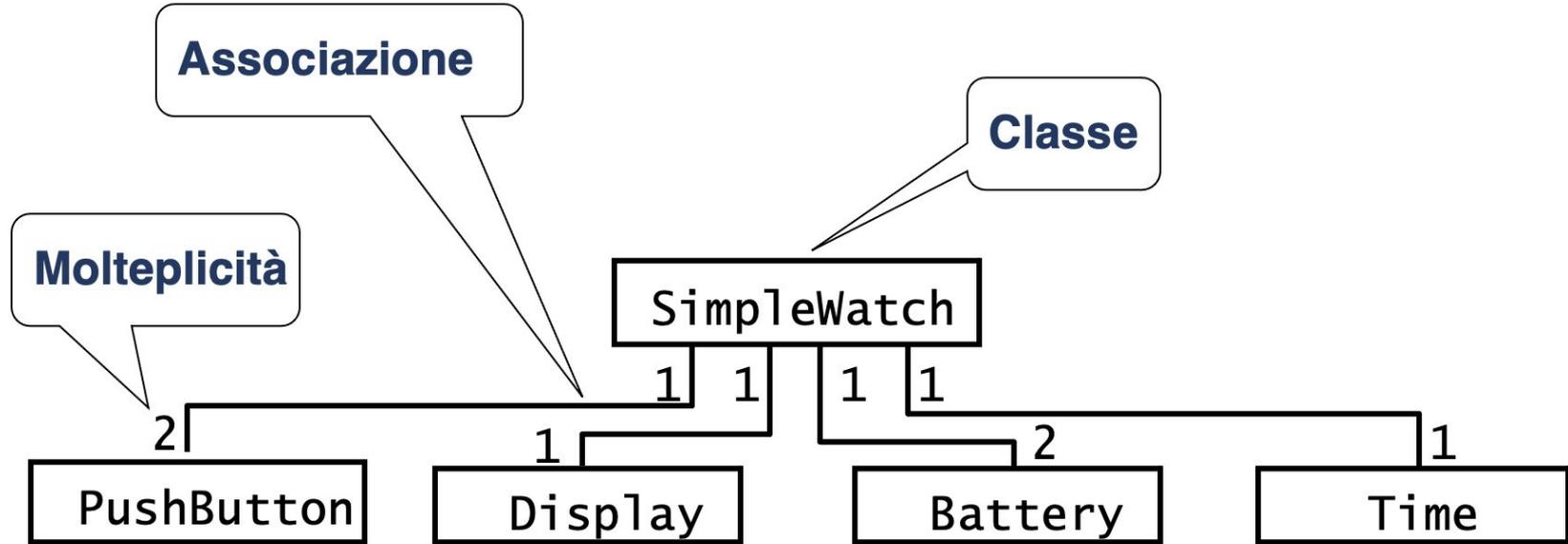


4. Modellazione con UML (II)

Paola Barra
a.a. 2022/2023

Diagramma delle classi

Diagrammi delle classi



I diagrammi delle classi rappresentano la struttura del sistema

Diagramma delle classi

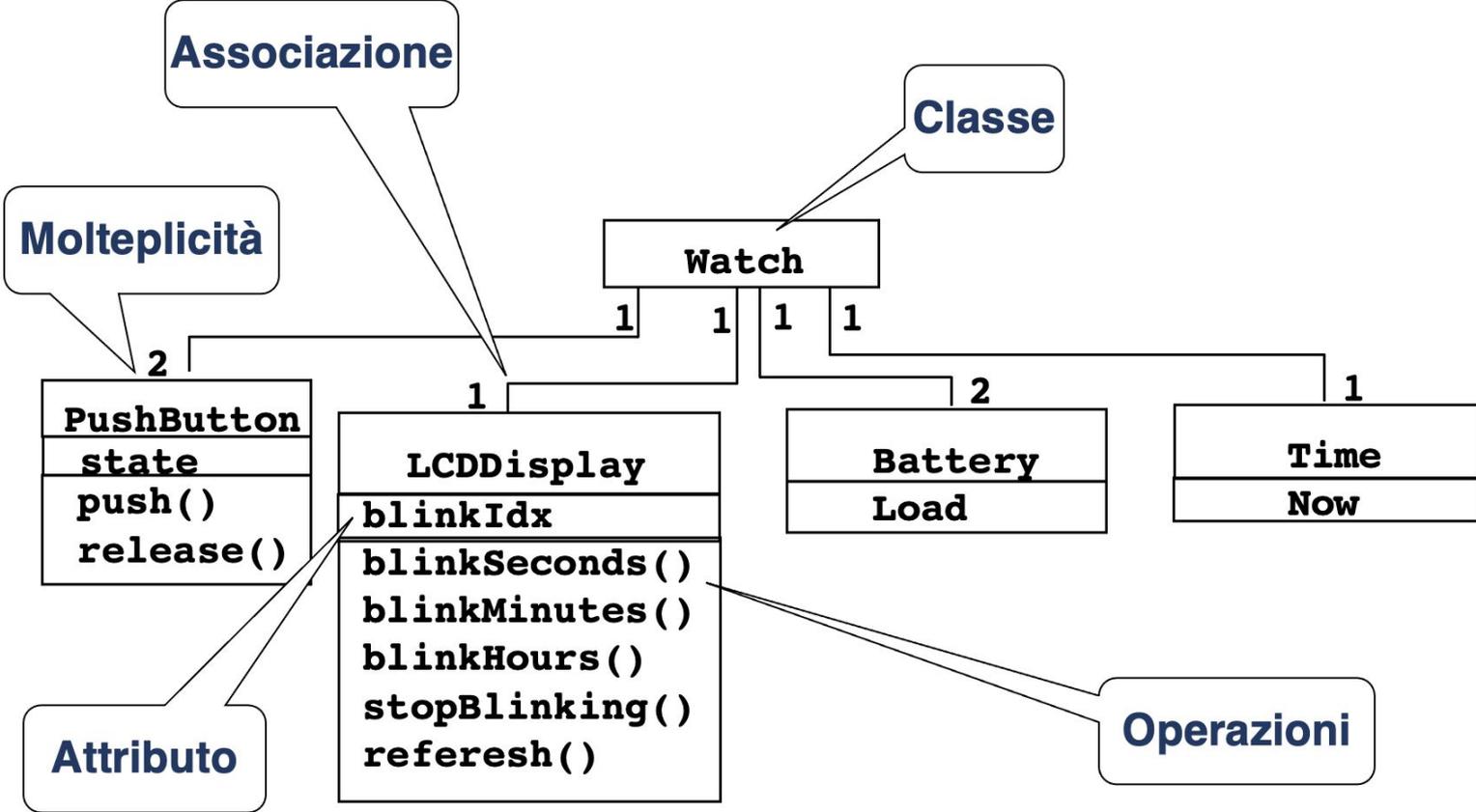


Diagramma delle classi

- Classi ed oggetti
 - I diagrammi delle classi descrivono la struttura del sistema in termini di classi ed oggetti
 - Le **classi** sono astrazioni che specificano gli attributi ed il comportamento di un insieme di oggetti
 - Una classe è una collezione di oggetti che condividono un insieme di attributi che contraddistinguono gli oggetti come membri della collezione
 - Gli **oggetti** sono entità che incapsulano lo stato ed il comportamento
 - Ogni oggetto ha un'identità mediante la quale ci si riferisce ad esso individualmente e che lo distingue dagli altri oggetti
 - In UML, classi ed oggetti sono rappresentati da riquadri composti da tre compartimenti
 - Parte alta: nome della classe o dell'oggetto
 - Centro: attributi
 - Parte bassa: operazioni
 - Le parti relative agli attributi e alle operazioni possono essere omesse per chiarezza

Classi e oggetti

- Un oggetto è un'entità caratterizzata da
 - Un'identità
 - Uno stato
 - Un comportamento
- Una classe descrive
 - un insieme di oggetti con caratteristiche simili
 - cioè oggetti che hanno lo stesso tipo

Convenzioni UML per classi e oggetti

- I nomi degli oggetti sono sottolineati per indicare che sono istanze
- I nomi delle classi iniziano con lettere maiuscole
- Agli oggetti, nei diagrammi degli oggetti, possono essere assegnati dei nomi (seguiti dalla loro classe) per semplificarne il riferimento
 - In questo caso, i nomi iniziano con lettere minuscole

Attore vs classe vs oggetto

- Attore

- Un'entità da modellare esterna al sistema, interagisce con il sistema ('passenger')

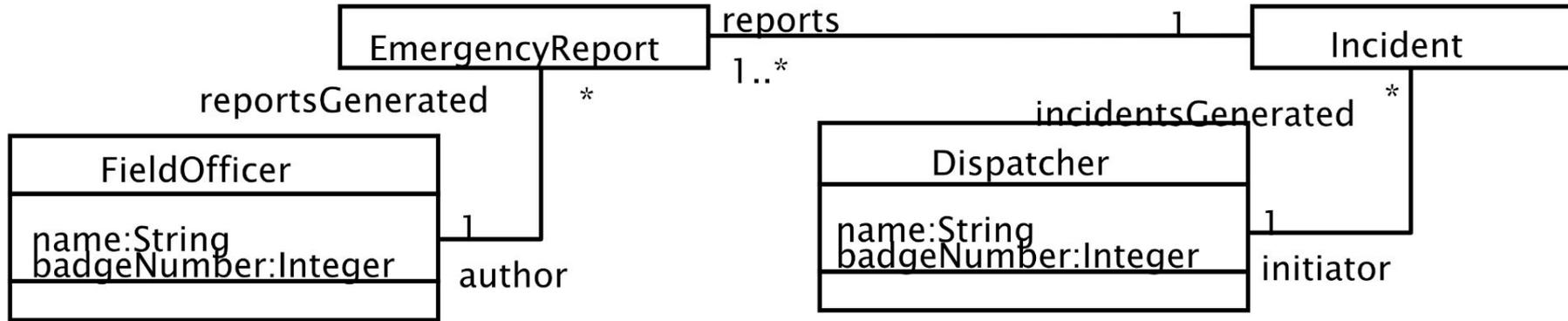
- Classe

- Un'astrazione che modella un'entità nel dominio applicativo o del dominio della soluzione ('User', 'Ticket distributor', 'Server')

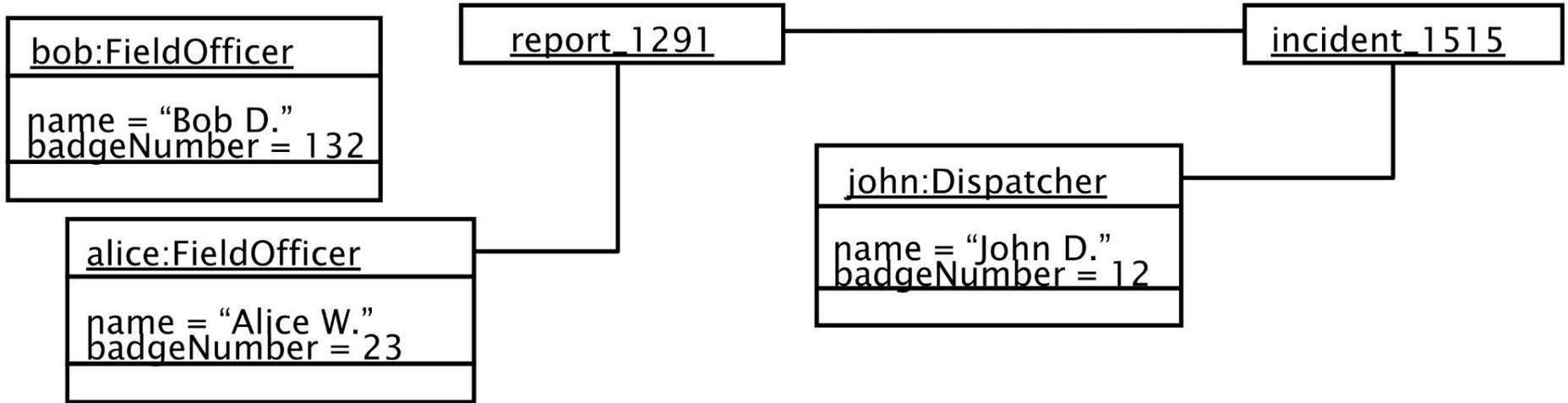
- Oggetto

- Una specifica istanza di una classe ('Joe, il passeggero che sta comprando un biglietto dal distributore di biglietti')

