

Modifieri

In questa lezione definiremo i **modifieri del linguaggio JAVA**, dove per modificatore si intendono le **parole chiave capaci di cambiare il significato di un componente di un'applicazione Java**.

Si possono anteporre alla dichiarazione di un componente di un'applicazione Java **anche più modifieri alla volta**, senza tener conto dell'ordine in cui vengono anteposti.

I **modificatori di accesso** regolano essenzialmente la visibilità e l'accesso ad un componente Java. Vediamone alcuni nel seguito:

public

Può essere utilizzato sia relativamente ad un membro (**attributo o metodo**) che ad una **classe stessa**

- • un **membro dichiarato pubblico** sarà accessibile da una qualsiasi classe situata in qualsiasi package
- • una **classe dichiarata pubblica** sarà anch'essa visibile da un qualsiasi package

protected

Può essere utilizzato relativamente ad un membro (**attributo o metodo**) di una **classe**

- Un membro protetto sarà **accessibile all'interno dello stesso package** ed in tutte le sottoclassi della classe in cui è definito, anche se non appartenenti allo stesso package.

default

- Un membro di una classe **per default** sarà accessibile solo dalle **classi appartenenti al package** dove è definito.
- Una classe appartenente ad un package definita senza il modificatore **public** sarà visibile solo dalle classi appartenenti allo stesso package.

private

Può essere utilizzato solo con i membri (**attributo o metodo**) di una **classe**, ed un membro privato sarà **accessibile** solo all'interno della classe.

Modificatore	Stessa Classe	Stesso Package	Sottoclasse	Dappertutto
public	sì	sì	sì	sì
protected	sì	sì	sì	no
(default)	sì	sì	no	no
private	sì	no	no	no

final

Questo **modificatore** è applicabile alle variabili, ai metodi e alle classi.

- • una **variabile** dichiarata **final** diviene una costante
- • un **metodo** dichiarato **final** non può essere riscritto in una sottoclasse (non è possibile applicare l'**override**)
- • una **classe** dichiarata **final** non può essere estesa

Il modificatore **final** si può utilizzare anche per **variabili e parametri locali**, i valori di tali variabili non sono modificabili localmente.

static

Un membro **statico** è **condiviso da tutte le istanze della classe**, quindi un membro **statico** può essere utilizzato anche senza **istanziare** la classe. Il suo utilizzo provocherà il caricamento in memoria della classe contenente il membro in questione.

Un **esempio di metodo statico** è il metodo **sqrt()** della classe **Math**, che viene chiamato tramite la sintassi: **Math.sqrt(numero);** **Math** è il nome della classe e non il nome di un'istanza di essa.

Una **variabile statica**, essendo condivisa da tutte le istanze della classe, assumerà lo stesso valore per ogni oggetto di una classe.

```
public class ClasseDiEsempio
{public static int a = 0;}
public class ClasseDiEsempioPrincipale
{public static void main (String args[])
{System.out.println("a = "+ClasseDiEsempio.a);
ClasseDiEsempio ogg1 = new ClasseDiEsempio();
ClasseDiEsempio ogg2 = new ClasseDiEsempio();
ogg1.a = 10;
System.out.println("ogg1.a = " + ogg1.a+ " ogg2.a = " + ogg2.a);
ogg1.a = 20;
System.out.println("ogg1.a = " + ogg1.a+" ogg2.a =" + ogg2.a);} }
```

L'output di questo semplice programma sarà:

```
a=0  
ogg1.a = 10 ogg2.a = 10  
ogg1.a = 20 ogg2.a = 20
```

Quindi se un'istanza modifica la variabile statica, essa risulterà modificata anche relativamente all'altra istanza, essa è condivisa dalle due istanze ed in realtà risiede nella classe.

Il modificatore **static** quindi prescinde dal concetto di **oggetto** e lega strettamente le variabili al concetto di **classe**, per questo a volte le variabili statiche sono definite come “**variabili di classe**”.

Una variabile dichiarata **static** e **public** può considerarsi una sorta di variabile **globale** perciò queste vanno evitate. Viceversa è utile a volte utilizzare costanti globali, definite con **public**, **static** e **final**. Nella classe **Math** sono definite due variabili **public**, **static** e **final**: **PI** ed **E**.

abstract

Questo **modificatore** è applicabile ai metodi e alle classi, sappiamo già che una **classe astratta** non può essere istanziata, ma nulla vieta però ad una classe astratta di implementare tutti suoi metodi.