

Progettazione Software

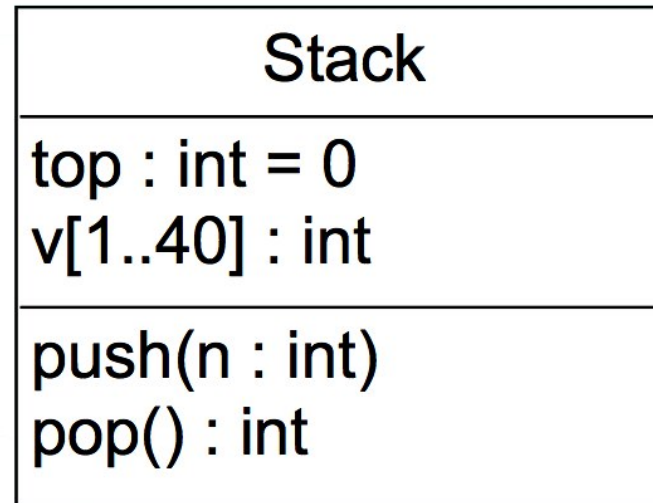
Modulo 5: Class Diagram

Class Diagram

- Il diagramma delle classi consente di rappresentare delle viste, a vari livelli di astrazione, sul modello globale di un dominio applicativo mediante:
 - **Entità presenti nel dominio applicativo:**
 - Classi
 - Interfacce
 - **Relazioni fra le entità, come:**
 - generalizzazione
 - associazione
 - aggregazione
 - composizione

Rappresentazione delle Classi

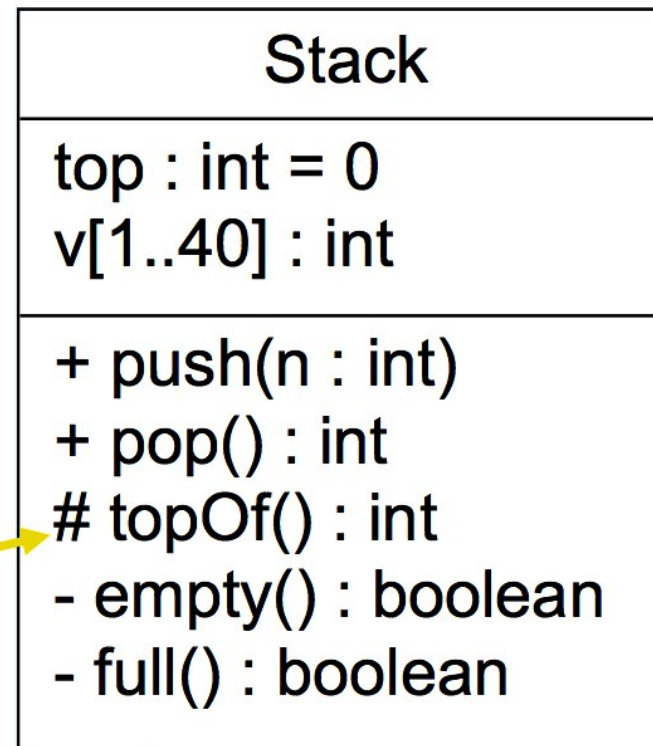
classe con nome
semplice



attributi

operazioni

classe con nome
composto (path)



operazioni
pubbliche

operazione
protetta

operazioni
private

simbolo di
visibilità

Definizione degli **Attributi**

- La sintassi di base per descrivere un attributo all'interno del simbolo di classe è la seguente:
 - `idAttributo [: tipo] [= valore iniziale]`
- Durante la specifica un attributo può essere rappresentato come:
 - `idAttributo`
 - `idAttributo : idTipo`
 - `idAttributo : idTipo = valoreDefault`

Attributi d'Istanza e di Classe

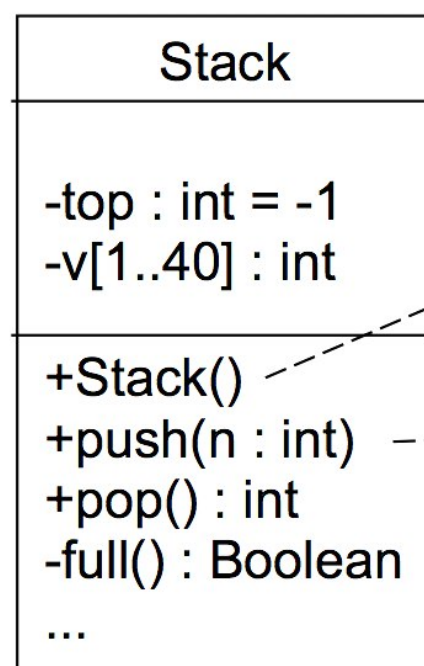
- Gli attributi dichiarati in una classe si classificano come:
 - attributi d'**istanza**
 - vengono **allocati per ogni istanza** della classe
 - vengono manipolati da qualunque operazione appartenente alla classe
 - attributi di **classe**
 - vengono **allocati una sola volta** e sono **condivisi** fra tutte le istanze della classe
 - in Java vengono dichiarati con la parola chiave static
 - in Java vengono inizializzati contestualmente alla loro dichiarazione.