Esempio di diagramm delle classi: classi partecipanti nel caso d'uso "Report Emergency"



Esempio di diagramma degli oggetti: oggetti partecipanti nello scenario “warehouseOnFire”

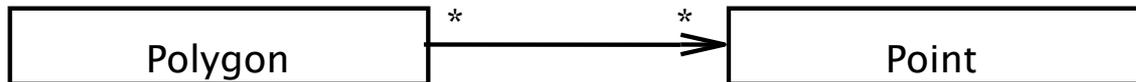


Associazioni e collegamenti (link)

- Un **collegamento** o **link** rappresenta una connessione tra due oggetti
- Le **associazioni** sono relazioni tra classi e rappresentano gruppi di link
- Nell'esempio *FRIEND*, ogni oggetto *FieldOfficer* ha anche una lista di *EmergencyReport* che sono stati scritti dal *FieldOfficer*
 - Nell'esempio del diagramma delle classi, la linea tra la classe *FieldOfficer* e la classe *EmergencyReport* è un'associazione
 - Nell'esempio del diagramma degli oggetti, la linea tra l'oggetto *alice:FieldOfficer* e l'oggetto *report_1291:EmergencyReport* è un link
 - Rappresenta uno stato del sistema per cui *alice:FieldOfficer* ha generato *report_1291:EmergencyReport*

Associazioni simmetriche e asimmetriche

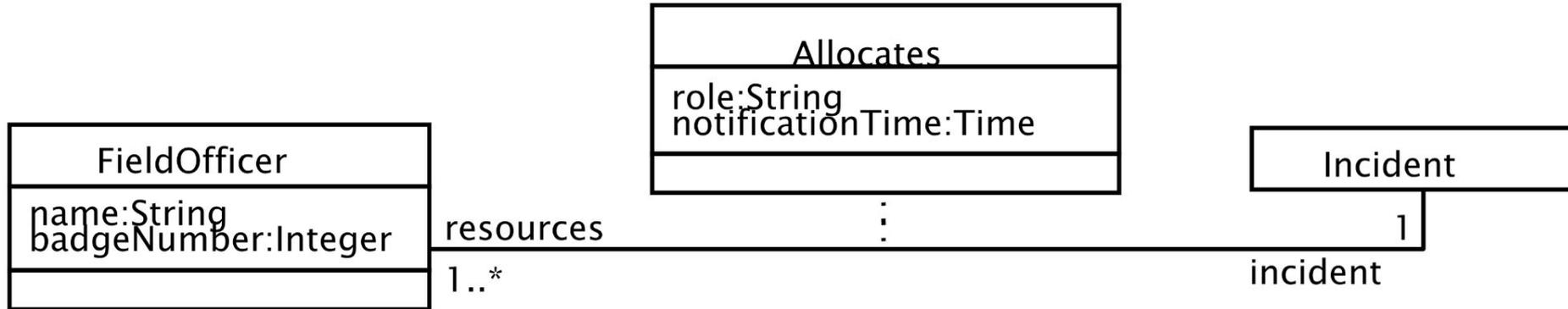
- Le associazioni possono essere bidirezionali (simmetriche) o unidirezionali (asimmetriche)
 - Nei diagrammi precedenti sono tutte simmetriche
- Una associazione asimmetrica è, ad esempio, quella tra le classi *Poligono* e *Punto*
 - La freccia di navigazione indica che il sistema supporta solo il verso da poligono a punto
 - Dato un poligono specifico, è possibile individuare tutti i punti che costituiscono il poligono. Dato un punto specifico, non è possibile individuare il poligono di cui il punto fa parte. Per convenzione, le associazioni senza frecce sono simmetriche



Classe di associazione

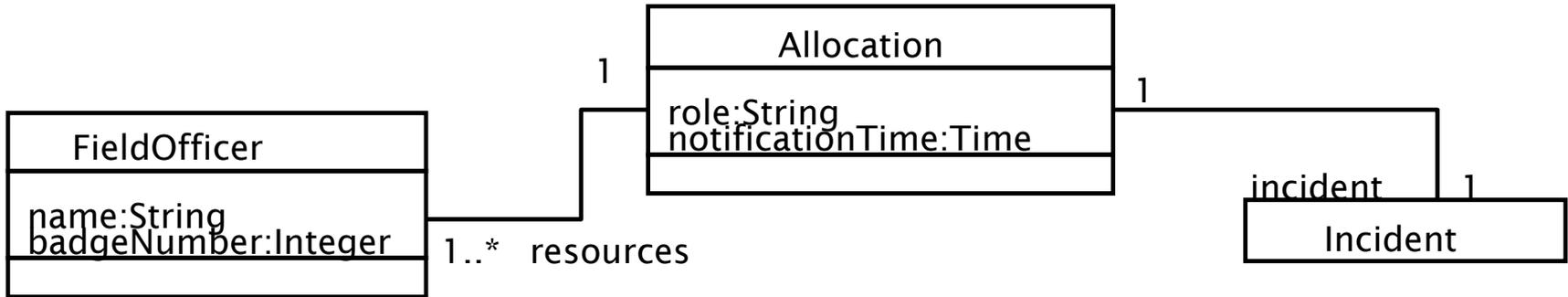
- Le associazioni sono simili alle classi poiché possono avere attributi ed operazioni
- Una tale associazione è chiamata classe di associazione
 - Disegnata con un simbolo di classe che contiene attributi e operazioni ed è connessa al simbolo di associazione con una linea tratteggiata
 - Ad esempio, l'allocazione dei *FieldOfficer* ad un Incidente è modellata come una classe di associazione con attributi ruolo (*role*) e ora di notifica (*notificationTime*)

Esempio di classe di associazione



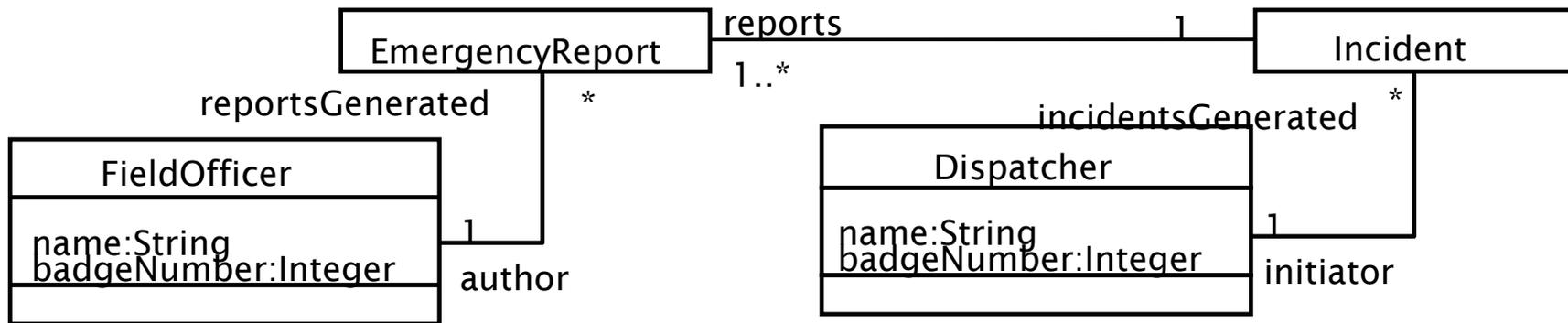
Modello alternativo di Allocation

Qualsiasi classe di associazione può essere trasformata in una classe e associazioni semplici



Ruoli

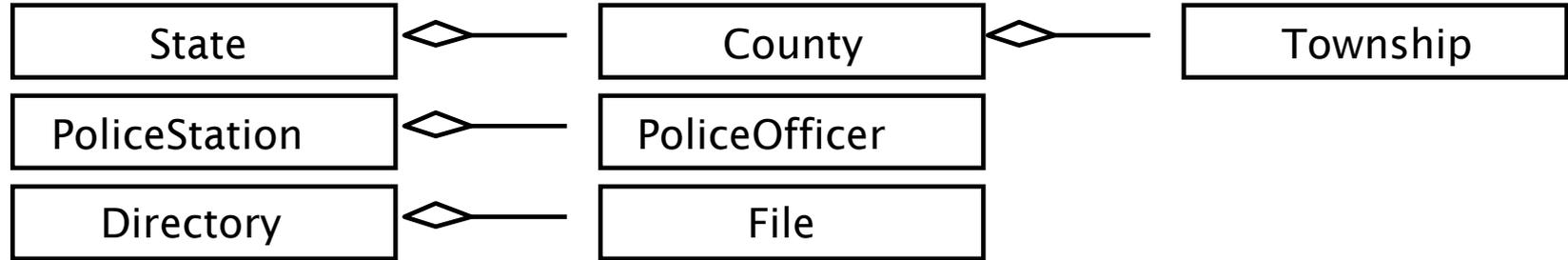
- Ciascuna estremità di un'associazione può essere etichettata con una stringa chiamata **ruolo**
- I ruoli dell'associazione tra le classi *EmergencyReport* e *FieldOfficer* sono autore (*author*) e rapporto generato (*reportGenerated*)
 - Etichettare le estremità delle associazioni con i ruoli consente di distinguere tra le multiple associazioni che si originano da una classe. Inoltre, i ruoli chiariscono lo scopo dell'associazione



Aggregazione

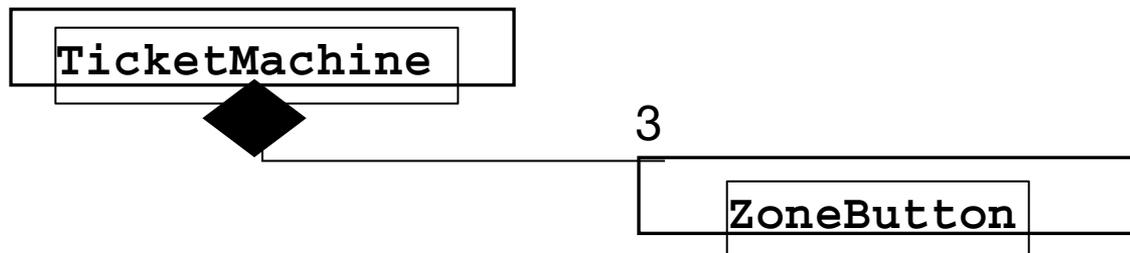
- Le associazioni sono usate per rappresentare un'ampia gamma di connessioni tra un insieme di oggetti
- Un tipo speciale di associazione si presenta frequentemente: le **aggregazioni** (denotate con una linea con testa di diamante)
 - Esempi:
 - uno *stato* contiene molti *paesi* che a loro volta contengono molte *città*
 - una *stazione di polizia* è costituita di un certo numero di *poliziotti*
 - una *directory* contiene un certo numero di *file*
 - Tali relazioni possono essere modellate con associazioni uno-a-molti
 - UML, invece, fornisce il concetto di aggregazione che consente di denotare aspetti gerarchici della relazione che può avere molteplicità sia uno-a-molti che multi-a-molti

