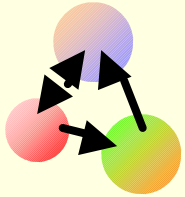
Progettazione orientata agli oggetti

Introduzione alle tecniche orientate agli oggetti

Modelli ad oggetti

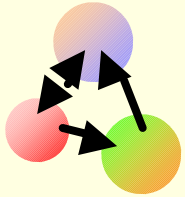
Oggetti, classi, associazioni, aggregazione

OO - Introduzione



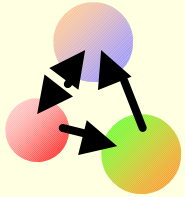
- Il compito del programmatore: collegare lo spazio del problema (problem space) con quello della soluzione (solution space)
- L'approccio ad oggetti: descrizione in termini dello spazio del problema non in quello della macchina

OO - Panoramica



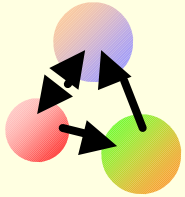
- Un sistema software viene realizzato mediante un insieme di oggetti che cooperano alla soluzione
- Gli oggetti comunicano tramite messaggi (una metafora che evidenzia la separazione tra gli oggetti)
- Gli oggetti sono descritti da classi, che ne definiscono le caratteristiche (metodi e proprietà)
- Le classi sono collegate tramite relazioni di varie tipologie
- Le relazioni tra le classi valgono anche per gli oggetti da esse definiti

Oggetti



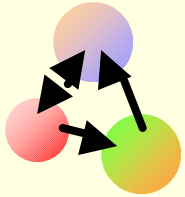
- Un sistema complesso viene suddiviso in oggetti collegati tra loro e cooperanti
- Gli oggetti sono scatole nere completamente blindate: l'unica parte visibile di un oggetto è la sua "superficie" (interfaccia) definita sulla base dei messaggi che scambia con gli altri oggetti
- Corrispondono agli "attori", alle "entità" fondamentali di un sistema o una situazione (termini definitori presi nel **problem space**)

Classi



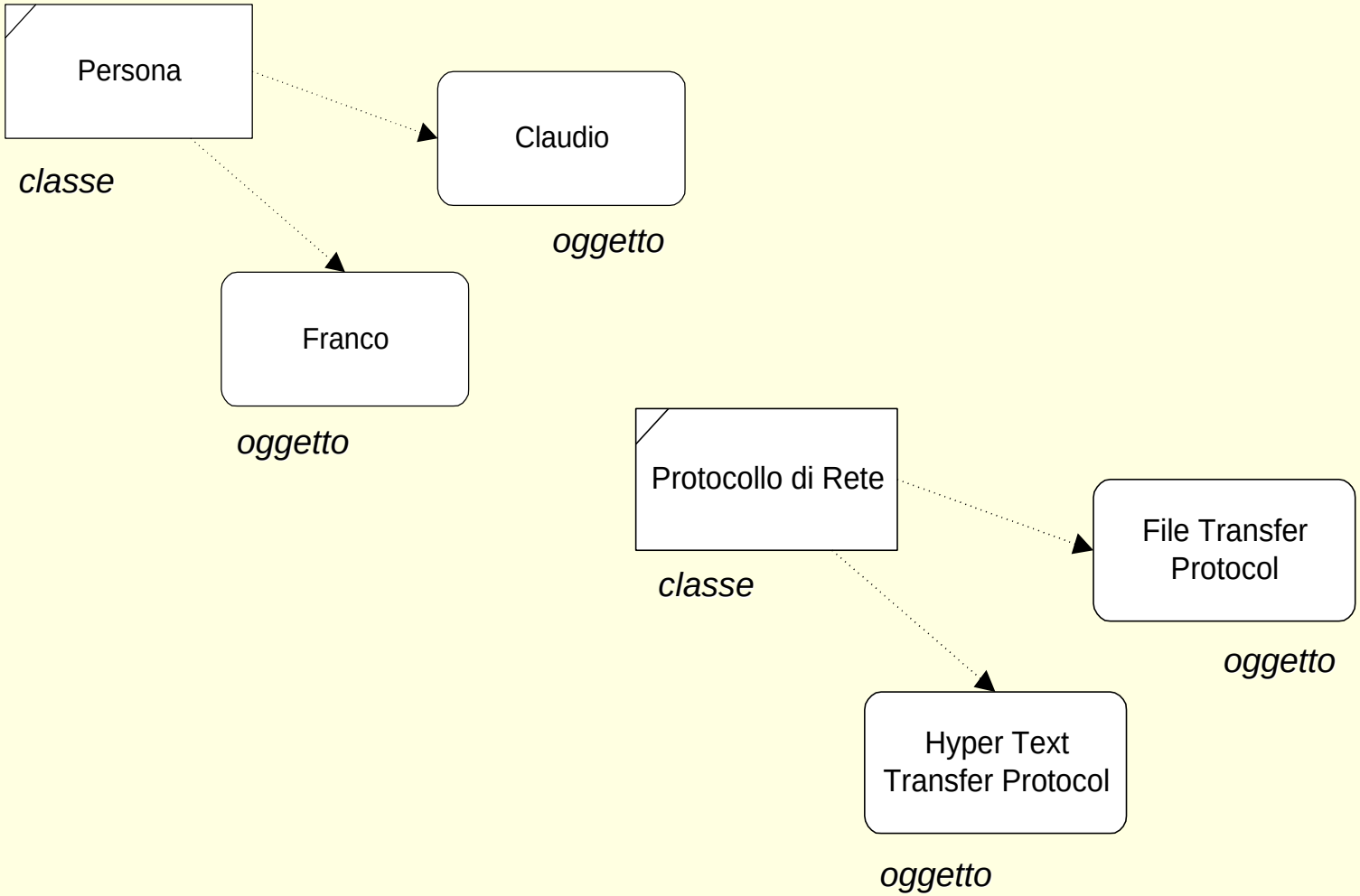
- Ogni oggetto deriva da una classe (si dice che è l'istanza di una classe)
- La classe descrive caratteristiche e comportamenti di un insieme di oggetti
- Le classi sono organizzate in gerarchie e sono in relazione tra loro

