

# Ingegneria del Software

## 7. Relazioni

Dipartimento di Informatica

Università di Pisa

A.A. 2014/15

# relazioni, 1

- Una relazione rappresenta un legame
  - tra due o più oggetti
  - normalmente istanze di classi diverse
- Dal punto di vista matematico
  - una relazione binaria tra  $A$  e  $B$  è un sottoinsieme del prodotto cartesiano  $A \times B$
- UML fornisce notazioni grafiche per
  - prodotti cartesiani, a livello di classificatori
  - coppie, a livello di istanza
  - anche triple, quadruple, etc (associazioni n-arie)

# relazioni, 2

- Tra elementi di un modello

## Tra classi

### ○ Associazione

- Aggregazione
- Composizione
- Associazione navigabile

## Tra oggetti

### ○ Collegamento

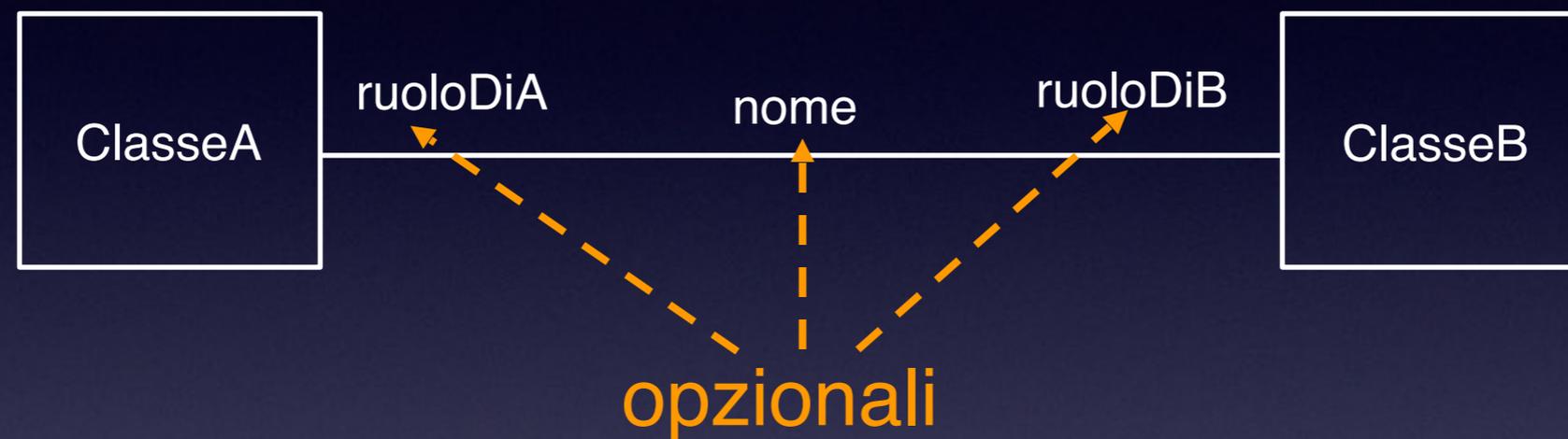
- Collegamento unidirezionale

### ○ Generalizzazione

- Dipendenza (uso, realizzazione, istanza)
- Flusso, estensione, realizzazione

[non ancora!]

# associazione (binaria) sintassi



- “The only necessary thing is the line, but usually you want at least one name, rarely all names”
- Il ruolo serve a descrivere meglio l’oggetto associato, soprattutto se gli oggetti sono della stessa classe

# associazione (binaria) esempio

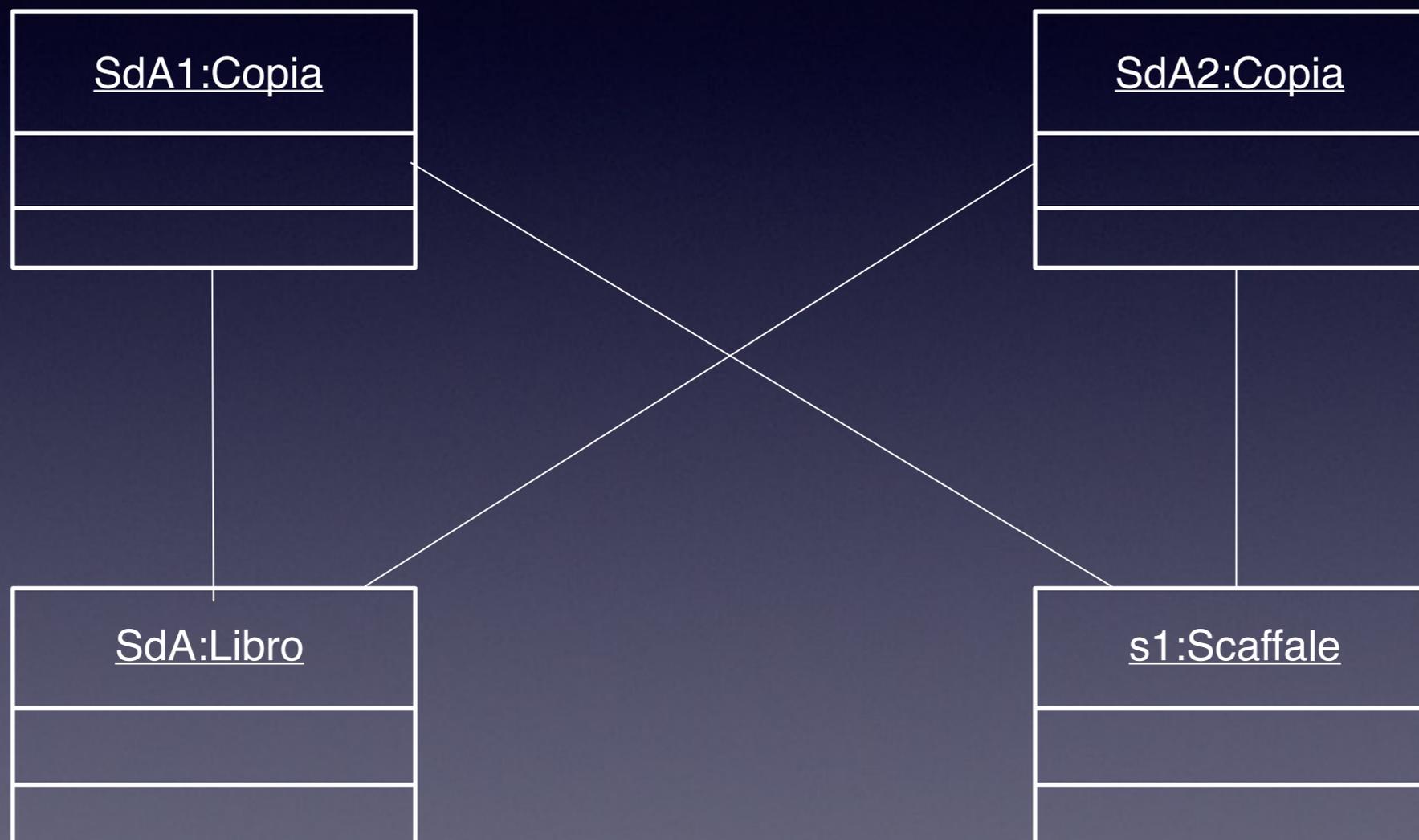


- se  $i$  è un Impiegato e  $c$  una Chiave...
- e se  $c$  è stata consegnata ad  $i$ ...
- si può dire che  $i$  è il *proprietario* di  $c$  e che  $c$  è la *chiave* dell'ufficio di  $i$