Esempi di Aggregazione



Composizione

- Un rombo solido denota una composizione
 - Una forma forte di aggregazione dove il tempo di vita delle istanze component è controllato dall'aggregato
 - Le parti non hanno esistenza autonoma ("il tutto controlla /distrugge le parti")

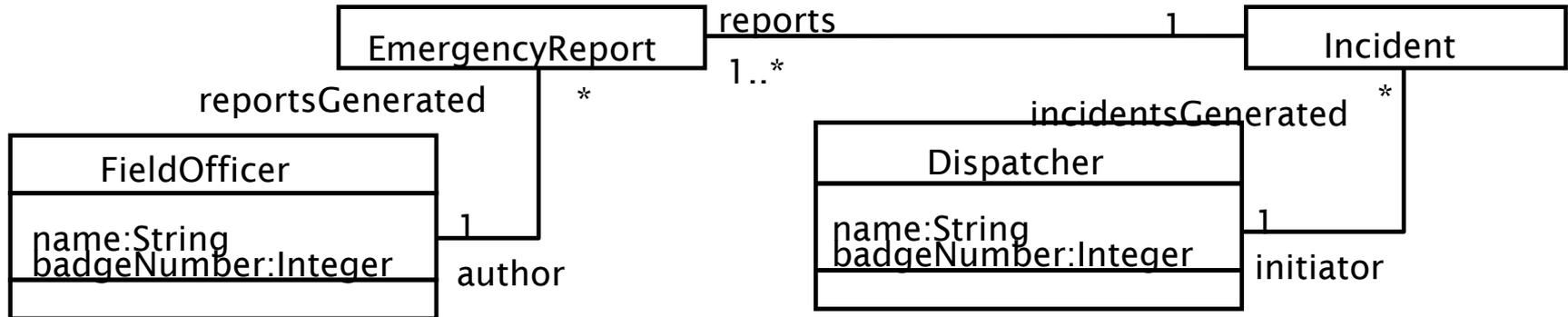


Molteplicità

- Ogni estremità di un'associazione può essere etichettata con un insieme di interi che indicano il numero di link che si originano da un'istanza della classe connessa all'estremità dell'associazione
- Questo insieme di interi è chiamato **molteplicità** dell'estremità dell'associazione

Esempio di molteplicità

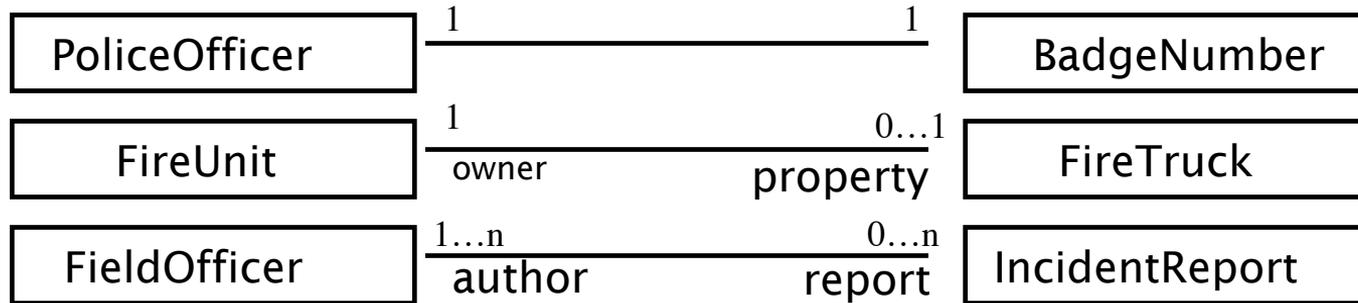
- L'estremità dell'associazione *author* ha una molteplicità pari ad 1
 - Significa che tutti gli *EmergencyReport* sono scritti esattamente da un *FieldOfficer* vale a dire, ogni oggetto *EmergencyReport* ha esattamente un link ad un oggetto della classe *FieldOfficer*
- La molteplicità dell'estremità dell'associazione *reportsGenerated* è “molti” ed indicata con un asterisco (*) che indica 0..n



Tipi di molteplicità

- Le associazioni solitamente usate nei diagrammi sono di tre tipi:
 - **Associazione uno-a-uno**
 - Ha molteplicità 1 su entrambe le estremità: esiste esattamente un link tra le istanze di ogni classe
 - **Associazione uno-a-molti**
 - Ha molteplicità 1 su di una estremità e 0..n o 1..n dall'altra estremità: denota il rapporto di composizione tra due classi
 - **Associazione multi-a-molti**
 - Ha molteplicità 0..n o 1..n su ambo i lati: indica che un numero arbitrario di link possono esserci tra le istanze di due classi

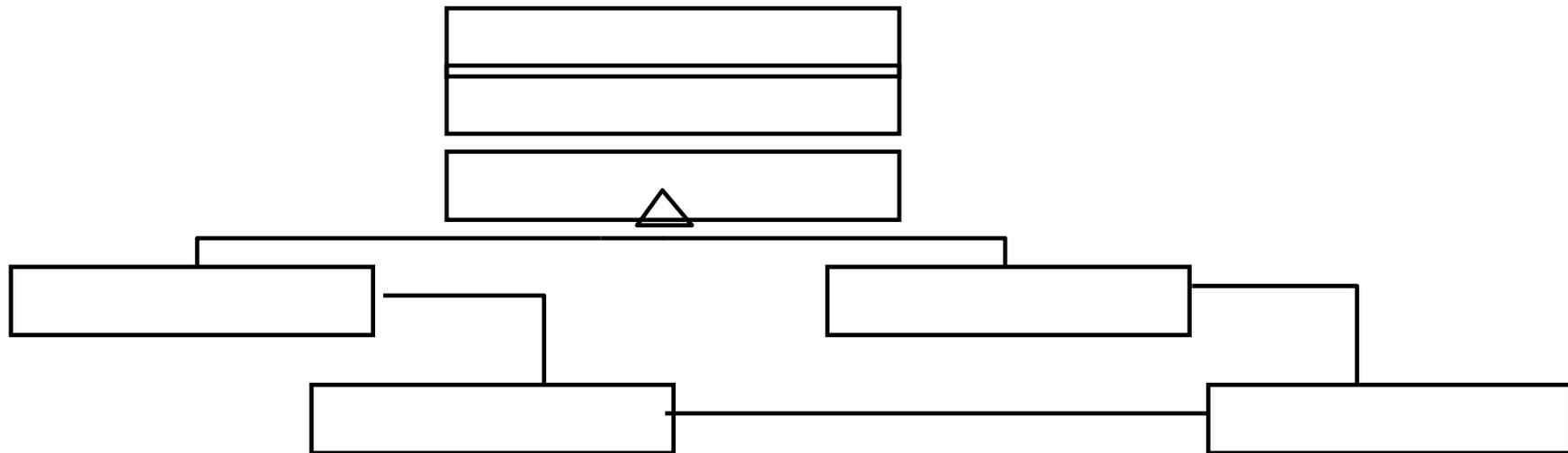
Esempi di molteplicità



Ereditarietà

- L'**ereditarietà** è la relazione tra una classe generale ed una o più classi specializzate
- L'ereditarietà consente di descrivere tutti gli attributi e le operazioni che sono comuni ad un insieme di classi
 - Esempio: *FieldOfficer* e *Dispatcher* hanno entrambi gli attributi *name* e *badgeNumber*. Tuttavia, *FieldOfficer* ha un'associazione con *EmergencyReport*, mentre *Dispatcher* ha un'associazione con *Incident*. Gli attributi comuni di *FieldOfficer* e *Dispatcher* possono essere modellati introducendo una classe *PoliceOfficer* che è specializzata da *FieldOfficer* e *Dispatcher*
 - *PoliceOfficer* è chiamata la **superclasse**; *FieldOfficer* e *Dispatcher* sono chiamate **sottoclassi**
- Le sottoclassi ereditano gli attributi e le operazioni della loro superclasse

Esempi di generalizzazione



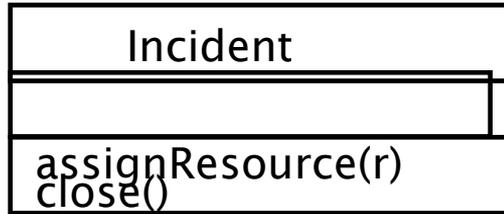
Classi astratte

- La classe *PoliceOfficer* è una classe astratta. Per distinguerla dalle classi concrete si scrive il nome in corsivo
 - Le classi astratte sono utilizzate nella modellazione orientata agli oggetti per classificare concetti collegati riducendo, quindi, la complessità totale del modello
 - Eliminano la ridondanza

Oggetti e operazioni

- Il comportamento di un oggetto è specificato dalle **operazioni**
- Un oggetto richiede l'esecuzione di un'operazione ad un altro oggetto inviandogli un messaggio
- Il messaggio è confrontato con il metodo definito dalla classe a cui l'oggetto ricevente appartiene o da una qualsiasi sua superclasse
- I metodi di una classe in un linguaggio di programmazione orientato agli oggetti sono le implementazioni di queste operazioni

Esempio di operazioni della classe **Incident**



Applicazione dei diagrammi delle classi

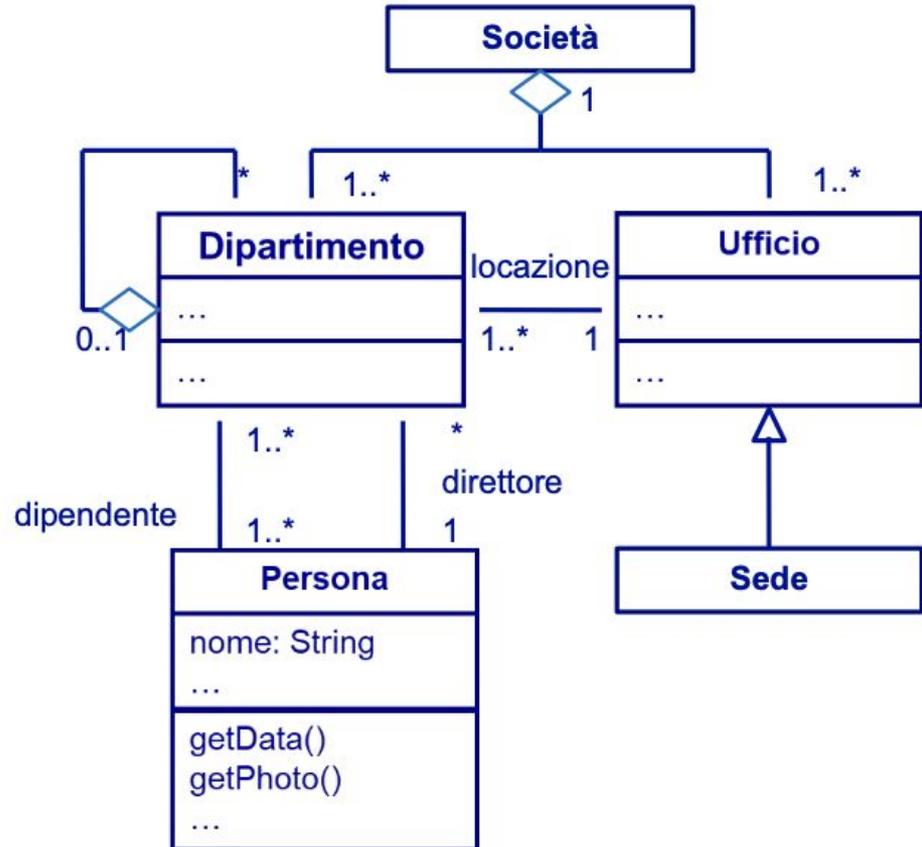
- Usati per descrivere la struttura del sistema
- Durante la fase di analisi gli ingegneri (del software) costruiscono diagrammi delle classi per formalizzare la conoscenza del dominio dell'applicazione
- Le classi rappresentano gli oggetti partecipanti individuati nei diagrammi dei casi d'uso e di interazione
 - Descrivono i loro attributi e le operazioni
- Lo scopo dei modelli di analisi è di descrivere il proposito del sistema e scoprire i suoi confini
 - Ad esempio, un analista può esaminare la molteplicità dell'associazione tra *FieldOfficer* e *EmergencyReport* e chiedere all'utente se ciò è corretto
 - E' possibile avere più di un autore per l'*EmergencyReport*?
 - Sono previsti rapporti anonimi?
- La fase di analisi tiene fuori concetti implementativi
 - I diagrammi di classe sono rifiniti durante la fase di progettazione del sistema e degli oggetti