Classi: fabbriche di oggetti

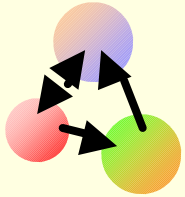


- Le classi sono “fabbriche” di oggetti
- Ogni volta che serve un oggetto, si usa la classe come uno “stampo” per generare un nuovo oggetto avente le proprietà definite dalla classe (si istanzia la classe nell’oggetto)
- Cfr. con tipi di dato e variabili (classe come tipo, oggetto come variabile)

Esempi


Persona
classe
Claudio
oggetto
Franco
oggetto
Protocollo di Rete
classe
File Transfer Protocol
oggetto
Hyper Text Transfer Protocol
oggetto

Modularità

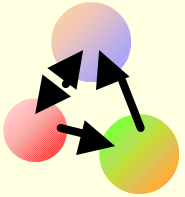


- E’ molto difficile descrivere un sistema complesso (controllo di reti idriche, rete di comunicazione, base di dati geografici ...) come una unica entità.

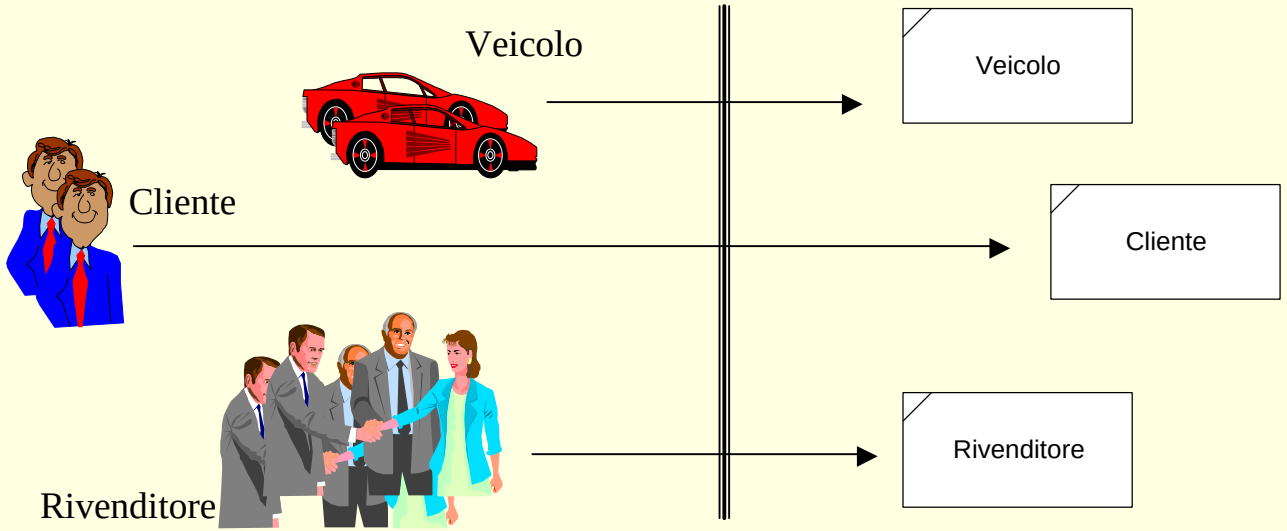
- Oggetti e classi permettono di decomporre il sistema in unità più piccole aventi comportamento e caratteristiche più semplici da comprendere

... divide et impera ...

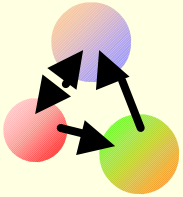
Minima distanza semantica



Le entità con cui abbiamo a che fare in un sistema OO sono una traduzione quasi diretta degli “attori” che compaiono nel problema: la distanza semantica tra modello e originale è molto piccola