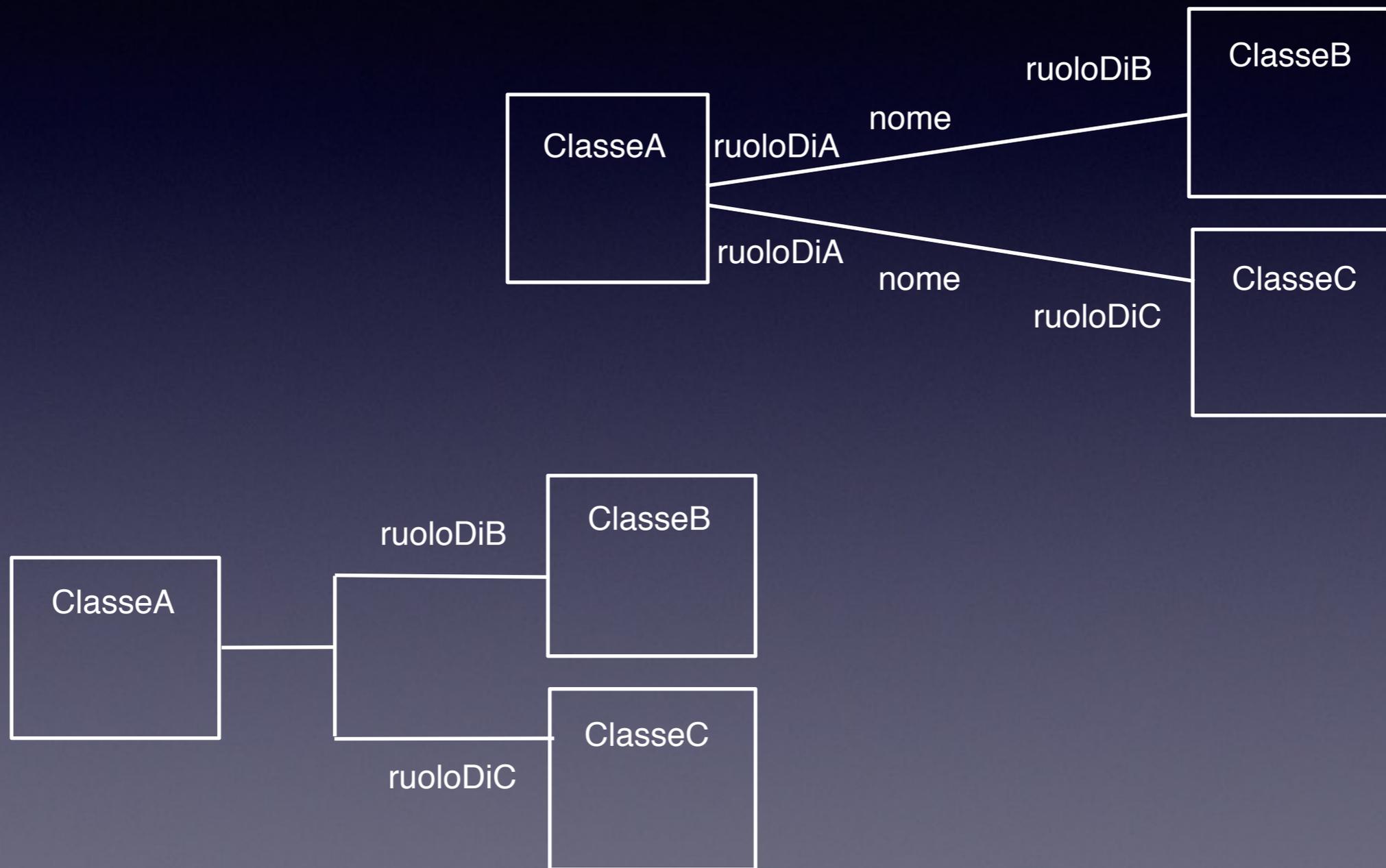
# associazione (binaria): nomi e ruoli

- nome e ruoli: lowerCamelCase
- nome: normalmente un verbo
- ruolo: normalmente un sostantivo
- formalmente opzionali...
  - è utile ci sia o il nome dell'associazione o l'indicazione dei ruoli
  - superfluo entrambi

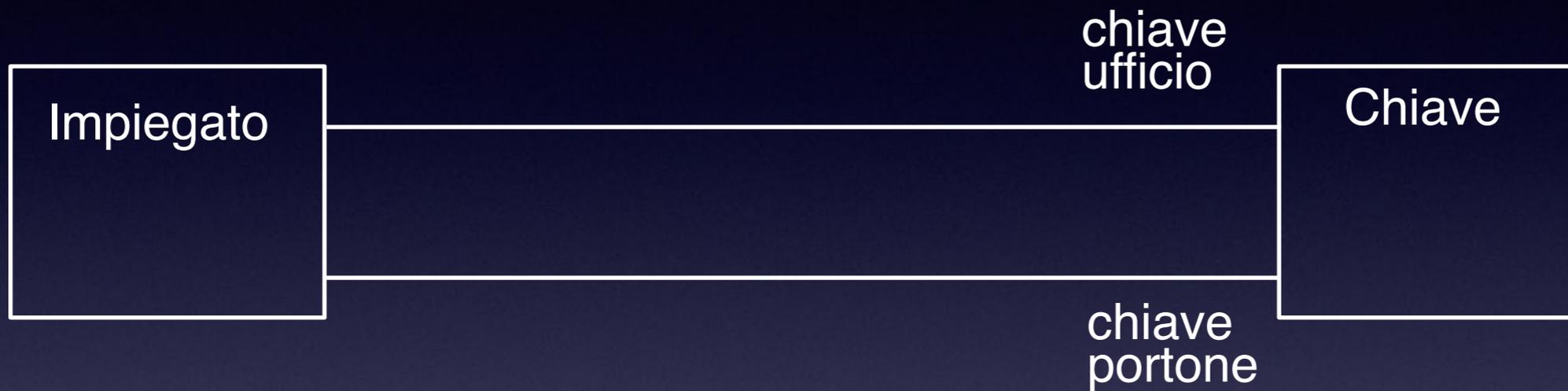
# a livello di istanza: collegamento



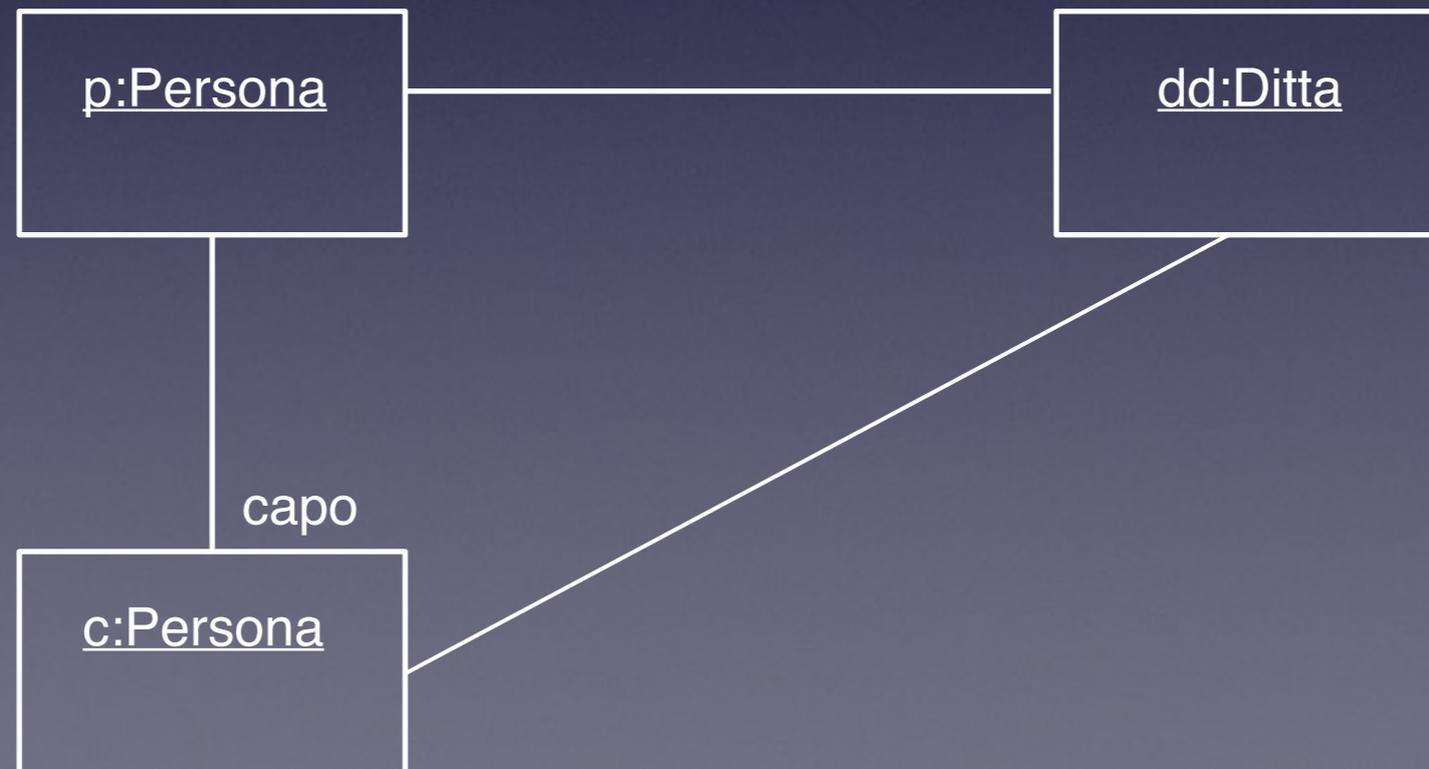
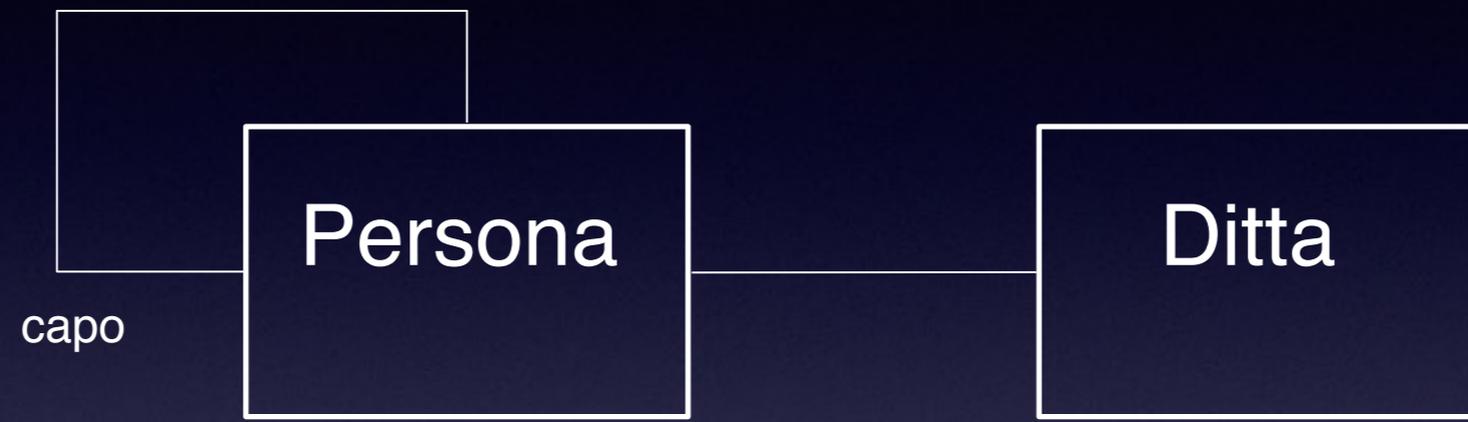
# a livello di istanza: stili