Diagramma delle classi

- Una classe cattura un concetto nel dominio del problema o della realizzazione
- Il diagramma delle classi descrive:
 - Il tipo degli oggetti che fanno parte di un sistema sw o del suo dominio
 - Le relazioni statiche tra essi: **gli elementi e le relazioni tra essi non cambiano nel tempo**
- I diagrammi delle classi mostrano anche le proprietà e le operazioni di una classe

ESEMPIO: classi

- Una società è formata da dipartimenti e uffici
- Un dipartimento ha un direttore e più dipendenti
- Un dipartimento è situato in un ufficio
- Esiste una struttura gerarchica dei dipartimenti
- Le sedi sono uffici



Sintassi della classe

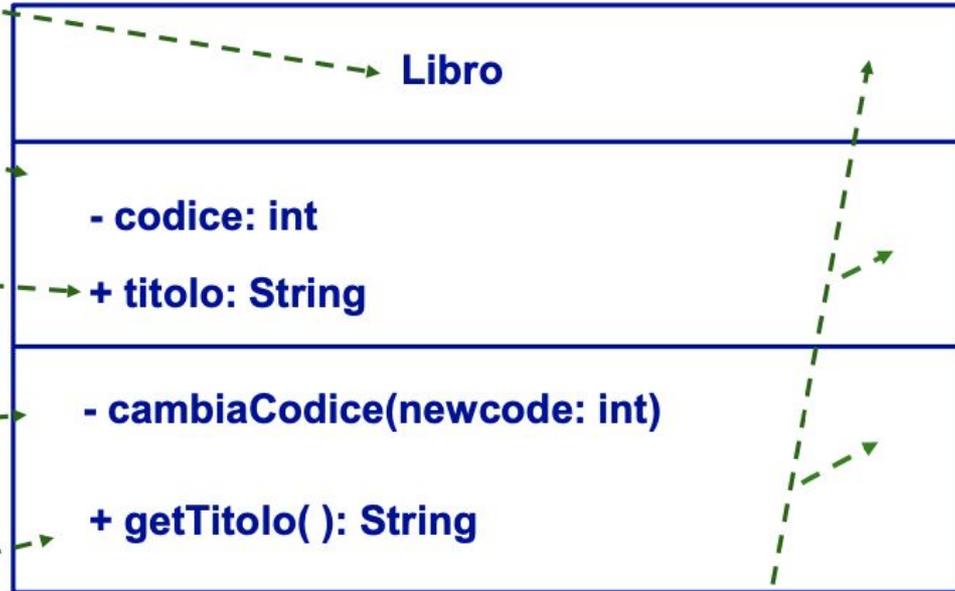
nome (maiuscolo e sempre al singolare)

attributo privato

attributo pubblico

operazione privata

operazione pubblica



sottosezioni
(compartments)

Usi del diagramma delle classi

Il diagramma delle classi puo' essere usato

- a diversi livelli di dettaglio
- in diverse fasi del progetto
- fino alla generazione del codice in linguaggi OO

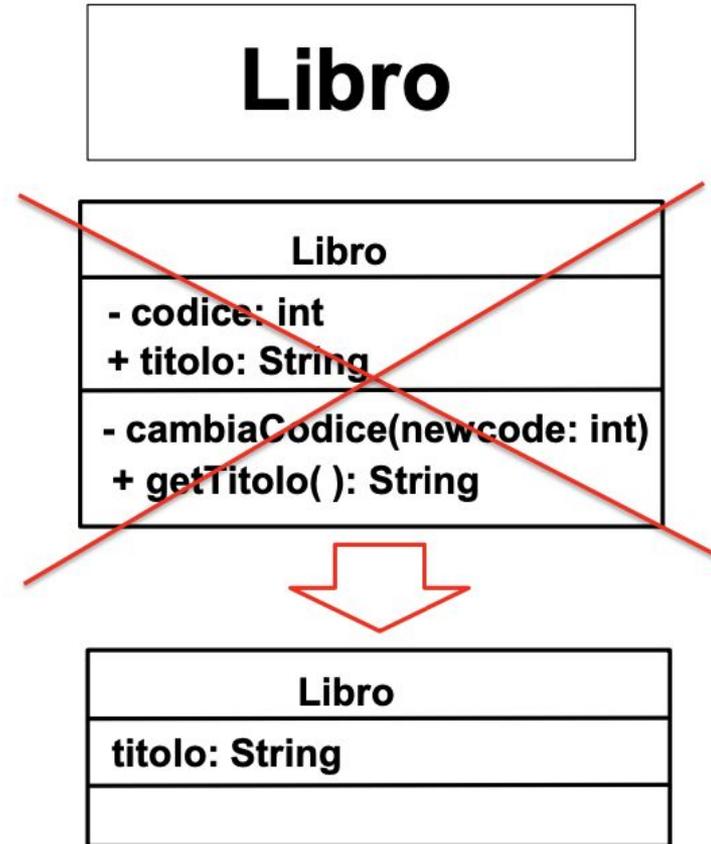
Semantica

- Un oggetto è un'entità caratterizzata da
 - Un'identità, uno stato, un comportamento
- (I valori de)gli attributi definiscono lo stato dell'oggetto
- Le operazioni definiscono il suo comportamento

Non è obbligatorio indicare attributi e operazioni

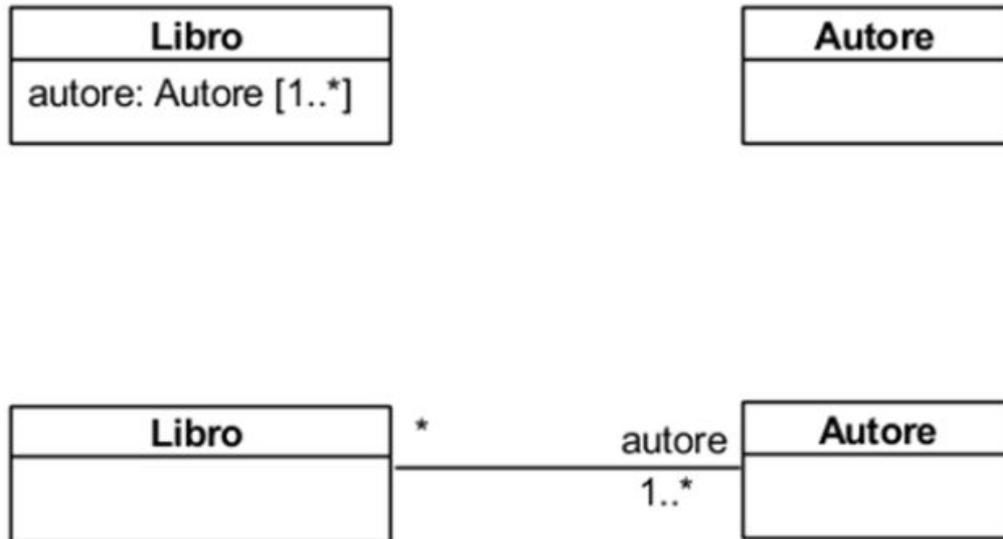
Quando si usa il diagramma delle classi per descrivere il dominio:

- le operazioni sono ritenute a un livello di dettaglio eccessivo e normalmente si omettono
 - In particolare MAI setters e getters
- gli attributi utili per caratterizzare l'elemento del dominio si specificano, dettagli implementativi no
- Le visibilità si omettono



Associazioni e attributi

Altro modo per rappresentare una proprietà



Relazioni

- Tra elementi di un modello
- Vedremo le seguenti:

Tra Classificatori	Tra oggetti
Associazione Aggregazione Composizione	Collegamento Aggergazione Composizione
Generalizzazione	(non definita)
Realizzazione	(non definita)
Dipendenza (d'uso, di istanza...)	

Associazione (binari): sintassi

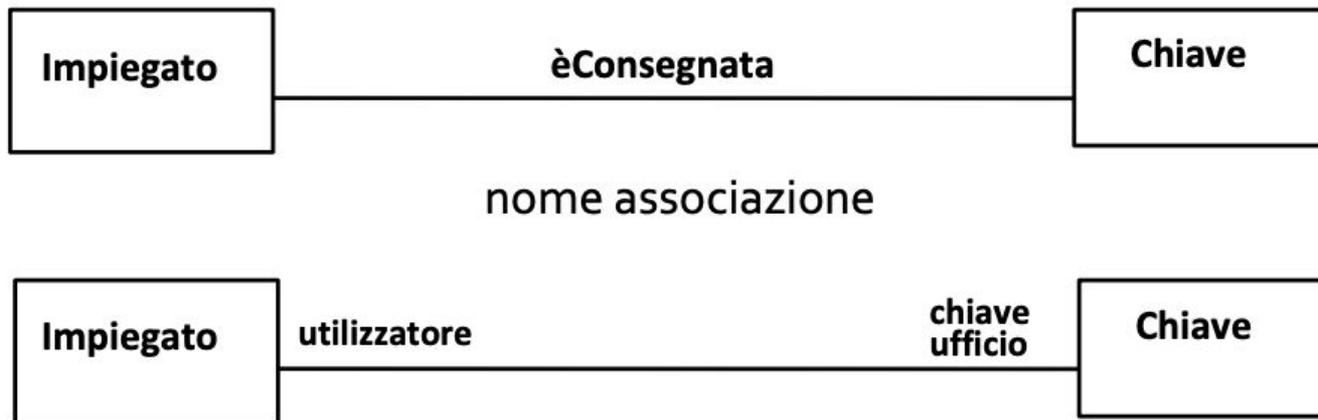


- Una linea retta tra le classi
 - A volte, e solo documentando il codice (non il dominio) una freccia → (eventualmente doppia) per specificare la navigabilità
- Almeno uno tra nome o ruoli, raramente entrambi.
 - Servono a caratterizzare la relazione