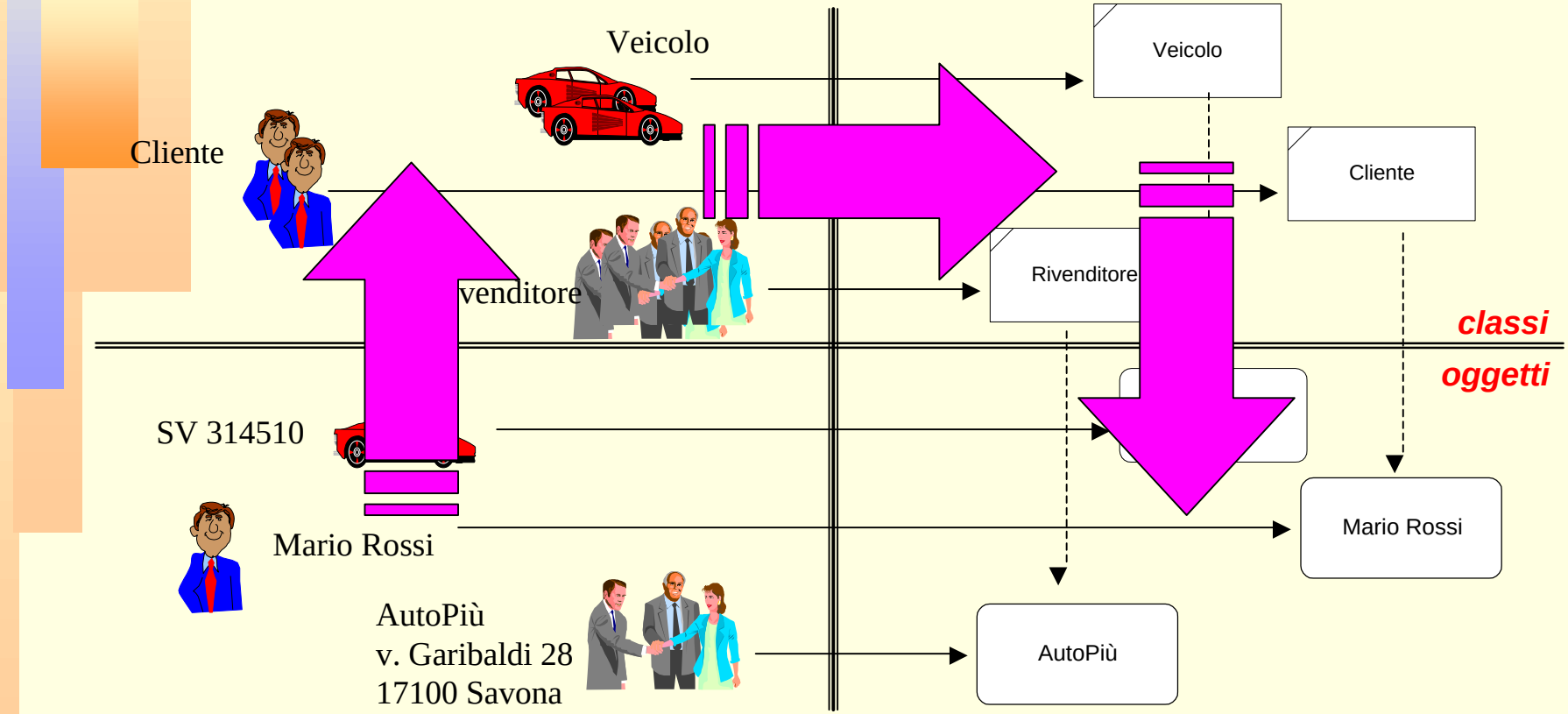


Diagrammi

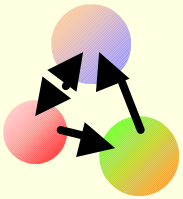


- **Diagrammi delle classi**
  - struttura logica del sistema
  - componenti e relazioni tra essi
  - descrizioni generiche di sistemi (casi generali)
- **Diagrammi degli oggetti**
  - particolari istanze di sistemi (casi particolari)

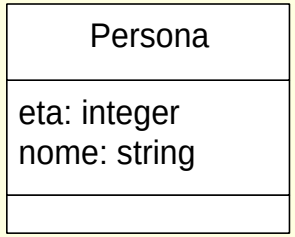
Esempio

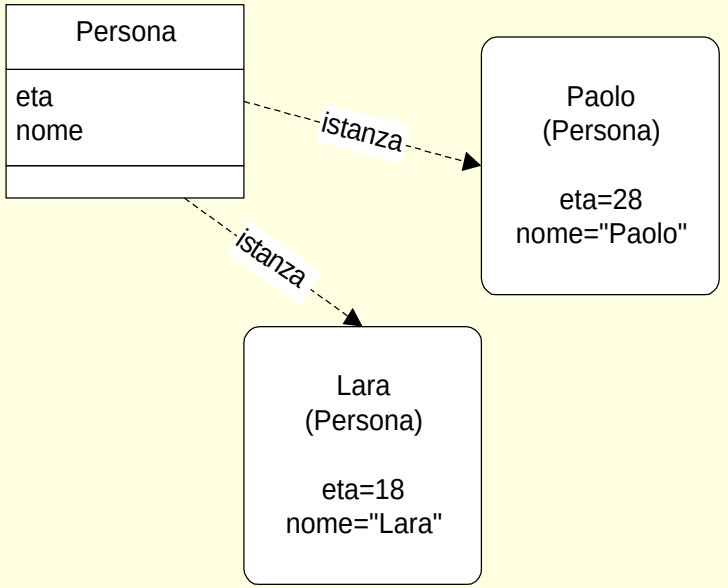

Veicolo
Cliente
venditore
SV 314510
Mario Rossi
AutoPiù
v. Garibaldi 28
17100 Savona
Veicolo
Cliente
Rivenditore
classi
oggetti
Mario Rossi
AutoPiù

Attributi

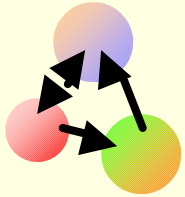


- Gli attributi modellano le caratteristiche fondamentali degli oggetti o delle classi (cfr. variabili)
- Ad ogni attributo in una classe corrisponde un valore in un oggetto