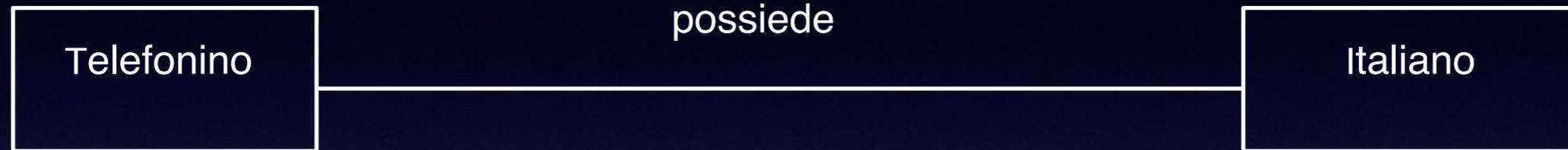
# per ruoli importanti, 1



# per ruoli importanti, 2



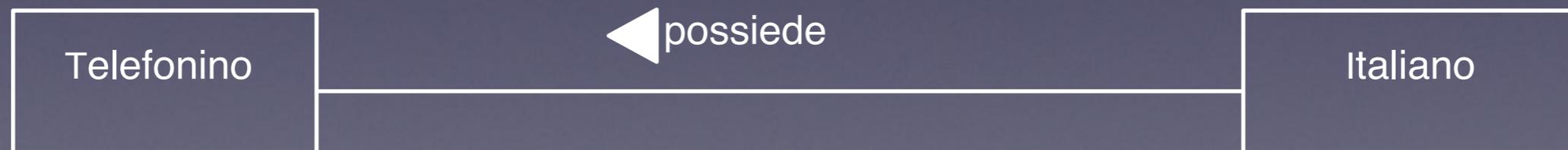
# direzione



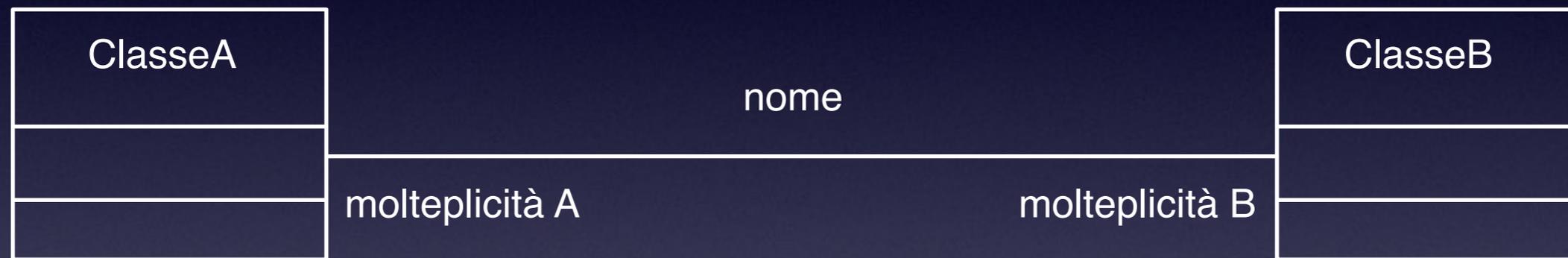
Potrebbe essere letto come

(ogni) telefonino possiede un italiano [Altan]

Si usa una freccia di *direzione di lettura* per disambiguare

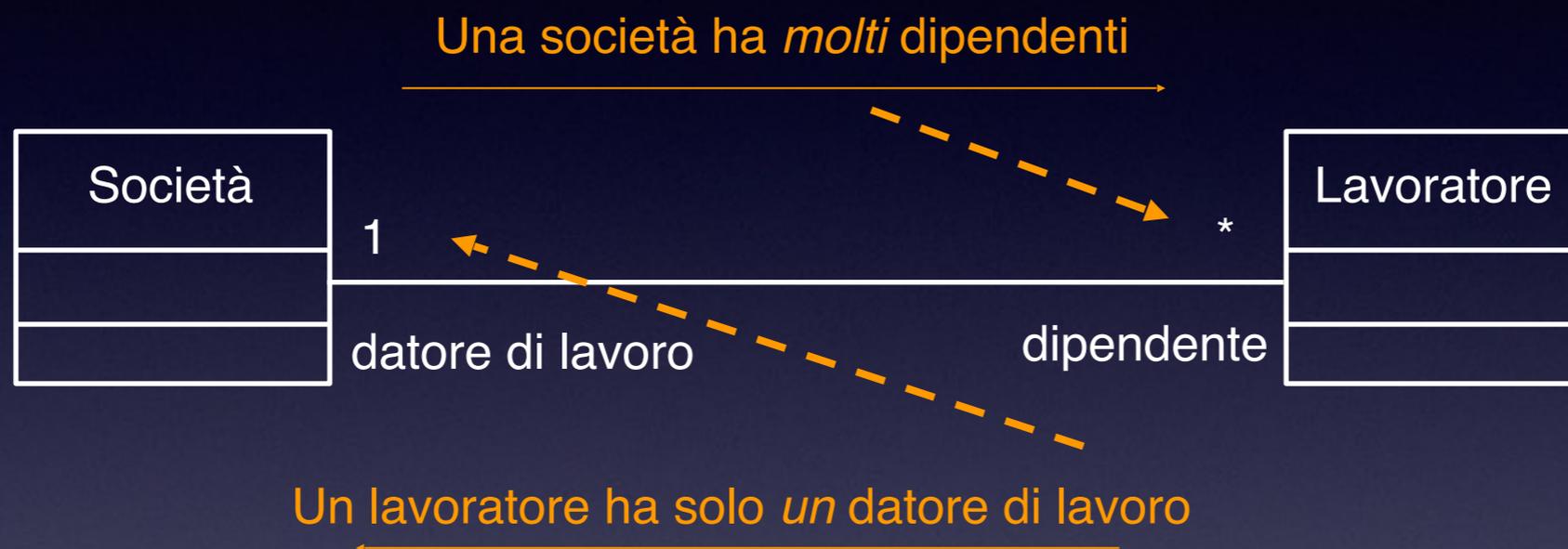


# vincoli di molteplicità



- Indica il numero di oggetti che sono coinvolti in una associazione IN UN DATO ISTANTE

