

Specifiche

Specifiche in UML per Classi,
relazioni, messaggi....

Specifiche per la classe

☰ Una classe può essere specificata

- testualmente (linguaggio naturale)
- graficamente (linguaggio di modellazione)

☰ in entrambi i casi la specifica conterrà

- il nome della classe
- gli attributi degli oggetti della classe
- i metodi per gli oggetti della classe

Specifiche per la classe

Metafora grafica per la Classe

Nome Classe

-Attributi Classe

+Metodi Classe()

Attributi degli oggetti

- ☞ Gli attributi rappresentano le caratteristiche degli oggetti di una classe.
- ☞ I singoli oggetti della classe sono caratterizzati dall'insieme dei valori che assumono gli attributi.
 - Tale insieme rappresenta lo “stato” dell'oggetto.
- ☞ In taluni casi un attributo può essere una caratteristica della classe, non degli oggetti.
 - Il valore di un tale attributo sarà comune a tutti gli oggetti della classe.

Metodi

- ☰ I metodi rappresentano, nel loro insieme, il comportamento degli oggetti di una classe.
 - Il comportamento effettivo di un oggetto dipenderà dal suo stato corrente.
- ☰ In alcuni casi uno o più metodi rappresentano un comportamento della classe. (Metodo per la classe)
 - l'invocazione di un metodo del genere comporterà una elaborazione indipendente dall'oggetto che la esegue.

Specifica degli attributi

📄 Un attributo è caratterizzato da

- Visibilità - Tipo Attributo- Nome Attributo- Valore predefinito
- La visibilità indica se un attributo è visibile o meno all'esterno della classe in cui è definito.
- Il tipo è un tipo concettuale sia semplice che complesso.
 - Esempi di tipi: Colore, Area, Kg, Km/h....
- Il valore predefinito è il valore che assume l'oggetto all'atto della sua creazione.

Visibilità degli attributi.

📄 La visibilità di un attributo può essere:

- Private simbolo ‘ - ‘
 - visibilità solo all’interno della classe
- Public simbolo ‘ + ’
 - visibilità anche all’esterno della classe
- Protected simbolo ‘ # ’
 - In una gerarchia Gen-spec: visibilità all’interno della classe base e di quella derivata.

📄 Per gli attributi propri della classe (non degli oggetti) si parla di attributi “statici”

- Essi sono specificati tramite il simbolo ‘ \$ ’

Visibilità degli attributi

- ☞ In generale, per il principio dell'information hiding gli attributi sono privati alla classe.
- ☞ In tal modo la struttura dati per l'attributo è nascosta all'interno della classe.
 - se l'attributo deve essere acceduto, la classe definirà degli opportuni metodi di accesso.

Specifica dei metodi

☰ Un metodo è specificato per mezzo di una “signature”

– **Visibilità -Nome(ListaParametri) : TipoRitorno**

- La visibilità indica se il metodo è visibile o meno all'esterno della classe in cui è definito.
- La lista dei parametri è una sequenza di parametri specificati come:
 - nome_parametro:Tipo_parametro =Valore_Predeterminato
- Il tipo di ritorno ed il tipo dei parametri formali è, come per gli attributi, un tipo concettuale.

