

The logo consists of the letters 'U', 'M', and 'L' in a bold, blocky font. Each letter is filled with a wood-grain texture and has a black outline. They are arranged horizontally on a light yellow background that has a dashed border and a hole on the left side, resembling a tag. A brown paper-like corner is visible in the top-left corner of the overall image.

UML

---

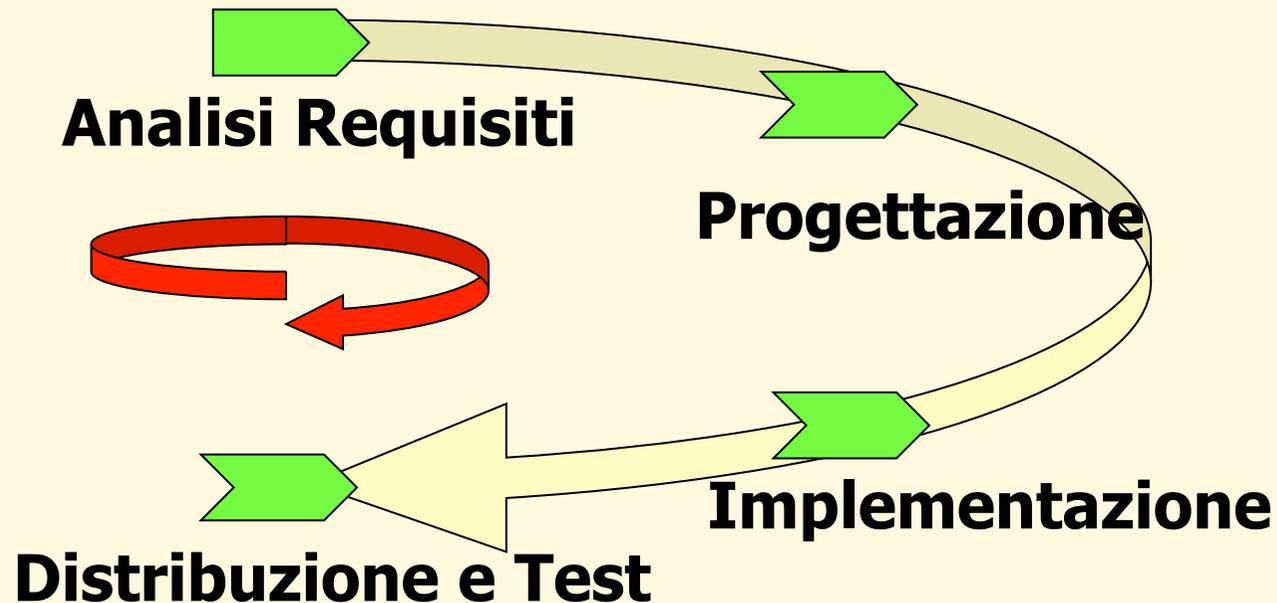
# **Unified Modeling Language**

**Autore: Giselda De Vita**

# Fasi d'elaborazione

La produzione di un sistema software segue diverse fasi di sviluppo.

UML serve a specificare, definire e progettare tutte queste fasi.



# Cos'è UML?

**E' un linguaggio visuale il cui scopo principale è quello di specificare, visualizzare, costruire e documentare un sistema software.**

**UML rappresenta concettualmente e fisicamente sistemi tramite diagrammi**

**Non è un linguaggio di programmazione**

# Casi d'Uso

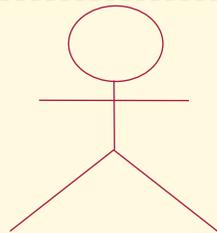
Un caso d'uso è un'interazione fra un attore, solitamente un utente, ed un programma.

I diagrammi di use case descrivono una serie di funzioni visibili all'utente che realizzano un obiettivo definito.

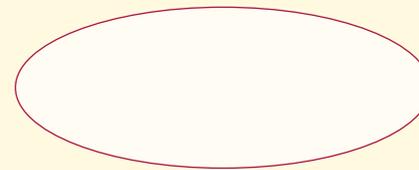
Gli elementi grafici che compongono questi diagrammi sono molto semplici.

La parte di maggiore importanza sono i documenti con specifiche testuali.

# Casi d'Uso - Simboli



**Attore**

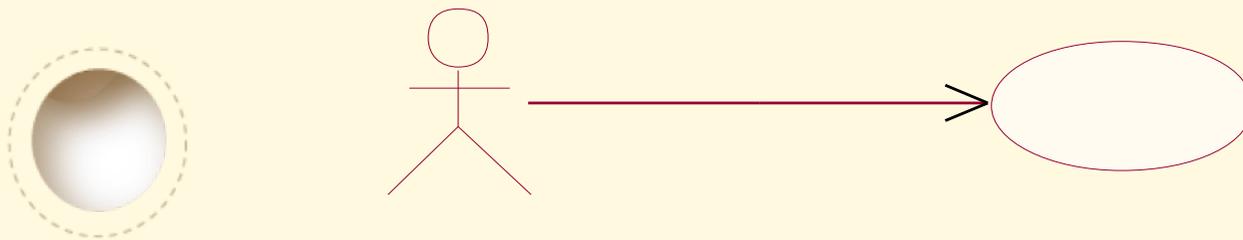


**Caso d'Uso**

Un attore è un'astrazione di un'entità esterna che interagisce direttamente con il sistema.