# un esempio di molteplicità



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

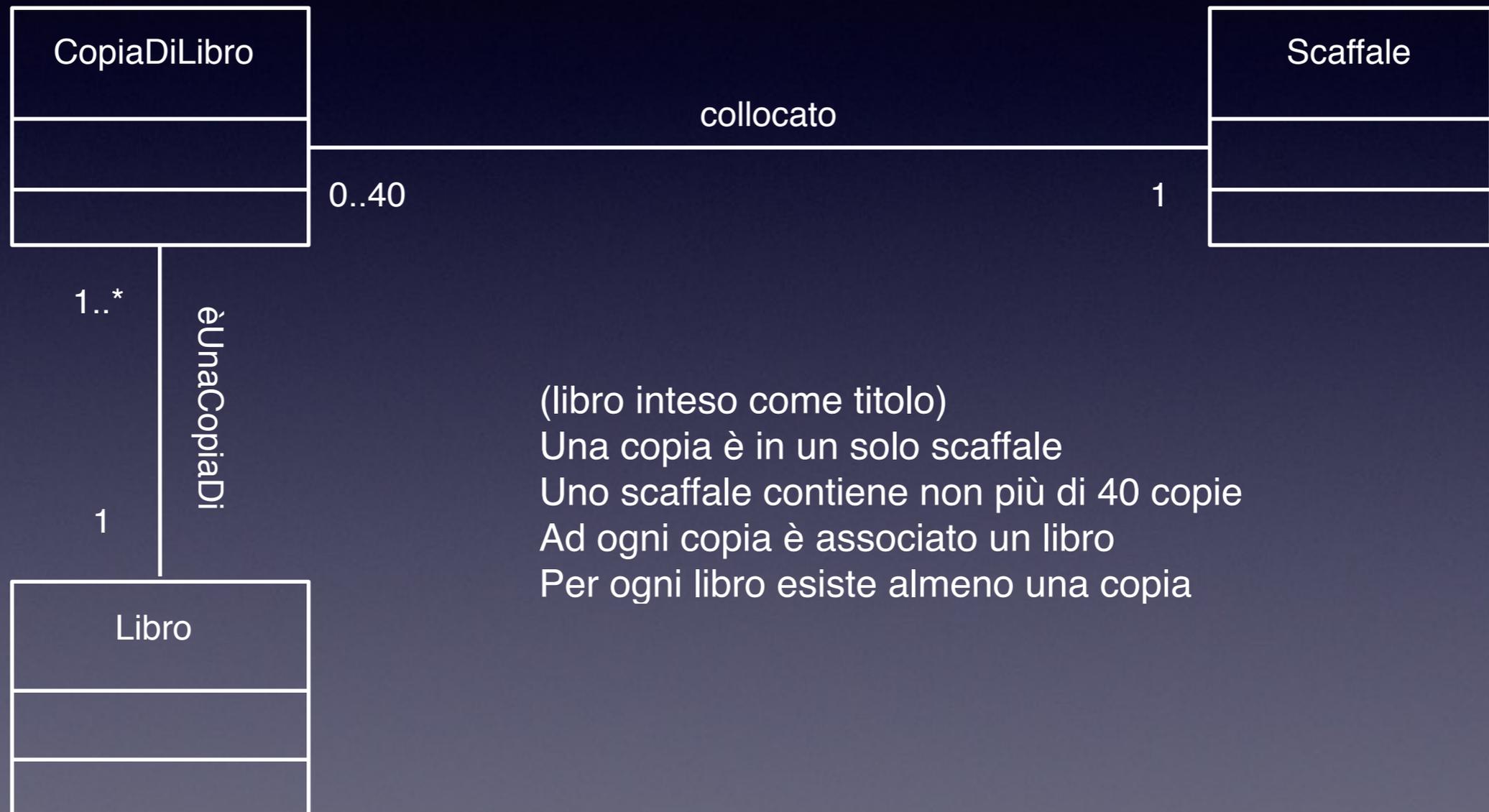
# attenti al nome!



# molteplicità: sintassi

- minimo..massimo
  - 0..2 nessun oggetto, un oggetto o 2 oggetti
  - 1..\* almeno un oggetto
  - \* equivalente a 0..\*
  - 1 un solo oggetto
- [notazioni obsolete: cancellare ultima riga tabella 9.1 in Arlow (UML1)]

# un altro 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

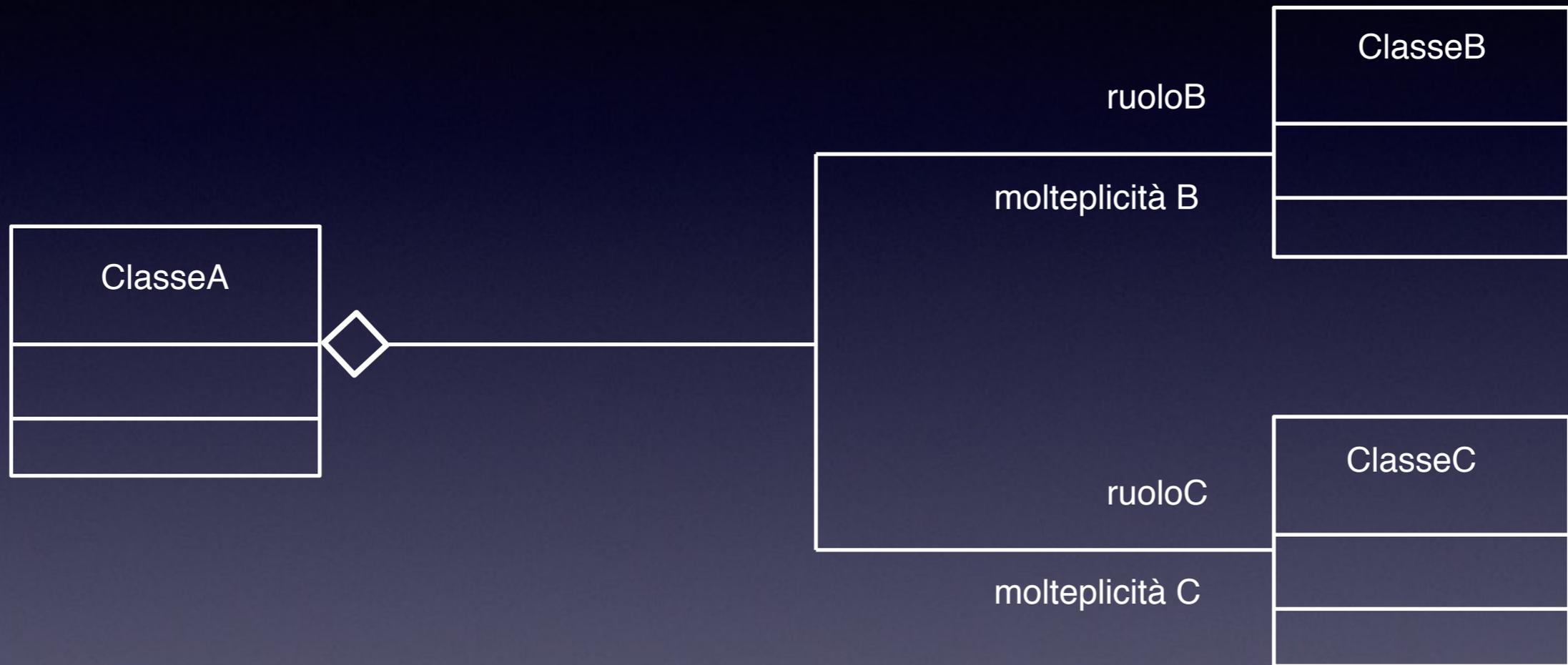
# sulle associazioni n-arie

In an  $n$ -ary association, the multiplicity is defined with respect to the other  $n-1$  ends. For example, given a ternary association among classes (A, B, C), then the multiplicity of the C end states how many C objects may appear in association with a particular pair of A and B objects. If the multiplicity of this association is (many, many, one), then for each possible (A, B) pair, there is a unique value of C. For a given (B, C) pair, there may be many A values, however, and many values of A, B, and C may participate in the association.

