UML Notes

Paradigmi OOP

1. **Abstraction:** usare le classi per astrarre la natura delle caratteristiche di un oggetto, eche é un'istanza della propria classe di appartenenza.
2. **Encapsulation:** Nascondere i dettagli del funzionamento di un oggetto; gli oggetti hanno accesso solo ai dati che hanno bisogno.
3. **Inheritance:** classi possono specializzare altre classi ereditando da esse e implementando solo la porzione di comportamento che differisce.
4. **Polynorohism:** invocare comportamento diverso in reazione allo stesso messaggio, a seconda di quell'oggetto che lo riceve.

Diagrammie e viste

Un diagramma è la rappresentazione grafica di un modello e fornisce una vista di un sistema (o una sua parte) per metterne in risalto le sue proprietà.

Viste:

1. **Logical:** mette in risalto la scomposizione logica del sistema tramite classi, oggetti e loro relazioni.
2. **Development:** mostra l'organizzazione del sistema in blocchi strutturali (packages, sottosistemi, librerie, ...).
3. **Process:** mostra i processi (o thread) del sistema in funzione, e le loro interazioni.
4. **Physical:** mostra come il sistema viene installato se eseguito fisicamente.
5. **Use case:** spiega il funzionamento desiderato del sistema.

UML fornisce i diagrammi divisi in due categorie:

- **Structure diagrams:** come è fatto il sistema; fornisce le vieste *Logical*, *Development* e *Physical*.
- **Behaviot diagrams:** come funziona il sistema; fornisce le viste *Process* e *Use case*.

Structure	Behavior
Class diagram	Use Case diagram
Object diagram	Activity diagram
Package diagram	State Machine diagram
Composite Structure diagram	Sequence diagram
Component diagram	Communication diagram
Deployment diagram	Interaction Overview diagram
	Timing diagram