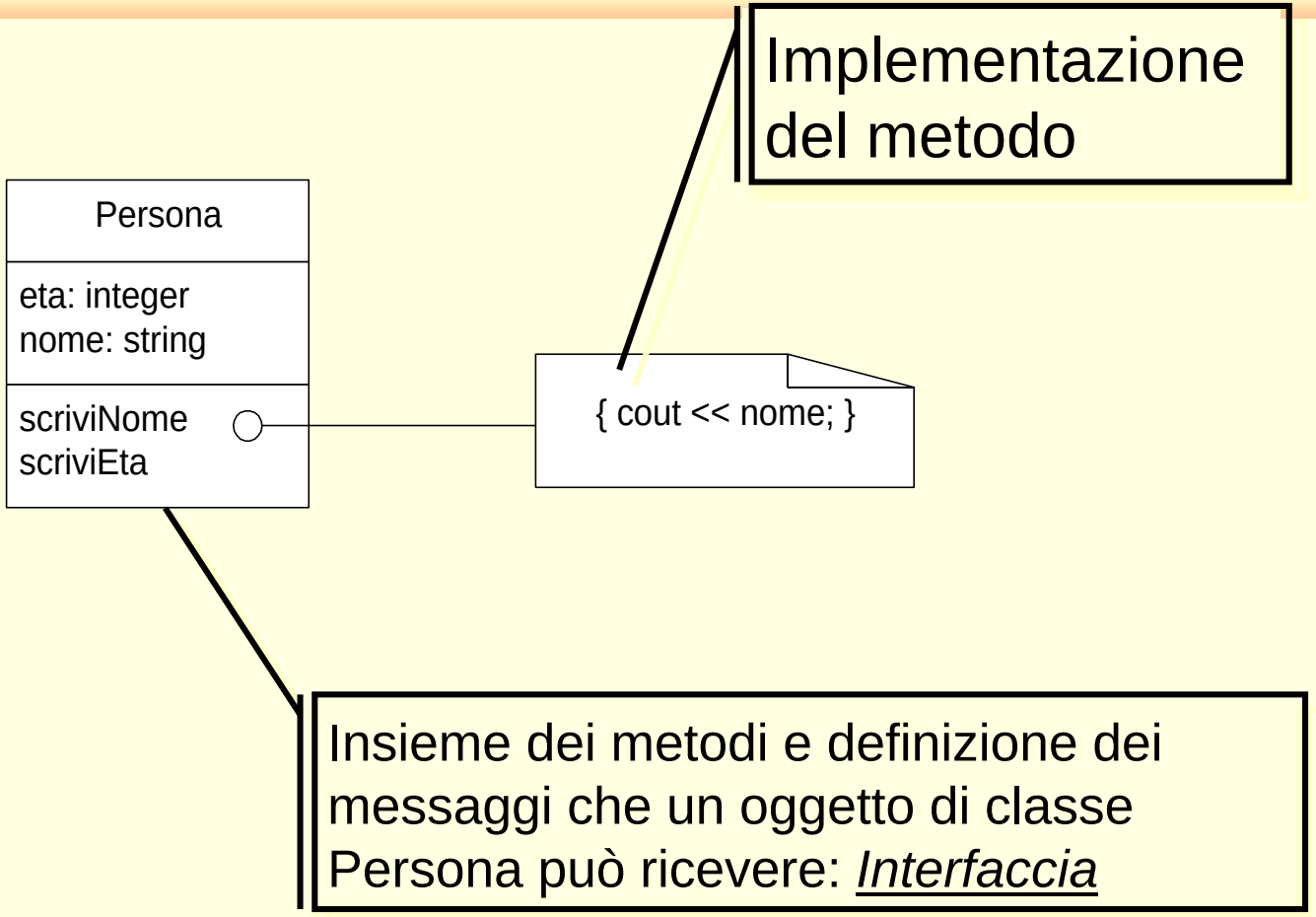


Messaggi e metodi

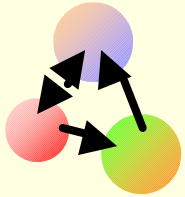


- Gli oggetti collaborano tra loro scambiandosi messaggi - qualunque scambio di informazione in un sistema ad oggetti è modellato in questo modo
- Ogni oggetto reagisce alla ricezione di un messaggio in un determinato modo, chiamato metodo (codice in C++)
- I metodi sono uguali per tutti gli oggetti di una stessa classe - il comportamento di un oggetto dipende dalla sua classe

Messaggi e metodi

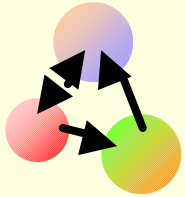


Incapsulamento



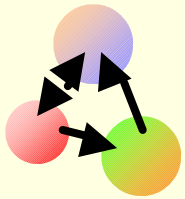
- Incapsulamento significa occultamento della informazione: ogni componente di un sistema deve conoscere del resto del sistema solo ciò che gli è indispensabile
- L'interfaccia di ciascun componente (l'insieme dei suoi metodi, a rigore) deve essere tale da rivelare il meno possibile della sua struttura interna (importanza delle interfacce)
- Progettare a compartimenti stagni significa facilitare il riuso