# ancora un esempio

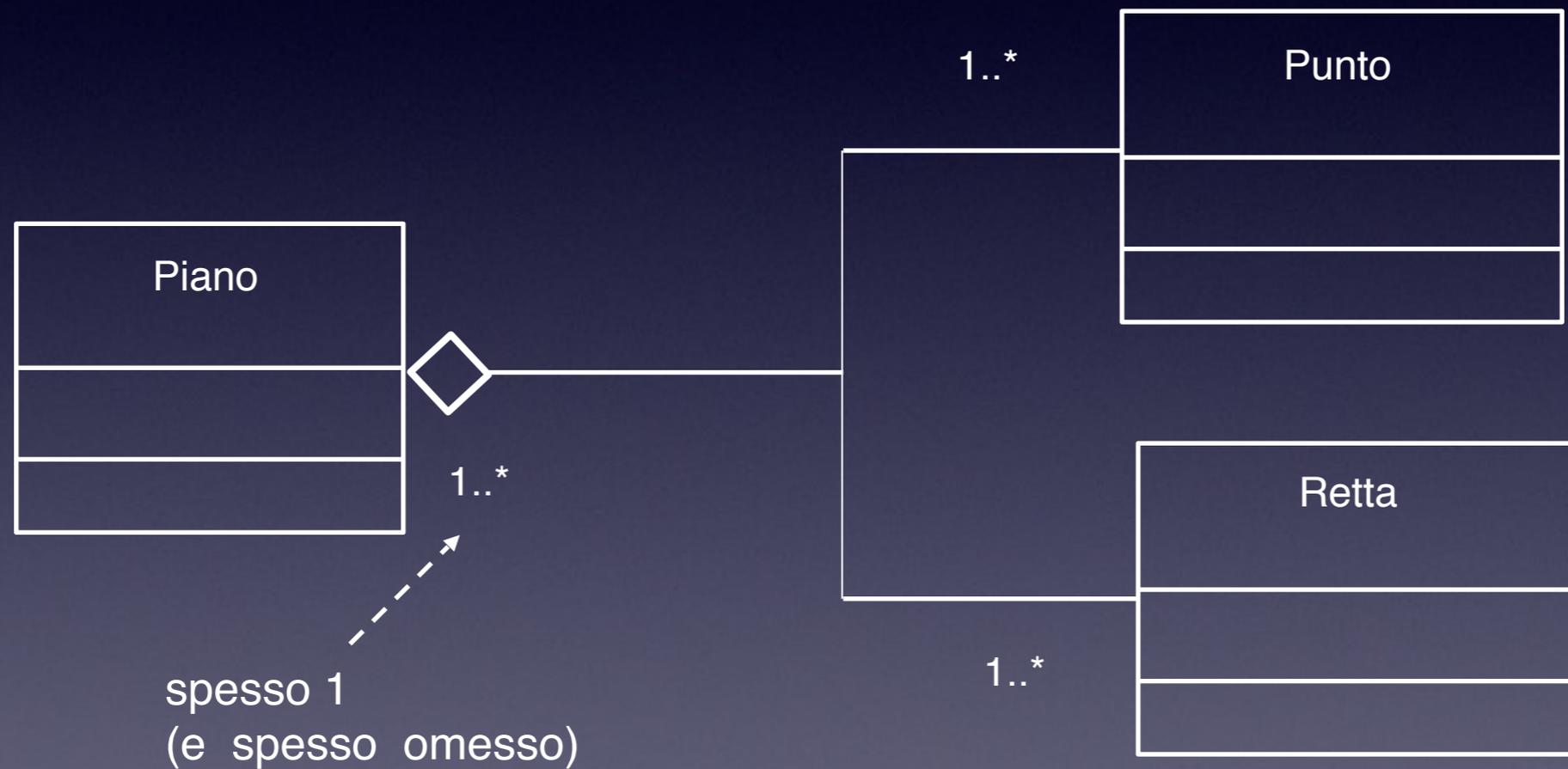


# aggregazione: sintassi

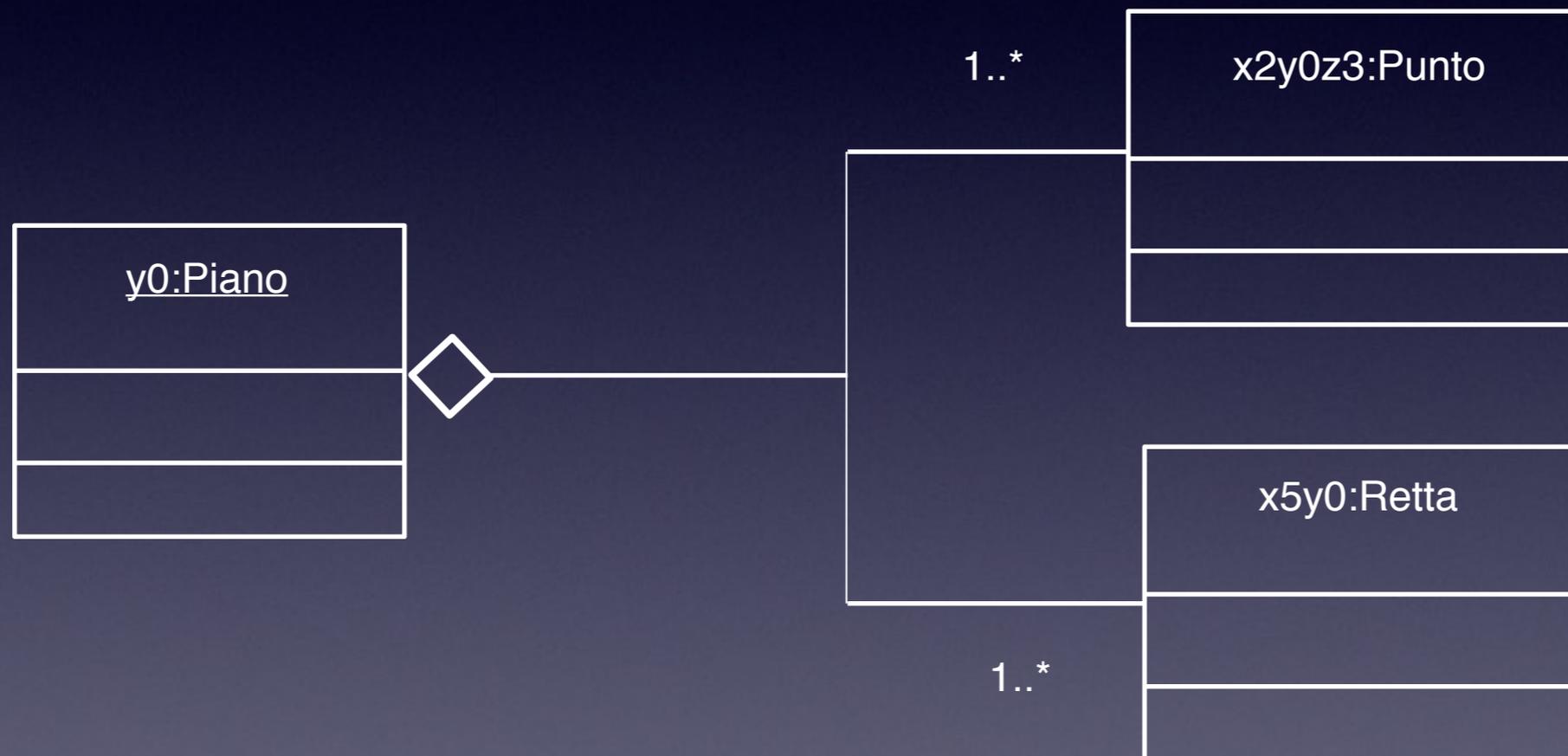


- rappresenta una relazione parte (B e C)/tutto (A)

# un esempio di aggregazione

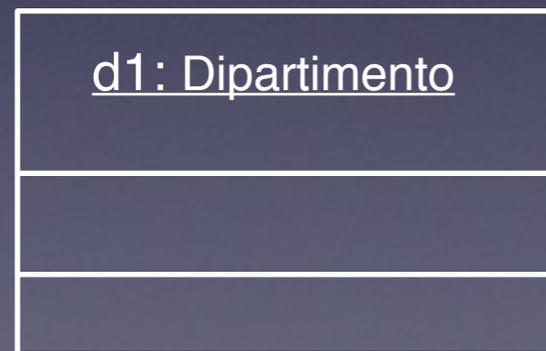
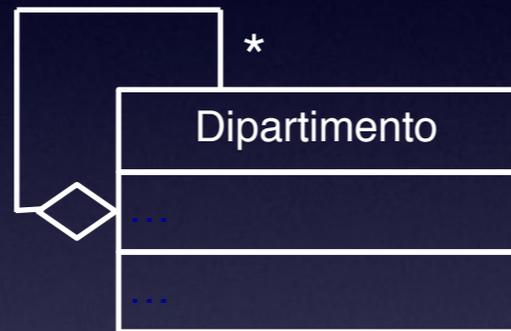


# a livello di istanza

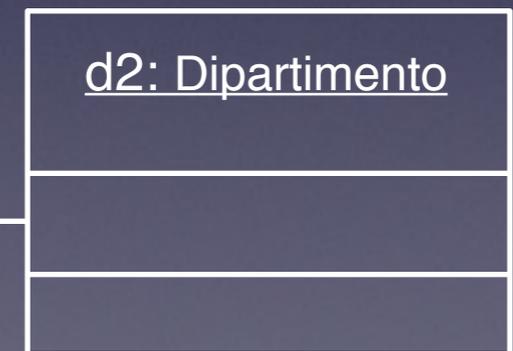


# per casi importanti

sotto-dipartimento



sotto-dipartimento



# un altro esempio



Tutti vivono in un sottomarino giallo

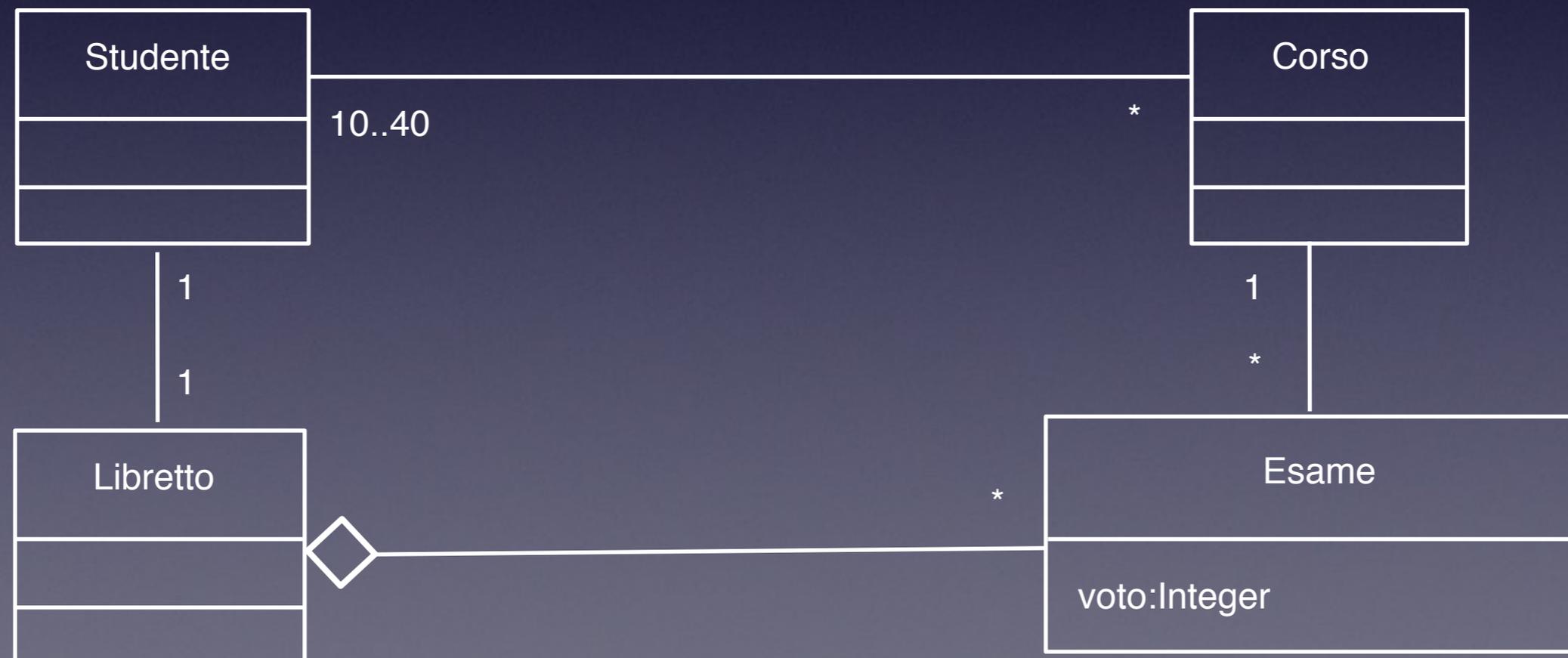
[assai indefinito]

