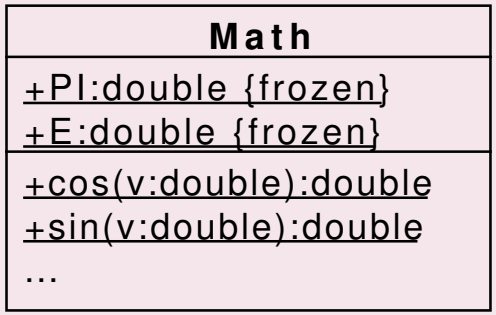
Diagrammi delle classi
Class Diagram
Concetti Avanzati

a cura di **Angelo Furfaro**
da “UML Distilled”
Martin Fowler

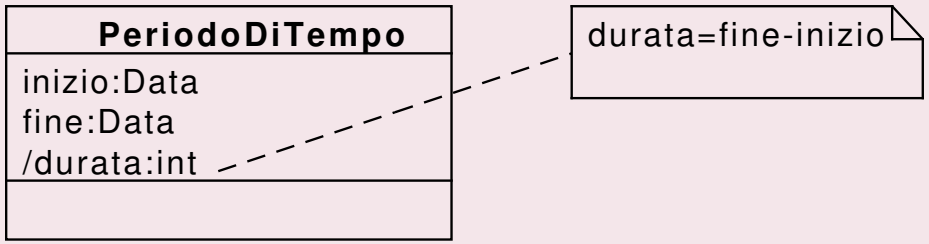
Dipartimento di
Ingegneria Informatica, Elettronica, Modellistica e Sistemistica
Università della Calabria, 87036 Rende(CS) - Italy
Email: a.furfaro@unical.it
Web: http://angelo.furfaro.dimes.unical.it

Operazioni ed attributi statici



- UML chiama **static** un'operazione o un attributo che si applicano ad una classe anziché alle sue istanze.
- Le caratteristiche statiche sono sottolineate.

Proprietà derivate

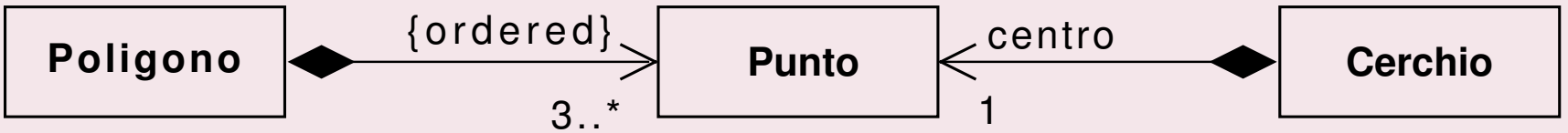


- Sono quelle che possono essere calcolate a partire da altri valori.
- La durata di un periodo di tempo può essere calcolata se si conoscono l'inizio e la fine.
- La derivazione può essere interpretata in due modi:
  - per differenziare valori calcolati da valori memorizzati;
  - come vincolo tra i valori delle proprietà coinvolte.

Aggregazione

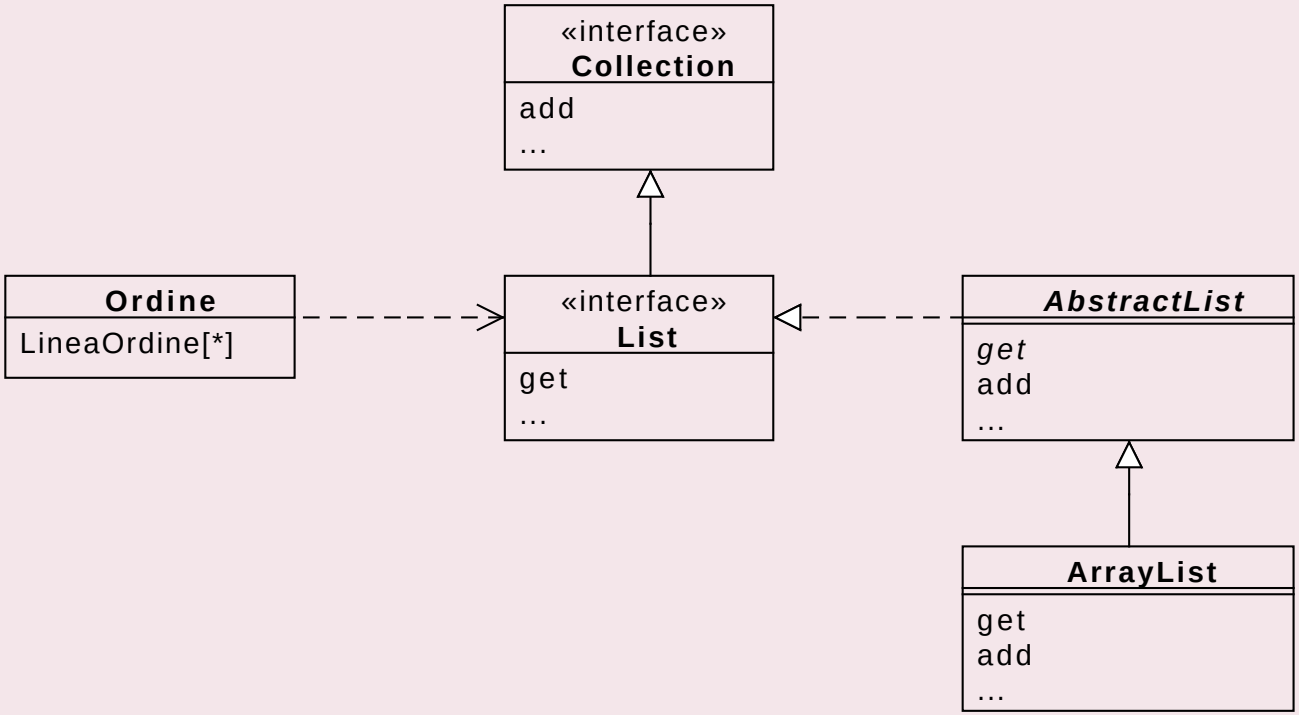
- L'**aggregazione** è la relazione "parte-di". UML non definisce una chiara semantica che la distingua dall'associazione.
- Nell'esempio, un certo numero di persone fa parte di un club.

