

# Programmazione II e Lab di PII

## UML

Angelo Ciaramella

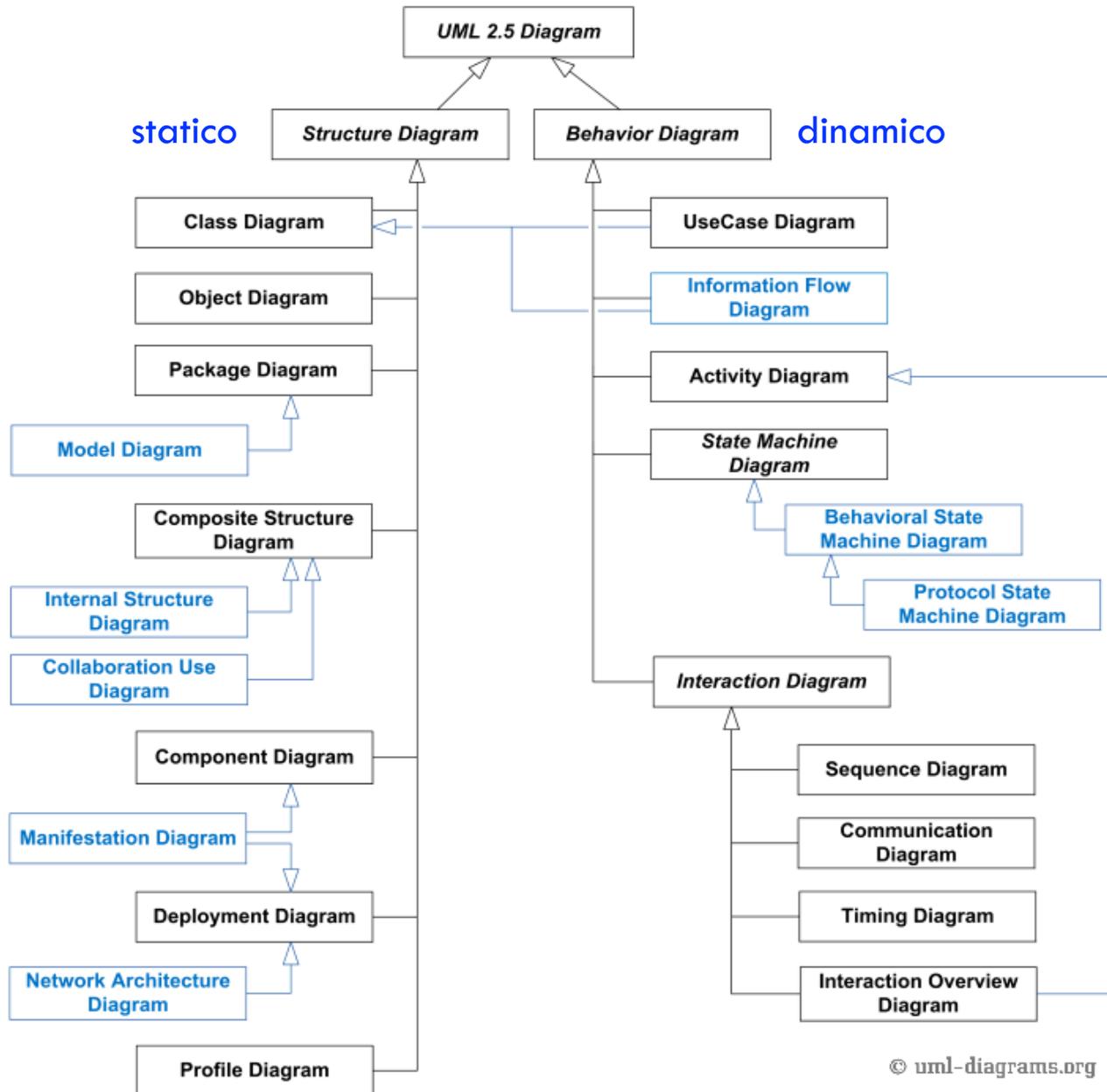
# Unified Modeling Language

---

- Unified Modeling Language (UML)
  - insieme di diagrammi formali per la descrizione di un problema o di una soluzione
  - prima versione sviluppata nel 1994
  - l'ultima versione è la 2.5
  - contiene elementi grafici (simboli) connessi



# UML 2.5



non sono ufficialmente inseriti

# Use Case Diagram

---

- Use Case Diagram

- interazioni tra l'ambito della descrizione e le entità a esso esterne

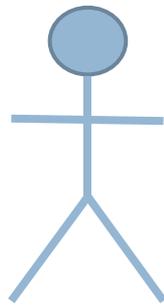
- Elementi principali

- attore
- caso d'uso (use case)