Le Operazioni

- Un'operazione descrive un **servizio** fornito da una classe.
- All'interno della classe, a un'operazione corrisponde un'implementazione, detta **metodo**, cioè la codifica degli algoritmi utilizzati per realizzare l'operazione.
- Un'operazione viene invocata inviando a un'istanza della classe un **messaggio**

Descrizione delle operazioni

- L'algoritmo di un'operazione può essere descritto utilizzando il simbolo delle note.
- La descrizione può essere effettuata in:
 - formato libero (testo)
 - con pseudocodice
 - con le istruzioni di un linguaggio di programmazione.



top = -1;

```
if (!full()) {  
    top = top + 1;  
    v[top] = n;  
}  
else  
    throw new ExStackFull();
```

descrizione delle
operazioni con le note

Operazioni di classe e di istanza

- operazioni d'**istanza**
 - vengono eseguite in seguito alla ricezione di messaggi dall'istanza
 - **possono manipolare qualunque attributo** dichiarato nella classe
- operazioni di **classe**
 - **possono manipolare solo gli attributi di classe** (attributi static)

Terminologia UML e Terminologia Java

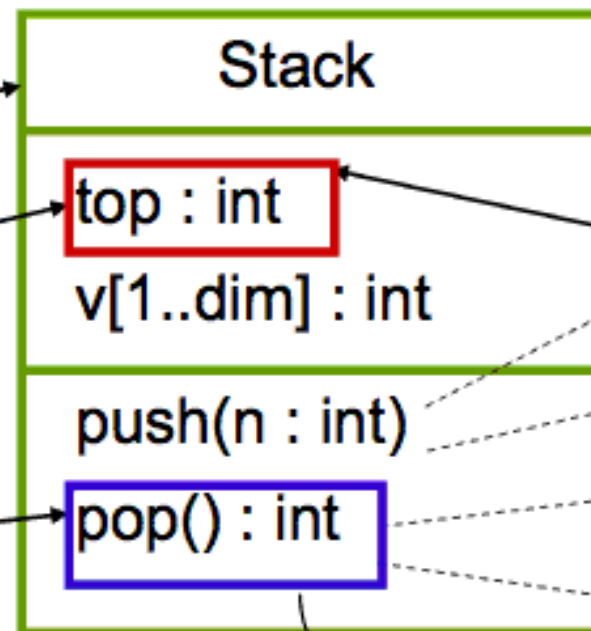
UML

classe

attributo

operazione

metodo



Java

classe

campo

metodo

Progettazione Software

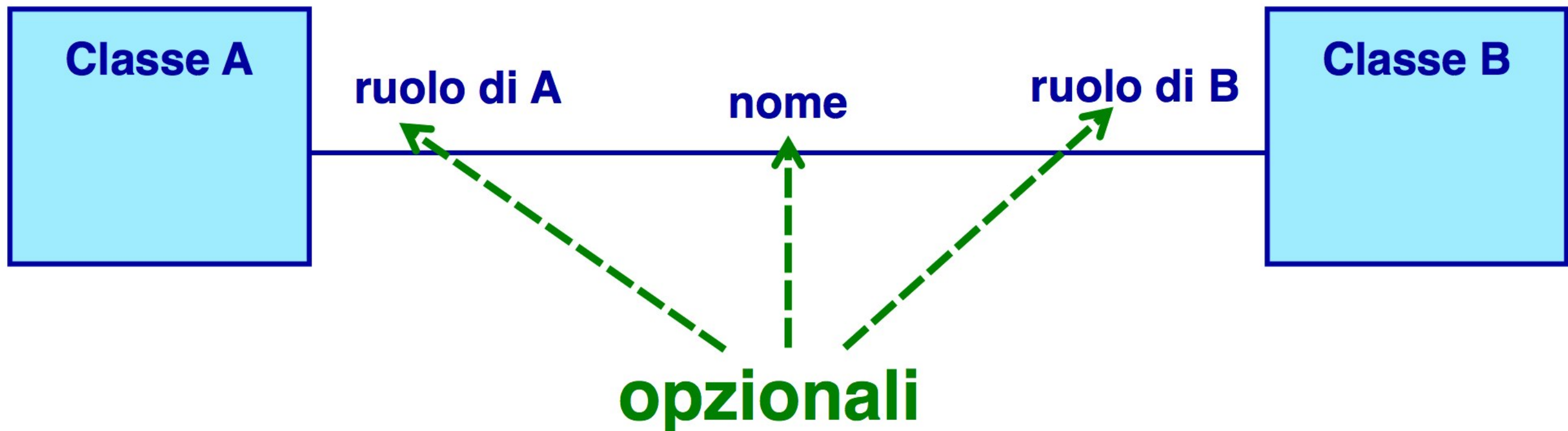
Modulo 6: Relazioni tra le Classi

Relazioni

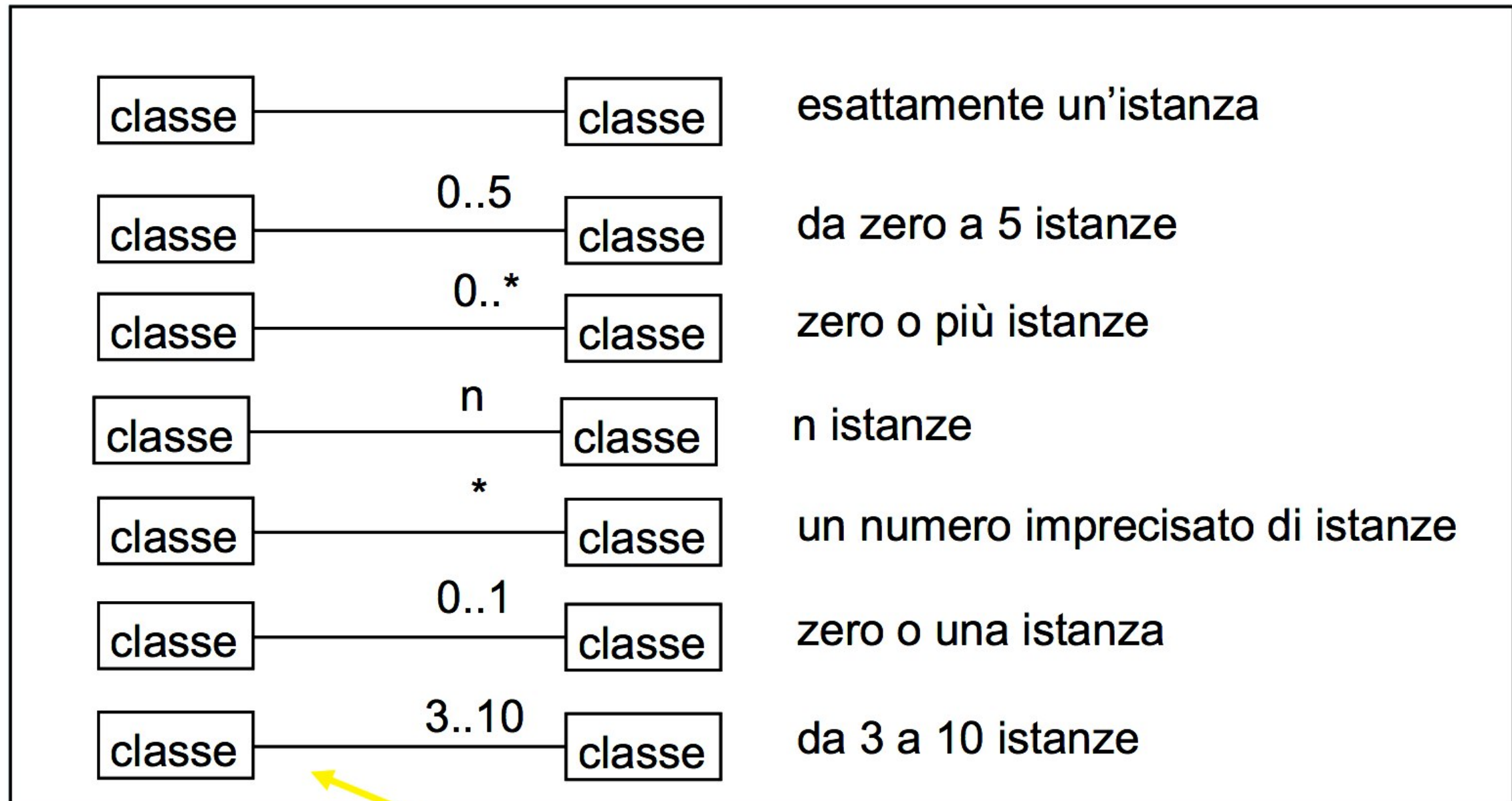
- Una relazione rappresenta un **legame** tra due o più classi e relative istanze
- Le **principali relazioni** previste in UML sono:
 - Associazione
 - Aggregazione
 - Composizione
 - Generalizzazione

Associazione

- L'associazione è la relazione basilare che intercorre tra due classi.
- È riassunta dal termine **HAS-A**
- È possibile definire il nome, la molteplicità e i ruoli delle classi coinvolte



Molteplicità delle Associazioni



quando non c'è niente per default assumiamo uno

Esempio di molteplicità

- Una società ha molti dipendenti, un dipendente ha un solo datore di lavoro



- Un oggetto Società potrà essere in relazione con 1 o più dipendenti
- Un oggetto Dipendente potrà essere in relazione con 1 solo oggetto Società

