

Dichiarazione e utilizzo di un metodo static

I metodi dichiarati nelle classi si possono distinguere in due tipologie:

- metodi di istanza
- metodi di classe (o **static**).

I **metodi di istanza** sono utilizzabili dalle istanze della classe. In pratica dalla classe si crea un oggetto e, utilizzando l'oggetto, si invoca l'esecuzione dei metodi di istanza. I metodi di istanza possono fare riferimento agli attributi dell'oggetto, leggendo o modificando il loro valore.

I **metodi di classe** esistono e possono essere utilizzati anche senza che venga creato un oggetto della classe che li contiene. Questi metodi non possono fare riferimento agli attributi definiti nella classe, a meno che anche questi siano statici.

I metodi di classe devono essere dichiarati inserendo la parola chiave *static* nell'intestazione del metodo.

L'esecuzione di un metodo statico da altre classi si attiva indicando il nome della classe seguito dal nome del metodo. Per esempio, il calcolo della radice quadrata di un numero (metodo *sqrt* della classe *Math*), si può eseguire scrivendo il comando *Math.sqrt(numero)*.

Per apprezzare meglio le differenze tra i metodi di istanza e di classe definiamo la classe *Operazione* che contiene un metodo *somma*.

```
class Operazione
{
    private int a, b;

    public Operazione(int a, int b)
    {
        this.a = a;
        this.b = b;
    }

    public int somma()
    {
        return a + b;
    }
}
```

```
class Operazione
{
    public static int somma(int a, int b)
    {
        return a + b;
    }
}
```

Il codice a sinistra definisce un metodo di istanza e due attributi: per la sua esecuzione è richiesta la creazione di una istanza e l'esecuzione del metodo d'istanza.

Il codice a destra definisce un metodo di classe con due parametri: per la sua esecuzione si usa direttamente il nome della classe seguito dal nome del metodo e dai parametri. Si noti che non è stato definito il costruttore in quanto la classe non verrà istanziata.

Per eseguire la somma di due numeri, si deve usare la classe *Operazione* nel seguente modo.

```
int risultato;

Operazione op = new Operazione(2,5);
risultato = op.somma();
```

```
int risultato;

risultato = Operazione.somma(2,5);
```

In generale, i vantaggi di un metodo statico, rispetto ad uno di istanza, sono:

- non si deve istanziare la classe per utilizzarlo;
- si risparmia memoria perché non si creano oggetti;
- non è necessario conoscere i dettagli della classe: il metodo usa solo i parametri e nessun attributo.

Solitamente i metodi statici vengono raggruppati in classi di *Utility*, accessibili in modo generico. Per esempio, si potrebbero raggruppare in una classe i metodi statici corrispondenti alle funzioni matematiche per calcolare i volumi di figure note.

Tra le classe di Utility di Java ci sono **Math**, per le funzioni matematiche, e la classe **System.out**, per le operazioni di stampa su standard output.

ESEMPIO

Dichiarare un metodo statico che confronta due array di numeri interi e restituisce il valore *true* se tutti gli elementi degli array sono uguali.

Per verificare l'uguaglianza, il metodo statico inizia confrontando le dimensioni dei due array. Se hanno la stessa dimensione prosegue con la verifica dei singoli elementi. Solo se tutti gli elementi sono uguali esegue l'ultima riga di codice e restituisce il valore *true*. Il codice completo della classe è riportato di seguito.

IMPLEMENTAZIONE DELLA CLASSE (*Utility.java*)

```
class Utility
{
    public static boolean confronta(int a[], int b[])
    {
        // Confronta la dimensione
        if (a.length != b.length)
            return false;

        for(int i=0; i<a.length; i++)
        {
            // Confronta i singoli valori
            if (a[i] != b[i])
                return false;
        }

        return true;
    }
}
```

PROGRAMMA JAVA (*ProgUtility.java*)

```
class ProgUtility
{
    public static void main(String args[])
    {
        int arr1[] = {4, 6, 7};
        int arr2[] = {3, 6, 7};

        if ( Utility.confronta(arr1,arr2) )
            System.out.println("Gli array sono uguali.");
        else
            System.out.println("Gli array sono diversi.");
    }
}
```