

# Costruttori e Distruttori

In Java

## Classe di esempio

```
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(){
        System.out.println ("E' stato distrutto un punto ");
    }
}
```

## Il costruttore

- Il costruttore è un metodo particolare che viene invocato alla creazione dell'oggetto e che 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 ");
}
```

## Costruttori Java

- Possono avere parametri di input che serviranno per effettuare le operazioni di inizializzazione alla creazione dell'oggetto.
- Possono esistere più costruttori con lo stesso nome, ma con numero e tipo di parametri differenti. In questo modo, sarà possibile creare l'oggetto invocando uno dei costruttori all'atto della creazione e passargli i parametri necessari nell'ordine e in numero uguale a quanto definito nella dichiarazione.
- Possono anche essere vuoti o non essere definiti. In questi casi, sull'oggetto creato non sarà effettuata alcuna operazione di inizializzazione. In particolare, se non è definito viene utilizzato il costruttore di default, assegnato automaticamente dalla JVM.

## Il distruttore

- Nella programmazione orientata agli oggetti, un distruttore è un particolare metodo che viene invocato automaticamente quando l'oggetto viene distrutto.
- 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.

## Il distruttore in Java

- 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 attivato dal Garbage collector

## 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*

## Esempio

```
Punto a;
a = new Punto(); // Eseguito costruttore
a = null;
System.gc();    // Attivato Garbage collector

// Il Garbage collector esegue il distruttore
```