Associazione orientata

- La relazione di associazione può essere orientata.
- Prendiamo questo esempio:



(ogni) cellulare possiede un italiano



Esempio: Ordine di Vendita

UTENSILERIA ITALIANA spa

ORDINE N° 117125/07 DEL 12/05/07

RIGA	CODICE	DESCRIZIONE	Q.tà	Prezzo
1	3712007	PUNTA M8x18	10	2.58
2	4805091	FRESA	20	7.30
3
4

Totale : 331,50€

Esempio : Ordine di Vendita

- **Ordine :**
 - Ha una data, un numero e un importo totale
 - Ha un cliente intestatario e una o più righe d'ordine
 - Deve essere spedito e chiuso
- **Riga d'ordine:**
 - Ha un numero di riga, un articolo e una quantità
- **Articolo:**
 - Ha un codice, una descrizione e un prezzo
- **Cliente:**
 - Ha un nome e un indirizzo

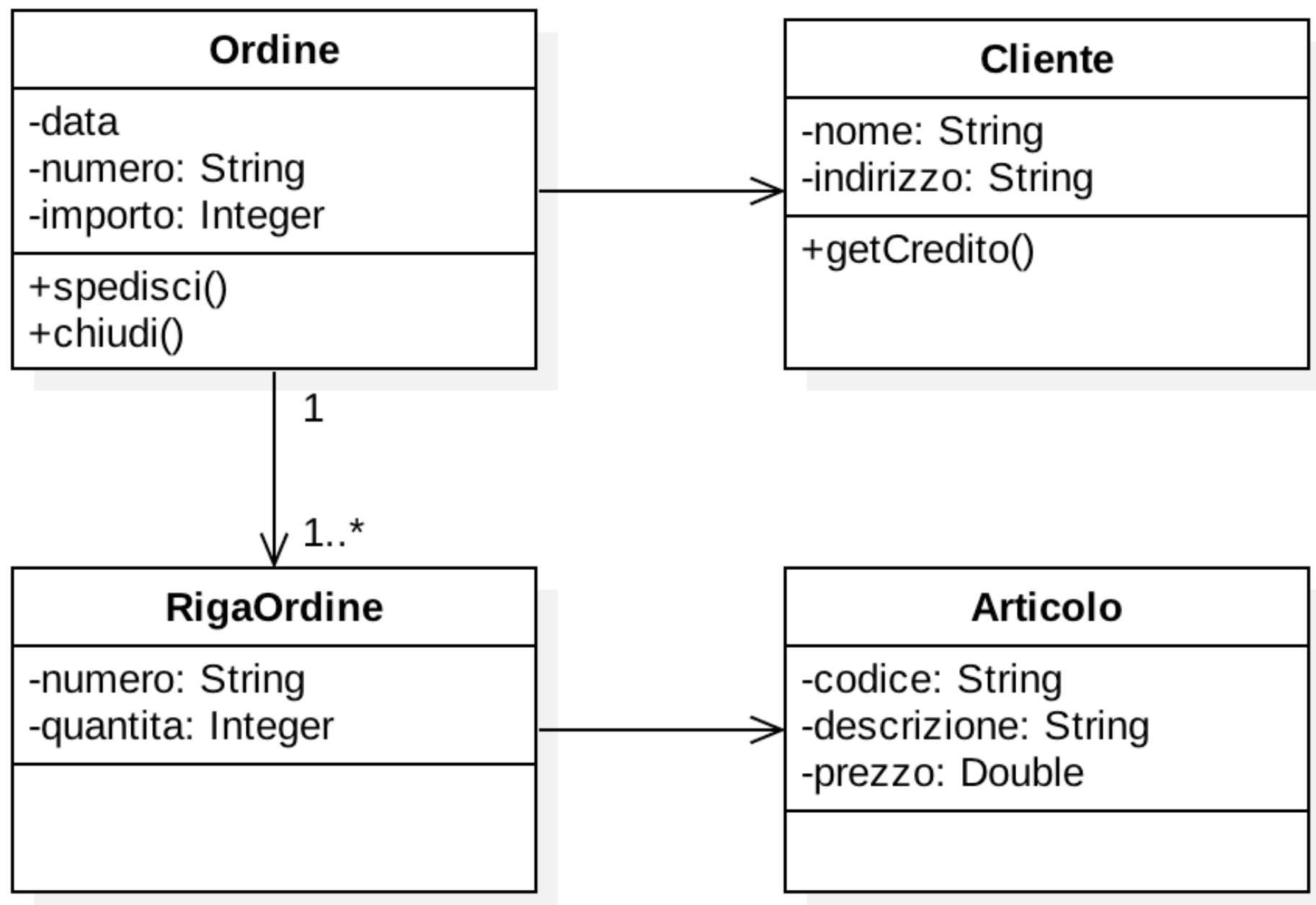
Esempio : Ordine di Vendita

- La classe Ordine ha:
 - un nome che lo identifica univocamente
 - attributi e operazioni