# Attore

- Attore
  - una persona
  - componente di un altro sistema

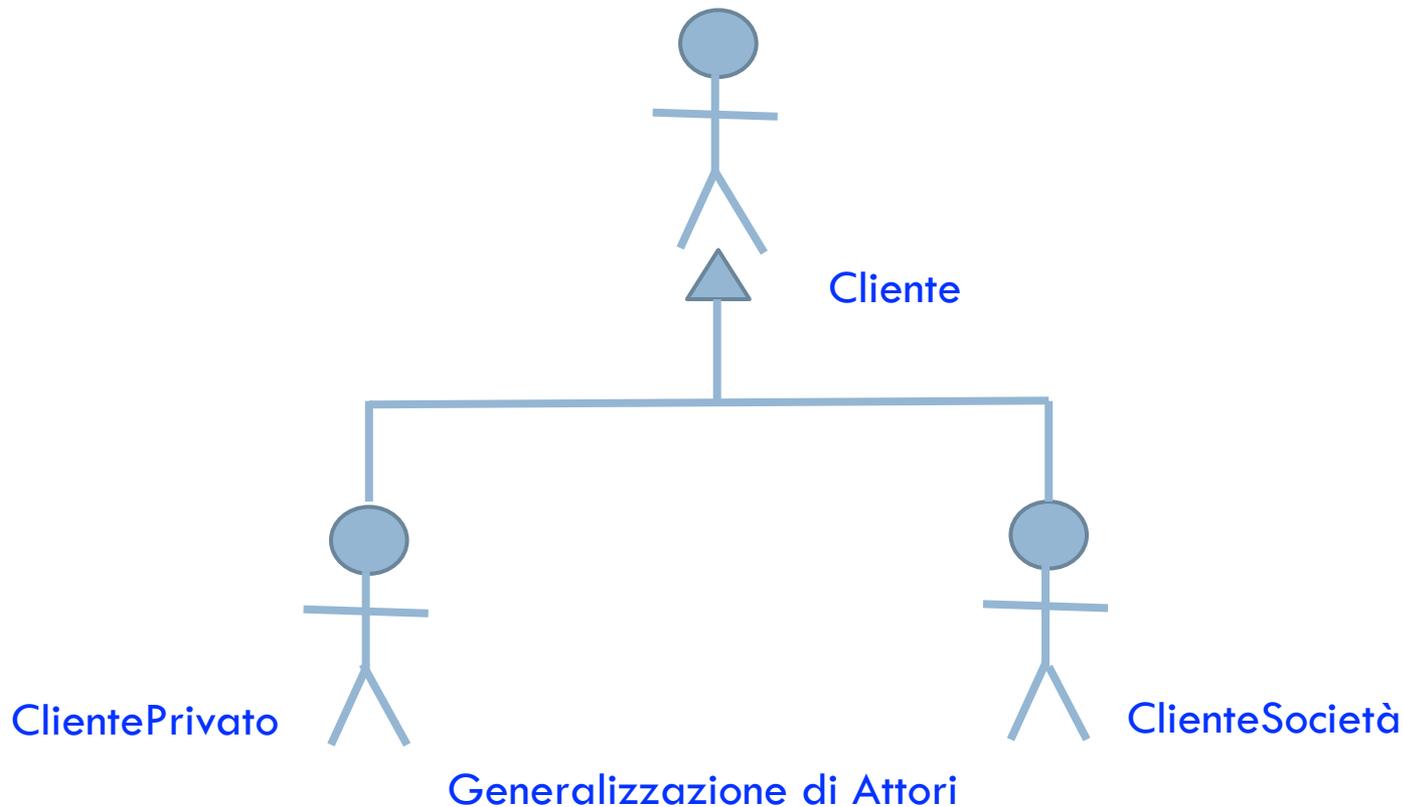


Rappresentazione entità Attore



# Generalizzazione

- Relazione di generalizzazione
  - Derivazione di entità figlie da un padre



# Caso d'uso

---

- Caso d'uso
  - Descrive una macro-funzionalità

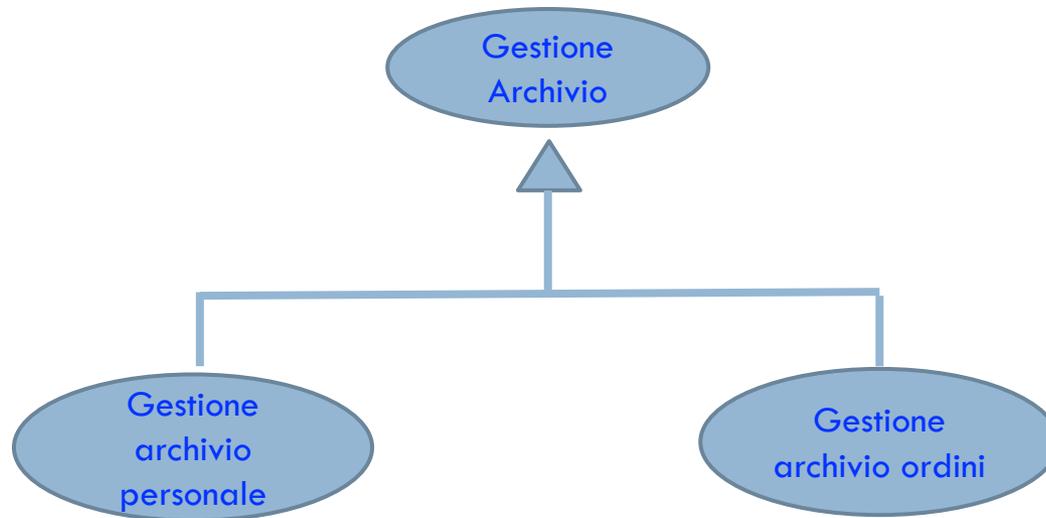


Caso d'uso



# Generalizzazione

- **Generalizzazione tra use case**
  - l'use case figlio eredita tutte le **caratteristiche** dello use case padre
  - l'use case figlio è **presente** in tutti gli scenari dello use case padre



Generalizzazione tra use case

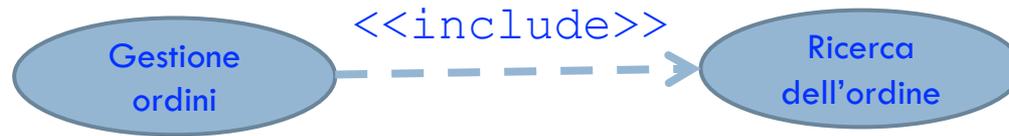


# Relazioni di inclusione ed estensione

- Le relazioni di **inclusione** ed **estensione**
  - **stereotipi** della relazione di dipendenza
    - **elementi** di UML per evidenziare proprietà peculiari
  - **inclusione**
    - per la **realizzazione dello use case che include**, è necessario che sia **realizzato lo use case incluso**
  - **estensione**
    - uno use case **estende un secondo use case (use case base)** quando **descrive in modo più ampio** e dettagliato una variante dello use case base