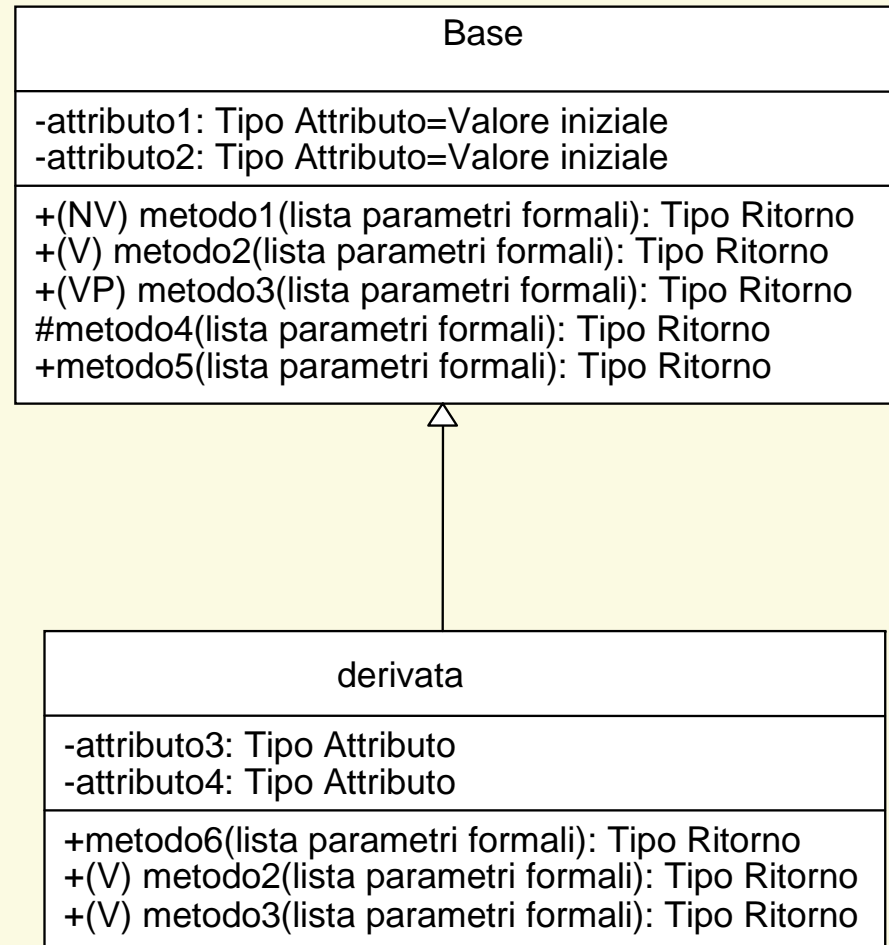
Visibilità dei metodi

- ☞ Anche per i metodi la visibilità può essere privata, pubblica o protetta.
 - Il formalismo è il medesimo
- ☞ I metodi privati possono essere usati solo dagli oggetti istanza della classe.
- ☞ I metodi pubblici possono essere usati anche da oggetti di altre classi.
- ☞ I metodi protected possono essere usati da:
 - gli oggetti della classe in cui sono definiti
 - gli oggetti istanze di eventuali classi derivate.

Visibilità dei metodi

- ☰ I metodi “pubblici” rappresentano, nel loro insieme, l’interfaccia della classe.
 - l’interfaccia indica quali risorse sono messe a disposizione dagli oggetti della classe nonché le modalità di utilizzo (signature dei metodi)
- ☰ I metodi "privati" servono per isolare “sottofunzionalità” comuni a più metodi pubblici.
 - Si evita la duplicazione di sottoparti comuni
- ☰ Per i metodi propri della classe (non dei singoli oggetti) si parla di metodi di tipo "static" e si usa il simbolo ‘ \$ ’

Specifiche per la relazione Gen-Spec



Specifiche per la relazione Gen-Spec

☰ La classe **derivata** eredita tutte le caratteristiche della classe **base**.

- Tutti gli attributi ed i metodi della classe base apparterranno anche alla classe derivata.

☰ La classe derivata potrà

- aggiungere altri attributi ed altri metodi.
- **ridefinire** i metodi virtuali della base
 - es. metodo2();
- **definire** i metodi VP della base
 - es. metodo3().

Specifiche per la relazione Gen-Spec

- ☞ La struttura dati delle classi derivate è l'unione della struttura dati definita nella classe base con quella definita nella classe derivata.
- Il codice che implementa i metodi (non ridefiniti) della base è condiviso dalla classe derivata.
- Il codice che implementa la ridefinizione dei metodi della classe base, è proprio della classe derivata.