L'esercito delle 12 scimmie



# un esercizio

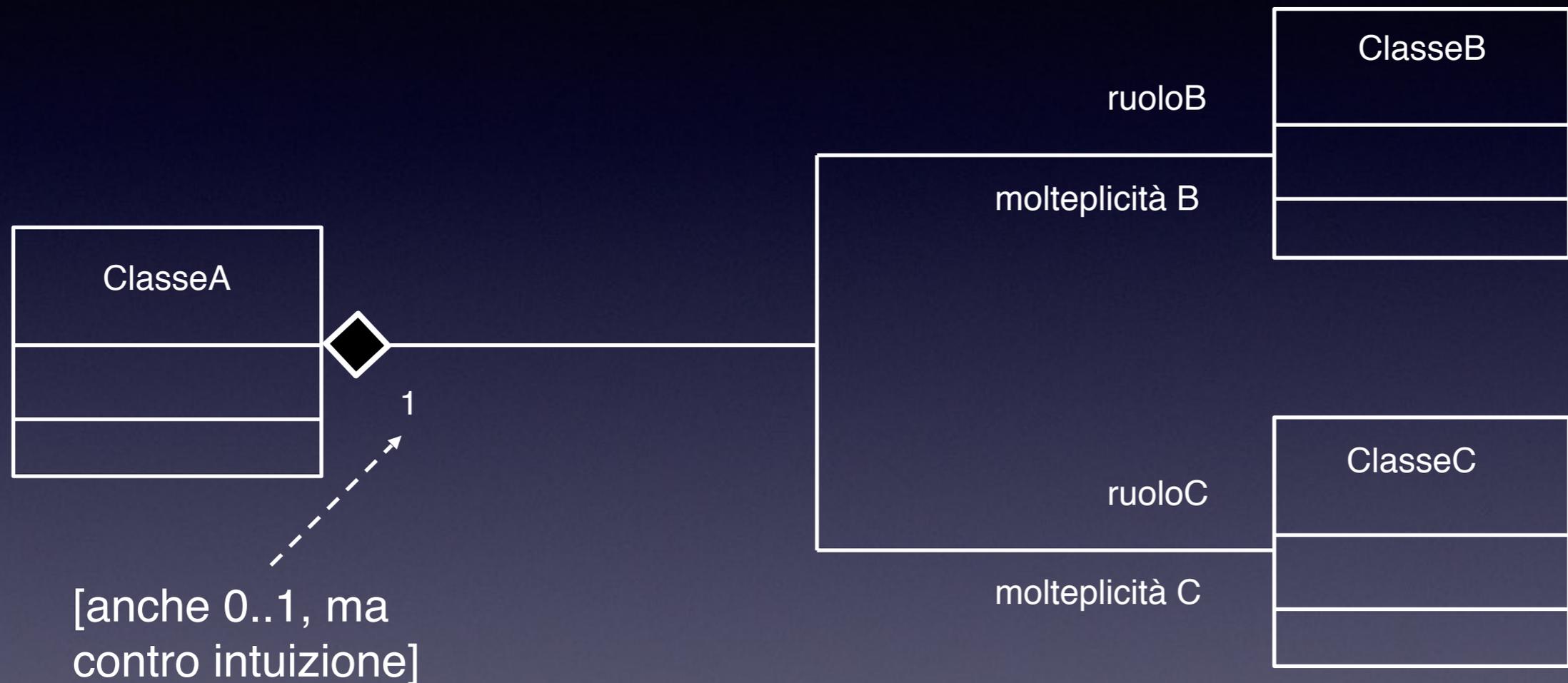
- fornire una rappresentazione delle relazioni fra
  - Studente, Libretto, Esame, Corso



# un altro esercizio

- fornire una rappresentazione degli alberi binari
  - radice e sotto-alberi...

# composizione: sintassi



- una forma di associazione più forte dell'aggregazione:  
LE PARTI NON HANNO SENSO SENZA IL TUTTO

# un esempio



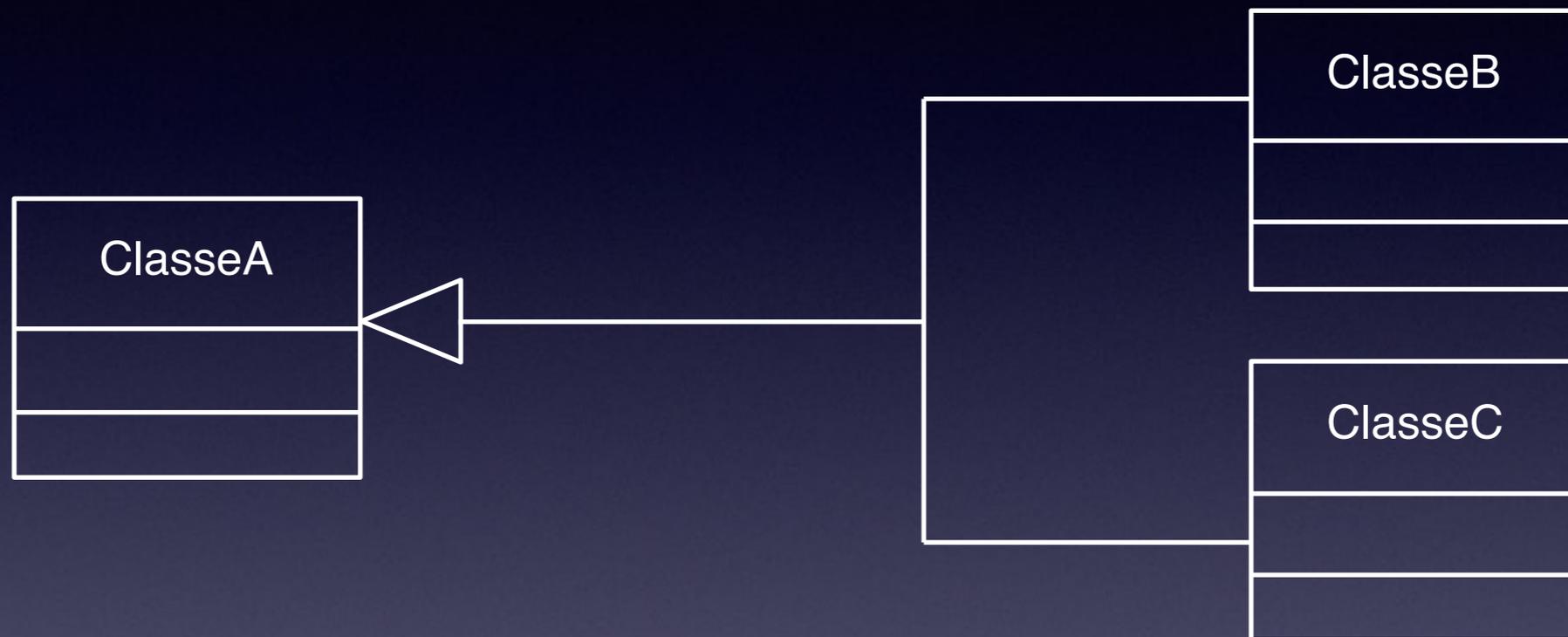
- Se cancello un poligono cancello anche i suoi lati (mentre se cancello un piano, i punti restano in quanto appartengono anche ad altri piani)
  - anche: directory e file, o ufficio e personale
- Prospettiva sw: il tutto crea e distrugge le parti

# attributo vs. associazione

- Semanticamente equivalenti
- Non usare mai entrambi

[...] the intent and usage of attributes and associations are usually different. Use attributes for data types—that is, for values with no identity. Use associations for classes—that is, for values with identity. The reason is that for objects with identity, it is important to see the relationship in both directions; for data types, the data type is usually subordinate to the object and has no knowledge of it.