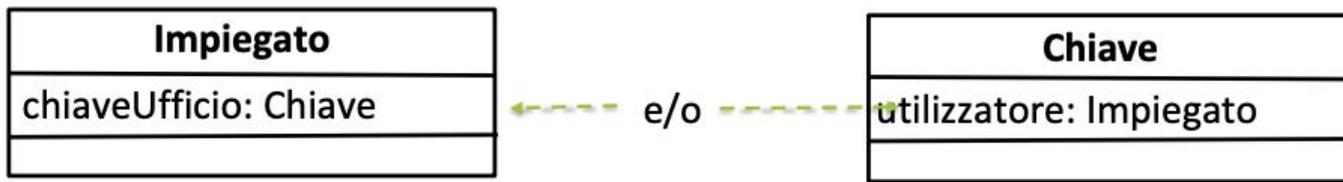
Associazione (binaria): nome e ruoli

- nome e ruoli: lowercase
- nome associazione: normalmente un verbo
eventualmente seguito da una freccia per capire da che parte si legge
- ruolo: normalmente un sostantivo
- Formalmente opzionali,
 - è utile ci sia o il nome dell'associazione o l'indicazione dei ruoli
 - inutile entrambi

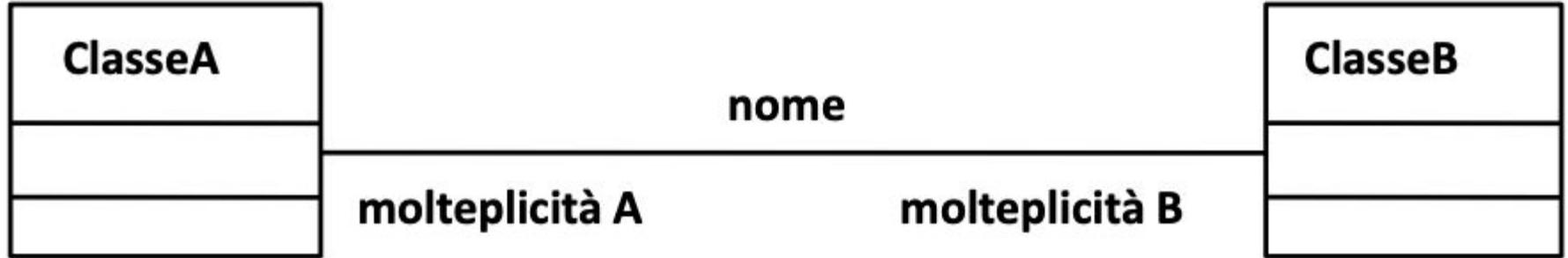
Associazione (binaria): esempio



- Si esplicitano i ruoli degli oggetti nella relazione
 - c è la chiave dell'ufficio di i
 - i ne è l'utilizzatore
- Quando si trasforma il modello in codice:



Associazioni: vincoli di molteplicità



**Numero di oggetti coinvolti nell'associazione
in un dato istante**

Molteplicità, un esempio



- Un oggetto Società può essere in relazione con molti oggetti Lavoratore
- Un oggetto Lavoratore può essere in relazione con un solo oggetto Società

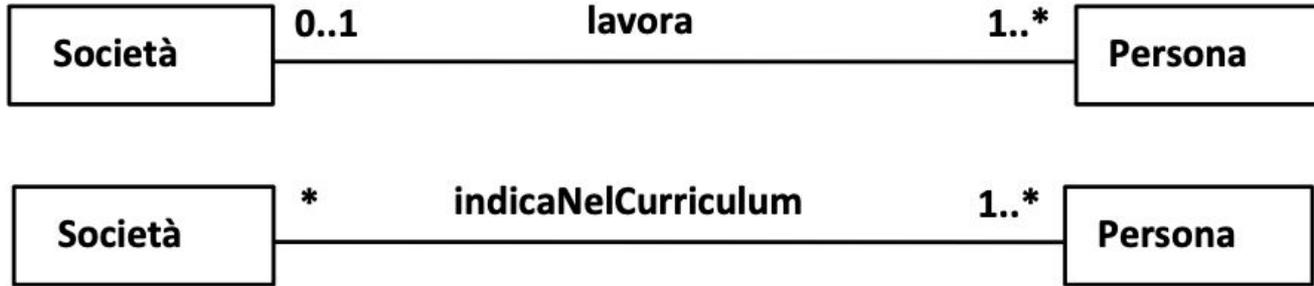
in un dato istante

Molteplicità delle relazioni

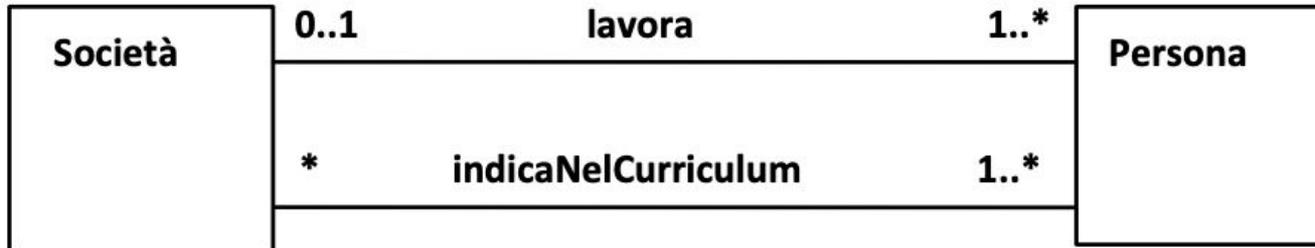
Le molteplicità si possono definire indicando gli estremi inferiore e superiore di un intervallo

- Esempio $2..4$ per le ruote di un veicolo
- l'estremo inferiore può essere zero o un numero positivo
- quello superiore un numero positivo o $*$ (indefinito)
- $n..n \equiv n$
- $0..* \equiv *$
- 1 è il default e si può omettere

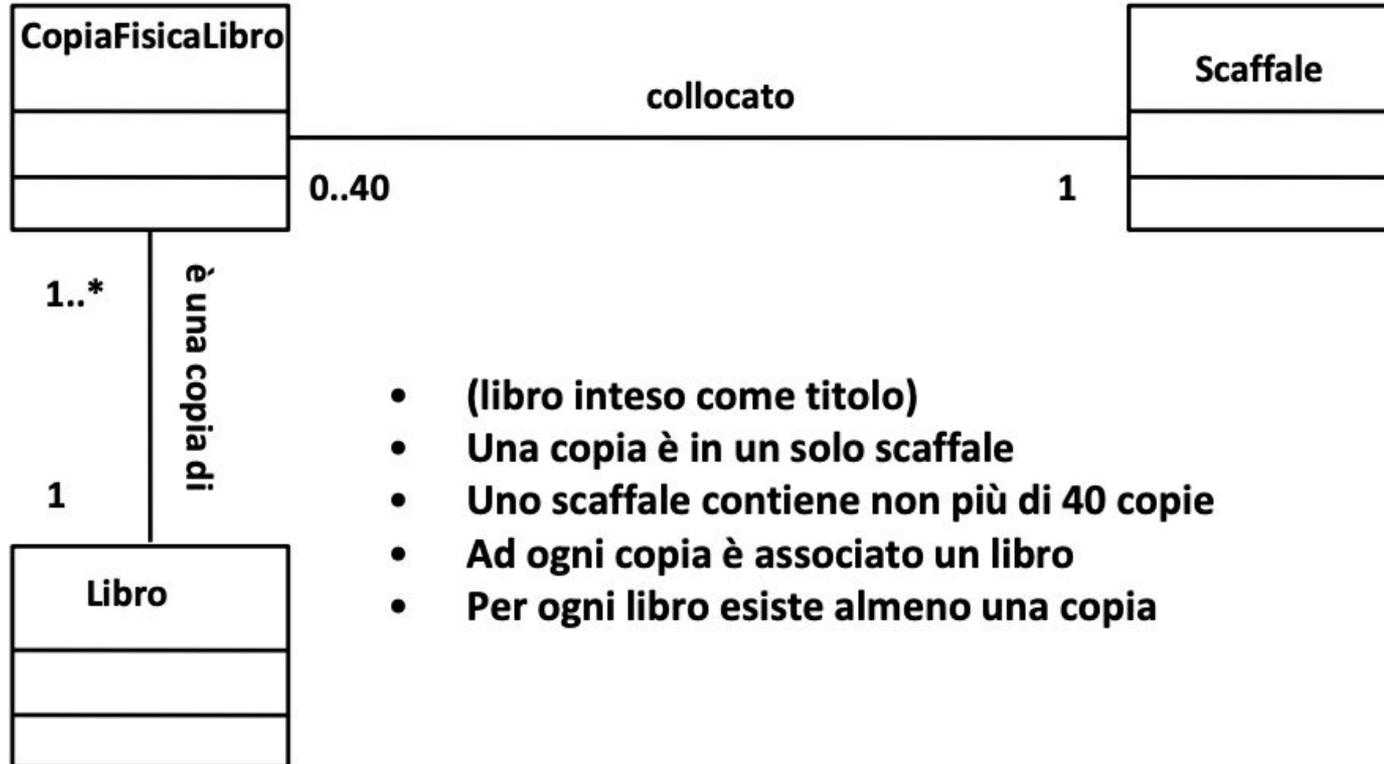
La molteplicità è legata al nome dell'associazione



Sintatticamente si possono anche avere più associazioni tra due classi



Molteplicità: esempio



- (libro inteso come titolo)
- Una copia è in un solo scaffale
- Uno scaffale contiene non più di 40 copie
- Ad ogni copia è associato un libro
- Per ogni libro esiste almeno una copia

Associazioni riflesse

In questo caso è fondamentale indicare il ruolo



Aggregazione e composizione

Aggregazione e Composizione sono tipi particolari di associazione

- entrambe specificano che un oggetto di una classe è **una parte** di un oggetto di un'altra classe
- Suggerimento: prenderle in considerazione quando il nome dell'associazione sarebbe del tipo:
 - fa parte di, appartiene....
- o, simmetricamente:
 - è composto da, possiede, ha, ...

Aggregazione vs Composizione

Aggregazione → relazione tra oggetti poco forte

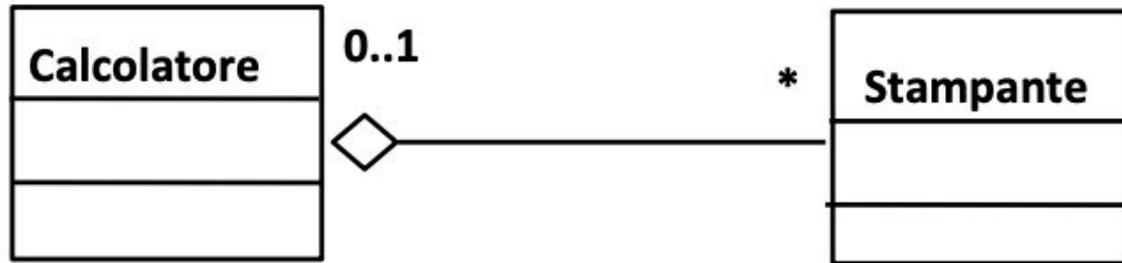
ovvero una relazione nella quale le classi parte hanno un significato anche senza che sia presente la classe tutto

Composizione → relazione tra oggetti forte

le classi parte hanno un reale significato solo se sono legate alla classe tutto

Sintassi e semantica con un esempio

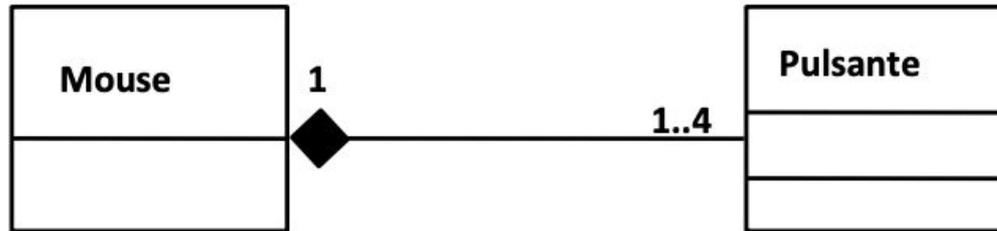
Aggregazione



- La stampante nel tempo può essere collegata a calcolatori diversi
- La stampante esiste anche senza calcolatore
- Se il calcolatore viene distrutto la stampante esiste comunque
- L'aggregazione non ha un nome