Composizione



- La **composizione** è una forma *forte* di aggregazione.
- Nella figura, un'istanza di `Punto` può essere parte di un `Poligono` oppure il centro di un `Cerchio` ma non entrambe le cose.
- Sebbene una classe possa essere componente di molte altre, ciascuna sua istanza può essere componente di un solo oggetto.
- La molteplicità lato oggetto composto è implicitamente **0..1**. Quando è specificata pari a **1** la classe componente può appartenere ad una sola altra classe.
- La cancellazione di un `Poligono` dovrebbe implicare la cancellazione di tutti i suoi punti.

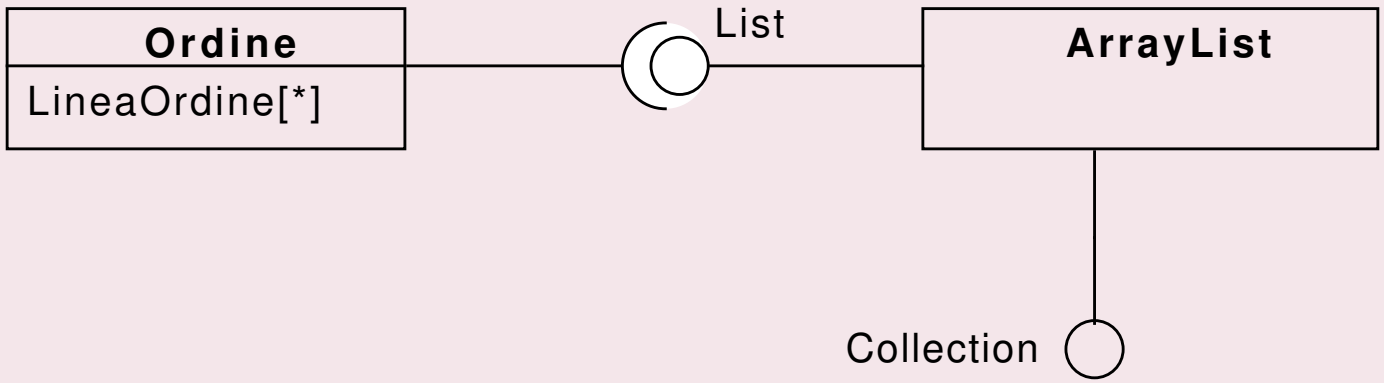
Interfacce e classi astratte



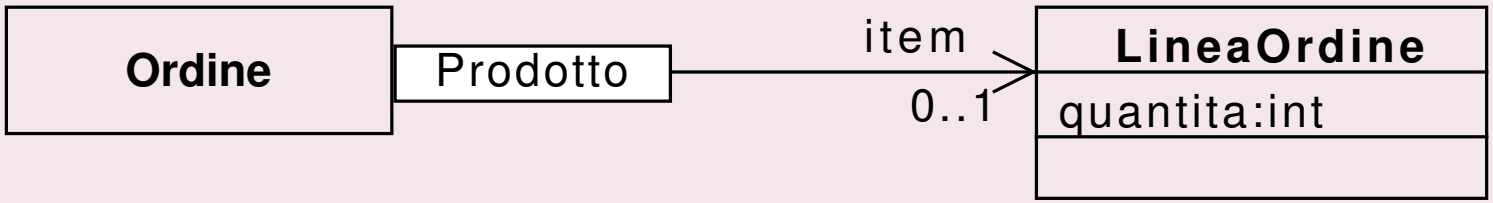
- Le classi astratte si indicano scrivendone il nome in corsivo.
- Anche le operazioni astratte sono scritte in corsivo. In alternativa si utilizza l'etichetta {`abstract`}.
- Le interfacce sono caratterizzate dalla presenza della parola chiave ≪`interface`≫ nel compartimento del nome.