# Relazioni di inclusione ed estensione



Relazione <<include>>



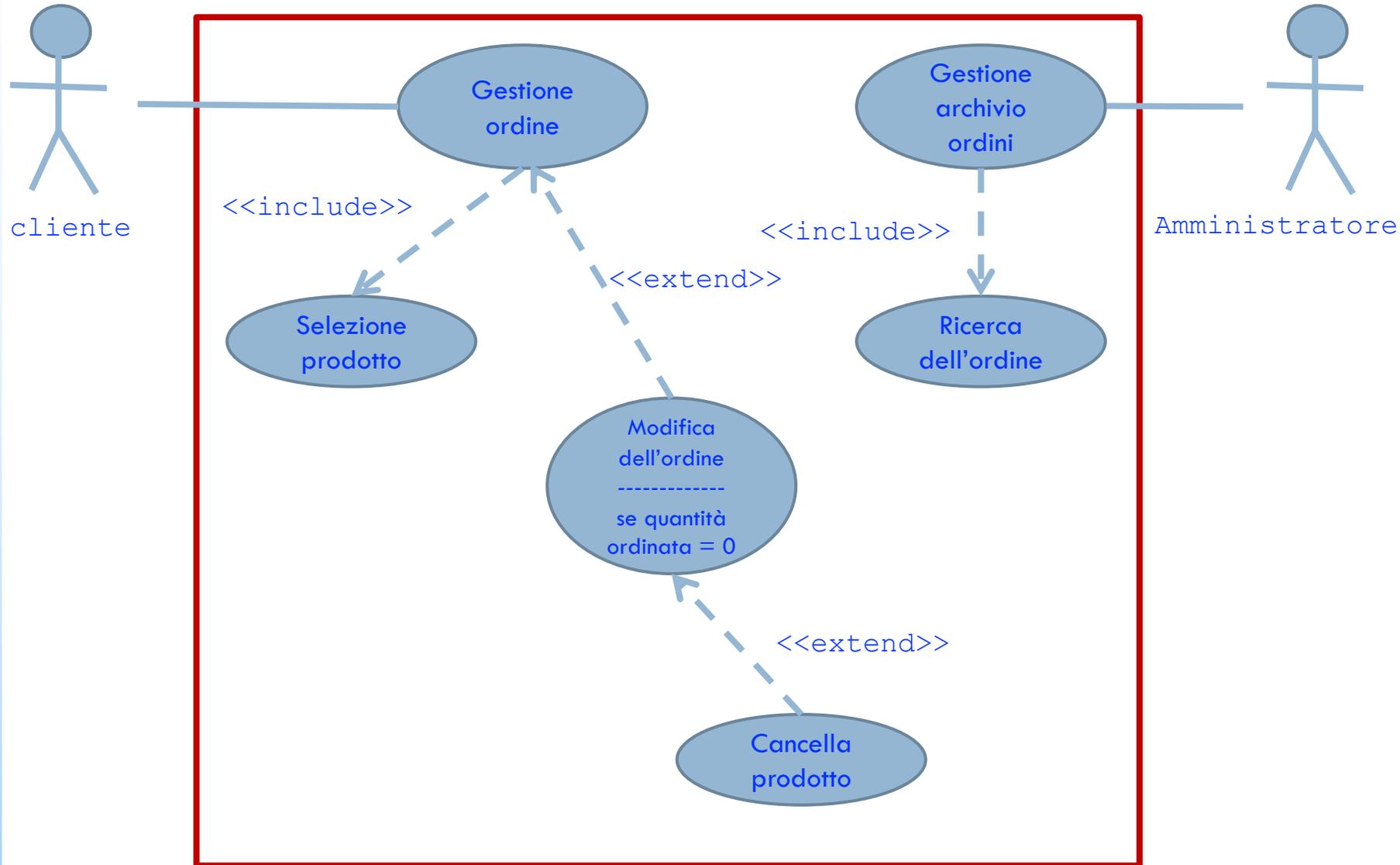
Relazione <<extend>>

use case base

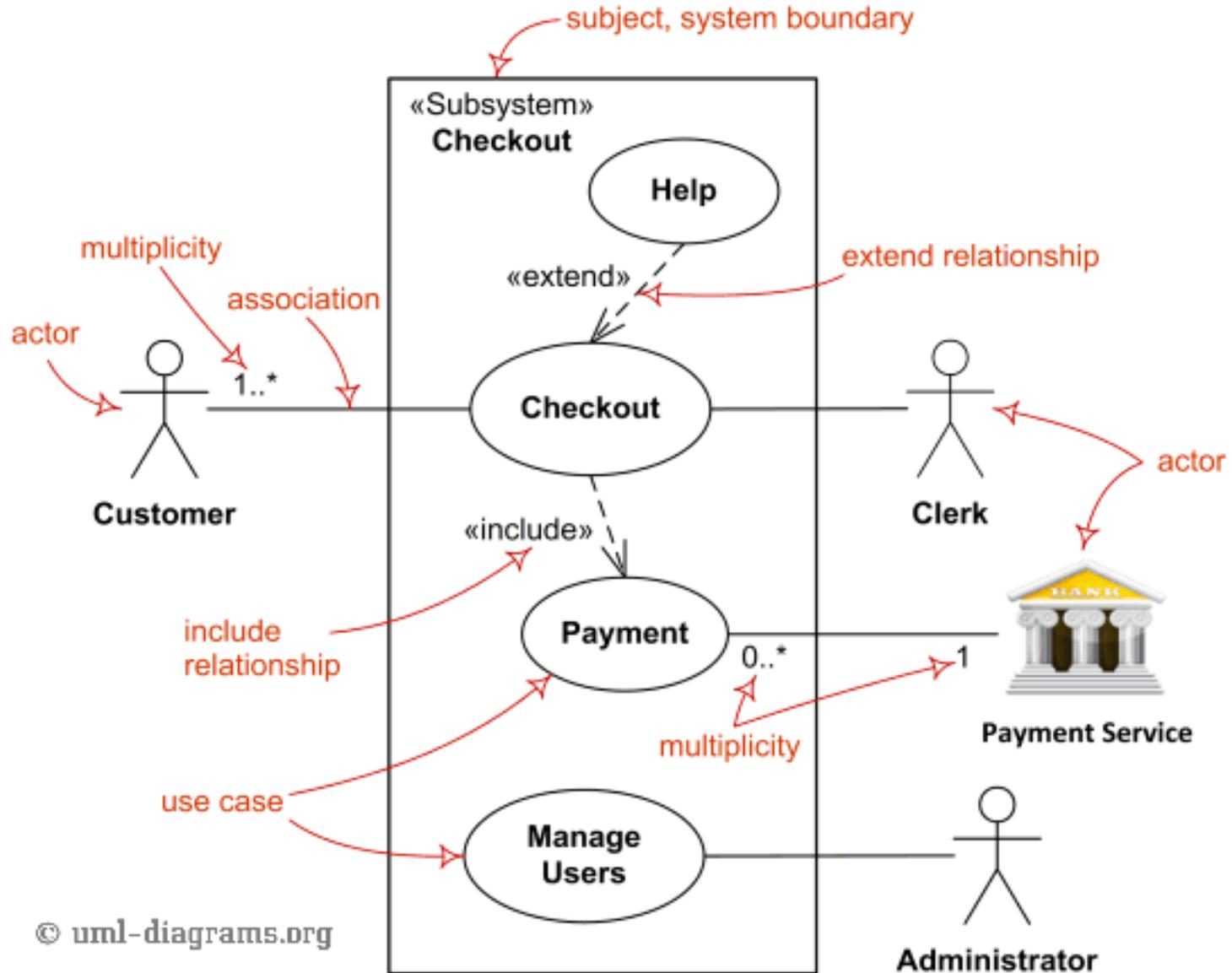
extension point, vincoli di esecuzione



# Esempio di use case diagram



# Esempio di use case diagram



© uml-diagrams.org

# Class diagram

---

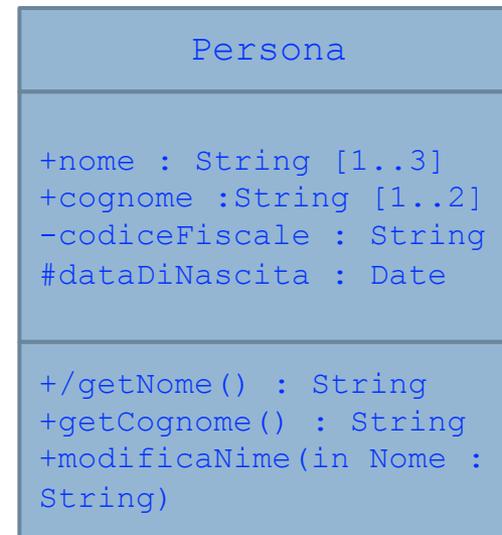
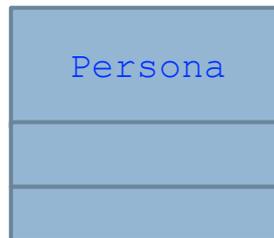
- Il class diagram
  - illustra l'ambito di **descrizione** da un punto di vista **statico**
  - evidenzia **caratteristiche** e **mutue relazioni**
    - classi
    - relazioni