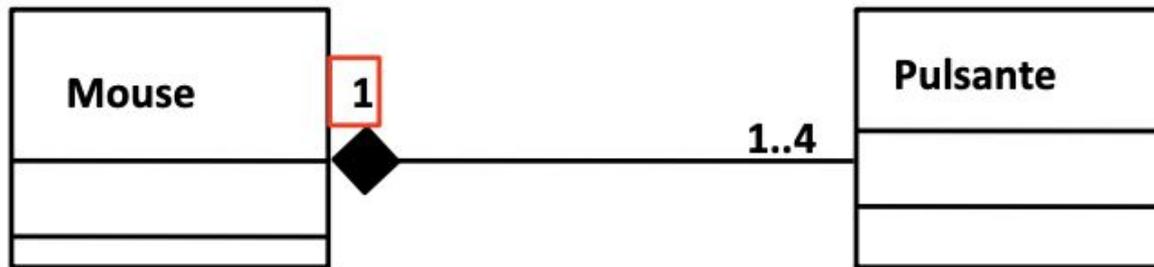
Sintassi e semantica con un e semplice

Composizione



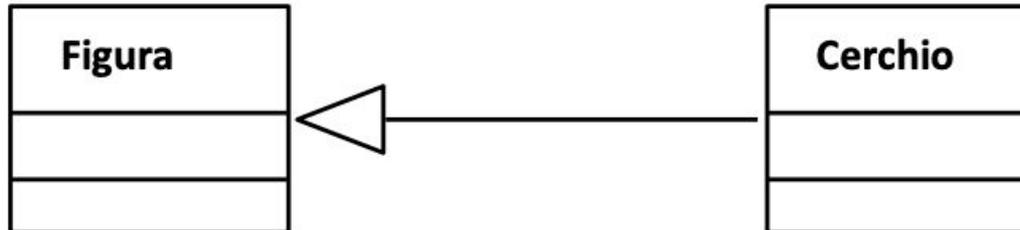
- Un pulsante appartiene a un solo mouse
- Non esiste senza il suo mouse
- Se il mouse viene distrutto vengono distrutti anche i pulsanti
- La composizione non ha un nome

Uno sguardo alle molteplicità



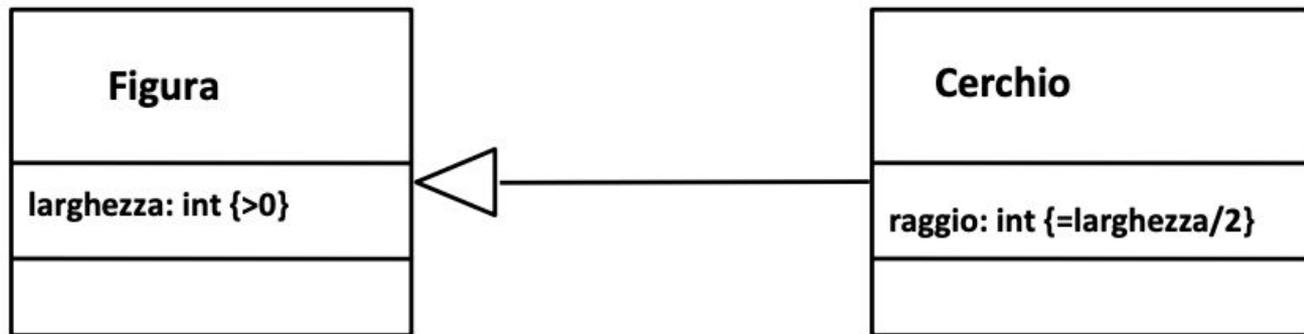
Generalizzazione

- Relazione tra un elemento generico e uno più specializzato
- L'elemento più specializzato è completamente consistente con quello più generico ma contiene più informazione
- Vale il principio di sostituzione della Liskov: l'elemento specializzato può essere usato al posto dell'elemento generico
- "è un tipo di"

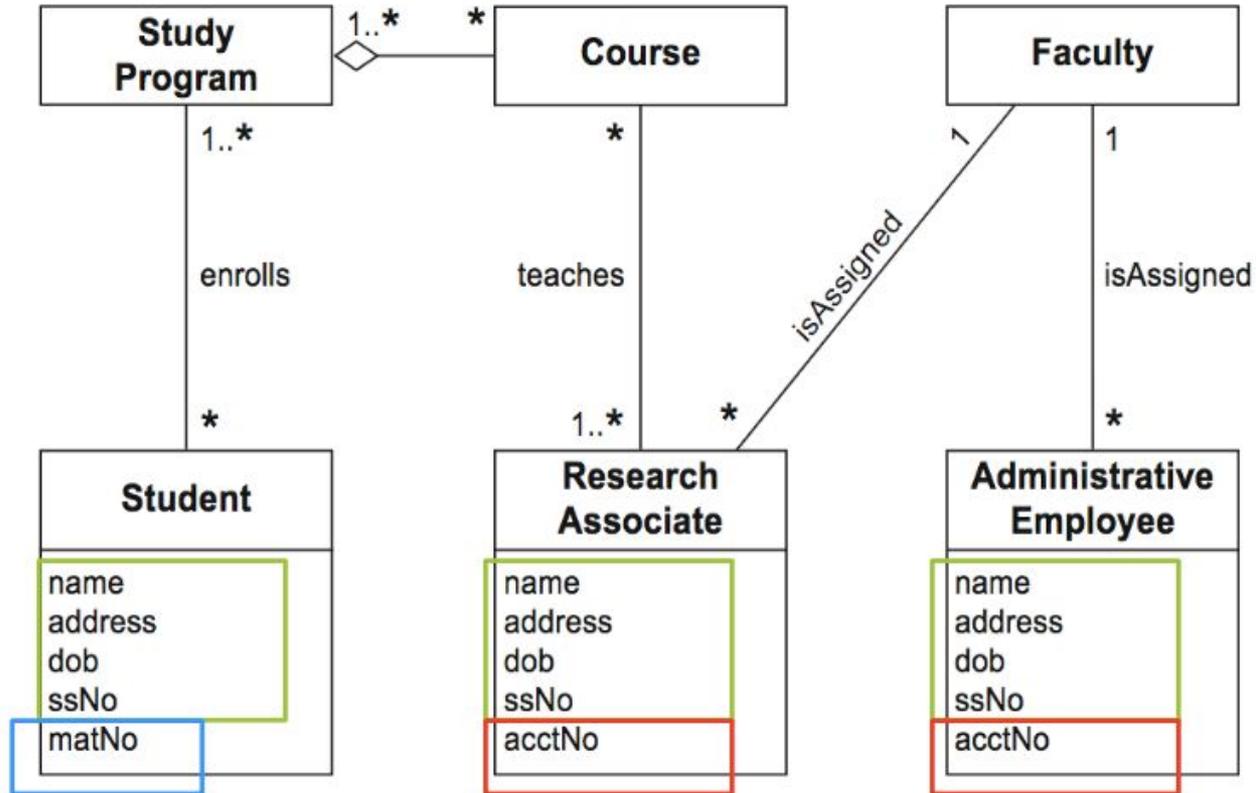


Ereditarietà di classe

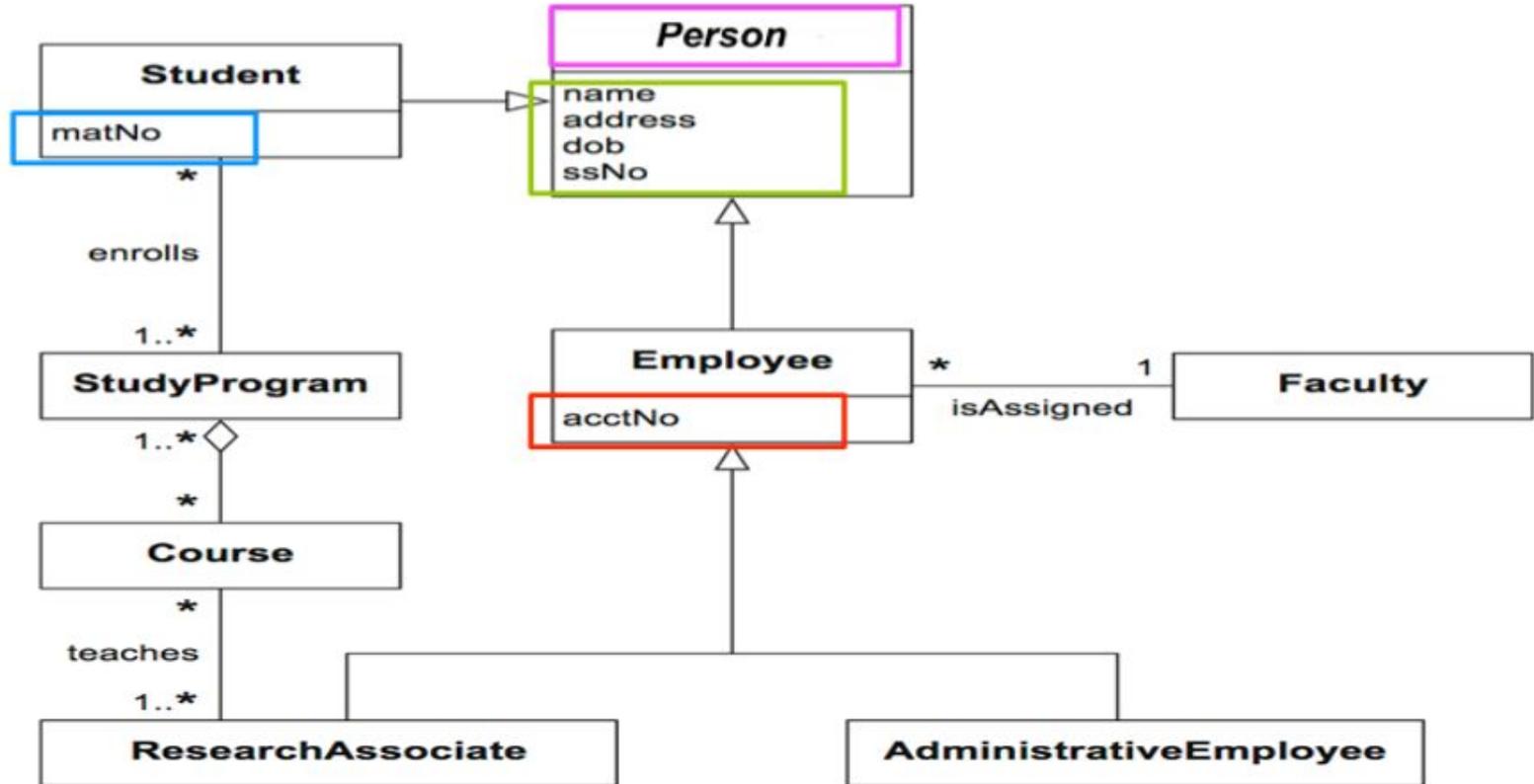
- Le sottoclassi ereditano tutte le caratteristiche della superasse:
 - attributi, operazioni, relazioni e vincoli
- Le sottoclassi possono aggiungere caratteristiche e ridefinire le operazioni