# generalizzazione: sintassi



- generalizzazione/specializzazione
  - super-classe/sotto-classe

# un esempio di generalizzazione



- (tutti) i gatti sono animali
- Minou è sia un Gatto che un Animale

# associazione: navigabilità



- Ci si può “spostare” da un corso ai suoi studenti [accessibilità]
  - legato alla “ownership”: il corso “conosce” lo studente
  - anche, la classe studente è riusabile senza la classe corso, non viceversa
- La navigabilità è indipendente dal verso di lettura del nome dell’associazione

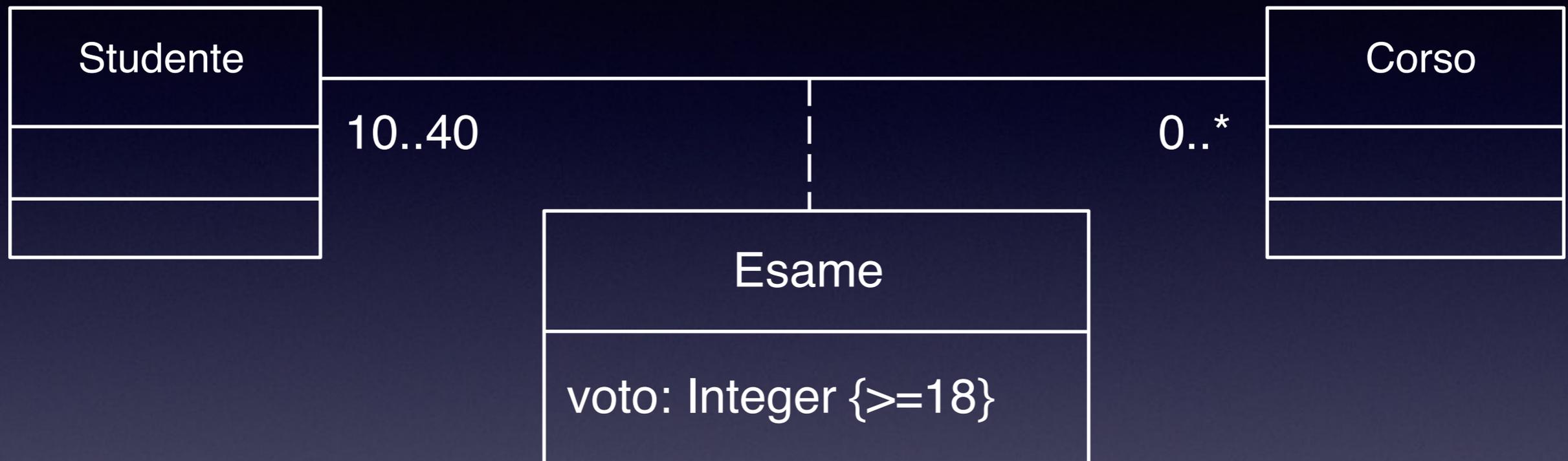
# associazioni qualificate



- Un qualificatore è un attributo di un'associazione:  
individua un oggetto tra quelli collegati dall'associazione

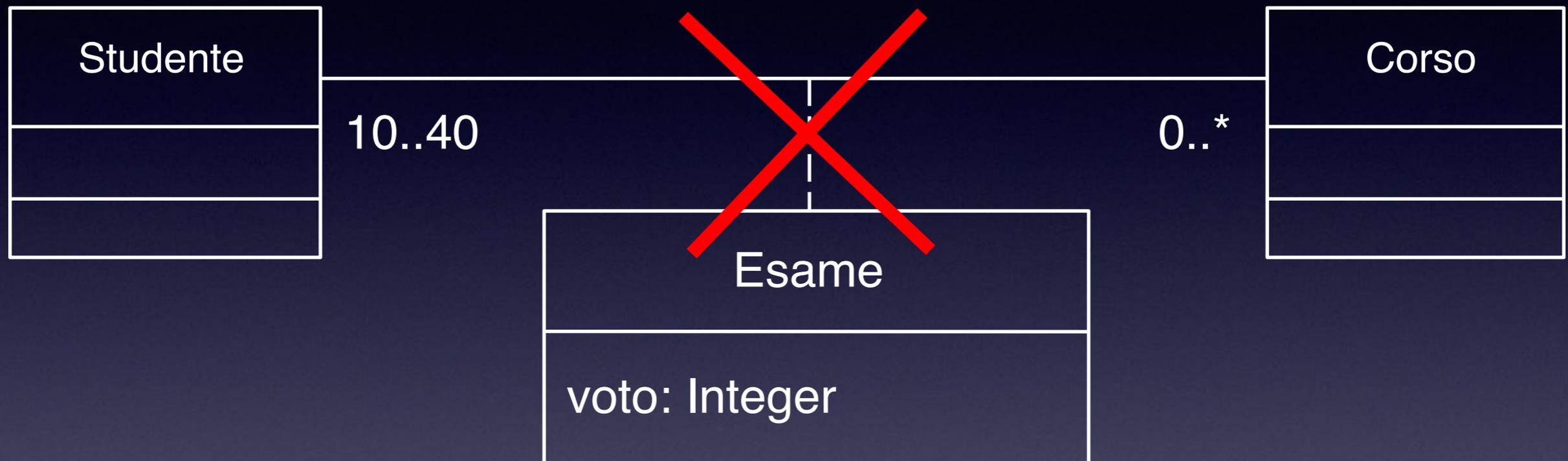


# classi associazione



- 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



- Volendo tenere traccia anche dei voti negativi...
- ...dovremmo avere più di un oggetto della classe associazione per ogni coppia di oggetti collegati fra loro...
- cosa che è vietata! [A meno di non usare {bag}... che non vedremo]

# dipendenze



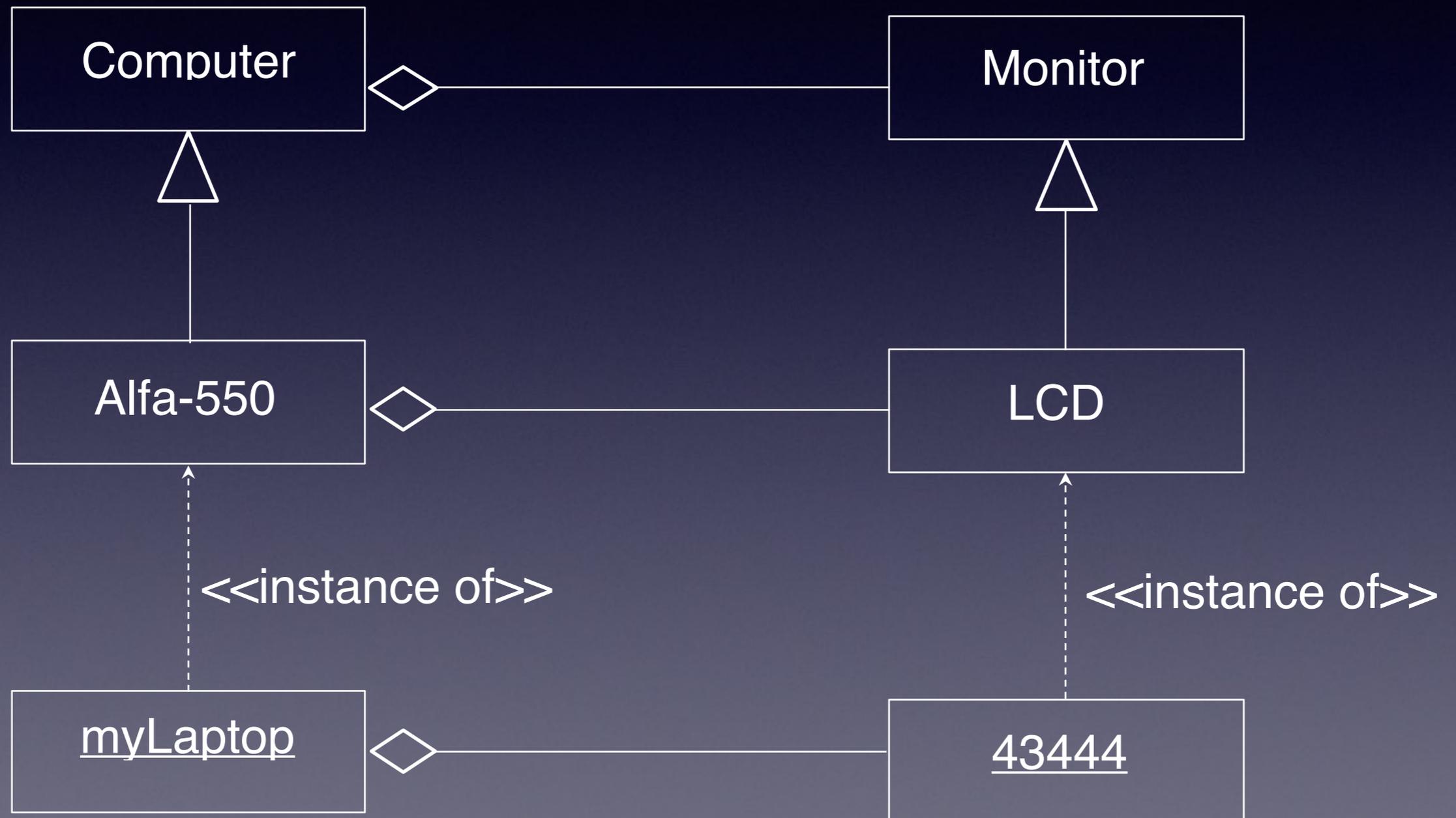
- Una relazione nella quale una modifica nel *fornitore* può influenzare il *cliente*
- Il cliente *dipende* dal fornitore