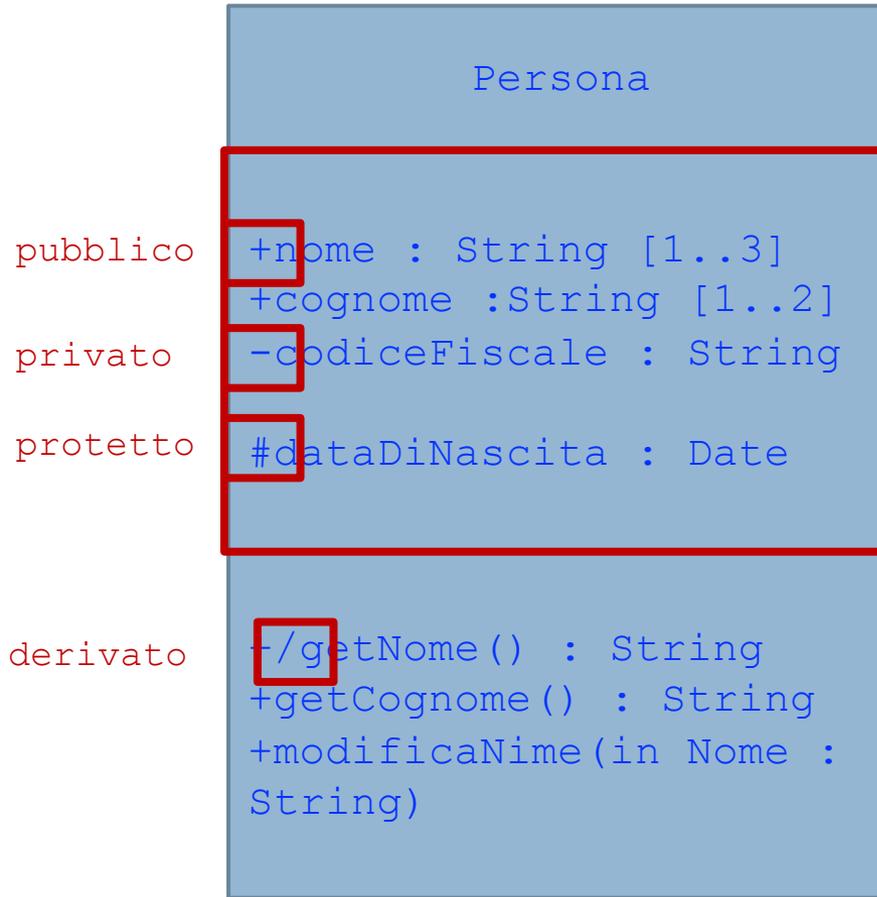
# Classe

## ■ Classe

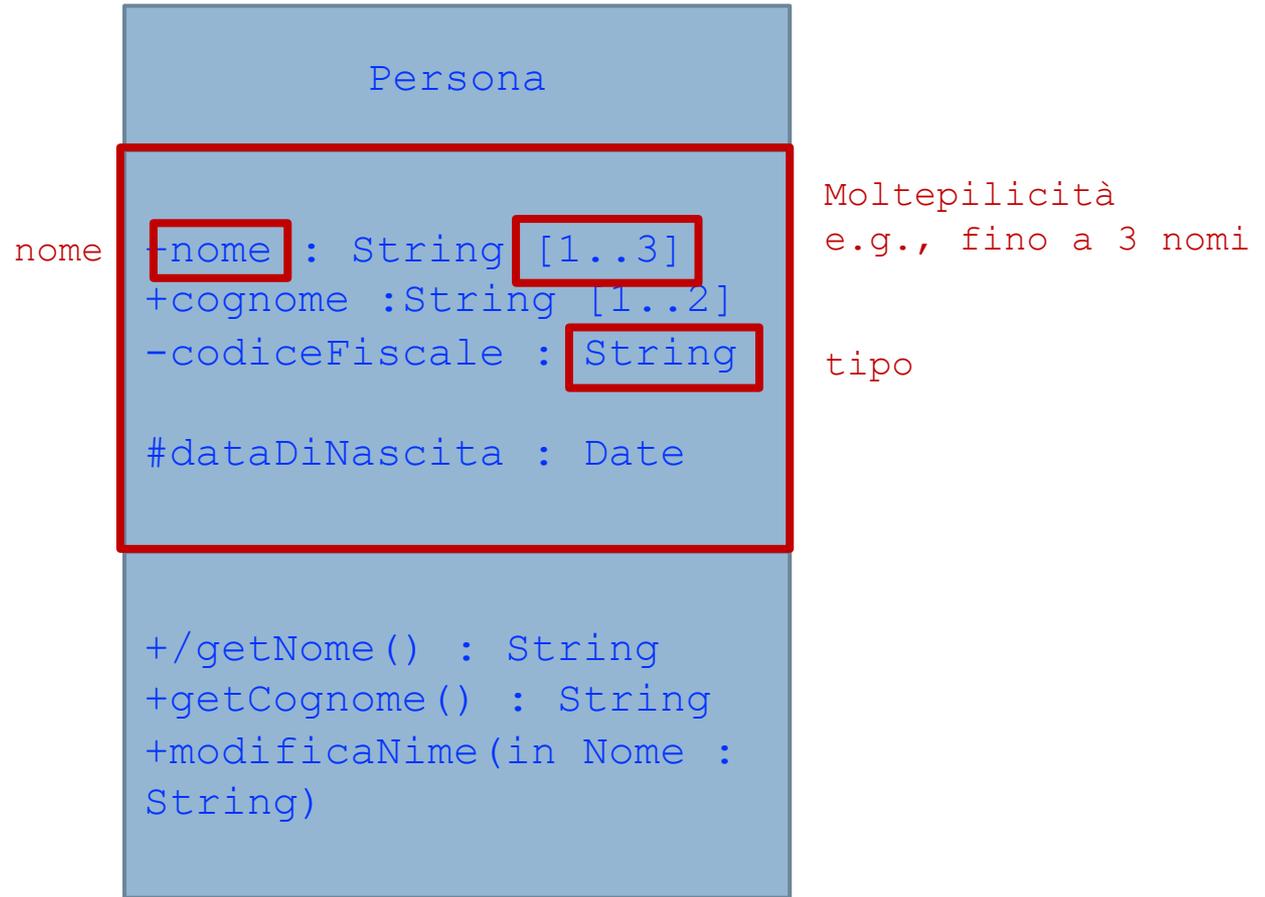
- descrive un insieme di **entità** dotate delle stesse caratteristiche e proprietà
  - oggetti



# Attributi

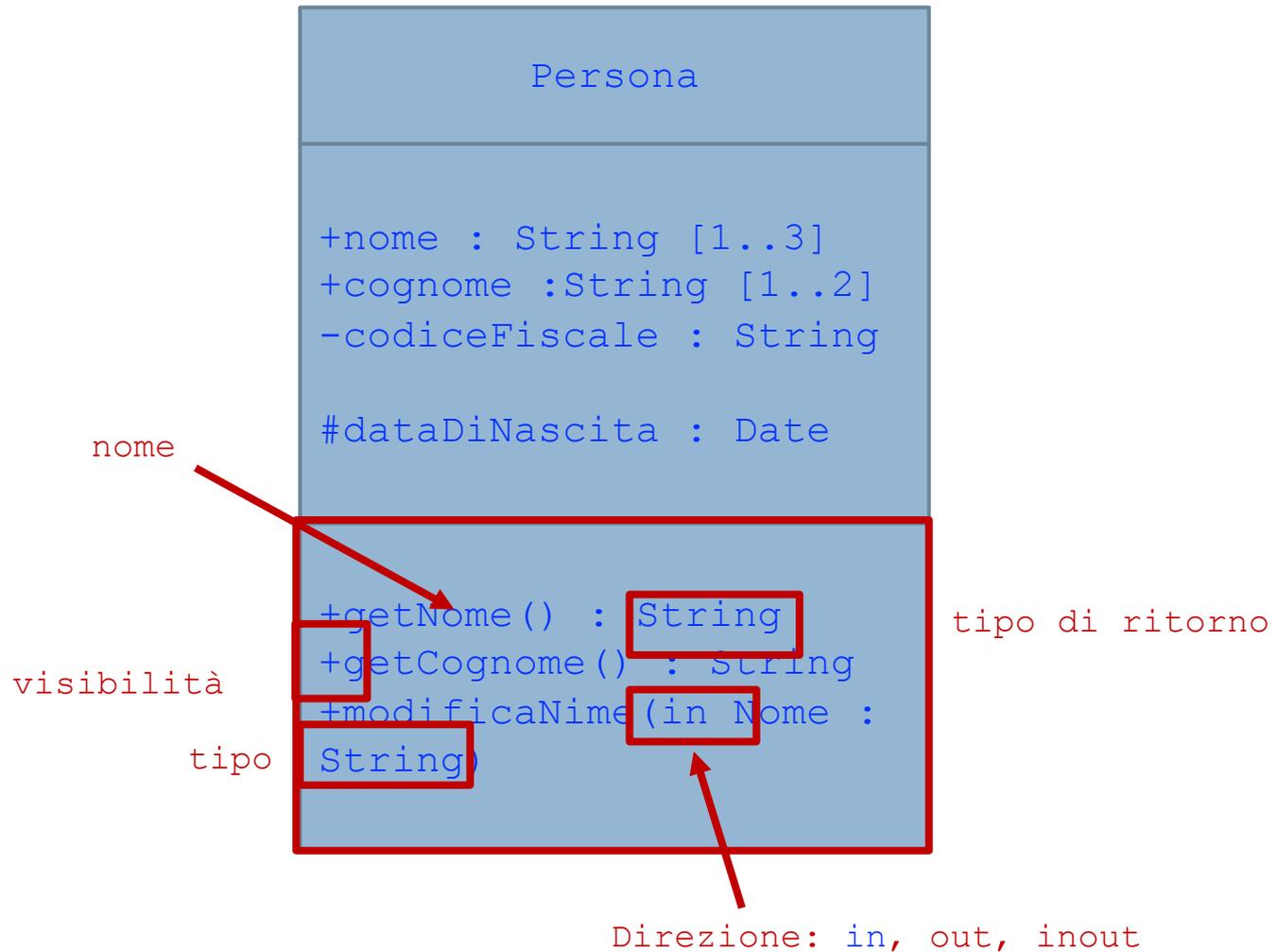


# Attributi



E' possibile avere anche un valore di Default per gli attributi

# Operazioni



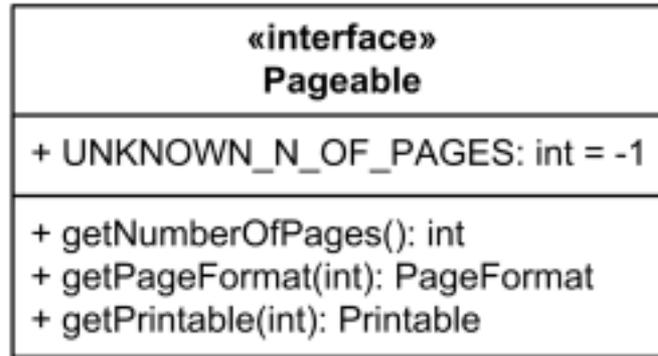
# Interfaccia

---

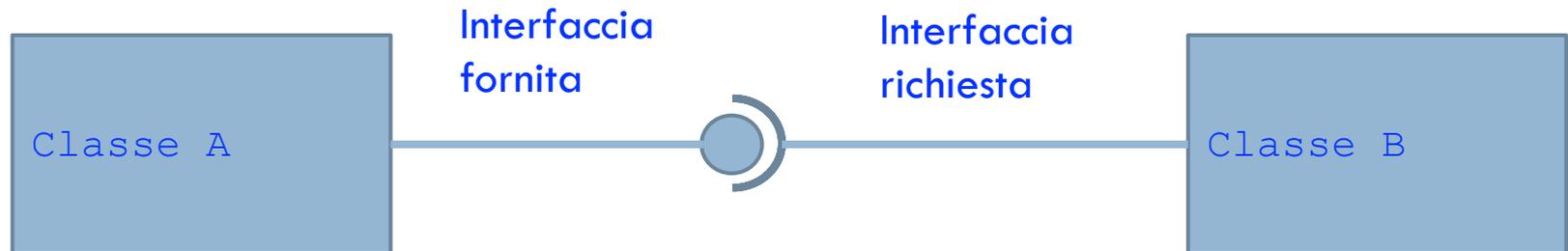
- **Interfaccia**
  - è una classe che **non ha implementazione**
  - presenta solo **dichiarazioni di operazioni**
  
- **Le interfacce sono di due tipi**
  - **Interfaccia fornita**
    - La classe fornisce le operazioni che vi sono dichiarate
  