Uno use case descrive una sequenza di azioni che portano ad un risultato osservabile e di valore per un particolare attore.

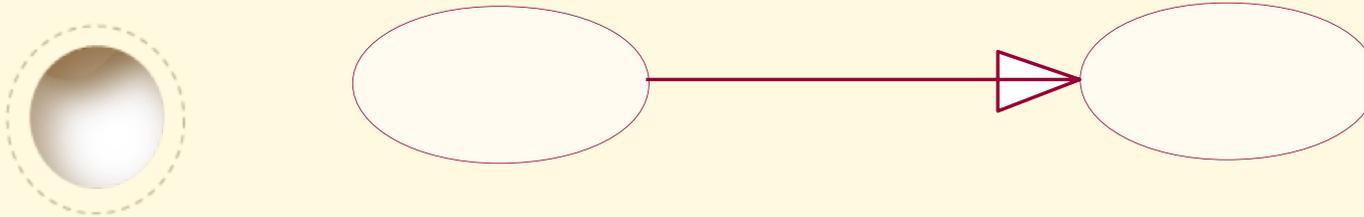
# Casi d'Uso - Diagrammi



Un diagramma di casi d'uso (**use case diagram**) è un diagramma che mostra le relazioni fra attori e use case all'interno del sistema.

Le relazioni fra un attore ed un use case viene indicata graficamente con una freccia

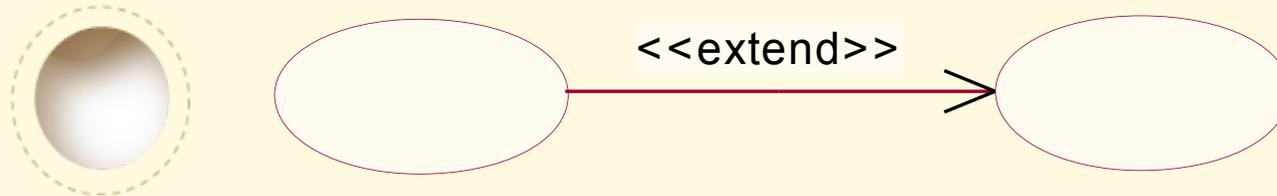
# Casi d'Uso - Relazioni



## Generalizzazione

Associa un use case specializzato ad un use case più generale. L' 'use case figlio eredita gli attributi e le sequenze di operazioni dall' use case padre e in più può aggiungere delle nuove operazioni.

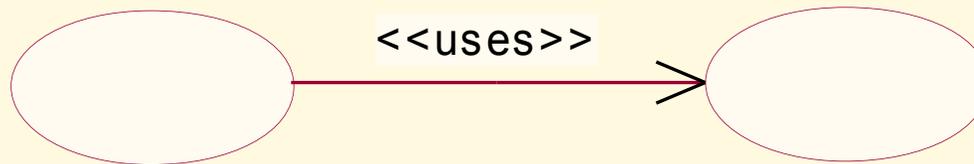
# Casi d' Uso - Relazioni



## Estensione

E' un tipo di relazione di dipendenza. Un use case estende un use case base aggiungendogli ulteriori sequenze d'azioni.