Ordine
<ul style="list-style-type: none">- cliente : Cliente- data : Date- numero : String- riga : RigaOrdine [1..*]- importo : int
<ul style="list-style-type: none">+chiudi()+spedisci()

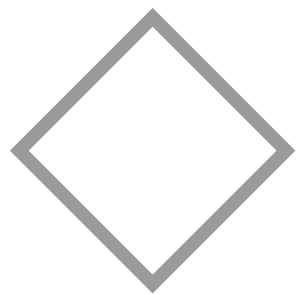
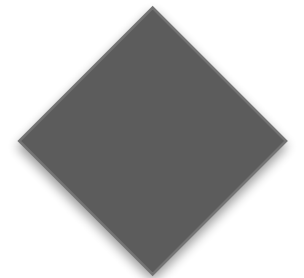
Esempio : Ordine di Vendita

- Una possibile soluzione con la relazione di associazione:

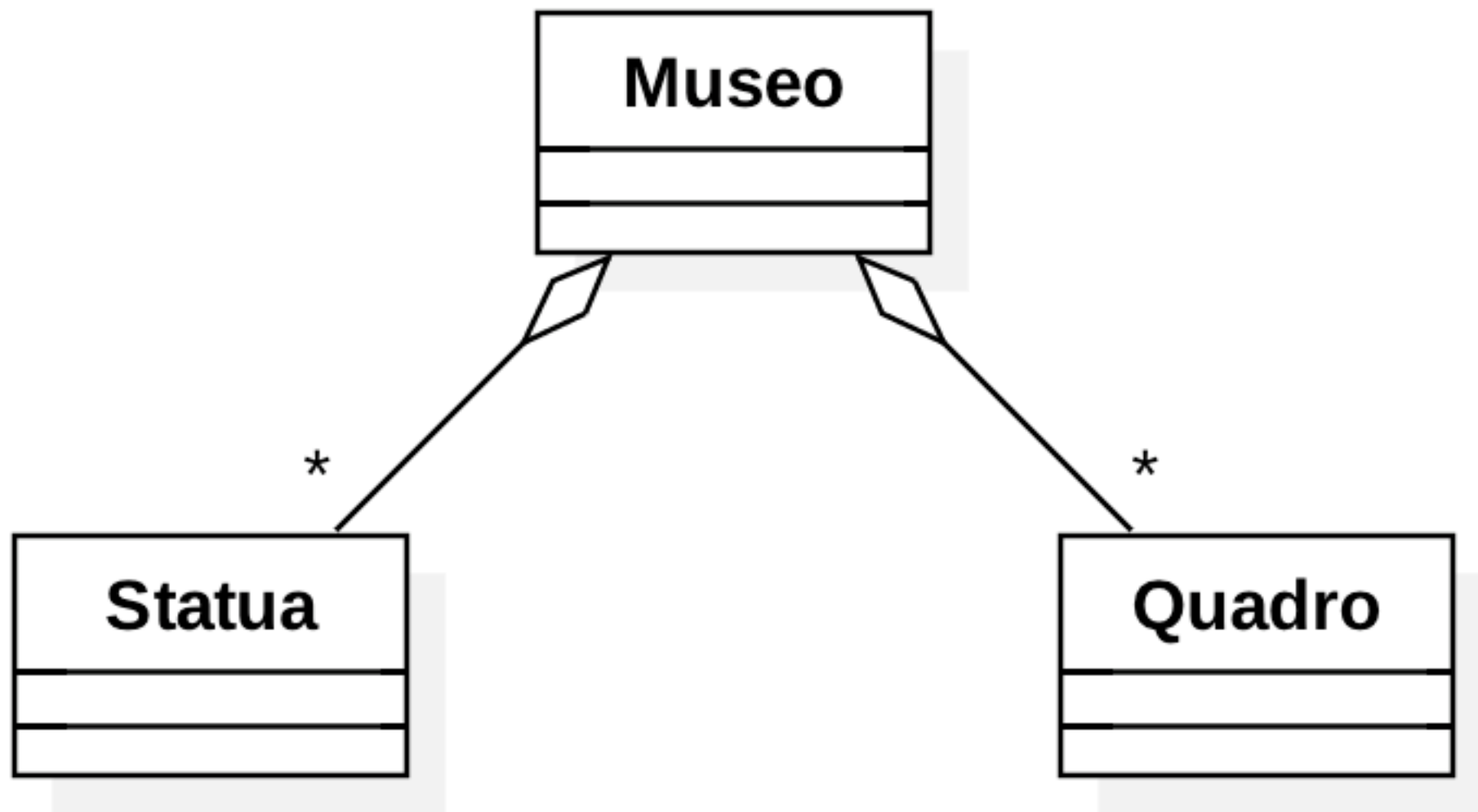


Aggregazione e Composizione

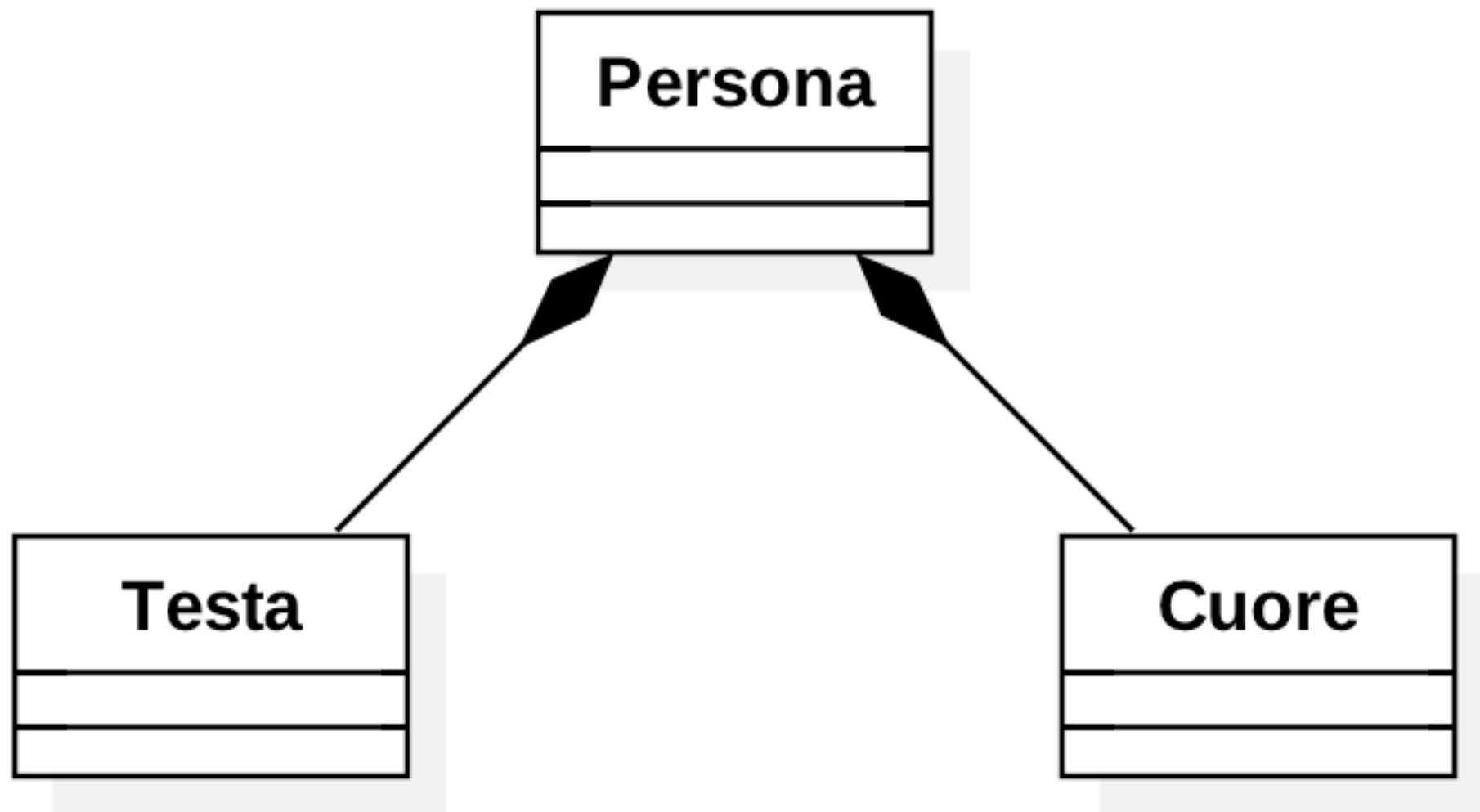
- Quando un oggetto ha un insieme di più oggetti, si possono incontrare due diverse situazioni:
 - L'oggetto contenitore è una **composizione** di più oggetti:
 - è una relazione **forte**: le classi parte hanno un reale significato solo se sono legate alla classe tutto
 - L'oggetto contenitore è un'**aggregazione** di più oggetti:
 - è una relazione **non forte**: le classi parte hanno un significato anche senza che sia presente la classe tutto



Aggregazione



Composizione



Generalizzazione

- Uno dei principi cardine della programmazione OO è l'**ereditarietà**, rappresentata in UML dalla relazione di Generalizzazione.
- L'ereditarietà è la **capacità di una classe (detta sottoclasse) di ereditare le caratteristiche e le operazioni definite in un'altra (superclasse), specializzandole.**
- La relazione è riassunta dal termine **IS-A**