Interfacce e classi astratte

- Una classe **fornisce** un'interfaccia se è sostituibile ad essa. Ciò si indica graficamente con una linea tratteggiata che termina con un triangolo vuoto dal lato dell'interfaccia. In Java ciò si realizza mediante la relazione di implementazione.
- Una classe **richiede** un'interfaccia se necessita di un oggetto conforme ad essa per funzionare (dipende da essa).
- Una notazione alternativa, più compatta, fa uso di *lollipop*, per indicare un'interfaccia fornita, e *socket*, per indicare un'interfaccia richiesta.



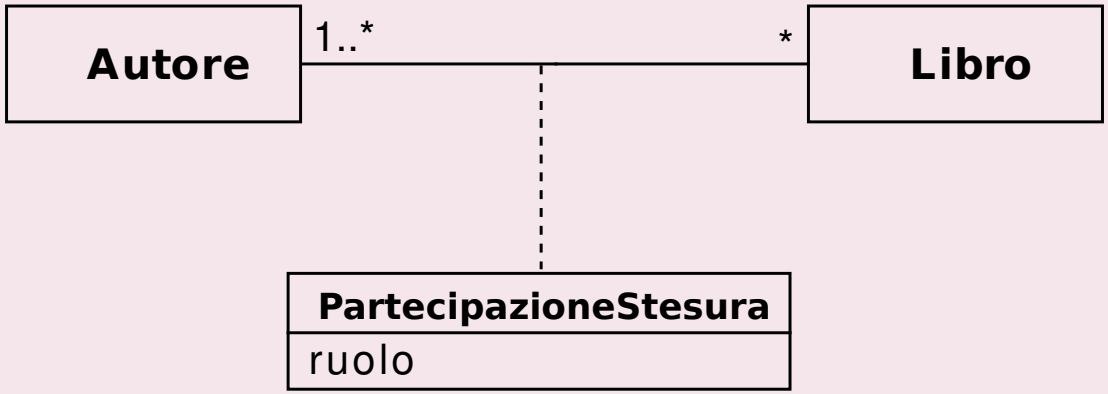
Associazioni qualificate



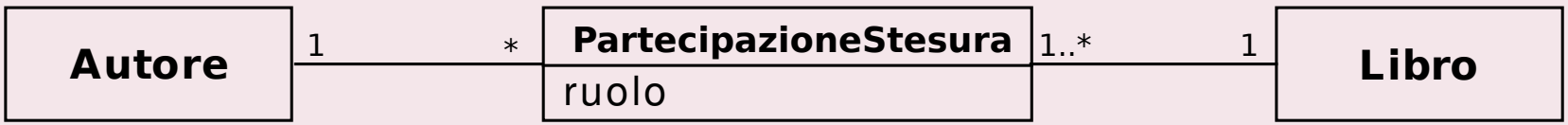
- Un'associazione qualificata è l'equivalente UML di concetti quali array associativi, tabelle hash, mappe, dizionari.
- Il qualificatore indica che per ogni istanza di `Prodotto` ci può essere una solo linea d'ordine collegata ad un'istanza di `Ordine`.
- La molteplicità va considerata nel contesto del qualificatore: un'ordine può avere più linee d'ordine ma al più una per ciascun prodotto.
- Dal punto di vista implementativo l'associazione qualificata in figura implica un'interfaccia simile alla seguente:

```
class Ordine ... {
    ...
    public LineaOrdine getElementoLinea(Prodotto p){ ... }
    public void addElementoLinea(int quantita, Prodotto p) { ... }
    ...
}
```