  - **Interfaccia richiesta**
    - La classe ha bisogno delle operazioni in essa
    - dichiarate per poter svolgere le elaborazioni



# Interfaccia



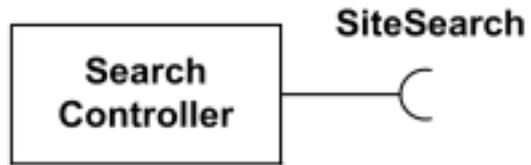
Esempio di interfaccia



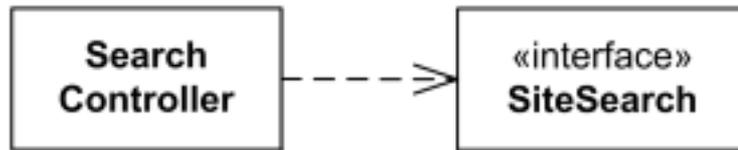
Esempio di interfacce



# Interfaccia



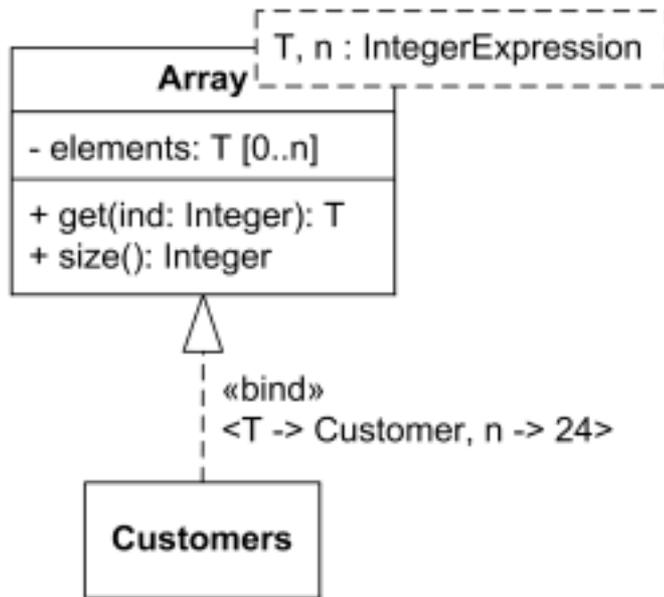
Esempio di interfaccia richiesta



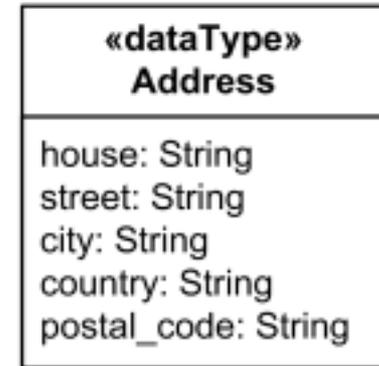
Esempio di interfaccia richiesta con l'interfaccia rappresentata da un rettangolo



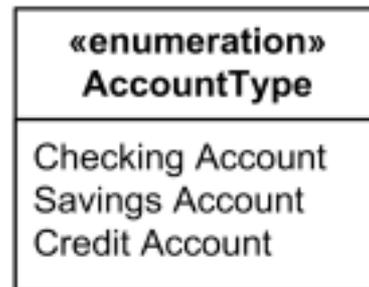
# Template, Enumerazioni e Dato



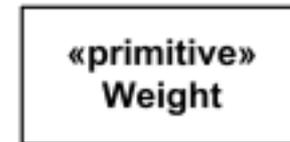
Classe Template



Tipo dato



Enumerazione



Dati primitivi



# Stereotipi di classe

---

## ■ Stereotipi di classe standard

### ■ Focus

- classe principale per gestire un flusso di controllo di più classi

### ■ Auxiliary

- classe ausiliare

### ■ Type

- specifica un dominio per gli oggetti e operazioni applicabili

### ■ Utility

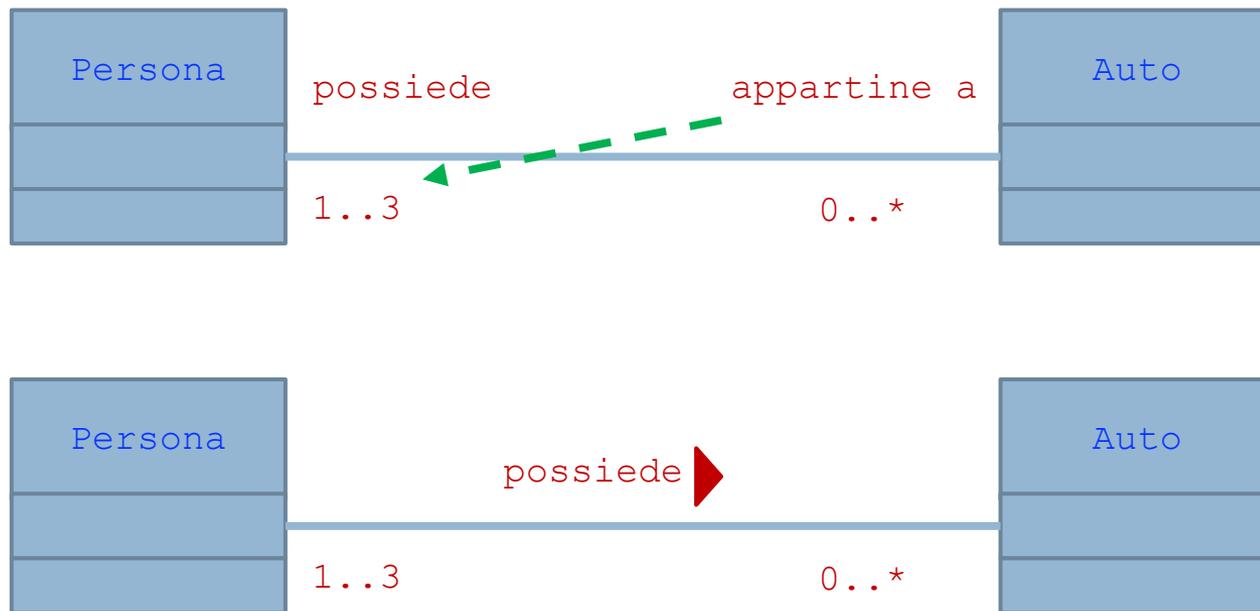
- classe con attributi e operazioni statiche



