

Object Oriented Programming - OOP

📅 Thu, 10 Mar

Oggetti

Un **oggetto** è una *black box* che incapsula **dati** e **metodi**.

Un oggetto è *caratterizzato* da:

▼ **Stato**

Ogni oggetto ha il suo *stato*, dato dall'insieme dei **valori** dei suoi **attributi**.

▼ **Comportamento**

Il *comportamento* di un oggetto determina come tale **interagisce** con il resto del *sistema*, è dato dai suoi **metodi**.

▼ **Identità**

Due oggetti, anche se si trovano nello stesso *stato*, mantengono la propria **identità distinta**.

Information Hiding

Tramite il principio dell'**information hiding**, si può *nascondere* la *struttura* dell'oggetto dal resto del *sistema*. È dunque possibile *distinguere* una parte di *informazioni* **pubblica** e una parte **privata**.

I *metodi* pubblici sono chiamati **comandi**.

OOP in Java

Per definire una classe in Java, si usa la seguente sintassi:

```
mod. visibilità [altri mod.] class NomeClasse {  
  
    attributi di istanza  
    attributi di classe  
  
    costruttori  
  
    metodi di istanza  
    metodi di classe  
}
```

Modificatori di visibilità

Il **modificatore di visibilità** può essere di quattro tipi, in ordine di protezione:

▼ **public**

Gli elementi *public* [+] sono visibili a **tutto** il *sistema*.

▼ **protected**

Gli elementi *protected* [~] sono visibili alla **classe** a cui *appartengono* ed alle sue **classi figlie**.

▼ **private**

Gli elementi *private* [-] sono visibili **solo** alla **classe** a cui *appartengono*.

La modalità di visibilità di **default** [#] permette a *tutte le classi* del **package** di *accedere alle informazioni*.

| Altri modificatori

Esistono *ulteriori modificatori* che attribuiscono alle classi *particolari proprietà*:

▼ `final`

Una classe *final* non può essere specializzata oltre, ovvero **non** può essere usata come **superclasse**.

▼ `abstract`

Una classe *abstract* **non** permette di **istanziare** un oggetto di tale classe.

È possibile *attribuire* ad una classe **più di uno** di questi *modificatori*.

| Attributi

Gli **attributi** di una classe possono essere di *due tipi*:

▼ **Istanza**

Un *attributo d'istanza* **cambia** per tutte le *istanze* di tale classe.

▼ **Classe**

Un *attributo di classe* è **condiviso** da tutte le *istanze* di tale classe.

| Costruttori

I **costruttori** sono delle *particolari operazioni* che servono ad **istanziare** gli *oggetti* di una *classe*. Può esistere **più di un costruttore** per *classe*.

| Metodi

I **metodi** di una *classe* possono essere di *due tipi*:

▼ **Istanza**

Un *metodo d'istanza*, di norma, lavora sugli **attributi di istanza**.

▼ **Classe**

Un *metodo di classe* può essere **invocato** anche **senza l'istanziamento** della *classe* a cui appartiene.