Associazioni

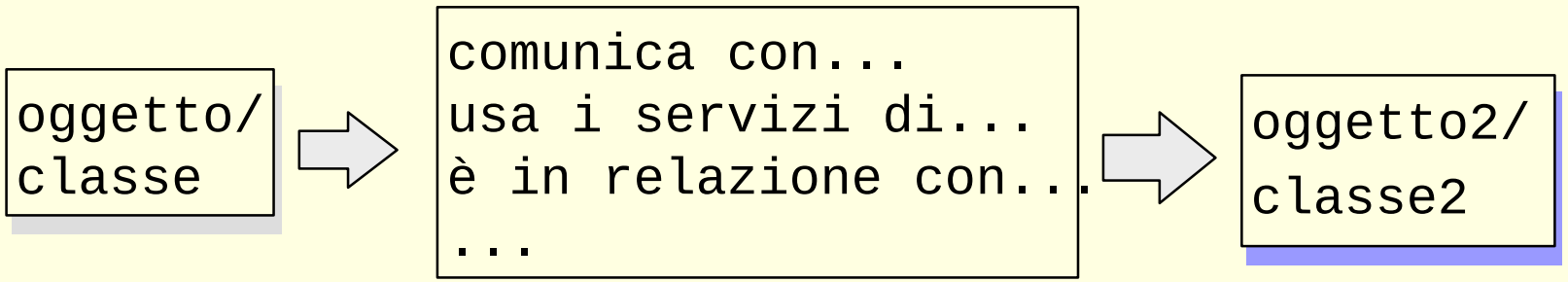


- Sono connessioni concettuali tra oggetti o classi
- Le associazioni sono generalmente bidirezionali
- Le associazioni in C++ vengono implementate con puntatori: per questo spesso è utile definire un verso di navigazione preferenziale della relazione

