



# CORSO JAVA AESYS

## Parte Prima

2022

[www.aesystech.it](http://www.aesystech.it)





# UML

- Definizione
- Use case diagram
- UML Class Diagram
- Component e deployment diagram
- State Diagram

# Definizione

Oggigiorno si sente parlare molto spesso di UML, ma non tutte le persone che parlano di UML sanno di che cosa realmente si tratti. Qualcuno pensa che sia un linguaggio di programmazione e quest'equivoco è dovuto alla parola "language". Qualcun altro pensa si tratti di una metodologia object-oriented e questo è probabilmente dovuto a cattiva interpretazione di letture non molto approfondite. Infatti si sente spesso parlare di UML congiuntamente a varie metodologie. Per definire quindi correttamente che cos'è UML, è preferibile prima definire per grandi linee che cos'è una metodologia.

Una metodologia object-oriented, nella sua definizione più generale, potrebbe intendersi come una coppia costituita da un processo e da un linguaggio di modellazione

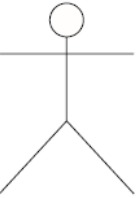
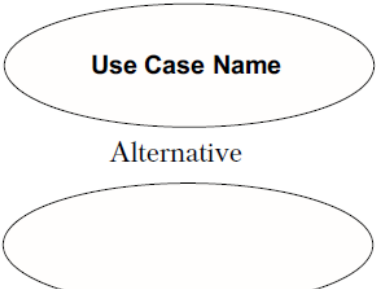
A sua volta un processo potrebbe essere definito come la serie di indicazioni riguardanti i passi da intraprendere per portare a termine con successo un progetto. Un linguaggio di modellazione è invece lo strumento che le metodologie utilizzano per descrivere (possibilmente con dei grafici) tutte le caratteristiche statiche e dinamiche di un progetto.


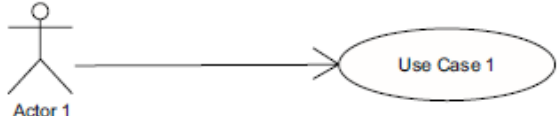
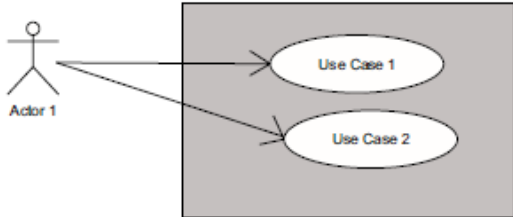
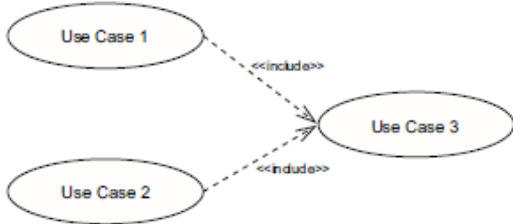
UML non è altro che un linguaggio di modellazione. Esso è costituito per linee generali da una serie di diagrammi grafici i cui elementi sono semplici linee, ovali, rettangoli, omini stilizzati e così via. Questi diagrammi hanno il compito di descrivere in modo chiaro tutto ciò che durante un progetto potrebbe risultare difficile o troppo lungo con documentazione testuale.

# Use case diagram 1/2

Unified Modeling Language Syntax Reference		
Nome Diagramma	Nomi degli Elementi	
Use Case Diagram	Actor	Use Case
	Relationship Link	System Boundary
	Inclusion	Extension
	Generalization	Actor Generalization
Diagramma dei casi d'uso: rappresentano le interazioni tra il sistema e gli utenti del sistema stesso.		