# Casi d' Uso - Relazioni

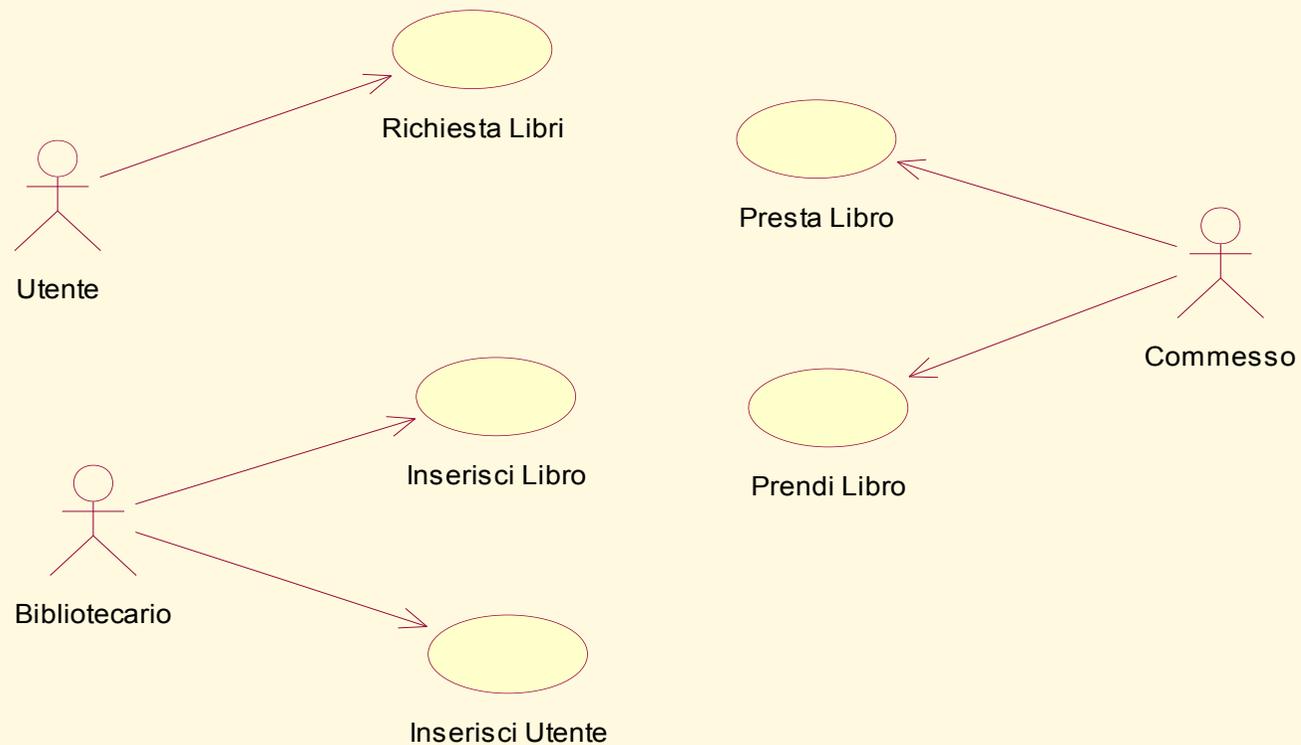


## Inclusione

E' un tipo di relazione di dipendenza che denota l'inclusione di una serie di operazioni descritte dall'use case "incluso".

# Casi d'Uso - Esempio

## Gestione di una Biblioteca



# Casi d' Uso - Esempio

## Documentazione

USE CASE	Richiesta Libri	
Scopo	Soddisfare la richiesta di una lista di libri presenti in Biblioteca	
Pre condizione		
Attori	Utente	
DESCRIZIONE	Passo	Azioni
	1	L'utente si collega al sistema
	2	Il sistema mostra la Form dove inserire l'ID
	3	L'utente inserisce l'ID
	4	Il sistema valida l'ID ( <b>Variazione 1</b> ) e mostra la form della richiesta
	5	L'utente inserisce i criteri di ricerca
	6	Il sistema ricerca i libri che corrispondono ai criteri di ricerca impostati e mostra all'utente la lista trovata ( <b>Variazione 2</b> )
	7	L'use case termina
ESTENSIONE		
		Condizione:
VARIAZIONI		Azioni
	1	Il sistema non riconosce l'ID specificato. Segnala l'errore e richiede di reinserirlo.
	2	Il sistema non trova nessun libro. Segnala che la ricerca non ha prodotto risultati e l'use case termina.

# Oggetti Classi - Oggetti

Un oggetto rappresenta un' entità fisica, concettuale o software all' interno del contesto del problema da analizzare.

Un oggetto è caratterizzato da:

- Identità
- Comportamento
- Stato

# Oggetti Classi - Classi

Una classe è la descrizione di un gruppo di oggetti con le stesse proprietà (**attributi**), gli stessi comportamenti (**metodi**), e le stesse relazioni con altri oggetti.

Un oggetto è un'istanza di una classe.

# Diagrammi di Interazione

Questi diagrammi mostrano come gli oggetti interagiscono per realizzare le specifiche funzionali descritte dagli Use Case.

Diagramma d'interazione è un termine generico che viene attribuito a diversi tipi di diagrammi che mettono in risalto le interazioni fra oggetti.

# Diagrammi di Sequenza

Il diagramma di sequenza mostra l'interazione tra oggetti organizzati in sequenza temporale.

In particolare mostra:

