



# costruttori e distruttori

Java

```
public class Punto {  
  
    private double x;           //ascissa del punto  
    private double y;           //ordinata del punto  
  
    public Punto() {           // Costruttore senza parametri  
        System.out.println ("E' stato istanziato un punto ");  
    }  
  
    public void finalize(){     // distruttore  
        System.out.println ("E' stato distrutto un punto ");  
    }  
  
}
```

- il **costruttore** è un metodo particolare che viene invocato alla **creazione** dell'oggetto
  - contiene tutte le istruzioni da eseguire per la sua **inizializzazione**
- in **Java** il costruttore **deve** avere lo **stesso nome** della **classe** e non deve essere specificato nessun tipo di ritorno per il metodo
- es.

```
public Punto() {    // Costruttore senza parametri
    System.out.println ("E' stato istanziato un punto ");
}
```

- il ***costruttore di copia*** consente di istanziare un nuovo oggetto come copia di un ***oggetto*** della stessa classe creato in precedenza e fornito come ***parametro***
- il codice di un costruttore di copia assegna agli attributi dell'oggetto in fase di creazione il valore degli ***omonimi attributi*** dell'oggetto passato come parametro

```
public class Punto {  
  
    private double x; //ascissa del punto  
    private double y; //ordinata del punto  
  
    public Punto(Punto p) { // Costruttore di copia  
        this.x = p.x;  
        this.y = p.y;  
    }  
  
    ...  
  
}
```

- il ***distruzione*** è un particolare metodo che viene invocato ***automaticamente*** quando l'oggetto viene ***distru***to (*non ha più riferimenti attivi*)
- è stato concepito per ***liberare*** le ***risorse*** utilizzate dall'oggetto, definite dalla sua classe e create dal corrispondente costruttore ed eventualmente le altre risorse allocate dinamicamente durante l'esecuzione dei metodi dell'oggetto stesso

- in Java il distruttore si implementa mediante il metodo *finalize*
- es.

```
public void finalize(){  
    System.out.println ("E' stato distrutto un punto ");  
}
```
- il distruttore viene *automaticamente attivato* dal Garbage collector

- per **garbage collection** (*letteralmente raccolta dei rifiuti*) si intende una modalità automatica di gestione della memoria, mediante la quale un sistema operativo, o un compilatore e un modulo di run-time, **liberano le porzioni di memoria** che **non** dovranno più essere successivamente **utilizzate** dalle applicazioni

*Wikipedia*



```
Punto a;  
a = new Punto();    // viene eseguito costruttore  
a = null;  
System.gc();        // viene attivato Garbage collector  
// in realtà è un suggerimento alla JVM ...  
// il Garbage collector esegue il distruttore
```