# Associazioni

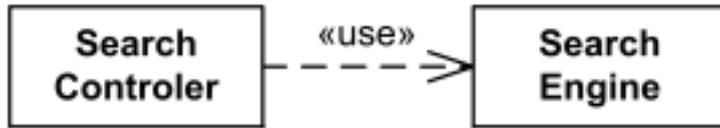
## ■ Associazione

- relazione statica che lega le classi tra di loro

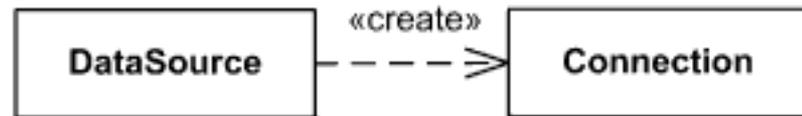


\* numero massimo non definito

# Dipendenza Usage



Relazione di uso



Relazione di creazione



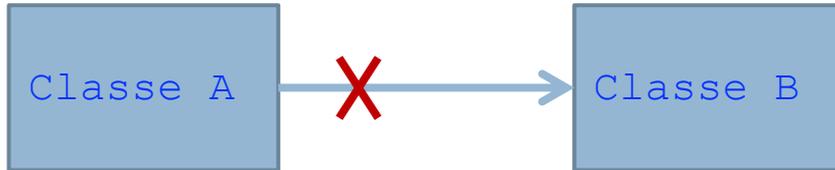
# Navigabilità



Navigabilità unidirezionale



Navigabilità didirezionale



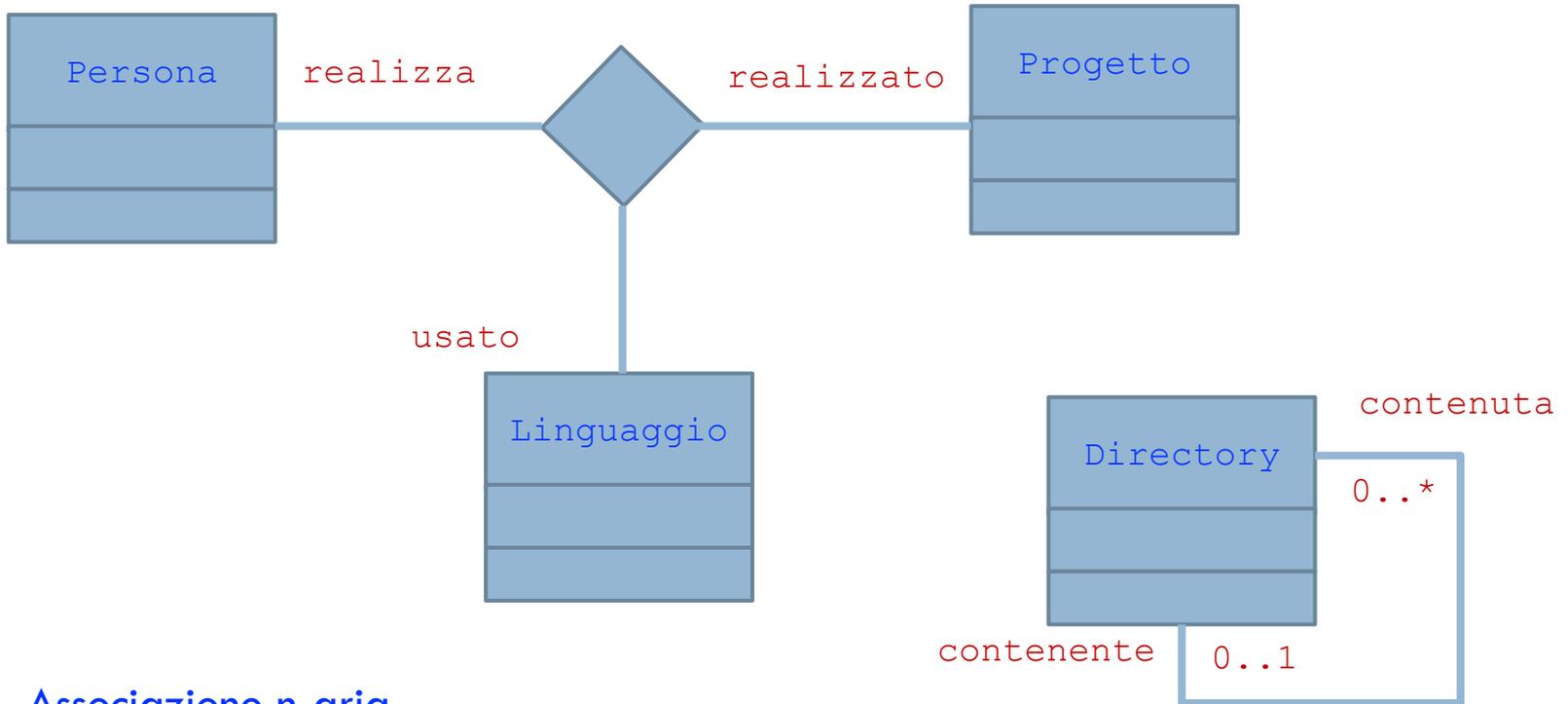
Non navigabilità



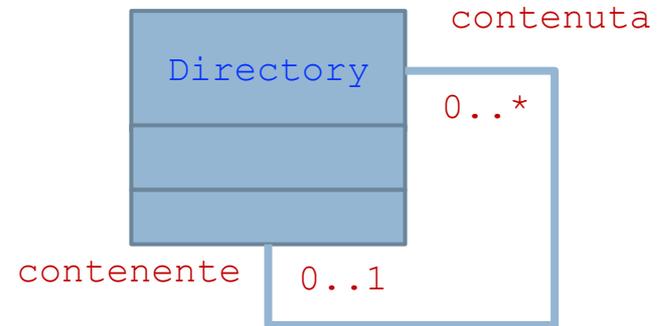
# Associazioni



Associazione binaria

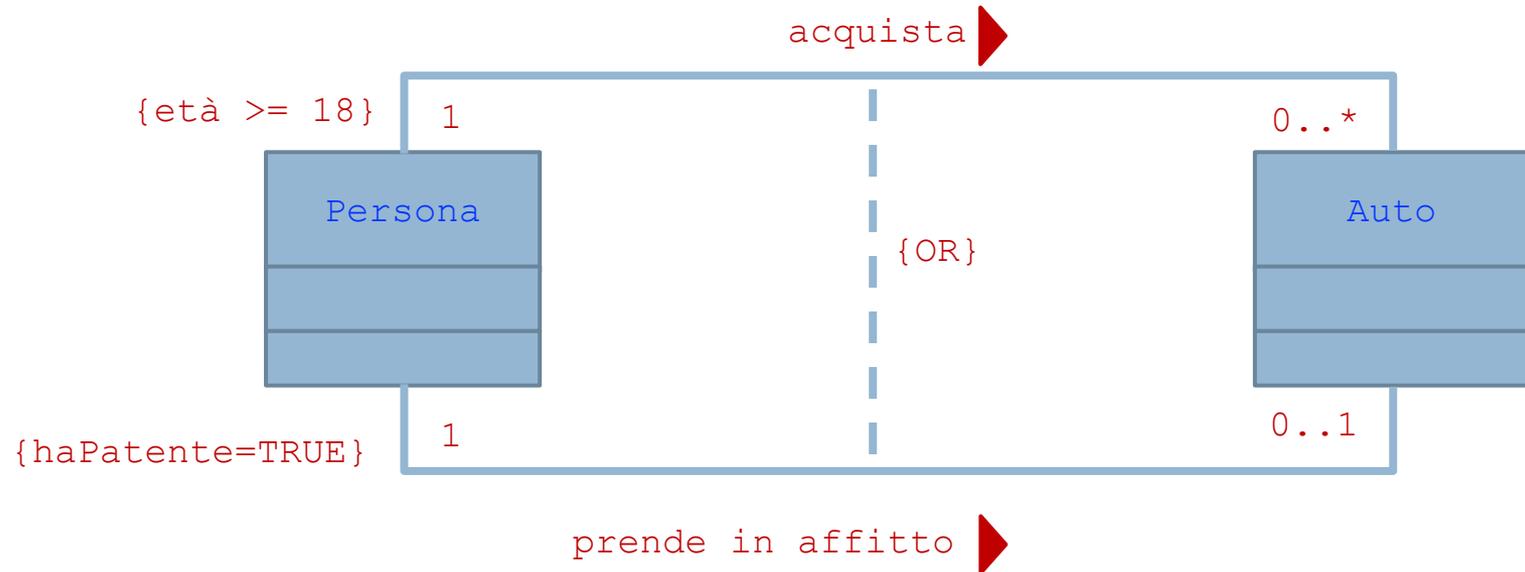


Associazione n-aria



Associazione riflessiva

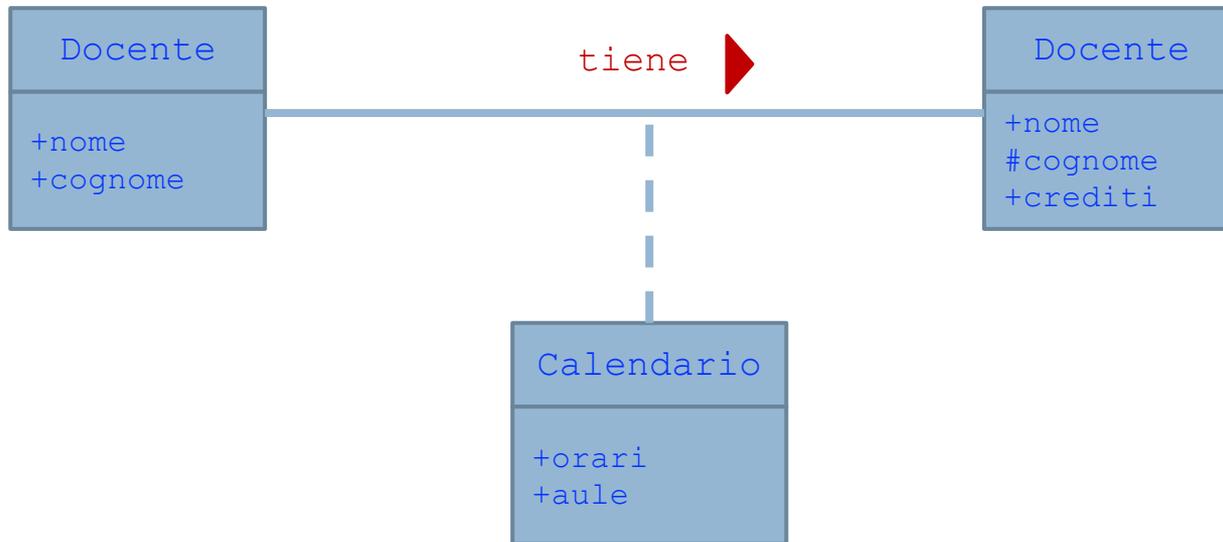