# Use case diagram 2/2

Use Case Diagram Syntax Reference	
Nome Diagramma	Nomi degli Elementi
Actor	 <p>Alternative</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <b>&lt;&lt;Actor&gt;&gt;</b>  <b>Actor Name</b> </div> <p>Actor Name</p>
<p>Attore: ruolo interpretato dall'utente nei confronti del sistema.  N.B.: un utente potrebbe non essere una persona fisica.</p>	
Use Case	 <p>Alternative</p> <p>Use Case Name</p>
<p>Caso d'uso: insieme di scenari legati da un obiettivo comune per l'utente.  Uno scenario è una sequenza di passi che descrivono l'interazione tra l'utenza ed il sistema. NB: è possibile descrivere scenari mediante diagrammi dinamici.</p>	

Use Case Diagram Syntax Reference	
Relationship link	 <p>Alternative</p> 
<p>Associazione (o relazione): associa logicamente un attore ad un caso d'uso.</p>	
System Boundary	
<p>Sistema (o delimitatore del sistema): delimitatore del dominio del sistema.</p>	
Inclusion	
<p>Inclusione: relazione logica tra casi d'uso, che estrae un comportamento comune a più casi d'uso.</p>	

# Class Diagram syntax reference

Unified Modeling Language Syntax Reference			
Class Diagram	Class / Object	Association / link	Navigability
	Attribute	Aggregation	Multiplicity
	Operation	Composition	Qualified Association
	Member Properties	Extension	Association Class
	Abstract Class / Interface	Implementation	Roles Names
<p>Diagramma delle classi: descrive le classi che compongono il sistema e le relazioni statiche esistenti tra esse.</p> <p>N.B.: quando un diagramma mostra gli oggetti del sistema, spesso ci si riferisce ad esso come diagramma degli oggetti (Object Diagram).</p>			

# UML Class Diagram 1/2

Nella tabella seguente sono riportati gli elementi del **diagramma delle classi** (**class diagram**).