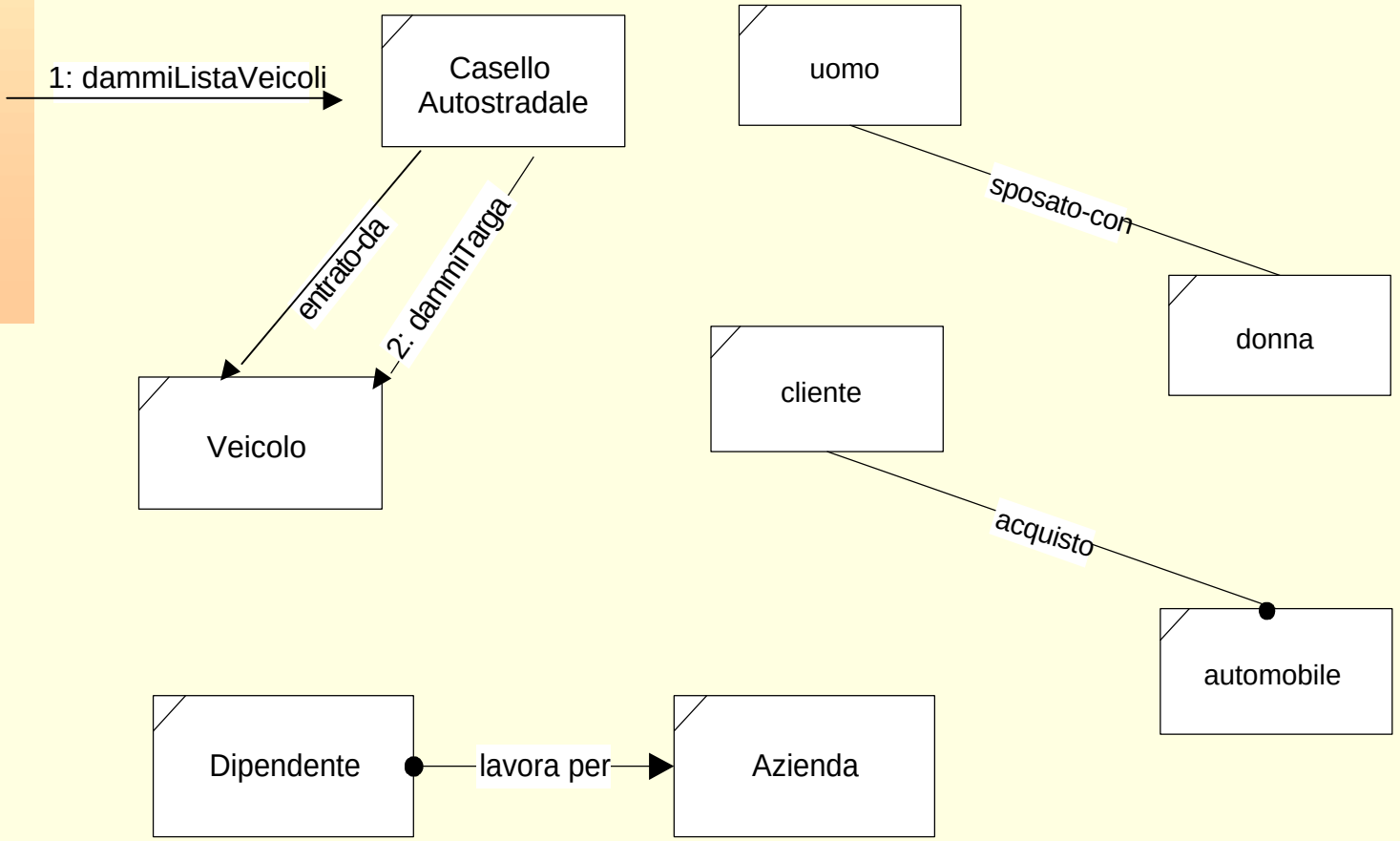
Associazioni



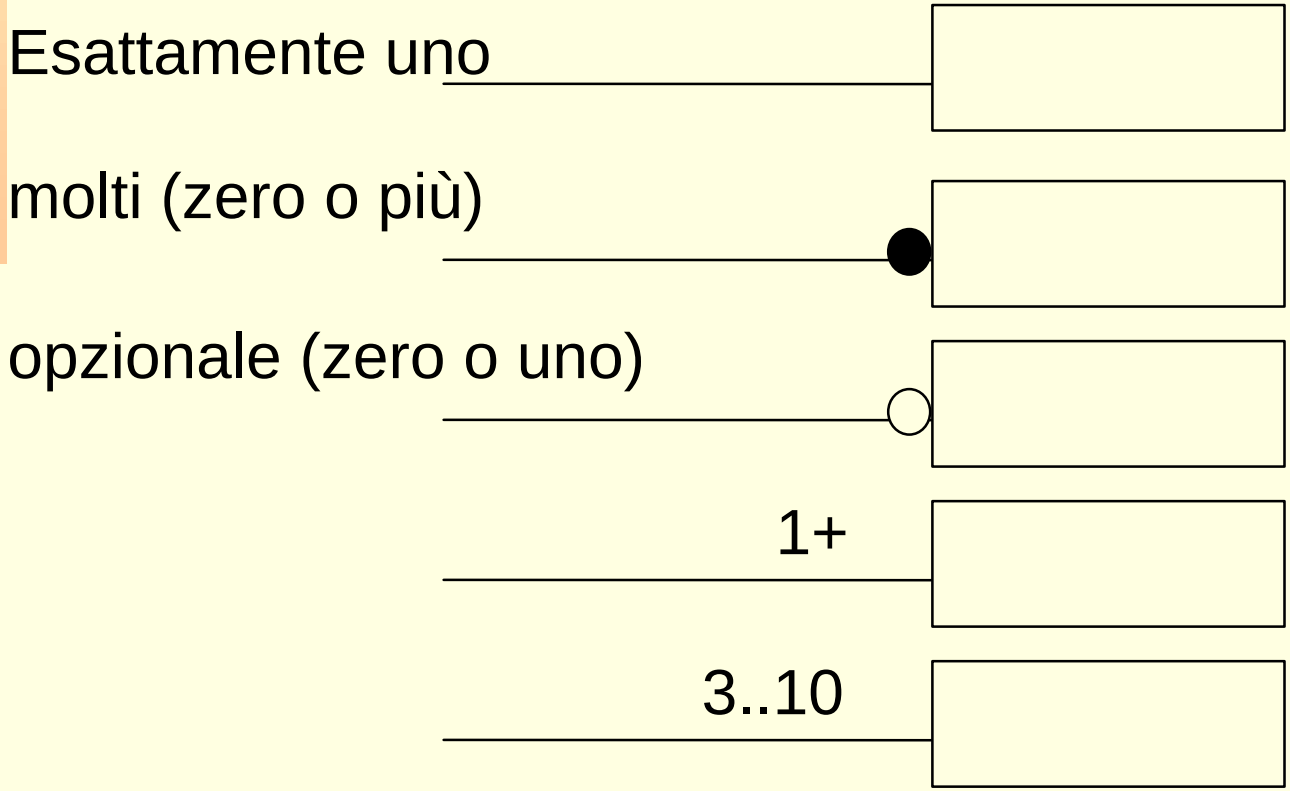
- La relazione di associazione implica una interazione tra oggetti o classi: due oggetti o classi in associazione di solito interagiscono attraverso l'invio di messaggi