# Vincoli



Vincoli sulle associazioni



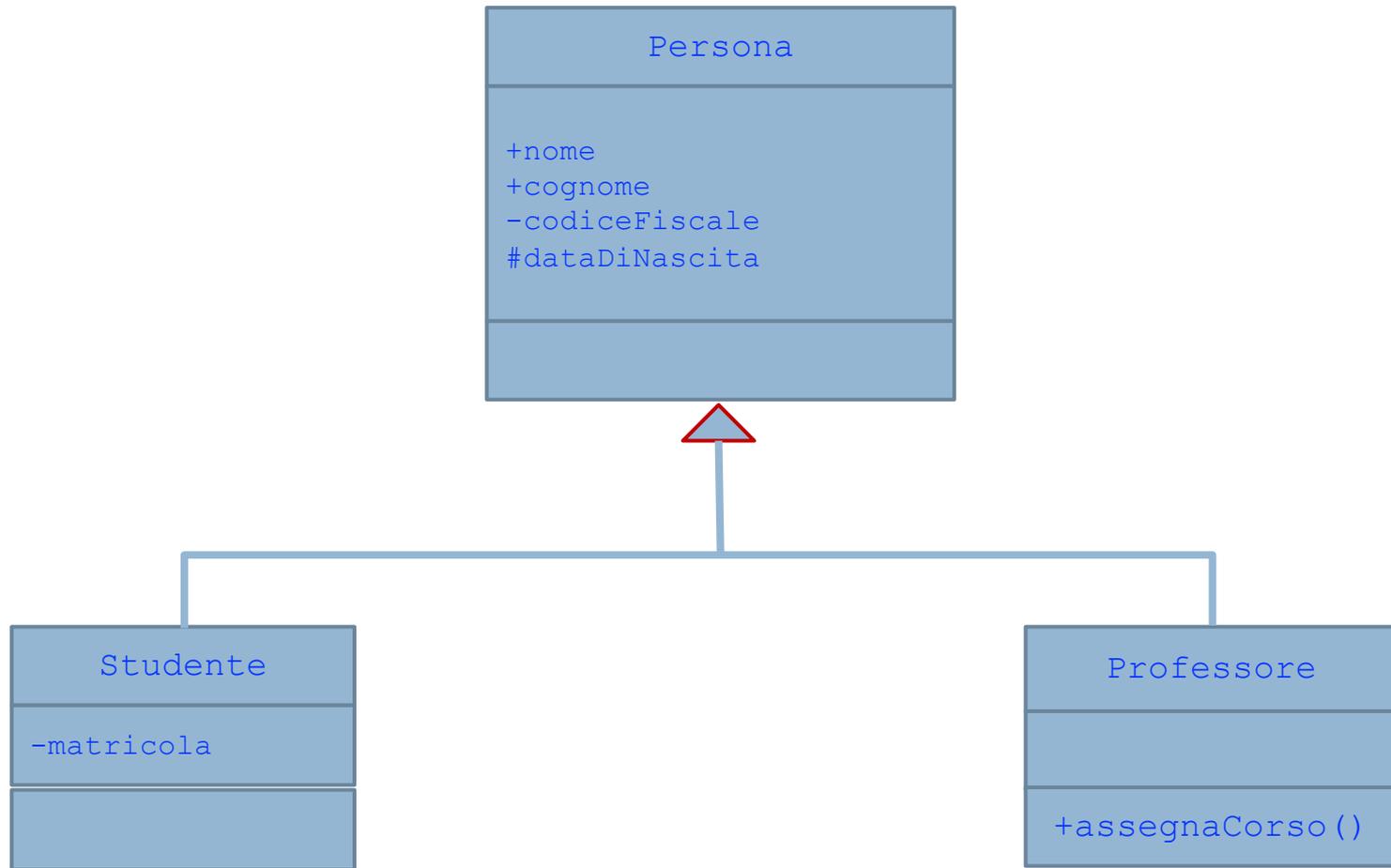
# Association class



Esempio di association class Calendario



# Relazioni



Esempio di generalizzazione



# Generalizzazione e vincoli

---

## ■ Vincoli

### ■ Overlapping

- un'istanza appartiene a due sottoclassi di due set distinti

### ■ Disgiunta

- Un'istanza appartiene ad un'unica sottoclasse

### ■ Completa

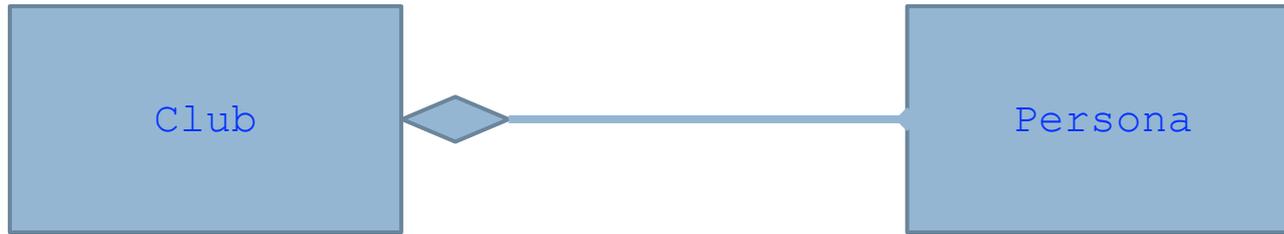
- Tutte le possibili istanze della superclasse appartengono a una delle sottoclassi definite

### ■ Incompleta

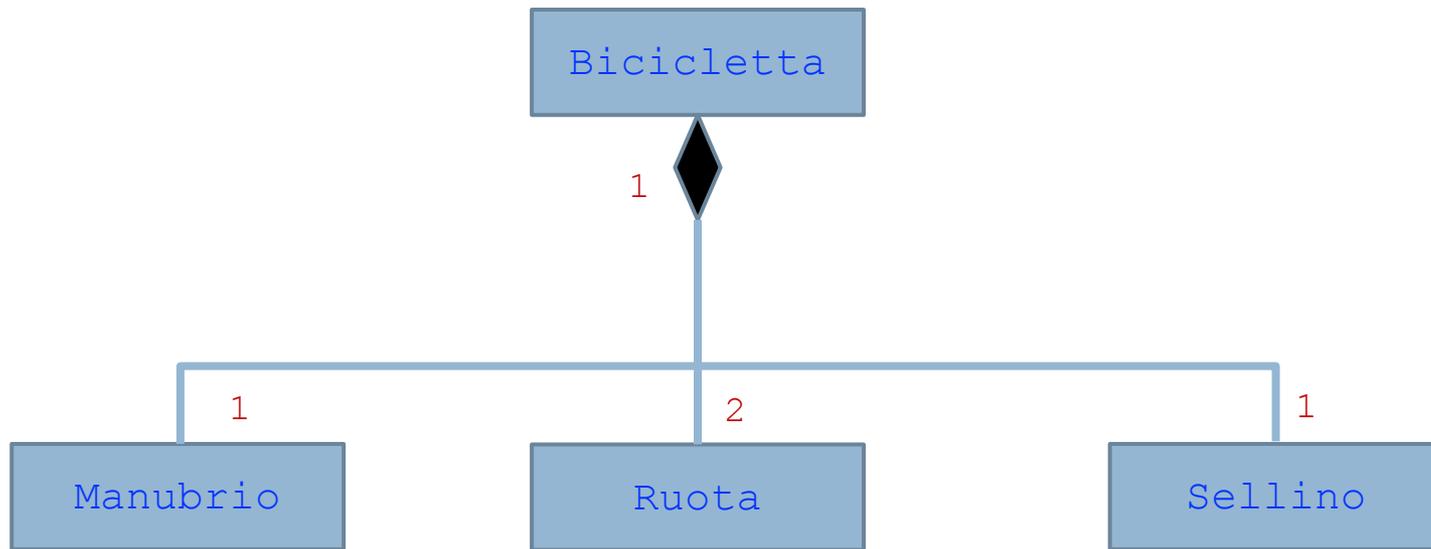
- Almeno un'istanza della superclasse non appartenga alle sottoclassi definite nella generalizzazione