# esempio di dipendenze d'uso



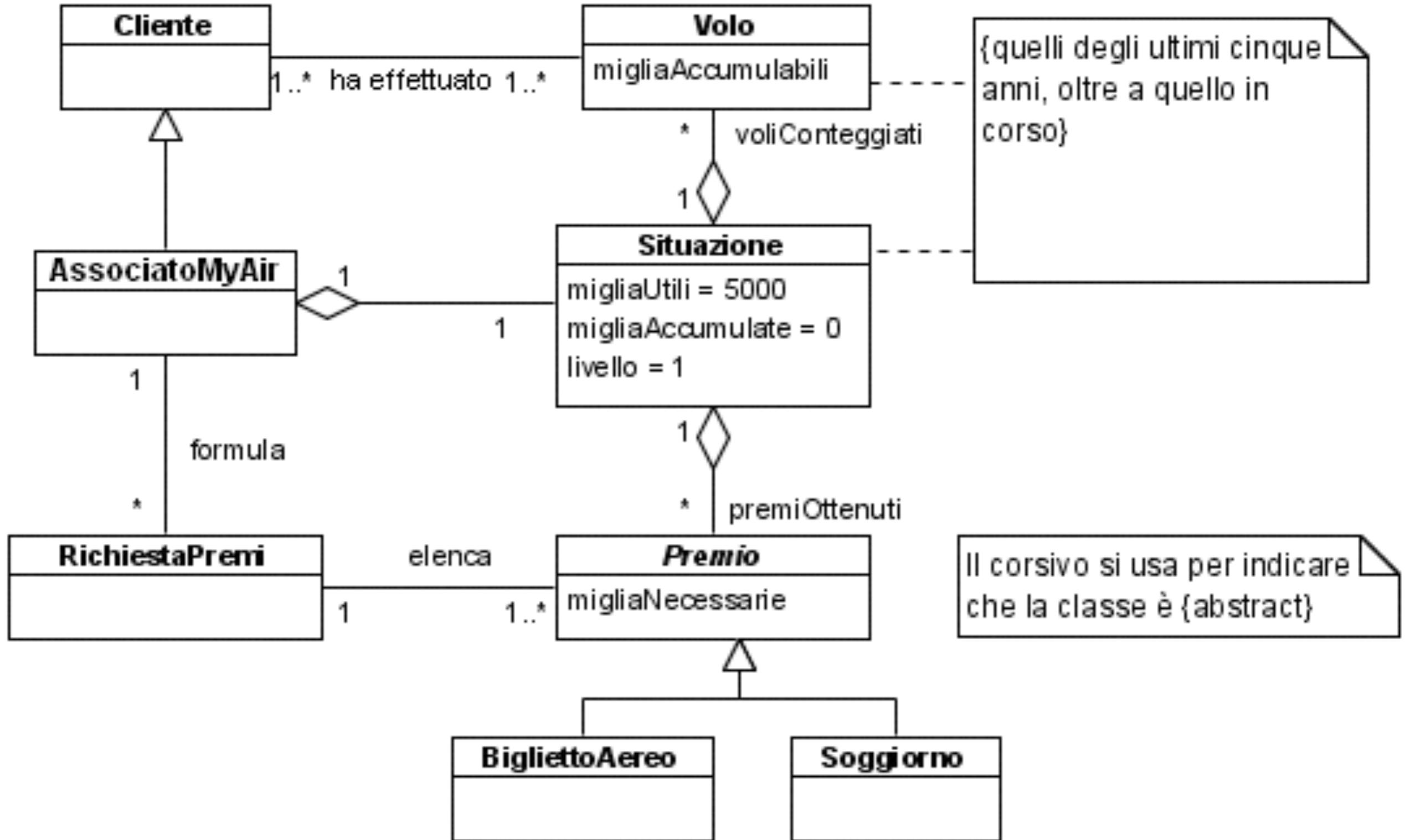
- Un parametro di un'operazione di A è di tipo B
- Un'operazione di A restituisce un oggetto di tipo B
- Un'operazione di A crea dinamicamente un oggetto di tipo B

# esempio paradigmatico



# un esempio: MyAir

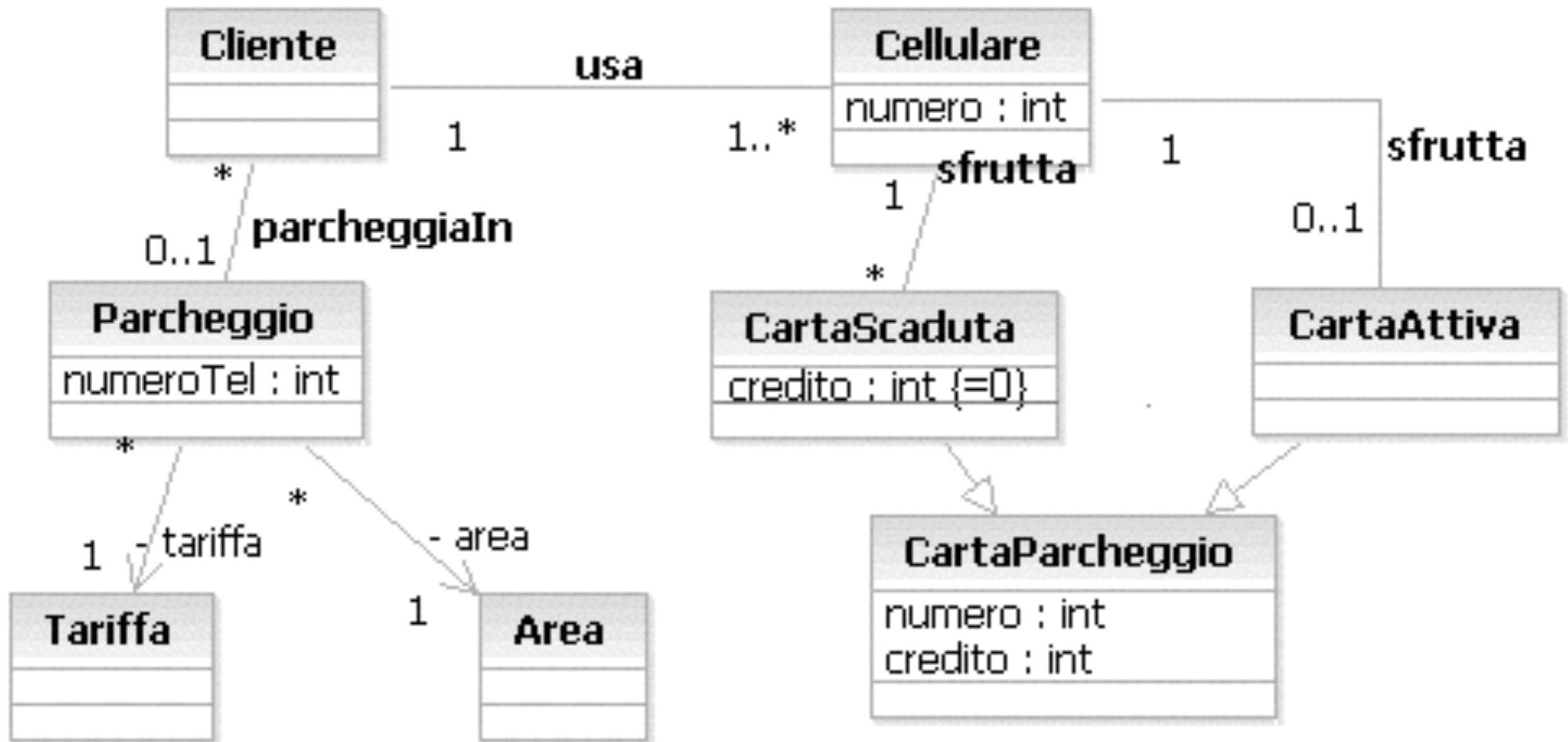
- Iscriviti al programma e da semplice cliente diventerai un associato MyAir, guadagnando immediatamente un bonus di 5.000 miglia utili
- Ogni volta che volerai con MyAir le miglia accumulabili del volo saranno sommate alle tue miglia utili, permettendoti di raggiungere in poco tempo le miglia necessarie 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 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 oro<sup>2</sup>
- Tutte le condizioni si riferiscono esclusivamente alle miglia accumulate in un anno. Il passaggio di livello 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



# un altro esempio: EasyPark

## ★ Soluzione per il pagamento del parcheggio via telefono cellulare

- Il cliente acquista una Carta Parcheggio prepagata e la 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
- 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, è identificato attraverso il proprio numero di telefono cellulare e il sistema attiva il pagamento della sosta
- Il Controllore verifica l'effettivo pagamento della sosta inserendo il numero della Carta Parcheggio in un applicativo fruibile tramite Pocket PC connesso a internet
- 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



- Si modelli anche la sosta, indicando inizio, fine e costo (mediante una classe associazione)

# enumerazioni

```
<<enumeration>>  
Nome
```

- Un tipo enumerabile dichiara un insieme di “letterali” che possono essere usati come valori
  - Boolean è un tipo enumerazione predefinito (con letterali false e true)
- Insiemi ordinati ma senza altre proprietà
  - di solito, sia il nome che i valori sono indicati