

Le specifiche di rappresentazione in UML

Il linguaggio **UML** (*Unified Modeling Language*) è un insieme di regole riconosciute a livello internazionale per specificare, rappresentare, progettare e realizzare software; in particolare, nel contesto della programmazione ad oggetti, per descrivere le classi e gli oggetti.

Le componenti e il comportamento di un sistema software sono rappresentati con il linguaggio UML tramite l'utilizzo di diversi diagrammi. Ogni diagramma ha le sue specifiche di rappresentazione, formate dai simboli grafici e dalle regole di utilizzo.

I **diagrammi** di UML sono raggruppati in tre categorie, in base a quali aspetti del sistema descrivono.

- I *diagrammi di struttura* descrivono le entità del sistema (diagrammi delle classi, degli oggetti, delle componenti e dei package).
- I *diagrammi di comportamento* descrivono gli attori che usano il sistema e le azioni che svolgono (diagrammi dei casi d'uso, delle attività e della macchina a stati).
- I *diagrammi di interazione* descrivono le interazioni tra le componenti del sistema (diagrammi di sequenza e di comunicazione).

Il **diagramma delle classi** (*class diagram*) fa parte dei *diagrammi di struttura* ed è usato per rappresentare graficamente le classi, con i loro attributi e metodi, e le relazioni tra di esse.

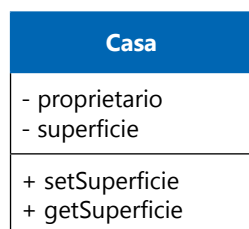
Il simbolo utilizzato per rappresentare una classe è un rettangolo suddiviso in tre zone, contenenti il nome della classe, in alto, l'elenco degli attributi, in centro, e l'elenco dei metodi, in basso.

Il livello di visibilità degli elementi della classe può essere espresso posizionando uno dei seguenti simboli prima del nome dell'attributo o del metodo:

- + *public*
- *private*
- # *protected*

Un attributo o un metodo statico sono evidenziati con la sottolineatura del relativo nome.

Il seguente diagramma mostra una classe con due attributi privati e due metodi pubblici.



Le relazioni tra le classi, messe in evidenza con il diagramma delle classi, sono:

- *generalizzazione o ereditarietà*, una classe (sopraclasse) è una generalizzazione di un'altra classe (sottoclasse), che ne eredita attributi e metodi;
- *implementazione*, una classe (in Java, un'interfaccia) descrive i metodi astratti che saranno implementati da una classe concreta;
- *associazione*, una classe contiene un riferimento ad un'altra classe;
- *composizione*, è una relazione di associazione in cui il riferimento indica che una classe (contenitore) è composta da altre (componenti), se il contenitore viene eliminato anche le componenti vengono eliminate;
- *aggregazione*, è una relazione di associazione in cui il riferimento indica che una classe (contenitore) ne aggrega altre (componenti), se il contenitore viene eliminato le componenti continuano a esistere;
- *dipendenza*, una classe dipende da un'altra se le modifiche ad una impongono una modifica all'altra.

Le relazioni tra le classi sono rappresentate in modo grafico con linee di diverso tipo come mostra la seguente tabella:

Relazione	Tipo di linea
Generalizzazione/ereditarietà	
Implementazione	
Associazione	
Composizione	
Aggregazione	
Dipendenza	

Il seguente diagramma mostra alcune relazioni tra le classi, in cui la classe *Casa* è una generalizzazione delle classi *Villa* e *Appartamento*, la classe *Casa* è associata alla classe *Persona* tramite il riferimento *proprietario*.