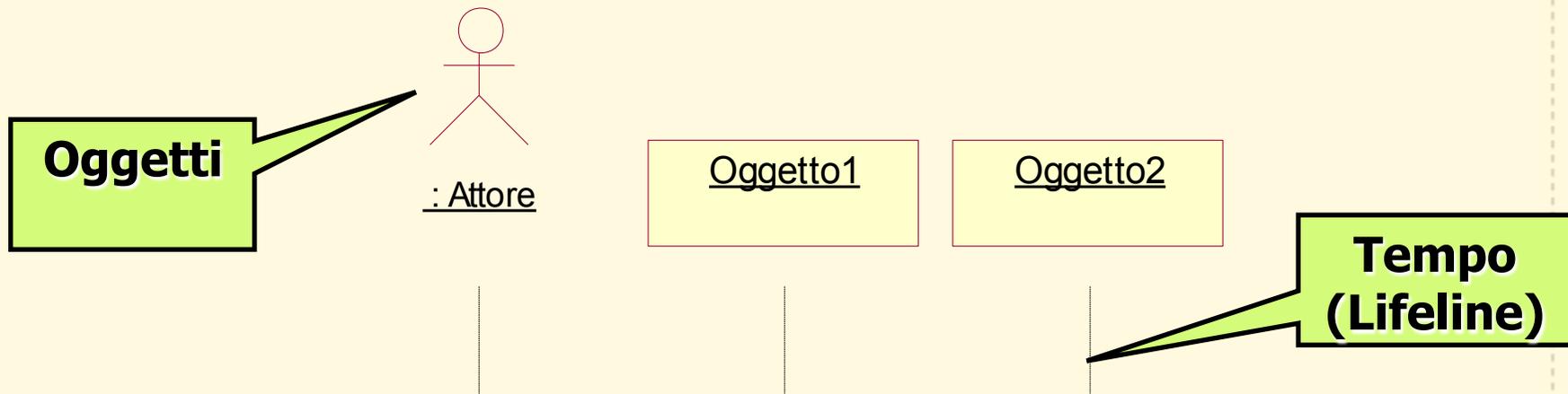
- Gli oggetti che partecipano all' interazione
- La sequenza dei messaggi scambiati.

# Diagrammi di Sequenza - Simboli

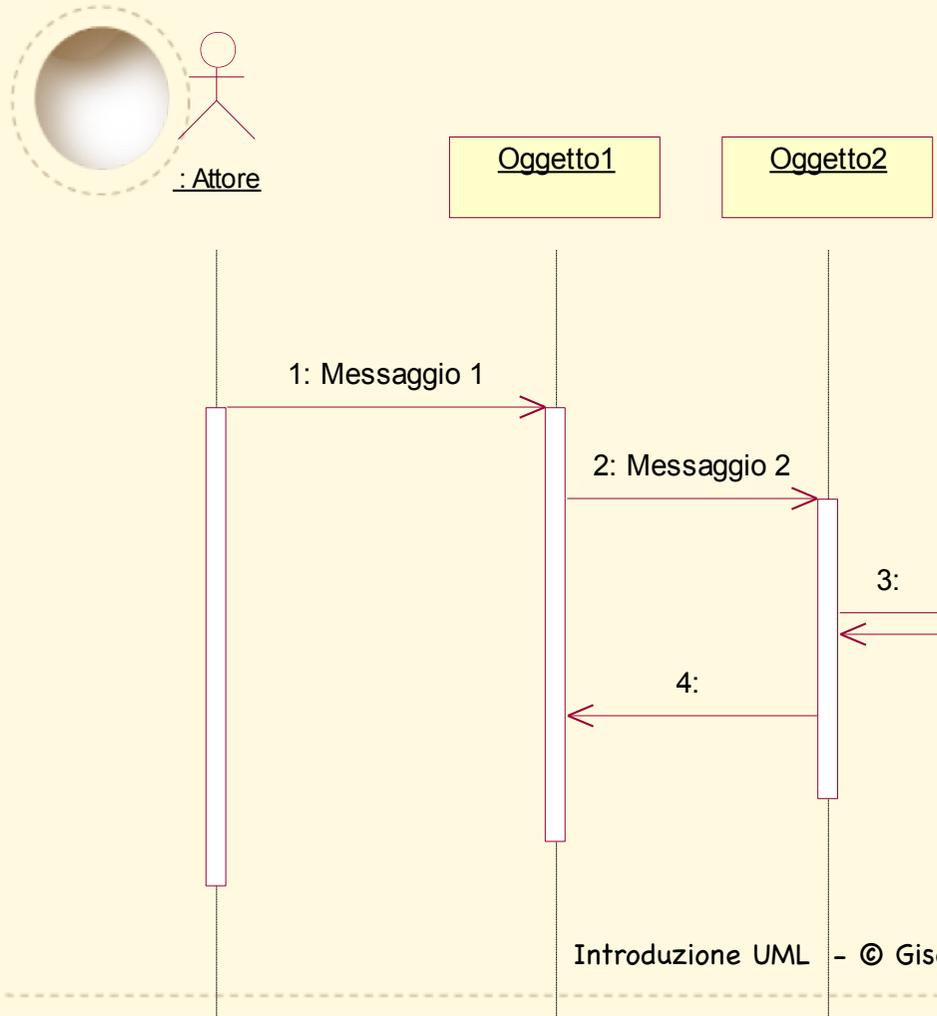
Un Diagramma di Sequenza ha due dimensioni:

La dimensione verticale rappresenta il tempo

La dimensione orizzontale riporta gli oggetti che partecipano all'interazione.