Specifiche per la relazione Gen-Spec

☞ La relazione di derivazione può essere di due tipi:

☞ Relazione di derivazione pubblica, **is-a**

- il tipo derivato è esattamente lo stesso tipo della classe base
 - vale il concetto di sostituzione pura: un oggetto della classe derivata può sostituire uno della classe base.

☞ Relazione di derivazione privata o protetta, **is-like**

- il tipo derivato aggiunge altri elementi.
 - la sostituzione non è perfetta.

Relazione Gen-Spec: is a.

☰ E' anche detta ereditarietà di interfaccia o sottotipazione (subtyping)

- tutte le classi specializzate sono vincolate a rispettare l'interfaccia ereditata
 - il comportamento delle classi specializzate è determinato dalla particolare implementazione dei propri metodi che può ridefinire i metodi definiti nella classe base.

☰ Significato concettuale:

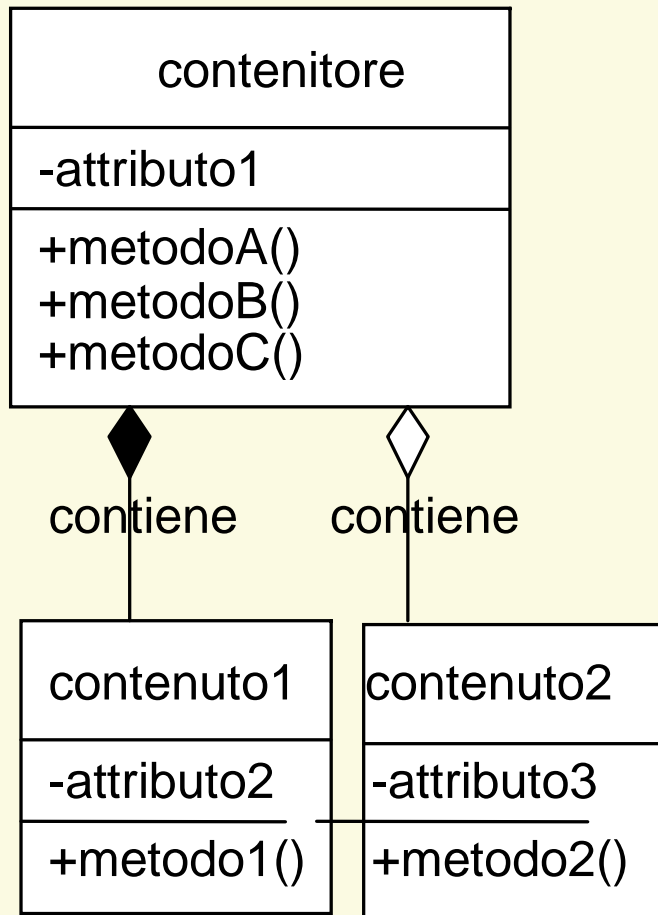
- una qualsiasi istanza della classe derivata è anche una istanza della classe base (un cerchio è una figura).

☰ Riuso a livello di modellazione.

Relazione Gen-Spec: is like.

- 📄 E' anche detta ereditarietà di implementazione: lo scopo principale non è quello di riuso "concettuale" ma di riuso di codice esistente.
- 📄 Tutte le classi derivate usano i metodi della classe base senza alterazioni (overriding)

Specifiche per la relazione di aggregazione



Specifiche per la relazione di aggregazione

☰ La relazione di aggregazione può essere di tipo

- “stretto” (si parla anche di composizione)
 - il ciclo di vita degli oggetti contenuti è legato a quello del contenitore
 - nell’esempio, tra contenitore e contenuto1
- “lasco”
 - il ciclo di vita degli oggetti contenuti è indipendente dal contenitore
 - essi nascono e muoiono con il contenitore
 - nell’esempio, tra contenitore e contenuto2

Specifiche per la relazione di aggregazione

- ☰ L'oggetto contenitore è proprietario degli oggetti contenuti
- ☰ Per accedere ai servizi messi a disposizione dai "contenuto" occorre "passare" per l'oggetto **contenitore**.
 - Ad esempio per invocare il metodo1 su un oggetto della classe contenuto1 occorre richiamare il metodoA dell'oggetto contenitore
 - quest'ultimo ridirigerà la richiesta all'oggetto contenuto1.