Esempio generalizzazione



Esempio generalizzazione



Dipendenze

Una relazione in cui le classi hanno ruolo di cliente e fornitore

- Il cliente dipende dal fornitore



- una modifica nel fornitore può influenzare il cliente

Dipendenze d'uso: esempi

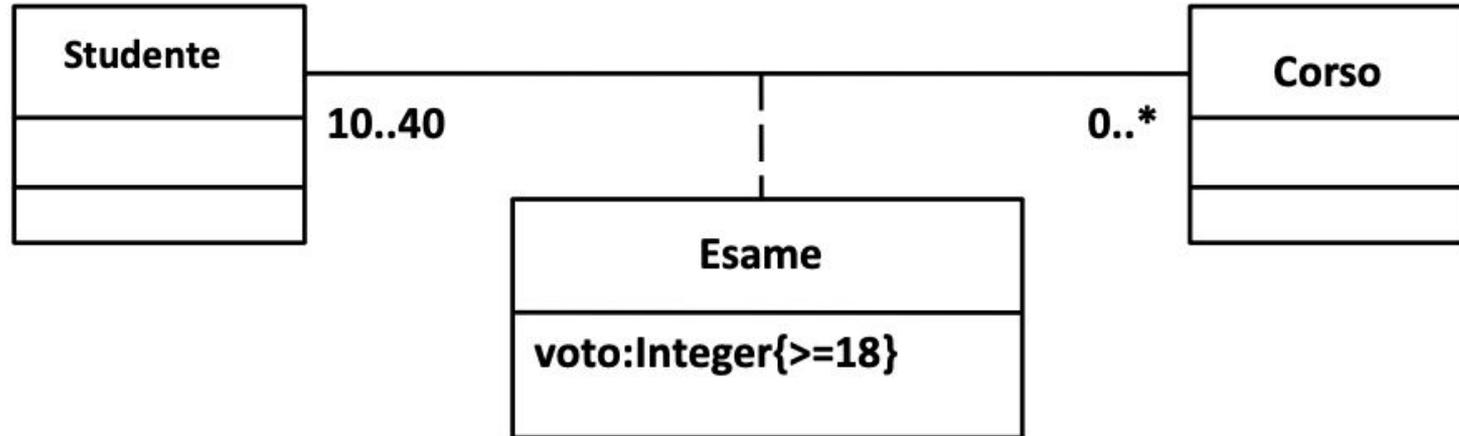
Le dipendenze più comuni

- Un parametro di un'operazione di A è di tipo B
- Un'operazione di A restituisce un oggetto di tipo B
- Un'operazione di A crea dinamicamente un oggetto di tipo B



in caso di creazione si usa <<create>> invece di <<use>>

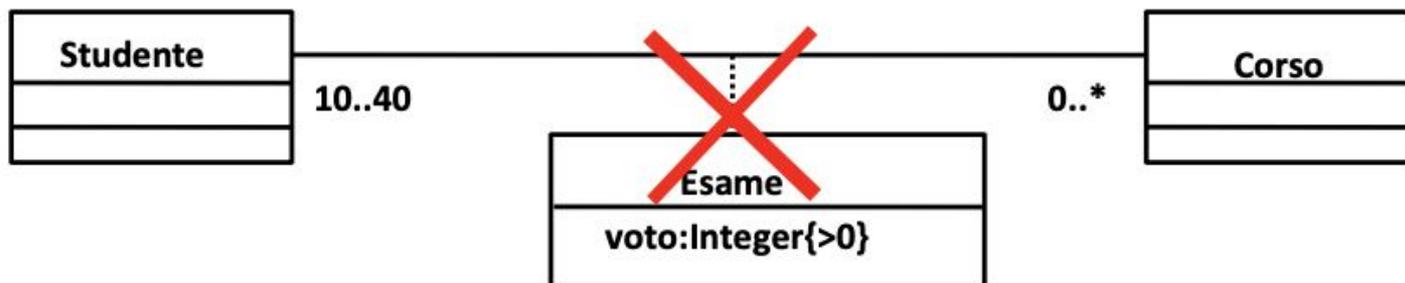
Classi associazione



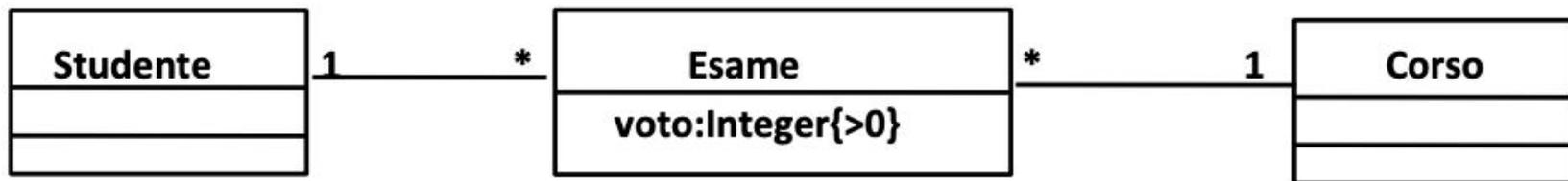
- **Un'associazione può avere attributi propri, rappresentati con una classe associazione.**
- **Le istanze sono collegamenti con attributi propri.**
- **Voto non è attributo né di Corso né di Studente**

Classi associazione (2)

Per ogni coppia di oggetti collegati tra loro può esistere un unico oggetto della classe associazione



Se vogliamo tenere traccia dei voti negativi non si possono usare le classi associazione



Identificazione delle classe

- Principali tecniche
 - Approccio data driven: tipico della fase di analisi
 - Si identificano tutti i dati del sistema e si dividono in classi (ad esempio mediante identificazione dei sostantivi)
 - Approccio responsibility driven: soprattutto durante la progettazione
 - Si identificano le responsabilità e si dividono in classi

Analisi nome-verbo

- Sostantivi → classi o attributi
- Verbi → responsabilità o operazioni
- Passi:
 - Individuazione delle classi
 - Assegnazione di attributi e responsabilità alle classi
 - Individuazione di relazioni tra le classi

Analisi nome-verbo

- Problemi ricorrenti :
 - Tagliare le classi inutili
 - Trattare i casi di sinonimia
 - Individuare le classi nascoste cioè le classi implicite del dominio del problema che possono anche non essere mai menzionate esplicitamente
 - In un sistema di gestione degli orari delle lezioni di un corso universitario, nella descrizione testuale potrebbe non essere mai nominata l'aula, che invece deve essere inserita nel modello

Individuazione classi: chiavi magnetiche

Per motivi di sicurezza, un'organizzazione ha deciso di realizzare un sistema secondo il quale a ogni dipendente è assegnata una chiave magnetica per accedere (aprire) determinate stanze. I diritti di accesso dipenderanno in generale dalla posizione e dalle responsabilità del dipendente. Quindi sono necessarie operazioni per modificare i diritti di accesso posseduti da una chiave se il suo proprietario cambia ruolo nell'organizzazione.