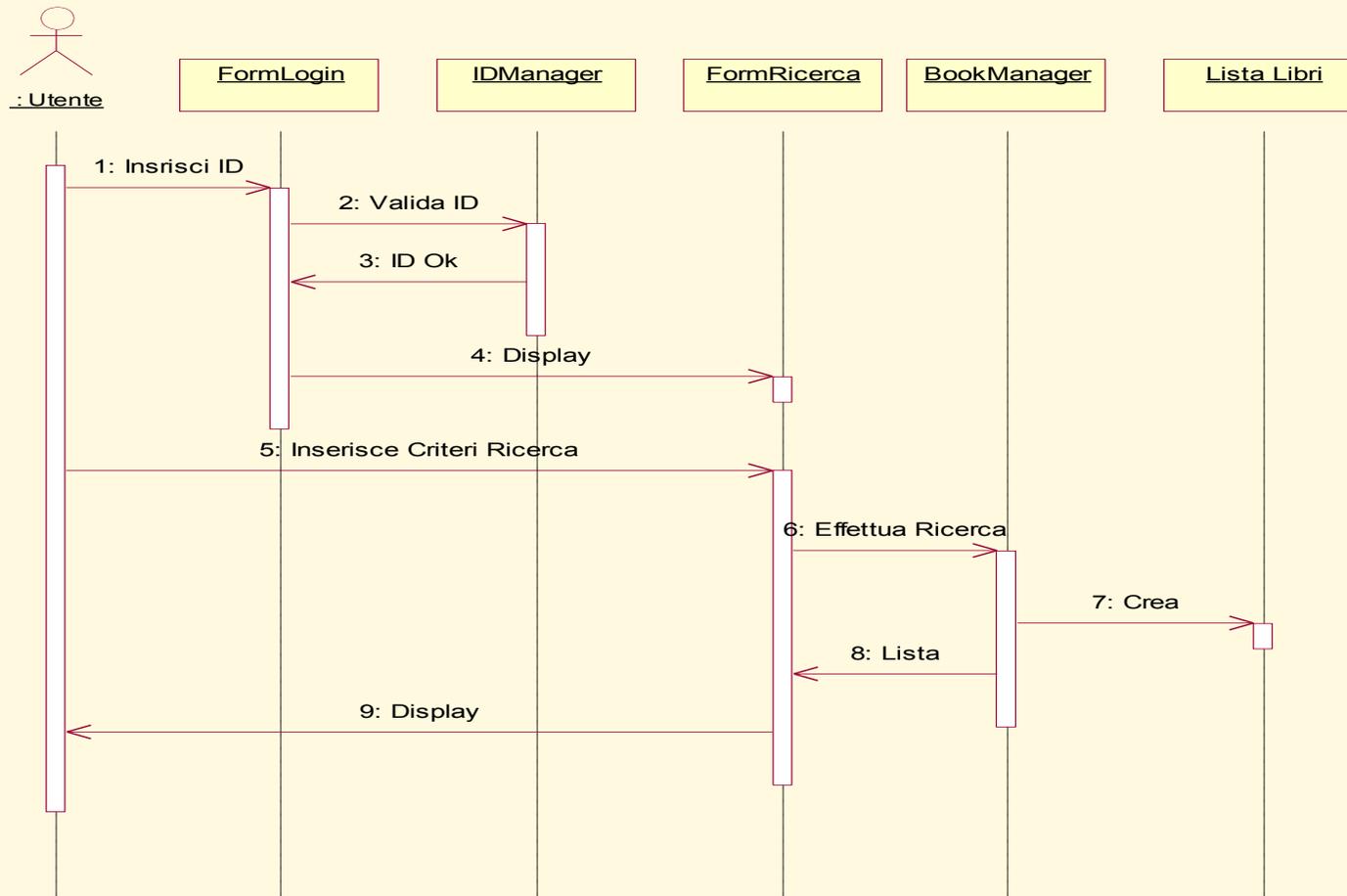


# Diagrammi di Sequenza - Relazioni



Un messaggio scambiato fra oggetti viene indicato con una freccia orizzontale che parte dalla lifeline del mittente e termina sulla lifeline del ricevente.

# Diagrammi di Sequenza - Esempio



# Diagrammi di Collaborazione

Nel diagramma di collaborazione viene descritto il flusso d'esecuzione come insieme di oggetti e loro interazioni.

Il sequence diagram ed il collaboration diagram esprimono la stessa quantità d'informazioni in maniera diversa.

Il collaboration non mostra il tempo come una dimensione separata.

