Introduzione alla OOP (Object Oriented Programming)

- È una nuova metodologia per lo sviluppo di applicazioni software

- La principale innovazione di questa metodologia sta nel superamento della tradizionale separazione tra dati e codice

- Questa metodologia si basa sulla descrizione delle caratteristiche intrinseche, dello Stato e del comportamento di un determinato oggetto
  Questi parametri servono per definire il modello dell'oggetto che stiamo descrivendo

- Per definire un modello dobbiamo acquisire due abilità:
  Saper effettuare una astrazione sulle caratteristiche traducendole in dati (attributi)
  Saper effettuare una astrazione di tipo funzionale per individuare le azioni da compiere (operazioni)

- Queste due abilità ci permetteranno di:
  progettare un oggetto (Object Oriented Design: OOD)
  programmare un oggetto (Object Oriented Programming: OOP)

- Una delle peculiarità della OOP e la possibilità di definire nuovi tipi di dati; quando noi pensiamo ai tipi di dati, istintivamente pensiamo ai valori che un particolare tipo di dato può assumere e alle operazioni che possiamo fare con quel dato; esempi:

  - Con i numeri reali alle operazioni di somma, sottrazione, divisione e moltiplicazione... ma anche al calcolo della parte intera i simboli che usiamo sono le cifre arabe

  - Con i numeri interi oltre alle operazioni aritmetiche, abbiamo l' operazione di modulo, che non è prevista i numeri reali (cifre arabe)

  - Con le stringhe pensiamo alle operazioni di concatenamento, estrazione di un singolo carattere o di sottostringa oppure il conteggio del numero dei caratteri o delle occorrenze di una sequenza (caratteri)

Questi esempi ci fanno capire che quando progettiamo un tipo di dato dobbiamo anche prevedere le operazioni che possiamo fare con esso

Generalizzazione

Un tipo di dato è definito quando riusciamo ad individuare un insieme di valori ammissibili e un insieme di operazioni che possono essere applicate a quel tipo di dato.

Consideriamo il tipo data e ci poniamo le seguenti domande:

- Ha senso fare la somma di due date? No
- Ha senso fare la differenza fra due date? La risposta è sì e il risultato è il numero di giorni che intercorrono fra le due date
- Ha senso sottrarre da una data un intero? La risposta è sì e il risultato è una data anteriore a quella iniziale
- Ha senso aggiungere ad una data un numero intero la risposta è ancora Sì è il risultato è una data posteriore a quella iniziale
- Le operazioni di moltiplicazione e di divisione per esempio fra le date non hanno senso.
- Altre operazioni che hanno senso con una data sono l'estrazione del giorno dalla data, l'estrazione del mese oppure l'estrazione dell'anno

Definizione

Un tipo di dato astratto (ADT Abstract Data Type) è ben specificato se per esso sono definiti:

- I dati (attributi)
- Le operazioni che si possono effettuare su quel dato (metodi)

Nella programmazione OOP per interagire con i dati, si usano esclusivamente i metodi previsti dal progettista dell'oggetto e esposti al programmatore

- Una caratteristica fondamentale della OOP é il principio di Information Hiding (nascondere le informazioni) detto anche incapsulamento che stabilisce la netta separazione fra la struttura interna di un oggetto e la sua interfaccia che saranno definiti in fase di progettazione

- Questa metodologia offre diversi vantaggi:
  - un ADT ben progettato può essere riutilizzato come componente in più progetti
  - una eventuale modifica da parte del progettista alla struttura interna dell'ADT rimane invisibile al programmatore
  - poiché il programmatore potrà interagire con l'ADT solo attraverso i metodi esposti, sarà garantita la integrità della struttura interna

Ricapitolando possiamo dire che un progettista deve:

- definire gli attributi e i metodi di un ADT
- decidere quali attributi e quali metodi devono essere resi pubblici (cioè messi a disposizione di un eventuale utilizzatore dell'ADT) e quali invece devono rimanere privati, cioè invisibili ad utilizzatore