

Diagramma UML delle classi


(Corrisponde al box (subject) dei diagrammi dei casi d'uso. Tutto ciò che non deve essere modellato sta fuori))

I diagrammi delle classi vanno spesso affiancati a diagrammi comportamentali.

Strutturale. Mostra una vista statica del modello: l'organizzazione del progetto del sistema. Non mostra informazioni temporali. Gli elementi corrispondono a concetti fondamentali del paradigma OO. (classi, oggetti, ereditarietà). Quindi:
> Si concentrano solo sulla struttura (classi, attributi, relazioni) e trascurano il comportamento del sistema.

Rappresenta: - I tipi di oggetti (classi) del sistema che corrispondono alle entità che esistono nel sistema. - Le relazioni *statiche* tra classi e i vincoli che si applicano a tali relazioni. - Le caratteristiche delle classi: proprietà (attributi) e responsabilità (metodi)

Sintassi

CLASSE Descrittore di un insieme di oggetti con gli stessi attributi (proprietà), operazioni (comportamenti) e relazioni con altri oggetti. (es: cane) Ogni oggetto è un'istanza di una sola classe. caratterizzato da uno stato = insieme del valore degli attributi. In base allo stato, due oggetti possono rispondere diversamente alla stessa operazione.  Rettangolo con tre diverse sezioni: nome, attributi, operazioni. Nome al singolare in **UpperCamelCase**. Se non ci sono attributi si lascia lo spazio vuoto. Solo il nome è obbligatorio

==Sintassi== visibilità nome molteplicità: tipo = valoreDefault
{proprietà}

Tre livelli di visibilità: - + **pubblico**: accesso esteso a tutte le classi - # **protetto**: accesso consentito solo alle classi che derivano dalla classe originale - - **Privato**: solo la classe originale può accedere

Molteplicità: Indica il quantitativo degli attributi di un certo tipo. Permette di indicare attributi come array o matrici, default = 1. Min e max racchiusi tra parentesi quadre. - [1..1], [0..1]... Gli elementi di una molteplicità a più valori sono considerati come un insieme. Se sono ordinati: {**ordered**}, se sono possibili duplicati: {**nonunique**}

tipo indica il tipo di dato. può essere primitivo (int, double, string..) o il nome di una classe definita nello stesso diagramma.

OPERAZIONE Corrispondono ai metodi di una classe, le operazioni invocabili sugli oggetti istanza della classe Sono le azioni che possono essere svolte da una classe di oggetti. Solitamente le operazioni che si occupano solo di modificare gli attributi non sono riportate perchè facilmente deducibili.

==sintassi:== `visibilità nome(listaParametri) : tipoRestituito{proprietà}`

![[Pasted image 20241104111949.png#invert|center|400]] ##### ASSOCIAZIONI Esprime una relazione fisica o concettuale tra le istanze (oggetti) delle classi. Caratterizzate da: - **nome**: etichetta dell'associazione. è un verbo - **ruolo**: Specificato sull'estremità dell'associazione prossima alla classe. Spesso usato in alternativa al nome. (es: dipendente - datore) - **molteplicità**: vincola il numero di oggetti di una classe che possono partecipare ad una associazione in ogni istante. Riportate sull'estremità dell'associazione vicino alla classe, nella forma min..max. - 0..1 partecipazione opzionale - 1 esattamente 1 - 0..* non c'è limite superiore - 1..* uno o più - 1..6 da uno a 6 - **verso di navigazione**: la classe di origine del verso di percorrenza ha la responsabilità di tenere traccia dell'associazione. Solitamente omessa nei progetti concettuali.

![[Pasted image 20241104114522.png#invert|center|400]]

![[Pasted image 20241104121057.png#invert|center|500]]

ASSOCIAZIONI vs ATTRIBUTI Le associazioni contengono info in più (molteplicità) Meglio non usare troppe associazioni 1:1, per non appesantire il diagramma. Si possono usare attributi per concetti secondari e associazioni per classi più significative. ![[Pasted image 20241104121553.png#invert|center|500]]