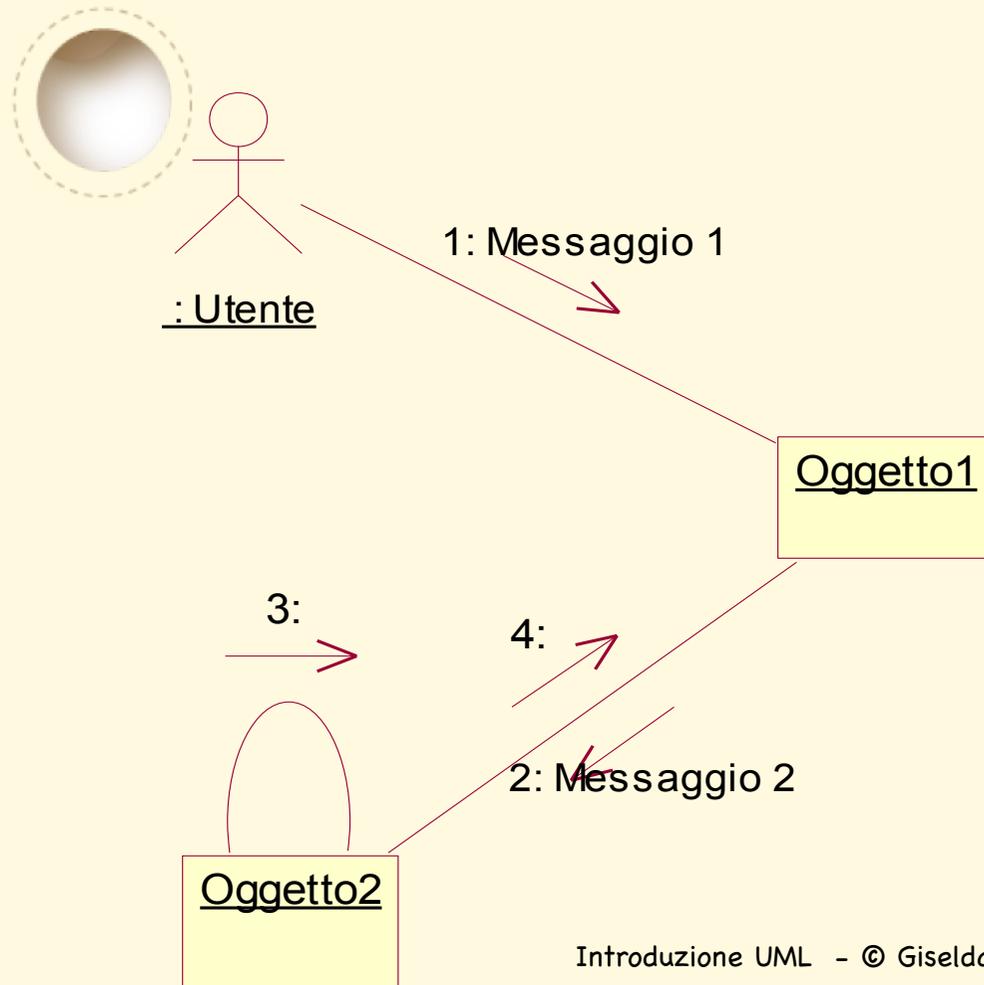
# Diagrammi di Collaborazione - Simboli

L'oggetto viene rappresentato tramite un rettangolo in cui viene specificata l'istanza della classe.

Oggetto:  
Classe

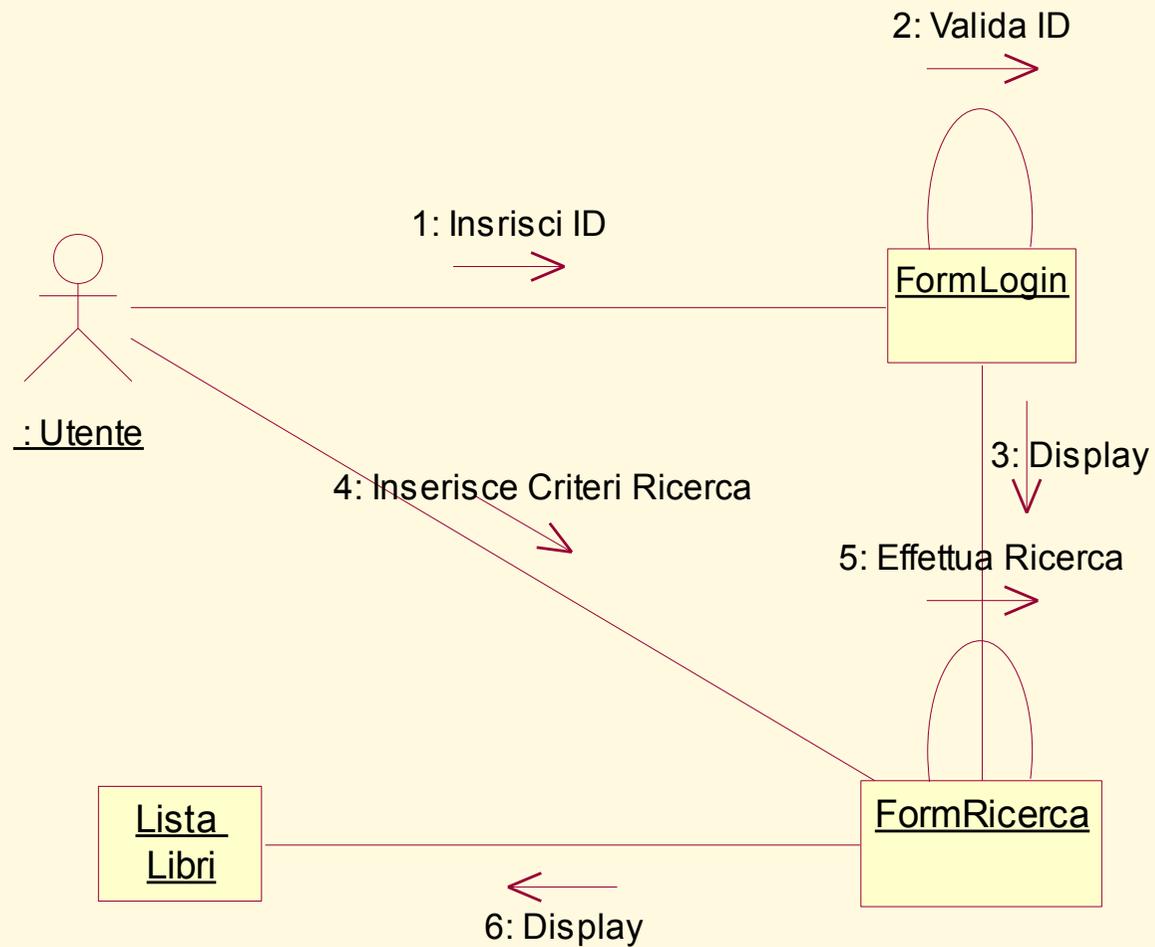
Le interazioni fra gli oggetti sono specificate da una linea.