Implementazione dell'aggregazione

☰ L'aggregazione "lasca" è realizzata per riferimento

- la classe contenitore conterrà una variabile membro di tipo riferimento per ognuna delle classi "contenuto".

☰ L'aggregazione "in senso stretto" è realizzata per inclusione

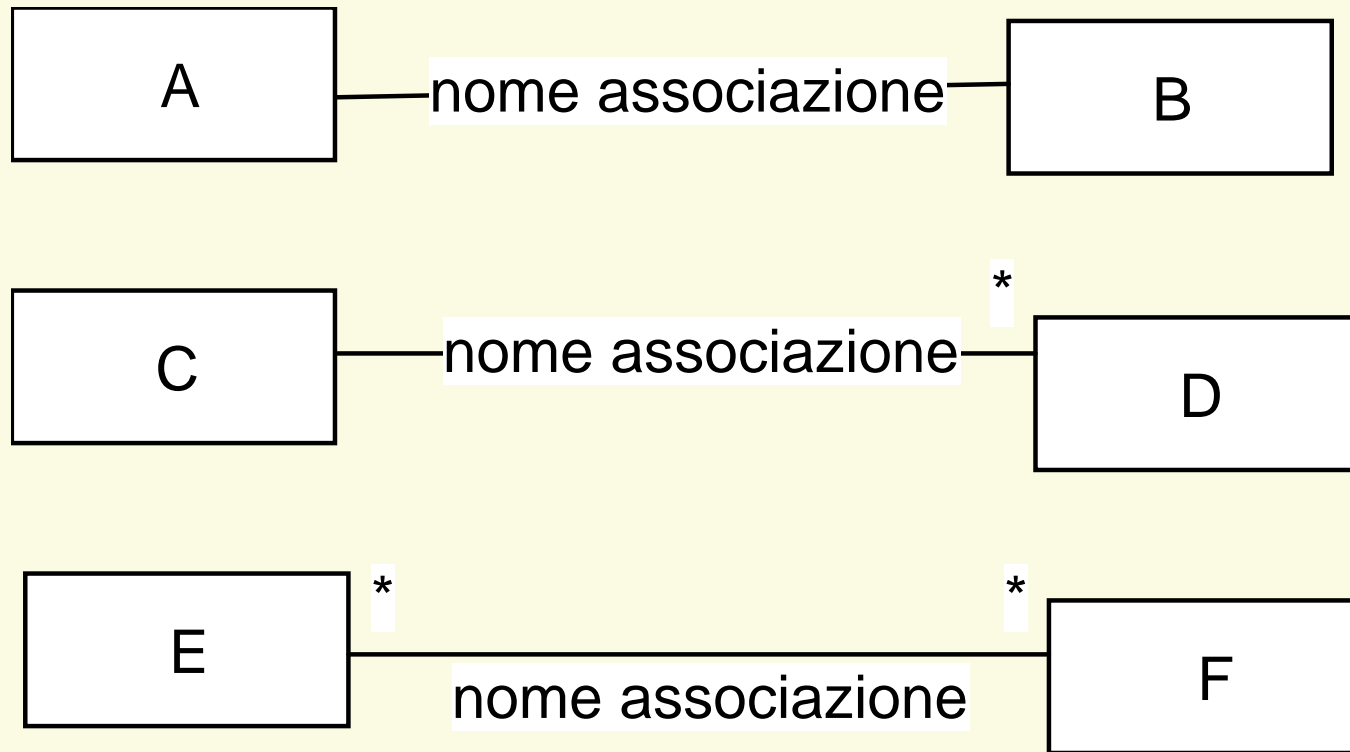
- la classe contenitore avrà come attributi, variabili di tipo delle classi contenuto (oggetti)