Ereditarietà e Polimorfismo

classi di un dominio applicativo



Aereo militare



Battello



Mongolfiera



Dirigibile



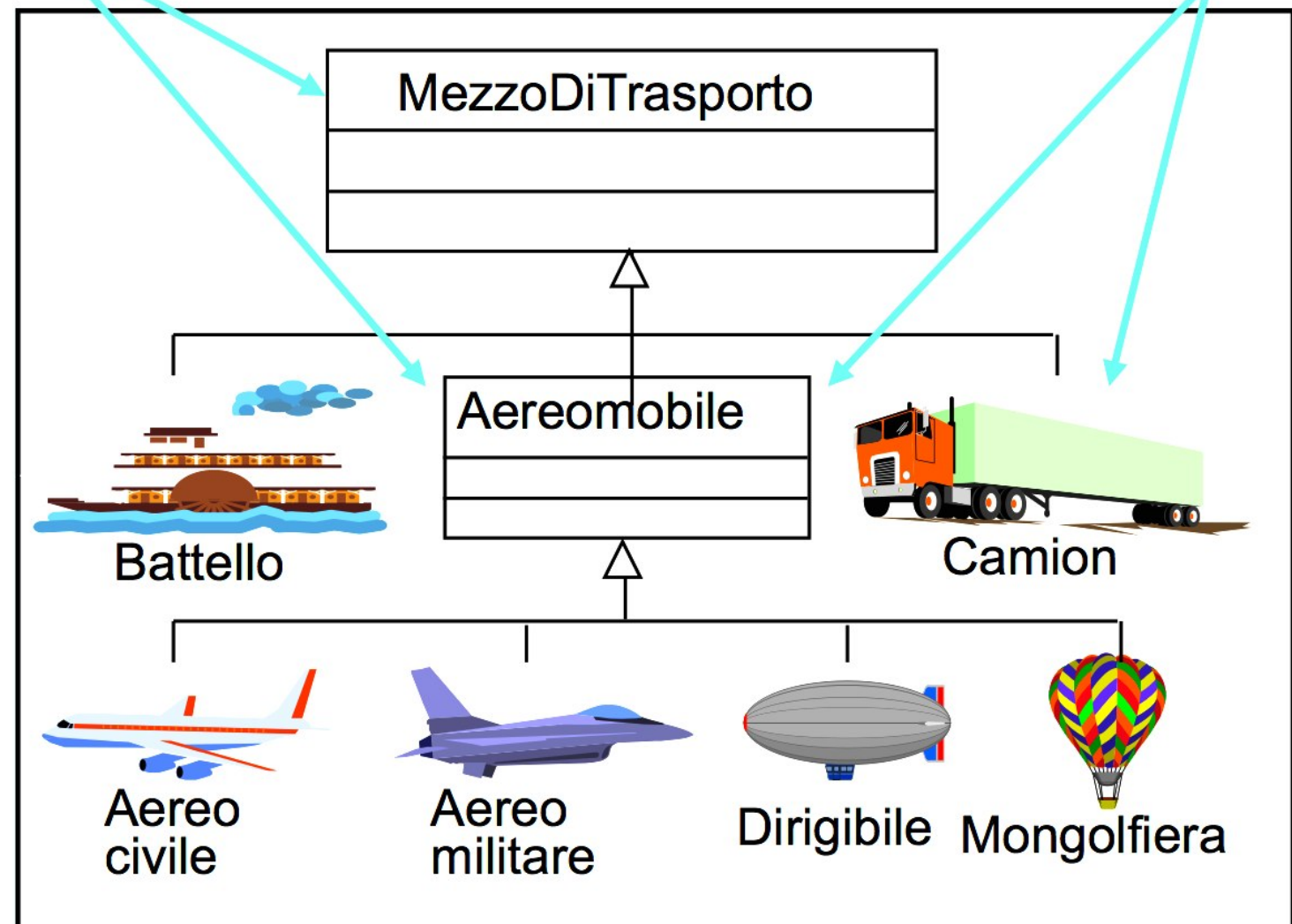
Camion



Aereo civile

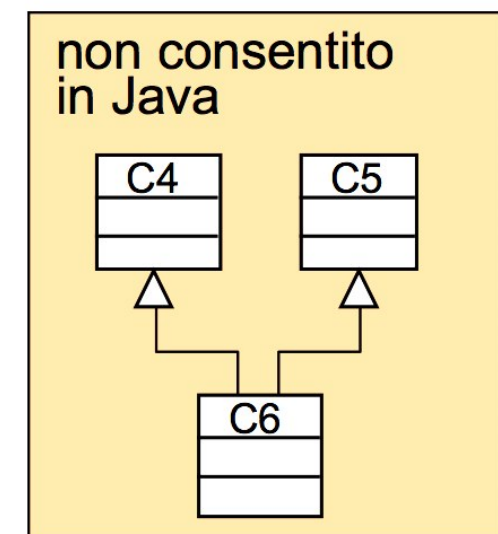
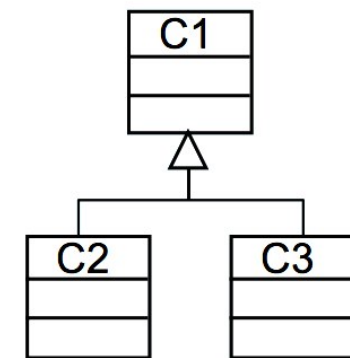
superclassi