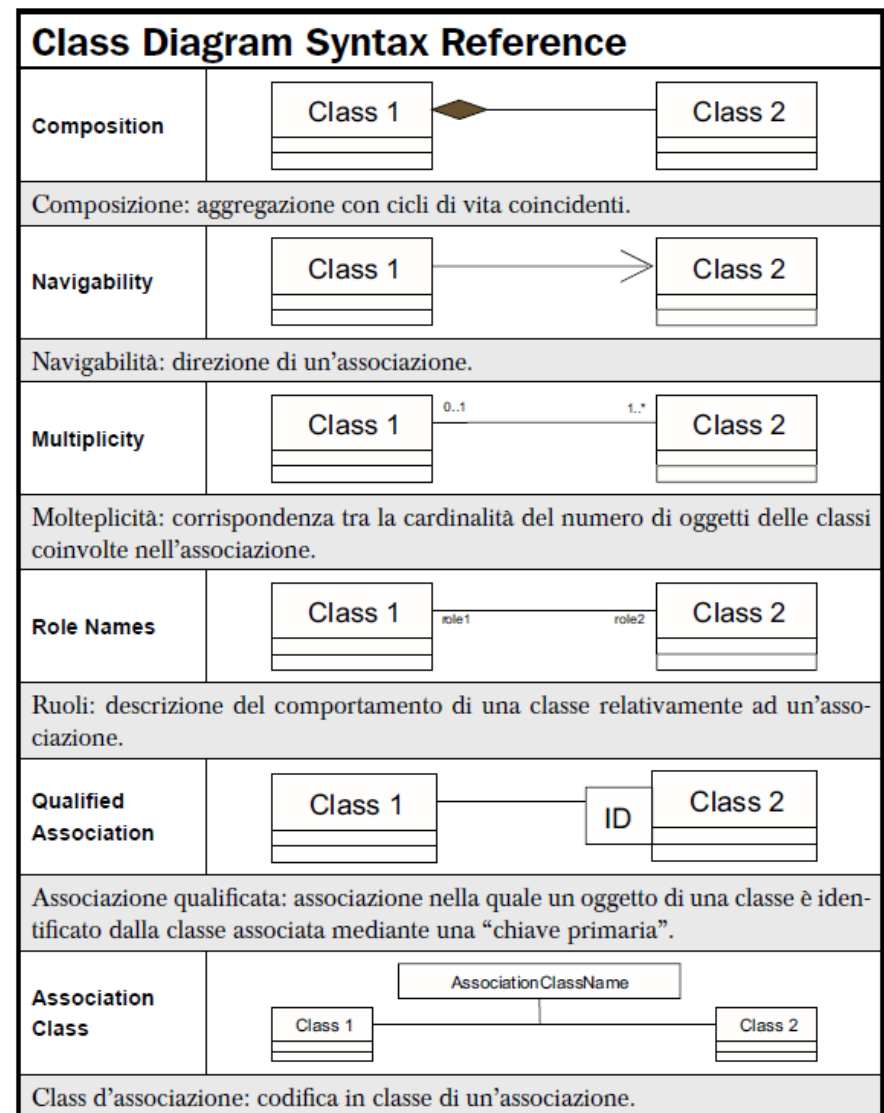
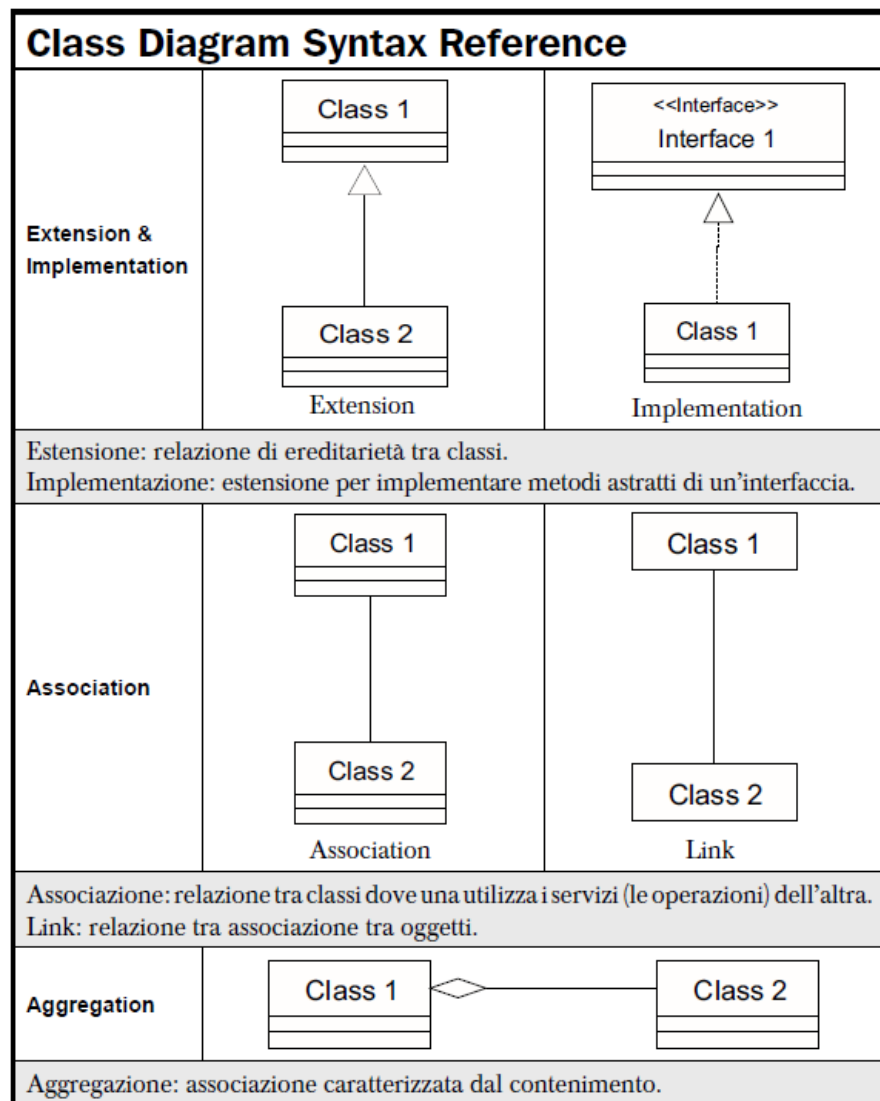
Class Diagram Syntax Reference	
Nome Diagramma	Nomi degli Elementi
Class & Object	<div> <div>ClassName</div> <div>-attributeName: typeName +operationName(): returnType</div> <div>Class</div> </div>
	<div> <div>ObjectName</div> <div>Object</div> </div>
Classe: astrazione per un gruppo di oggetti che condividono stesse caratteristiche e funzionalità. Oggetto: creazione fisica (o istanza) di una classe	
Attribute	<div> <div>ClassName</div> <div>-attributeName: typeName</div> </div>
Attributo (o variabile d'istanza o variabile membro o caratteristica di una classe): caratteristica di una classe/oggetto.	