# Diagrammi di Collaborazione - Relazioni



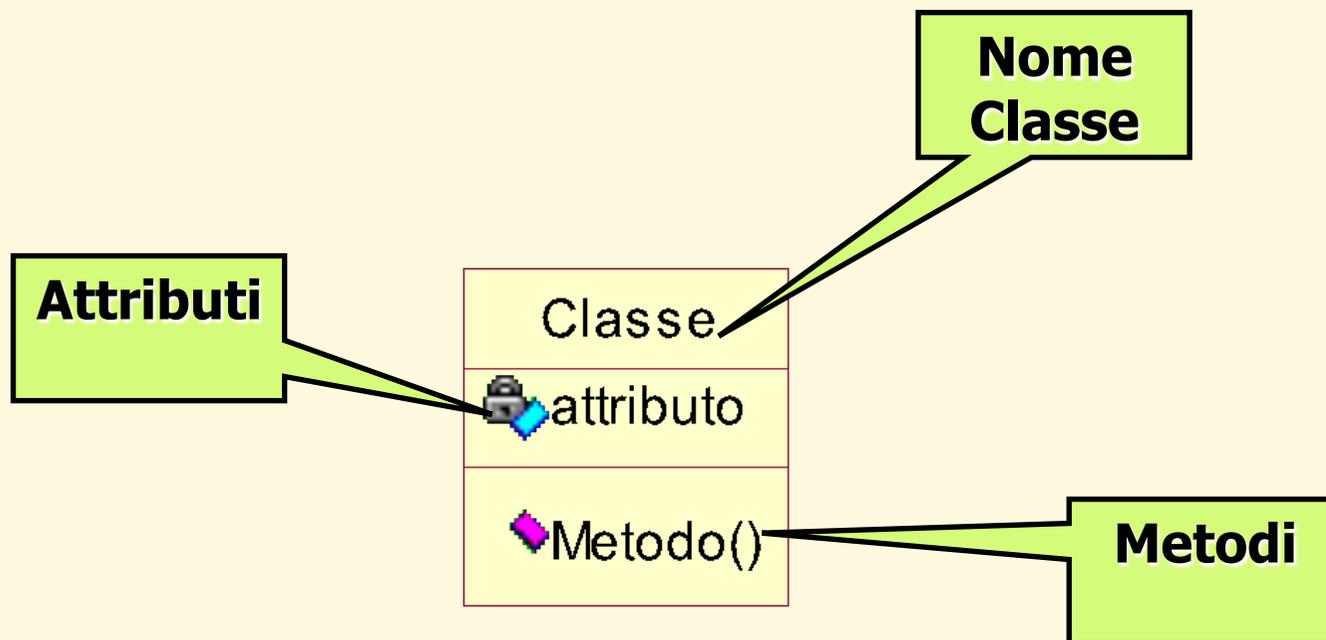
Al di sopra della linee di congiunzione fra due oggetti, si pone un messaggio che viene numerato a partire da 1 in ordine crescente al fine di dare un ordine alle iterazioni

# Diagrammi di Collaborazione - Esempio



# Diagramma delle Classi - Simboli

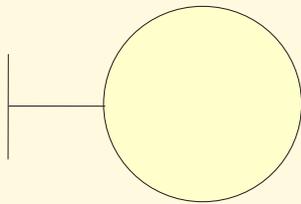
Una classe viene rappresentata da un rettangolo in 3 compartimenti



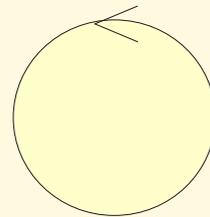
# Diagramma delle classi - Tipi di classe

Uno stereotipo è una tipizzazione di una classe.