Specifiche per la relazione di associazione

☰ In una relazione tra due classi si può specificare

- il nome dell'associazione
 - semantica del collegamento tra oggetti istanza delle classi associate
- la cardinalità dell'associazione
 - uno a uno, uno a molti, molti a molti
- il ruolo giocato dalle classi associate
- la navigabilità
 - monodirezionale, bidirezionale

Specifiche per la relazione di associazione



Associazione uno ad uno

☰ Se tra due classi A e B esiste una associazione uno a uno, ogni oggetto istanza di A è collegato ad un oggetto istanza di B.

- il collegamento tra gli oggetti si realizza tramite una variabile membro di tipo riferimento alla classe associata
- il nome della variabile sarà il ruolo specificato per la classe associata.

☰ Se la navigabilità è bidirezionale, la responsabilità di tenere traccia del collegamento è affidata ad entrambi gli oggetti collegati.

Associazione uno a molti

📄 Ogni oggetto della classe A è collegato con più oggetti di B

- l'oggetto a di A avrà quindi n riferimenti, uno per ognuno degli oggetti associati.
- la variabile che traduce l'associazione sarà quindi un array statico o dinamico di riferimenti agli oggetti associati.
- il nome di tale variabile potrà essere il ruolo (al plurale) che gioca la classe associata.

Associazione molti a molti.

☰ In questo caso entrambi gli oggetti istanza delle classi associati tradurranno l'associazione con un array di riferimenti alla classe associata.



Esempi

Esempio di classificazione

Gli oggetti:

Ferrari
GTV

Fiat
Punto

Mercedes
Classe A

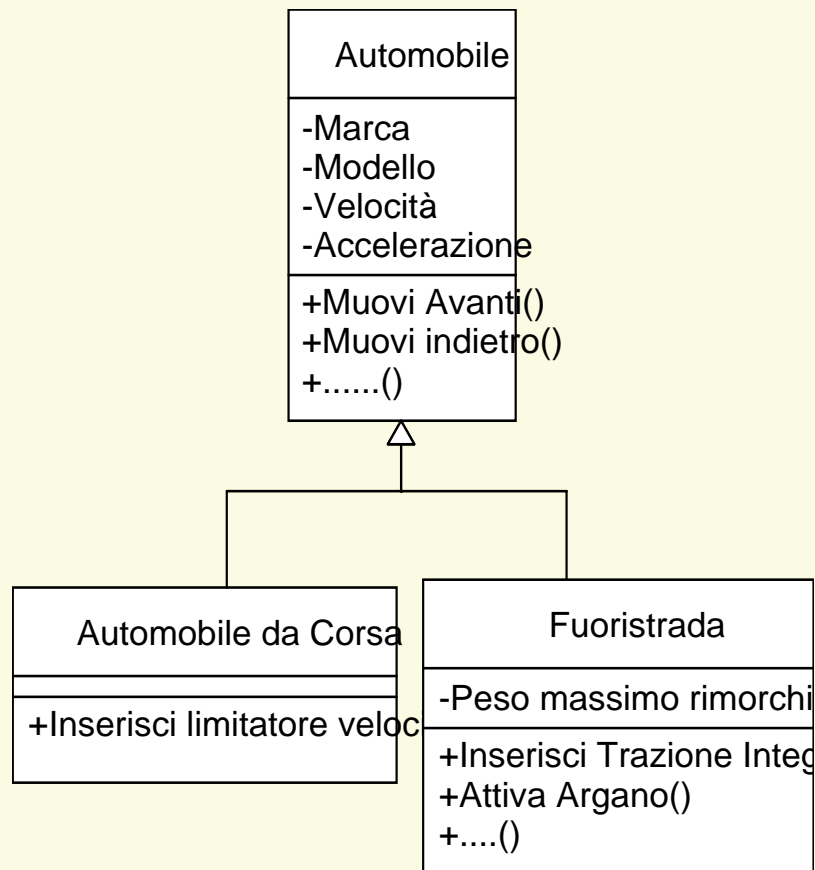
Renault
Twingo

Sono entità distinguibili,
ognuno con una propria marca,
un proprio modello,
un proprio colore

