

## Programmazione orientata agli oggetti (OOP)

- Linguaggio C
  - Paradigma **imperativo** (procedurale) basato sul concetto di *divide et impera*
- Linguaggio Java o C++
  - Programmazione orientata agli **oggetti**
  - Permettono anche la programmazione di tipo **imperativo**

## Che cos'è un oggetto ?

- Definizione da vocabolario:

*“Un oggetto è un'entità che si possa immaginare dotata di determinate **caratteristiche** e **funzionalità**”.*

- Il confinamento di *informazioni* e *funzionalità* in oggetti permette livelli maggiori di **astrazione** e semplifica la gestione di sistemi complessi

## Gli oggetti come modello naturale

- Gli oggetti nel software possono rappresentare entità del mondo reale
- Il primo uso di linguaggi a oggetti è stato fatto per effettuare simulazioni

Interazioni tra diversi componenti

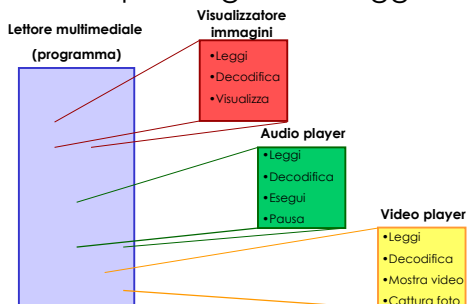


scambio di messaggi

## Simulazione: un videogioco



## Esempio di un programma scritto secondo il paradigma ad oggetti



## Caratteristiche di un oggetto

- Un oggetto possiede:
  - Attributi
  - Dati interni
  - Capacità' di eseguire azioni
  - Capacità' di interagire con altri oggetti attraverso l'invio di messaggi (chiamate a metodi)

## L'oggetto *audio player*

- **Attributi:**
  - ✓ Interfaccia grafica
  - ✓ Bottoni di controllo
- **Dati Interni:**
  - ✓ File audio da riprodurre
- **Azioni:**
  - ✓ Leggi il file da disco
  - ✓ Decodifica il file con il decoder mp3
  - ✓ Riproduci il brano
  - ✓ Interrompi la riproduzione

## L'oggetto *Letto re Multimediale*

- **Attributi:**
  - ✓ Interfaccia grafica
- **Dati Interni:**
  - ✓ Visualizzatore di immagini
  - ✓ Audio player
  - ✓ Video Player
- **Azioni:**
  - ✓ Crea nuovo Visualizzatore imm.
  - ✓ Crea nuovo audio player
  - ✓ Crea nuovo video player
  - ✓ Audio player -> Leggi il file da disco
  - ✓ Audio player -> Riproduci il brano
  - ✓ Audio player -> Interrompi

## Il concetto di *classe*

- Modello o schema dettagliato dal quale creare un oggetto
- Una classe e' una sorta di stampo per i dolci (ne definisce le caratteristiche)
- L'oggetto e' il dolce che viene prodotto con lo stampo (classe)
- Quando viene creato un oggetto da una classe si dice che e' stata creata una *istanza della classe*
- Diversi oggetti creati dalla stessa classe hanno tutti lo stesso comportamento

## Oggetti e Classi di Oggetti



## Programmare con gli oggetti

- Il programmatore ha il compito di scrivere il codice che definisce le classi da cui verranno creati gli oggetti che costituiranno il programma
- Il programmatore deve definire le relazioni tra le varie classi che costituiscono il programma

## Definire una classe

- Quale deve essere il comportamento della classe?
  - Cosa deve fare?
  - Con quali altre classi deve dialogare?
- Quale deve essere lo stato interno della classe?
  - Quali dati deve contenere?
  - Quali attributi deve contenere?

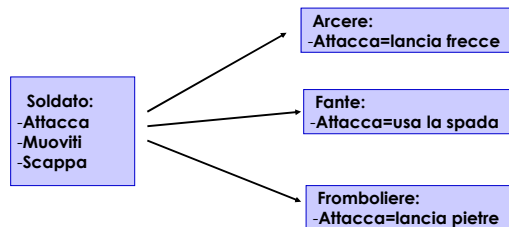
## Relazioni tra le classi

- *Uso*: E' una relazione ovvia e generale, nell'esempio del lettore multimediale, la classe *lettore multimediale* usa le classi *audio player*, *video player*, ecc)
- *Contenimento*: e' un caso particolare di uso; la classe *lettore multimediale* oltre ad usare *audio player* la contiene anche

## Ereditarietà

- Denota specializzazione, ad esempio una classe *Arciere* sarebbe un *Soldato* specializzato nell'attaccare il nemico scagliando delle frecce. Questa nuova classe **erediterebbe** le caratteristiche di *Soldato* e in più ne aggiungerebbe di nuove (attaccare con le frecce)

## Es. Ereditarietà



## Caratteristiche di un linguaggio a oggetti

- *Incapsulamento*: di una classe deve essere visibile solamente la sua interfaccia (gli attributi e le azioni utilizzabili dagli altri; es. riproduci filmato)
- *Polimorfismo*: e' la classe stessa a preoccuparsi di utilizzare il giusto metodo per effettuare un'azione

## Es. polimorfismo

- La classe *Soldato* ha la funzionalità *attacca*.
- I diversi tipi di soldati implementano la funzionalità *attacca* in modo diverso provocando conseguenze diverse:
  - un arciere scaglia una freccia
  - un fante colpisce di spada
  - un cavaliere lancia una lancia
- Il giocatore/computer chiedere ad ogni soldato di *attaccare* indipendentemente dal tipo di soldato
- Il programma stesso sceglie per ogni tipo di soldato qual'e' l'azione da compiere in seguito ad un comando *attacca*.

## Perche' programmare ad oggetti ?

- Programmare a oggetti...
  - ✓ non velocizza l'esecuzione dei programmi...
  - ✓ non ottimizza l'uso della memoria...

E allora perché programmare a oggetti?
- ✓ Programmare a oggetti facilita la progettazione, lo sviluppo e il mantenimento di sistemi software molto complessi!

## Vantaggi della programmazione ad oggetti

- La programmazione ad oggetti consente di:
  - ✓ Ridurre la **dipendenza** del codice di alto livello dalla rappresentazione dei dati
    - l'accesso ai dati è mediato da un'**interfaccia**
  - ✓ **Riutilizzare** del codice di alto livello
    - definendo opportune **classi astratte**
  - ✓ Sviluppare moduli indipendenti l'uno dall'altro

## Difetti del software non mantenibile

- **Rigidità**
  - ✓ Non può essere cambiato con facilità
  - ✓ Non può essere stimato l'impatto di una modifica
- **Fragilità**
  - ✓ Una modifica singola causa una cascata di modifiche successive
  - ✓ I banchi sorgono in aree concettualmente separate dalle aree dove sono avvenute le modifiche
- **Non riusabilità**
  - ✓ Esistono molte interdipendenze, quindi non è possibile estrarre parti che potrebbero essere