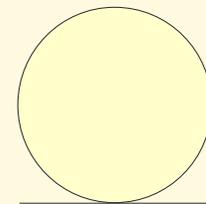
In una generica applicazione a tre livelli gli stereotipi comuni sono:



**Boundary**



**Control**



**Entity**

# Diagramma delle classi - Tipi di classe

**<<boundary>>  
Interfaccia**

Classe d' **interfaccia** che modella la comunicazione tra l'utente ed il sistema

**<<control>>  
Gestione**

Classe di **gestione** che modella il comportamento di controllo per uno o più casi d'uso

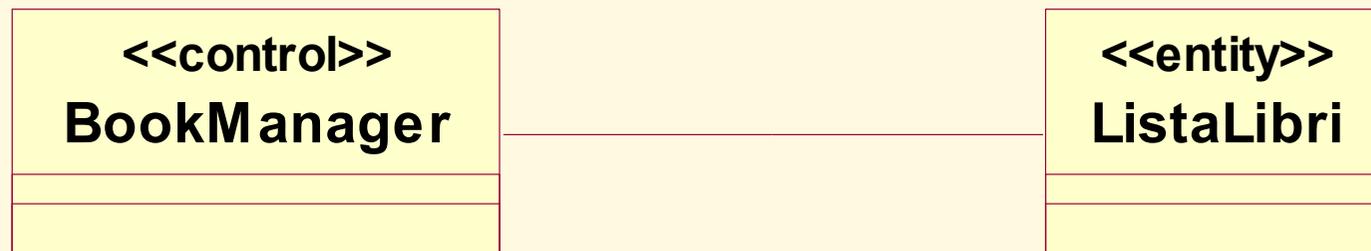
**<<entity>>  
Entità**

Classe di **entità** che modella i dati e i comportamenti associati.

# Diagramma delle classi - Relazioni

Una relazione di Associazione è un collegamento semantico bi-direzionale tra classi.

La rappresentazione grafica è una linea che collega le due classi.



# Diagramma delle classi – Relazioni

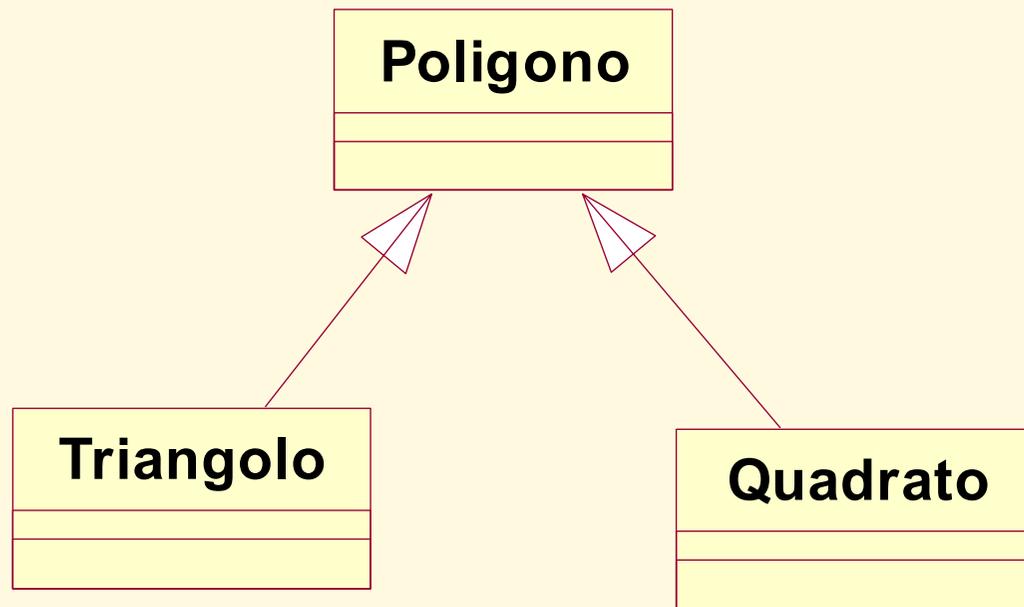
Una relazione di **Aggregazione** è una speciale relazione di associazione che relaziona una parte aggregante e le sue parti costituenti.

La **Molteplicità** è il numero di istanze di una classe con relazione ad una istanza di un'altra classe.



# Diagramma delle classi - Relazioni

L' Ereditarietà è una relazione di una classe (classe **figlia**) che condivide la struttura e i comportamenti di un' altra classe (classe **padre**).



# Diagramma delle classi - Relazioni

L' **Aggregazione** è una relazione del tipo: **Ha un**  
L' **Ereditarietà** è una relazione del tipo : **E' un**

**Esempio:**

- Il Poligono **ha** 3 o più segmenti (Aggregazione)
- Un triangolo **è** un poligono (Ereditarietà)
- Un segmento **non è** un poligono
- Un poligono **non è** un segmento
- Il poligono **non ha** un triangolo

# Diagramma delle classi - Esempio