Organizzazione

Dipendente

Ruolo

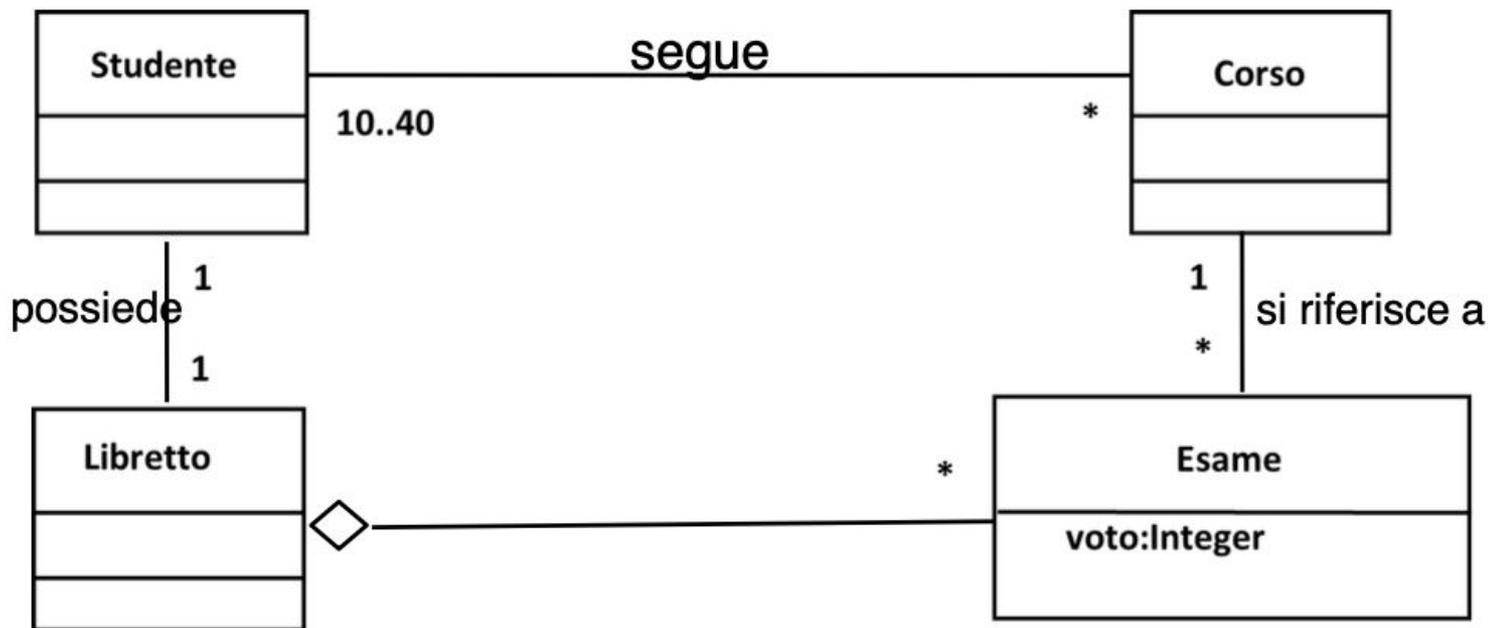
Chiave

Stanza

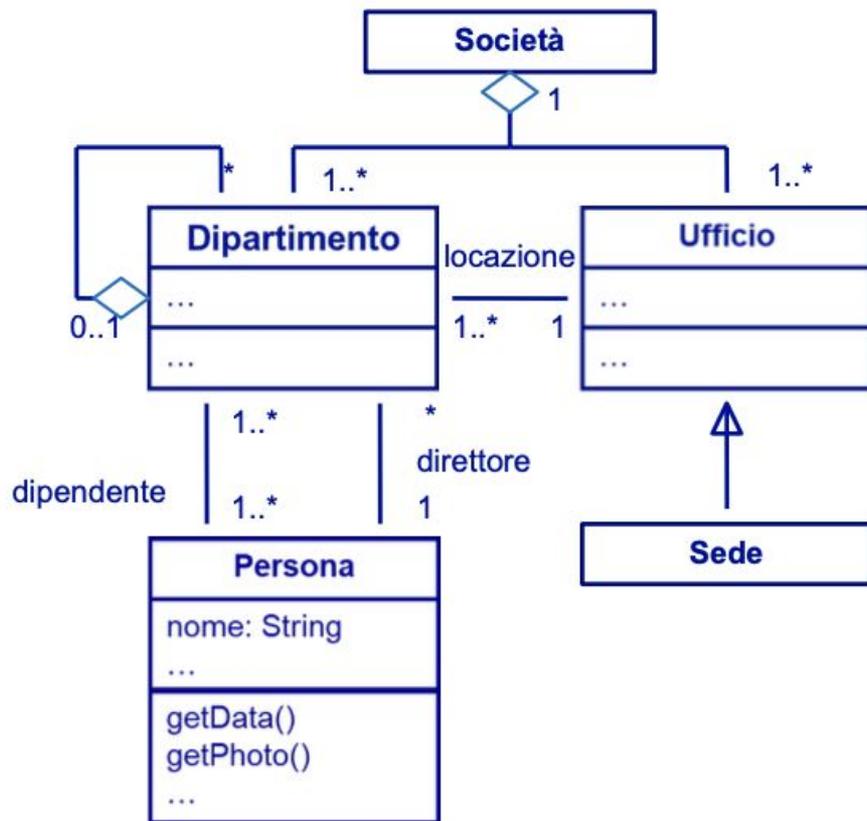
DirittiDiAccesso

Esempio

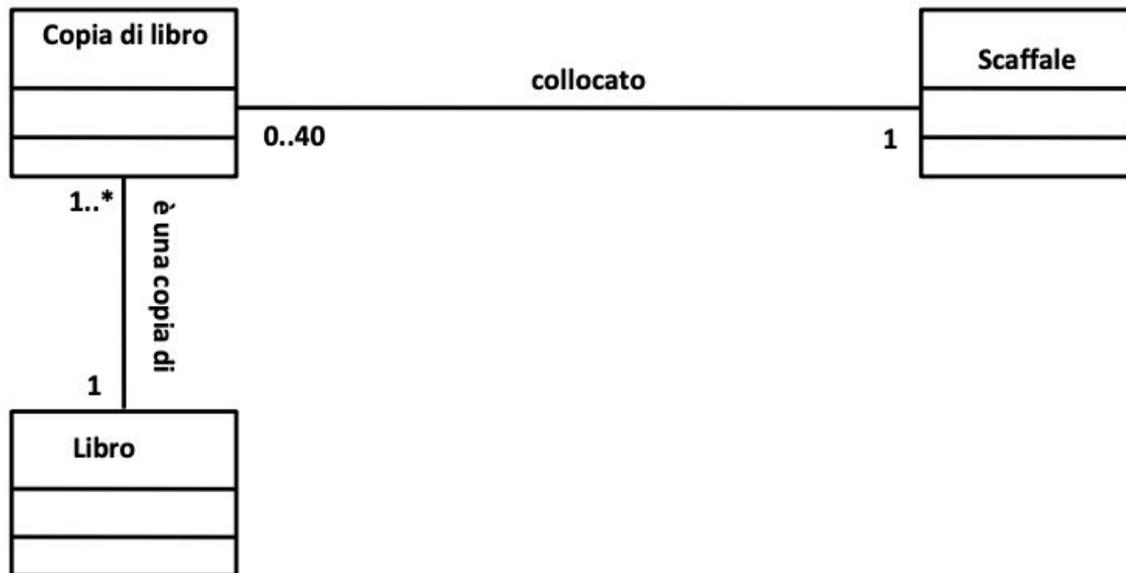
■ Studente, Libretto, Esame, Corso



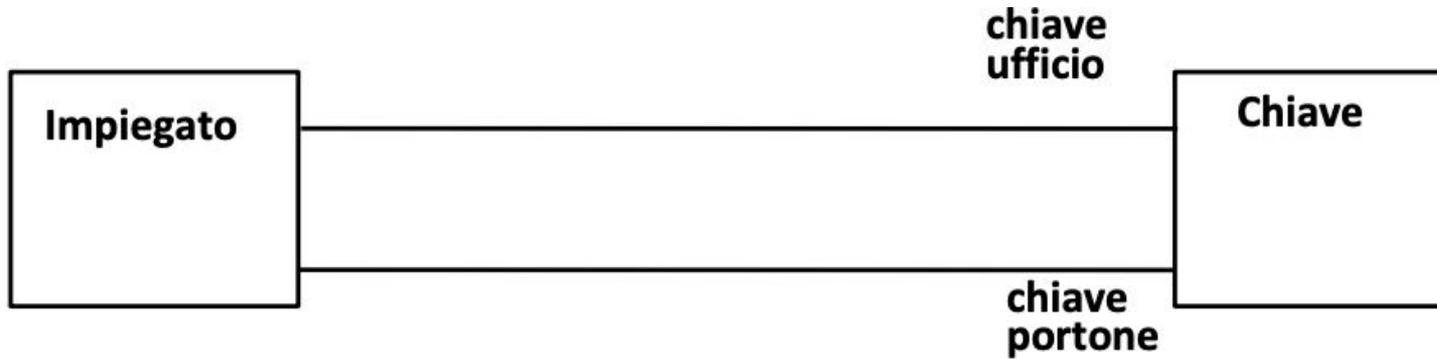
Esempio: classi



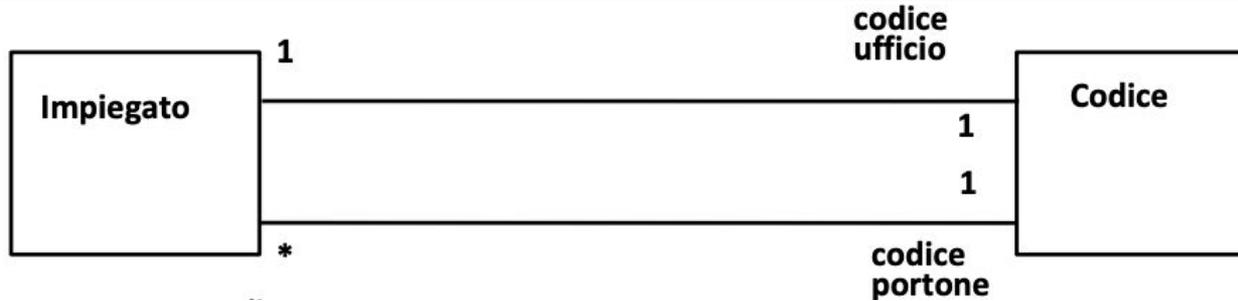
Esempio



Associazione: caso particolare - ruoli importanti



Molteplicità: esempio



Ruoli e aggregazione: un caso speciale

- Nel diagramma delle classi



Un esempio con generalizzazione e aggregazione

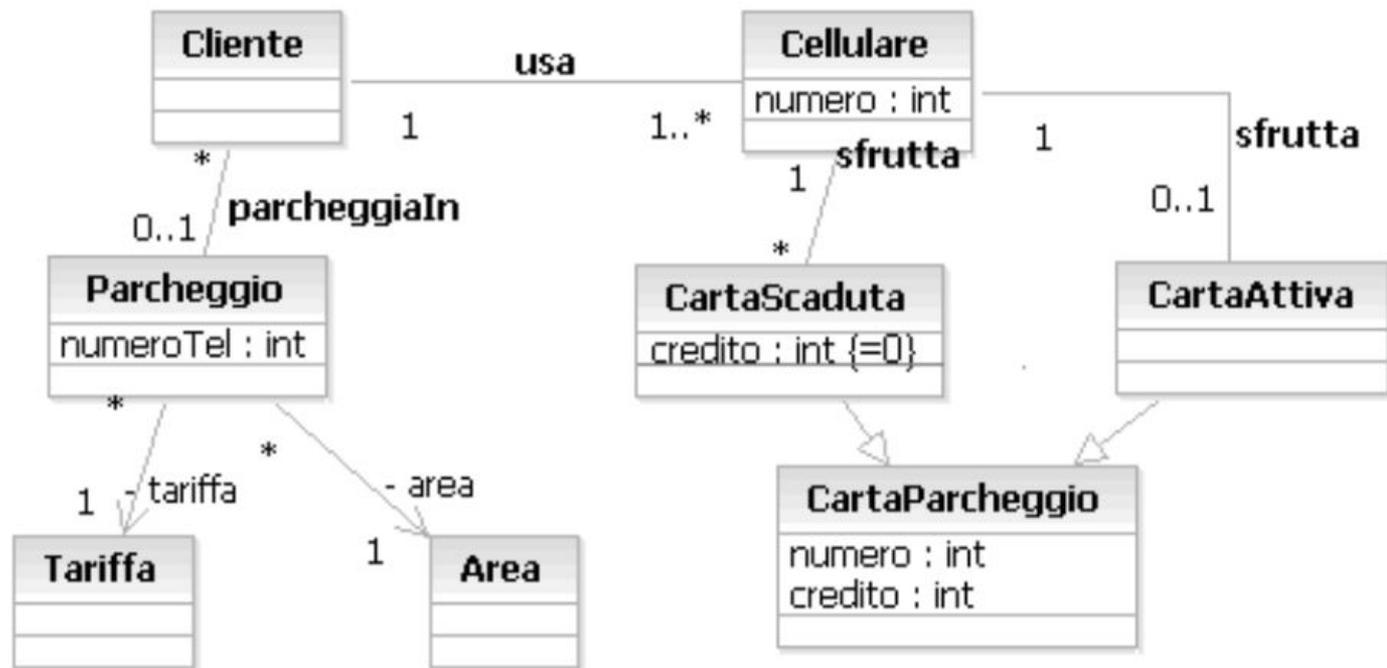


Esercizio: My Air

- Iscriviti al programma e da semplice cliente diventerai un associato MyAir, guadagnando immediatamente un bonus di 5.000 miglia utili.
- Ogni volta che volerai con MyAir le miglia accumulabili del volo saranno sommate alle tue miglia utili, permettendoti di raggiungere in poco tempo le miglia necessarie per richiedere uno dei nostri premi (omaggio biglietti aereo o soggiorni in località da sogno).
- I premi riscossi danno luogo a una diminuzione immediata delle miglia utili. La situazione è aggiornata il 31 dicembre, mantenendo solo le miglia dei voli effettuati negli ultimi 5 anni.
- Inoltre se accumulerai almeno 15.000 miglia (miglia accumulate) sarai promosso dal livello standard al livello argento. Se invece accumulerai almeno 100.000 miglia entrerai a far parte del ristretto numero di associati del livello oro².
- Tutte le condizioni si riferiscono esclusivamente alle miglia accumulate in un anno. Il passaggio da un livello all'altro è effettuato il 31 dicembre. La permanenza nel livello da un anno all'altro è soggetta al rispetto degli stessi requisiti per entrare nel livello. Il bonus iniziale non concorre al raggiungimento delle miglia richieste per cambiare o mantenere un livello.

Easy Park

- Soluzione per il pagamento del parcheggio via telefono cellulare.
- 1. Il cliente acquista una Carta Parcheggio prepagata e l'attiva indicando il proprio numero di cellulare. Durante l'attivazione, il sistema trasferisce sulla nuova carta l'eventuale credito residuo su una carta già associata al numero di telefono indicato.
- 2. Il cliente parcheggia ed espone sul cruscotto la Carta Parcheggio. Nel cartellone del Parcheggio verifica qual è il numero di telefono che identifica l'area e la tariffa. Il cliente telefona a questo numero, il cliente è identificato attraverso il proprio numero di telefono cellulare e il sistema attiva il pagamento della sosta.
- 3. Il Controllore controlla l'effettivo pagamento della sosta inserendo il numero della Carta Parcheggio in un applicativo fruibile tramite Pocket PC connesso a internet o Telefono Cellulare.
- 4. Disattivazione della sosta con chiamata via cellulare: l'utente chiama il numero associato al parcheggio, il sistema riconosce l'utente e disattiva il pagamento. Inoltre il sistema comunica vis SMS la disattivazione, la somma pagata, la durata della sosta e il residuo presente sulla Carta Parcheggio.



**Modellare anche sosta, con inizio, fine costo
(classe associazione)**