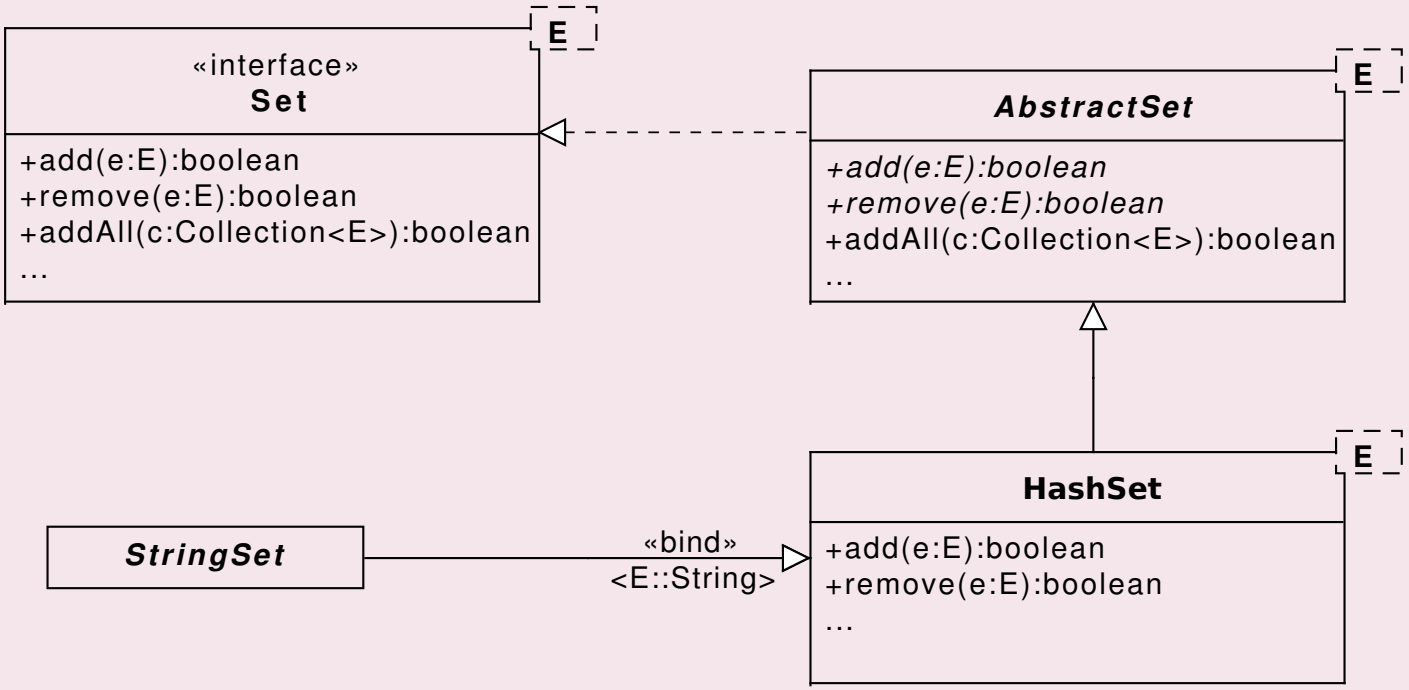
Classi di associazione (classi associative)



- Le classi associative permettono di aggiungere attributi, operazioni ed altre caratteristiche alle associazioni.
- Nella figura l'associazione tra `Autore` e `Libro` specifica il ruolo svolto dall'autore nella stesura del libro: autore principale, autore secondario, traduttore.
- La figura seguente mostra un modo alternativo per rappresentare tale informazione. Si noti come si spostano le molteplicità.
- *Ci può essere una sola istanza della classe associazione per ogni coppia di oggetti associati.*

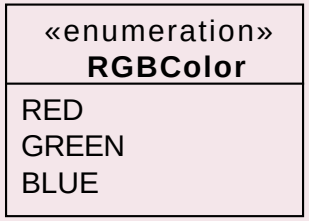


Classi Parametriche



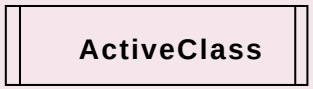
- Molti linguaggi supportano, con i dovuti distinguo, la nozione di **classe parametrica** (i template in C++, i generici in Java)
- La notazione UML è quella riportata in figura (rif. Java collection framework).
- Un uso particolare di una classe parametrica si chiama **derivazione**.
- La relazione tra elemento legato e template si chiama raffinamento ed è indicata con lo stereotipo ≪bind≫

Enumeration



- Le enumerazioni sono utilizzate per modellare tipi che possono assumere solo un insieme finito di valori prefissati.
- Sono rappresentate da una classe marcata con la parola chiave ≪enumeration≫ che riporta l'elenco dei valori.

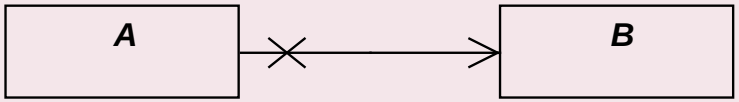
Classi Attive



- Ogni istanza di una classe attiva esegue e controlla il proprio thread.

Ancora sulle associazioni

Non navigabilità



Associazioni riflessive