Essi possono essere concettualmente rappresentati attraverso la classe:

Automobile
-Marca -Modello -Velocità -Accelerazione
+Muovi Avanti() +Muovi indietro() +.....()

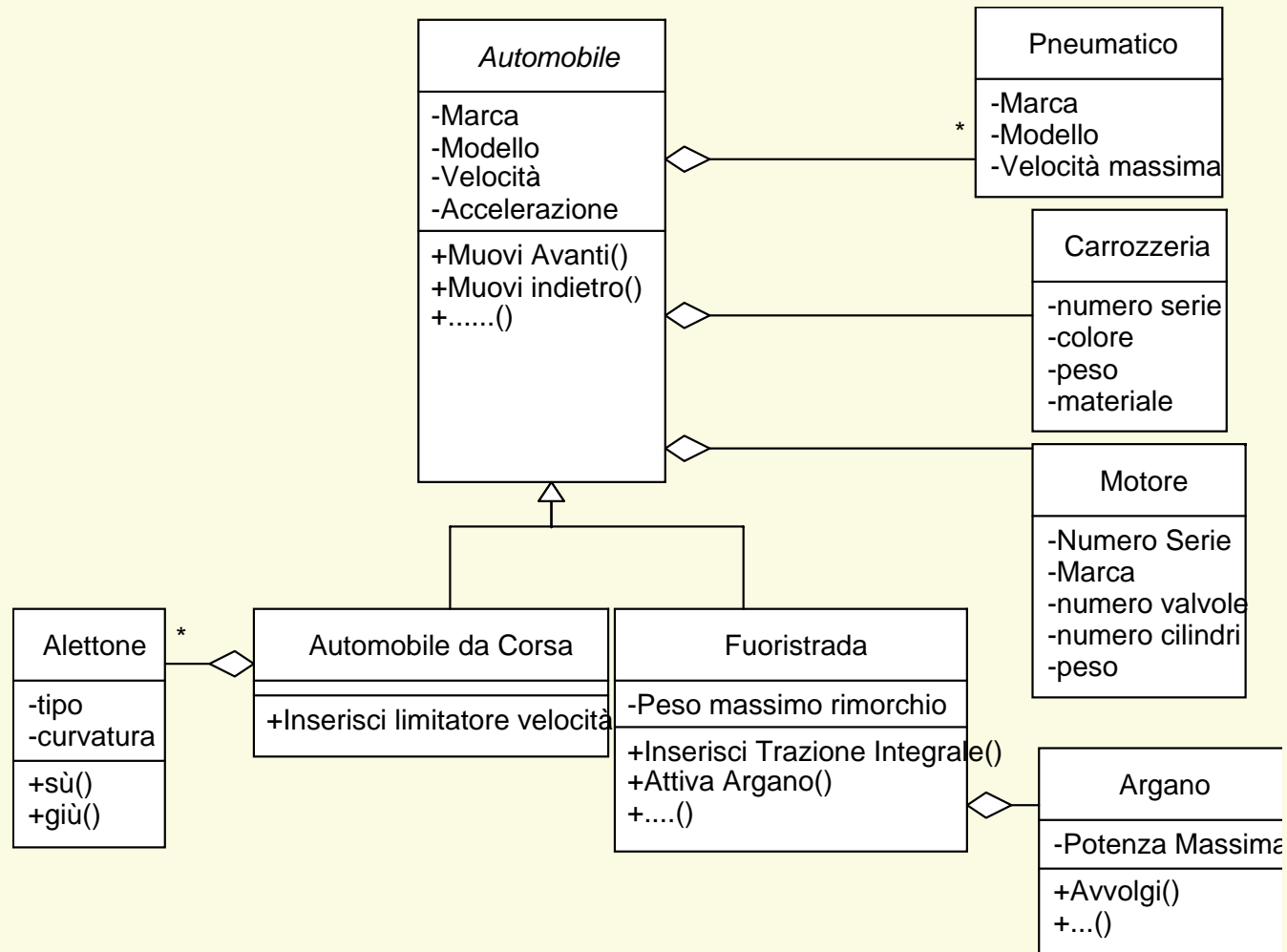
Esempio di classificazione



La classe Automobile da Corsa è una particolare particolare della classe Automobile così come lo è la classe Fuoristrada.

La classe Automobile cattura comportamenti e caratteristiche comuni.

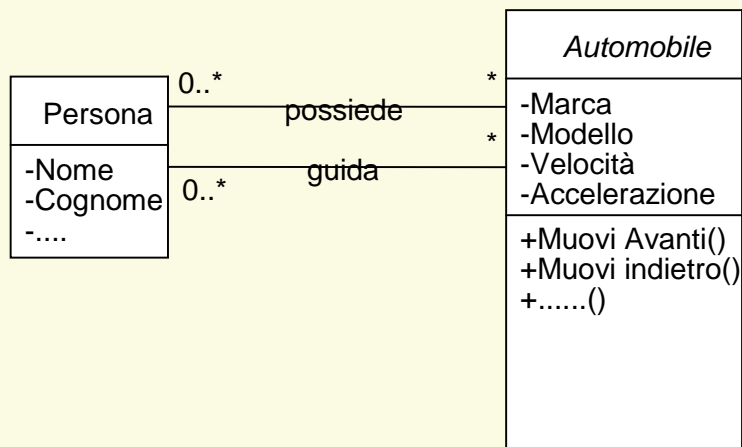
Esempio di aggregazione



Esempio di aggregazione

- ☰ Un'automobile può essere percepita come l'aggregazione delle sue parti costituenti: il motore, la carrozzeria, i pneumatici.