sottoclassi



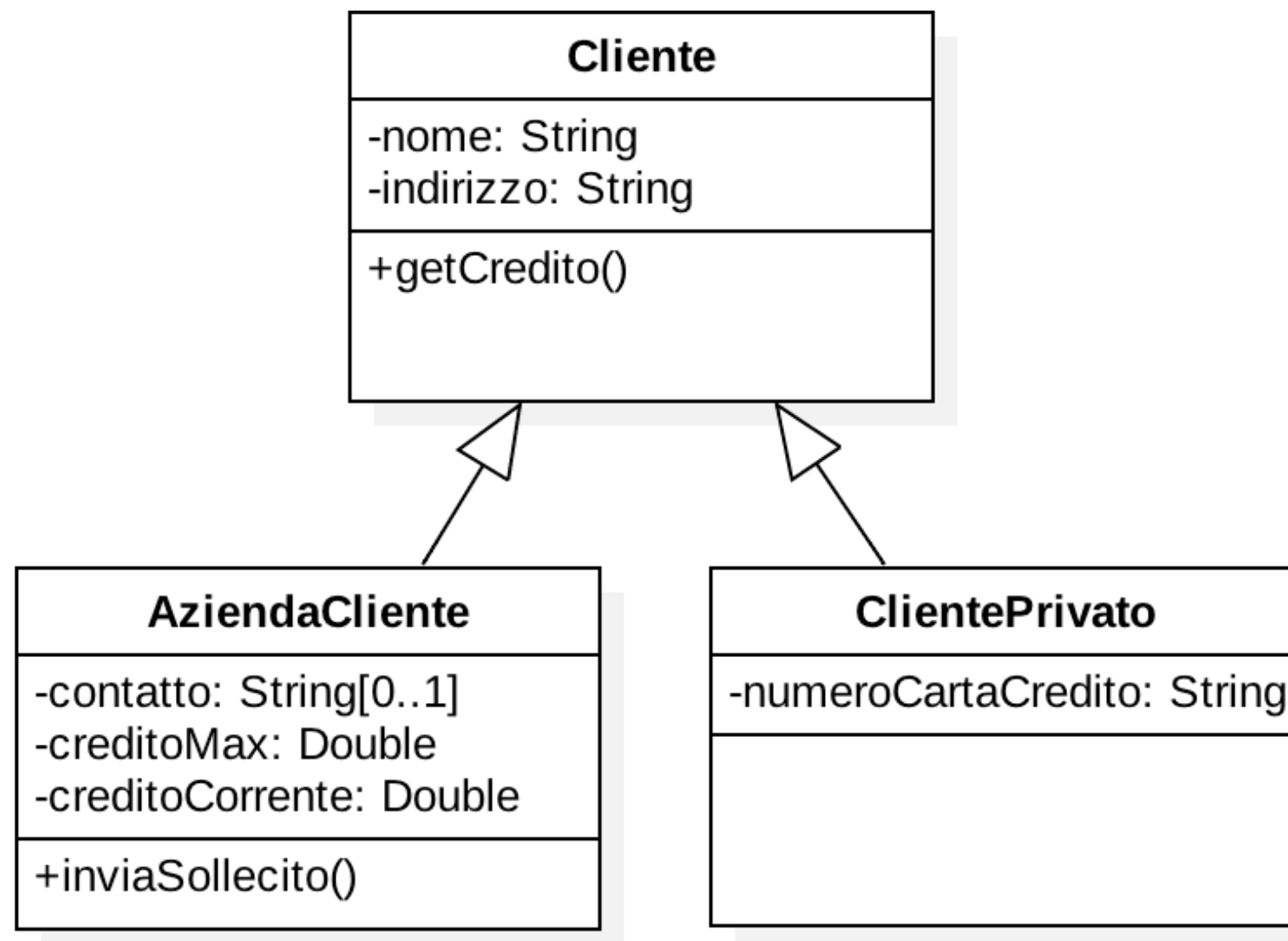
Generalizzazione

- L'ereditarietà (e la generalizzazione) può essere
 - **Singola**
 - una classe specializza una sola superclasse di un livello superiore
 - una superclasse può generalizzare una o più sottoclassi di un livello inferiore
 - **Multipla**
 - una classe specializza almeno due superclassi di un livello superiore



Esempio: Ordine di Vendita

- Tornando all'esempio, presupponiamo che i clienti possano essere privati o aziende:



Esempio: Ordine di Vendita

