo deve creare degli oggetti `Punto`, assegnarne le coordinate, e poi invocare i metodi della classe `Punto` verificando che il risultato sia conforme a quanto ci si aspetta.

Esercizio 4

Si aggiunga alla classe `Punto` sviluppata nell'esercizio 3 un metodo `public static Punto piuVicinoOrigine(Punto[] v)` che, dato un array v di riferimenti a oggetti `Punto`, restituisce il punto nell'array avente la distanza minima dall'origine degli assi,

oppure null se l'array v non contiene alcun punto.

Modificare il main della classe ProvaPunto dell'esercizio 3 per testare il funzionamento del metodo scritto.

Esercizio 5

Si aggiunga alla classe Punto sviluppata nell'esercizio 3 un metodo public static String[] converti(Punto[] v) che, dato un array v di riferimenti a oggetti Punto, restituisce un array di stringhe che rappresentano le coordinate dei punti (per calcolare la stringa corrispondente a un punto, usare il metodo toString sviluppato nell'esercizio 3).

Modificare il main della classe ProvaPunto dell'esercizio 3 per testare il funzionamento del metodo scritto.

Realizzato con **Drupal**