Class Diagram Syntax Reference		
Operation	ClassName	
	+operationName(): returnType	
Operazione (o metodo o funzione membro): funzionalità di una classe/oggetto.		
Member Properties	+ memberName “public member”	
	# memberName “protected member”	
	- memberName “private member”	
	<u>memberName</u> o \$memberName “static member”	
	operation o operation {abstract} “abstract operation”	
	/attributeName “derived attribute”	
Public, protected, private: modificatori di visibilità. Membro statico (o membro della classe): membro della classe. Operazione astratta: “signature” del metodo (metodo senza implementazione). Attributo derivato: attributo ricavato da altri.		
Abstract Class & Interface	ClassName	
	Abstract Class	
	<<Interface>> InterfaceName	
	Interface	
Classe astratta: classe che non si può istanziare e che può dichiarare metodi astratti. Interfaccia: struttura dati non istanziabile che può dichiarare solo metodi astratti (e costanti statiche pubbliche).		

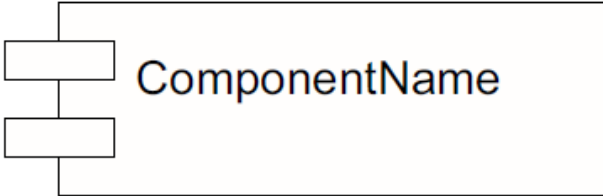
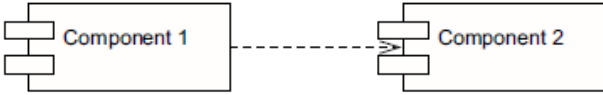

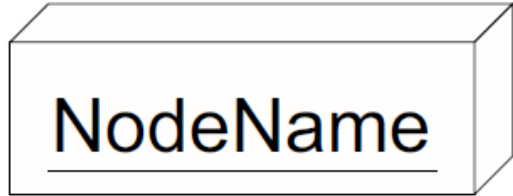


# UML CLASS DIAGRAM 2/2




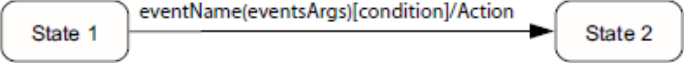



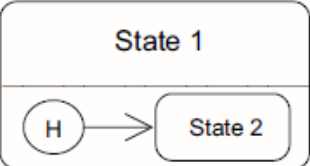


# Component e deployment diagram

Component & Deployment Diagram Syntax Reference	
Nome Diagramma	Nomi degli Elementi
Component	
Componente: modulo software eseguibile, dotato di identità e con un'interfaccia ben specificata, di cui di solito è possibile uno sviluppo indipendente.	
Dependency	
Dipendenza: relazione tra due elementi di modellazione, nella quale un cambiamento sull'elemento indipendente avrà impatto sull'elemento dipendente.	
Link	
Associazione (o relazione): relazione logica nella quale un partecipante (componente o nodo o classe) utilizza i servizi dell'altro partecipante.	
Node	
Nodo: rappresentazione di una piattaforma hardware.	

# State diagram

Nella seguente tabella sono riportati gli elementi del **diagramma degli stati** (**state diagram** o **state transition diagram**).

State Transition Diagram Syntax Reference	
Nome Diagramma	Nomi degli Elementi
State	
Stato: stato di un oggetto. Può essere caratterizzato da azioni (transizioni interne).	
Transition	
Transizione: attività che termina portando un oggetto in uno stato.	
Start & End	<div>   </div>
Stato iniziale: punto iniziale di uno state transition diagram. Stato finale: punto terminale di uno state transition diagram.	
Action	
Azione: attività che caratterizza uno stato.	
History	
Stato con memoria: stato capace di ripristinare situazioni precedenti.	