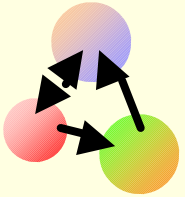
Associazioni



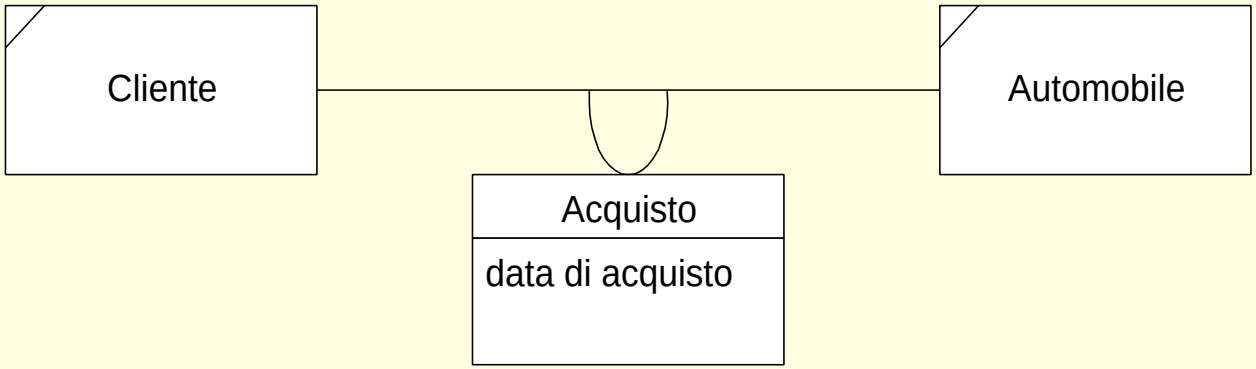
Molteplicità



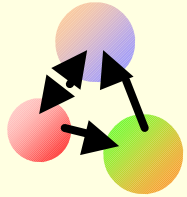
Associazioni e attributi

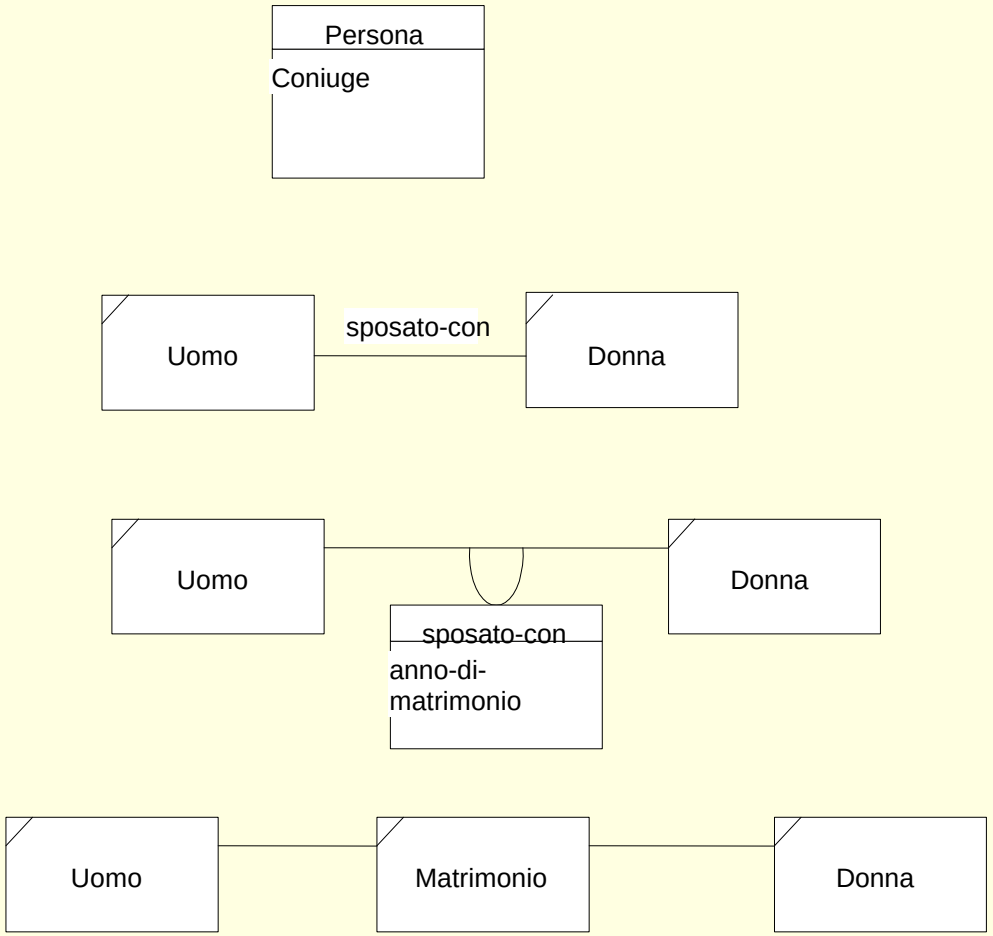


- Una associazione può essere caratterizzata da attributi

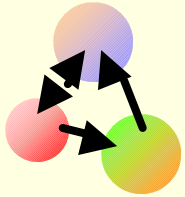


Associazione, attributo o classe?



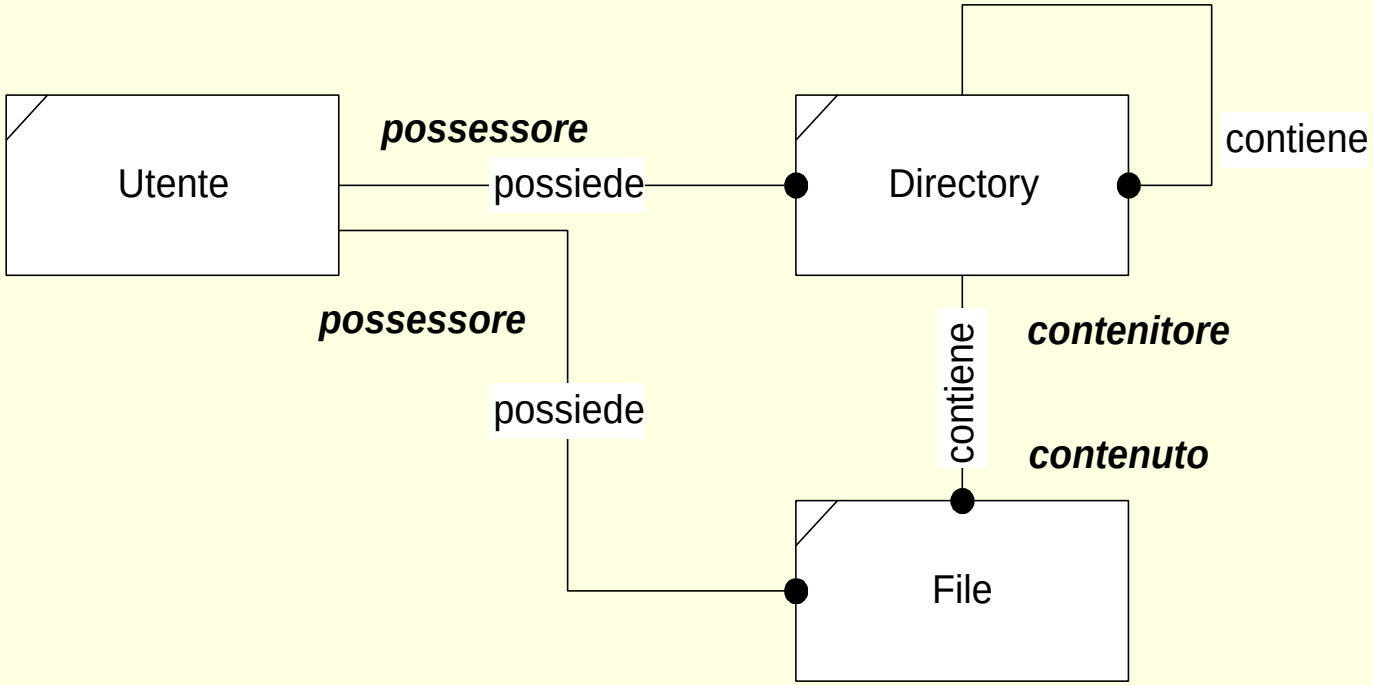


Ruoli

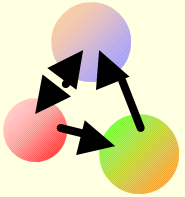


- Un ruolo è un nome che identifica univocamente un estremo della associazione
- Un ruolo viene utilizzato per attraversare l'associazione senza menzionarla eliminando eventuali ambiguità di interpretazione
- Un ruolo spesso è rappresentato con un sostantivo e una associazione con un verbo