- ☰ Una istanza della classe Automobile è un aggregato di Pneumatico (molti), Motore (uno), Carrozzeria (una).
- ☰ Una istanza della classe Fuoristrada è anch'esso un aggregato di oggetti: tutti quelli che compongono la superclasse Automobile (Fuoristrada è (is a) una Automobile) più un oggetto di tipo Argano.

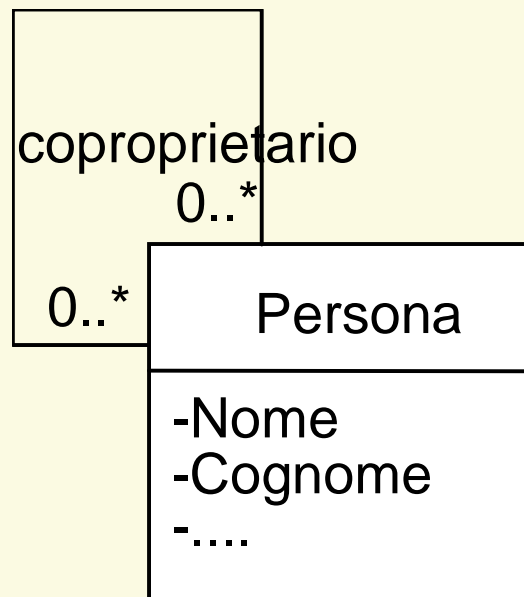
Esempio di associazione



Significato della relazione:

- Un oggetto di tipo persona:
 - possiede e guida una o più automobili
- un oggetto di tipo automobile può essere guidata e posseduta da zero o più persone

Esempio di associazione ricorsiva



- Ad ogni oggetto istanza di persona può essere collegato un oggetto di medesimo tipo che gioca il ruolo di coproprietario.