

Soluzione di sottoproblemi

Spesso i problemi che incontriamo nella realtà sono complessi. A livello dell'analisi e del progetto, è conveniente affrontare la soluzione utilizzando la tecnica di progettazione top-down, che suggerisce di suddividere il problema complesso in un insieme di sottoproblemi autonomi tra loro. All'analisi top-down segue la successiva fase di sviluppo bottom-up in cui ogni sottoproblema è progettato in modo indipendente. Questa impostazione si riflette nella creazione:

1. di un insieme di sottoprogetti, che contengono il modello dei dati e il corrispondente processo risolutivo di ogni singolo sottoproblema (sottoalgoritmo);
2. dall'algoritmo principale (*main*), il cui compito è quello di richiamare i diversi sottoalgoritmi.

In Java, a livello della realizzazione, i sottoprogetti sono codificati mediante metodi richiamati dal metodo *main()* di un'applicazione che implementa l'algoritmo principale. Nel seguito seguiremo questo modello tipico del paradigma procedurale, con l'obiettivo di imparare a costruire metodi che realizzano elaborazioni autonome, progettate con un proprio modello dei dati e sottoalgoritmo. Queste conoscenze saranno però utilizzate nel prossimo modulo per poter definire le elaborazioni sugli oggetti, mediante metodi da inserire (incapsulare) nel corpo della dichiarazione della classe, che definisce il comportamento degli oggetti stessi.

I metodi

Il metodo (method) è l'unico costrutto del Java, a disposizione dei programmatori, che permette di codificare una parte dell'elaborazione (modello dei dati + sottoalgoritmo) svolta da una classe.

In una classe, un metodo realizza una particolare elaborazione, definita dal programmatore, in cui possiamo riconoscere:

1. un input, definito da un insieme di variabili di ingresso, denominate parametri;
2. una elaborazione, che traduce il modello dei dati e il sottoalgoritmo;
3. un output, ovvero il risultato dell'elaborazione fornito dal metodo stesso.

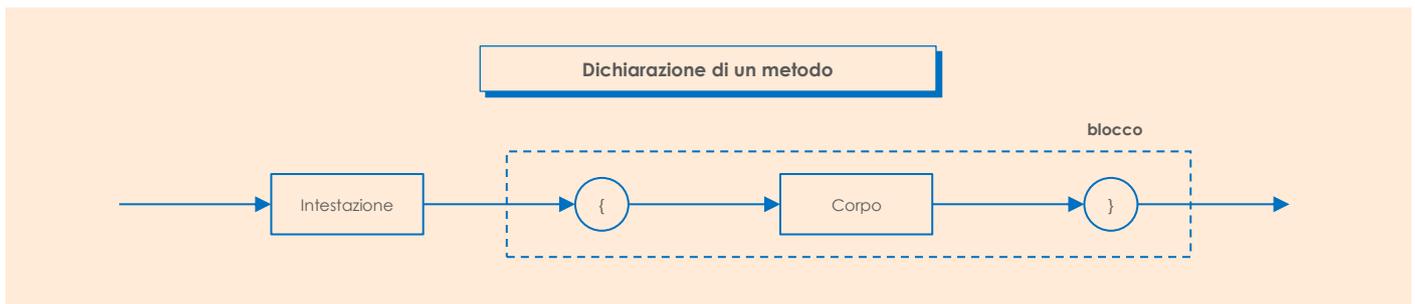
Un metodo si comporta come una funzione (*function*) ovvero un particolare sottoprogramma (in altri linguaggi, *procedure* o *subroutine*) e sé stante che restituisce un solo valore. In un sottoprogramma, i dati di ingresso devono essere assegnati, come studieremo successivamente, al momento del suo richiamo. Vedremo che in Java, per realizzare alcuni compiti, i parametri e/o il risultato fornito dal metodo possono mancare.

Dichiarazione di un metodo

Un metodo Java è formato da:

1. una intestazione,
2. un corpo,

e si dichiara sulla base del diagramma sintattico riportato di seguito.



L'intestazione si costruisce sulla base del grafo sintattico seguente.