# Composizione e aggregazione



Relazione di aggregazione



Relazione di composizione (solo parti intere)

# Composizione vs Aggregazione

---

## ■ aggregazione

### ■ relazione non forte

- relazione nella quale le classi parte hanno un significato anche senza che sia presente la classe tutto

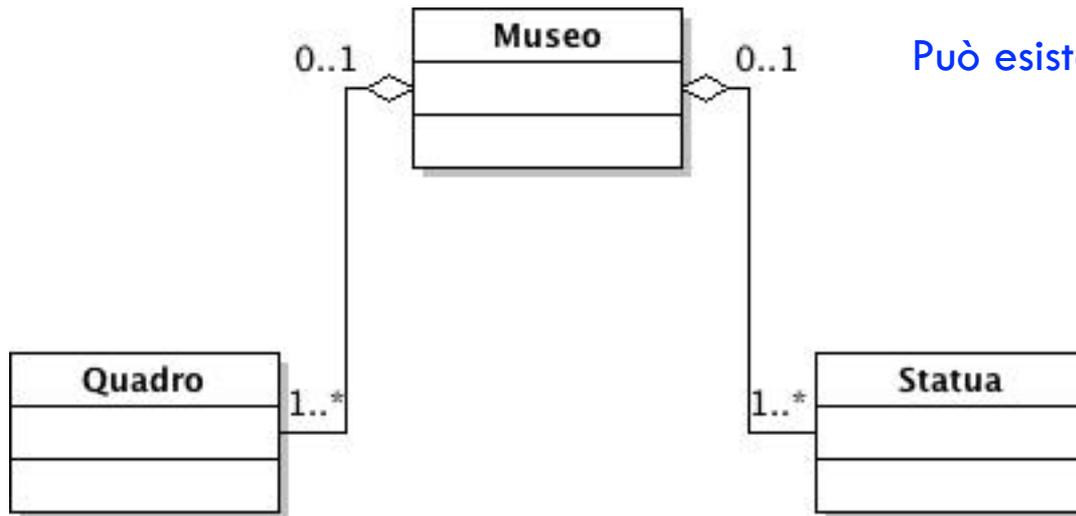
## ■ composizione

### ■ relazione forte

- relazione nella quale le classi parte hanno un reale significato solo se sono legate alla classe tutto

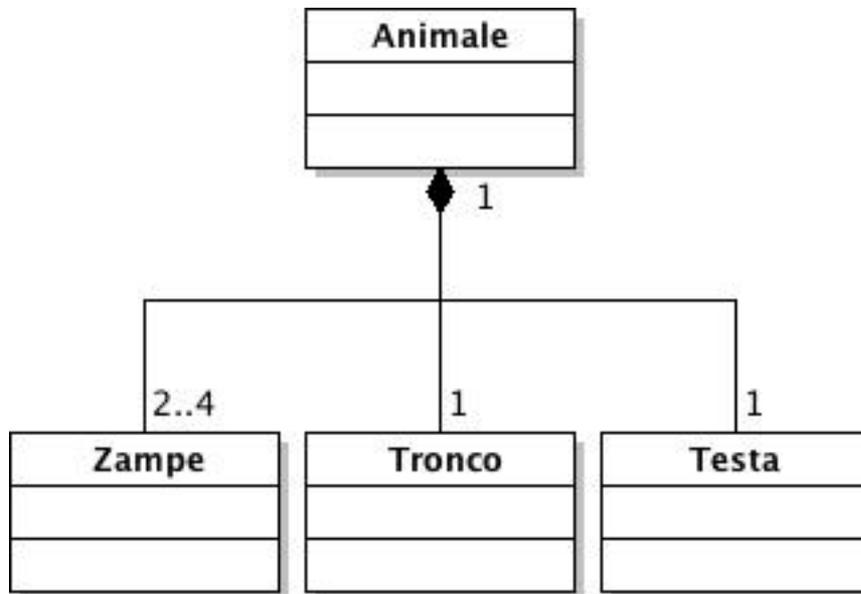


# Composizione vs aggregazione



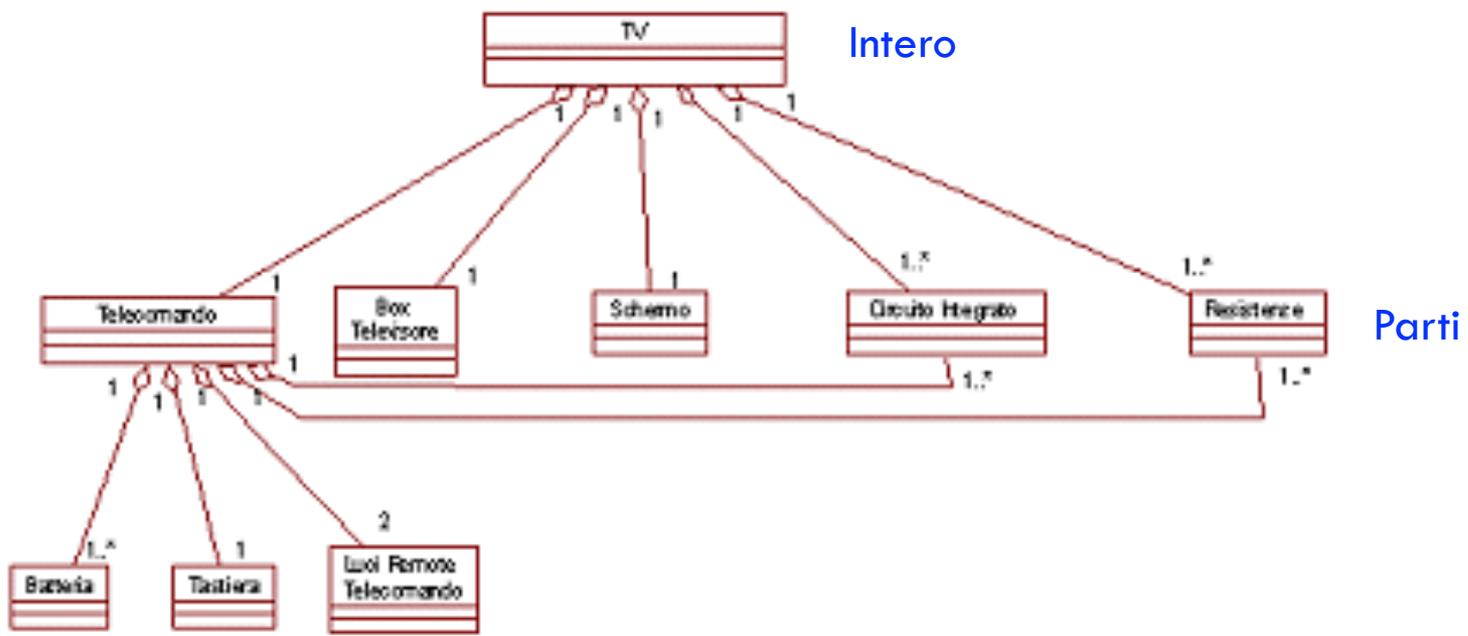
Può esistere senza alcune classi parte

Relazione di aggregazione

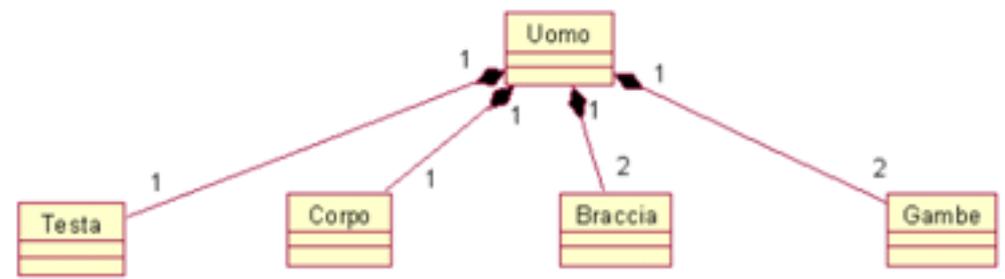


Relazione di composizione  
(appartenenza solo ad un intero)

# Composizione e aggregazione



Relazione di aggregazione



Relazione di composizione (appartenenza solo ad un intero)

# Domain diagram

