

UML

introduzione

UML - Introduzione

- ◆ Nel mondo odiernosviluppo di applicazioni Object Oriented.
- ◆ Esigenza di avere un **linguaggio universale** per modellare gli oggetti
- ◆ linguaggio **UML**, **Unified Modeling Language**.
- ◆ L'UML è un metodo per descrivere l'architettura di un sistema in dettaglio.

UML - Introduzione

- ◆ La forza dell'Unified Modeling Language consiste nel fatto che chiunque sia coinvolto nel sistema di sviluppo possa capire ed esaminare in modo efficiente il sistema e prendere parte alla sua costruzione in modo attivo.
- ◆ **esempio:** acquistare un appartamento 4 camere da letto, 3 bagni e di circa 180 metri quadri.
- ◆ In realtà vorremmo dire tutti i dettagli possibili (finestre, porte, ecc.) servendoci, a tale scopo, della mappa della casa.
 - Idem per la "macchina" dello sviluppo di sistemi software. Invece di leggere la piantina della casa, in questo caso ci si servirà dei diagrammi UML che non fanno altro che documentare in modo esauriente e a tutti i livelli le caratteristiche tecniche che dovranno essere implementate.

UML - Introduzione

I vantaggi del linguaggio UML

◆ Elenchiamo, qui di seguito, alcuni dei benefici derivanti dall'utilizzo del linguaggio UML:

📁 Un sistema software viene disegnato e documentato ancor prima che ne venga scritto il relativo codice.

📄 la scrittura del codice stessa è resa più agevole ed efficiente. I costi di sviluppo, dunque, si abbassano notevolmente .

📄 è più facile prevedere e anticipare eventuali "buchi" nel sistema.

📄 Avere una chiara idea, a chiunque sia coinvolto nello sviluppo, di tutto l'insieme che costituisce il sistema.

📄 Più facile effettuare eventuali modifiche future al codice.

UML - Introduzione

Componenti UML

- ◆ Svariati elementi grafici per la creazione dei diagrammi. L'UML utilizza delle regole per combinare i componenti nella creazione dei diagrammi.
- ◆ L'obiettivo dei diagrammi è quello di costruire molteplici viste di un sistema tutte correlate tra di loro.
- ◆ Il linguaggio UML consiste di nove diagrammi di base, ma è possibile costruire e aggiungere diagrammi differenti.

UML - Introduzione

Class Diagram

- ◆ **Definizione:** Una classe è una categoria o un gruppo di oggetti (con questo termine includiamo, per comodità anche gli esseri viventi) che hanno attributi simili e comportamenti analoghi. Esempio: computers, automobili, piante, animali, ecc.
- ◆ I Class Diagrams forniscono le rappresentazioni utilizzate dagli sviluppatori.

UML - Introduzione

Object Diagram

Un oggetto è una istanza di una classe - ovvero un qualcosa di specifico che ha dei valori determinati per i suoi attributi e dei comportamenti specifici.

UML - Introduzione

Use Case Diagram

Uno Use Case (caso d'uso) è una descrizione di un comportamento particolare di un sistema dal punto di vista dell'utente. Requisiti del sistema dal punto di vista utente. Simbolo particolare per l'actor (l'utente o un altro sistema che interagisce).

- ◆ **Definizione:** L'actor è l'entità che interagisce con uno use case facendo partire la sequenza di azioni descritte dallo use case stesso.

UML - Introduzione

State Diagram

Gli State Diagrams rappresentano gli stati degli oggetti ad un determinato istante, durante il funzionamento del sistema, ed i loro cambiamenti nel tempo.

Stato iniziale (Start State) e stato finale (End State).

Per esempio, ogni persona può essere identificato dai seguenti stati: neonato, infante, bambino, adolescente, adulto, anziano.

UML - Introduzione

Sequence Diagram

I class diagrams e gli object diagrams rappresentano informazione statica.

Il sequence diagram mostra le dinamiche, basate sul tempo, delle varie interazioni tra gli oggetti.

UML - Introduzione

Activity Diagram

Le attività che si riscontrano all'interno di use case o all'interno del comportamento di un oggetto accadono, tipicamente, in una sequenza ben definita. Tale sequenza viene rappresentata con gli activity diagrams.

UML - Introduzione

Collaboration Diagram

Gli elementi di un sistema lavorano insieme per realizzare e soddisfare le necessità del sistema. Un linguaggio di modellazione deve avere un modo per rappresentare tale cooperazione.

UML - Introduzione

Component Diagram

Ognuno nel team di lavoro si occupa di lavorare su un componente differente. Il component diagram descrive questa importante caratteristica.

UML - Introduzione

Deployment Diagram

Il Deployment Diagram mostra l'architettura dal punto di vista fisico e logistico di un sistema.

i computer e i vari dispositivi presenti,
le connessioni che intercorrono tra di essi,
software installato su ogni macchina.

Domanda

- ◆ Ma è proprio necessario sviluppare tutti e nove i diagrammi che l'UML mette a disposizione?
- La risposta a questa domanda può variare in relazione alla complessità del Sistema che si intende costruire