Ruoli

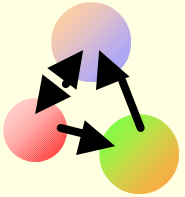

Utente
possessore
possiede
Directory
contiene
possessore
possiede
contiene
contenitore
contenuto
File

Aggregazione

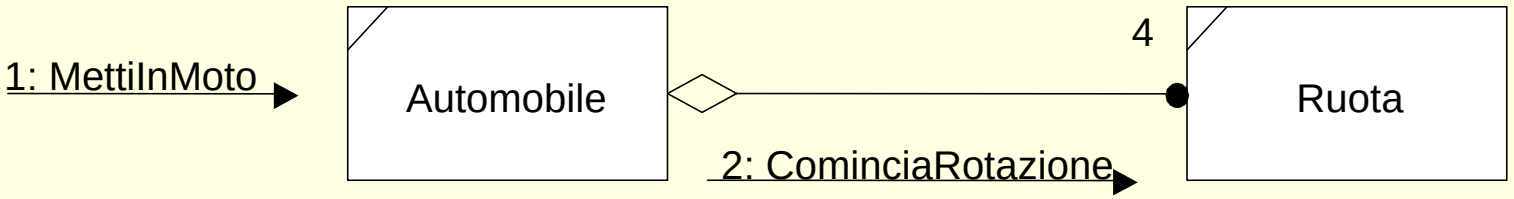


- E’ una associazione che indica “contiene”, “è costituito da”...

Aggregazione

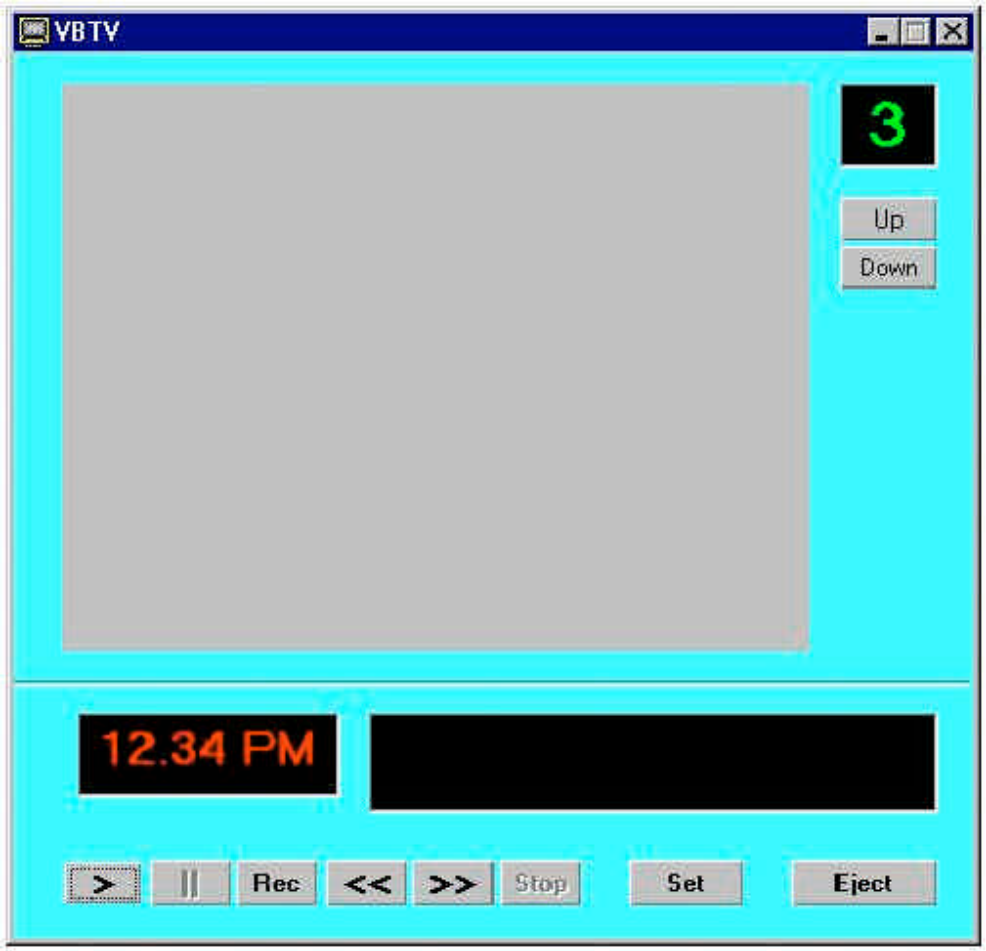


- Gli oggetti componenti diventano proprietà esclusiva dell'oggetto composto che diventa così la loro interfaccia per i messaggi ricevuti da altri oggetti

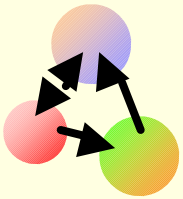


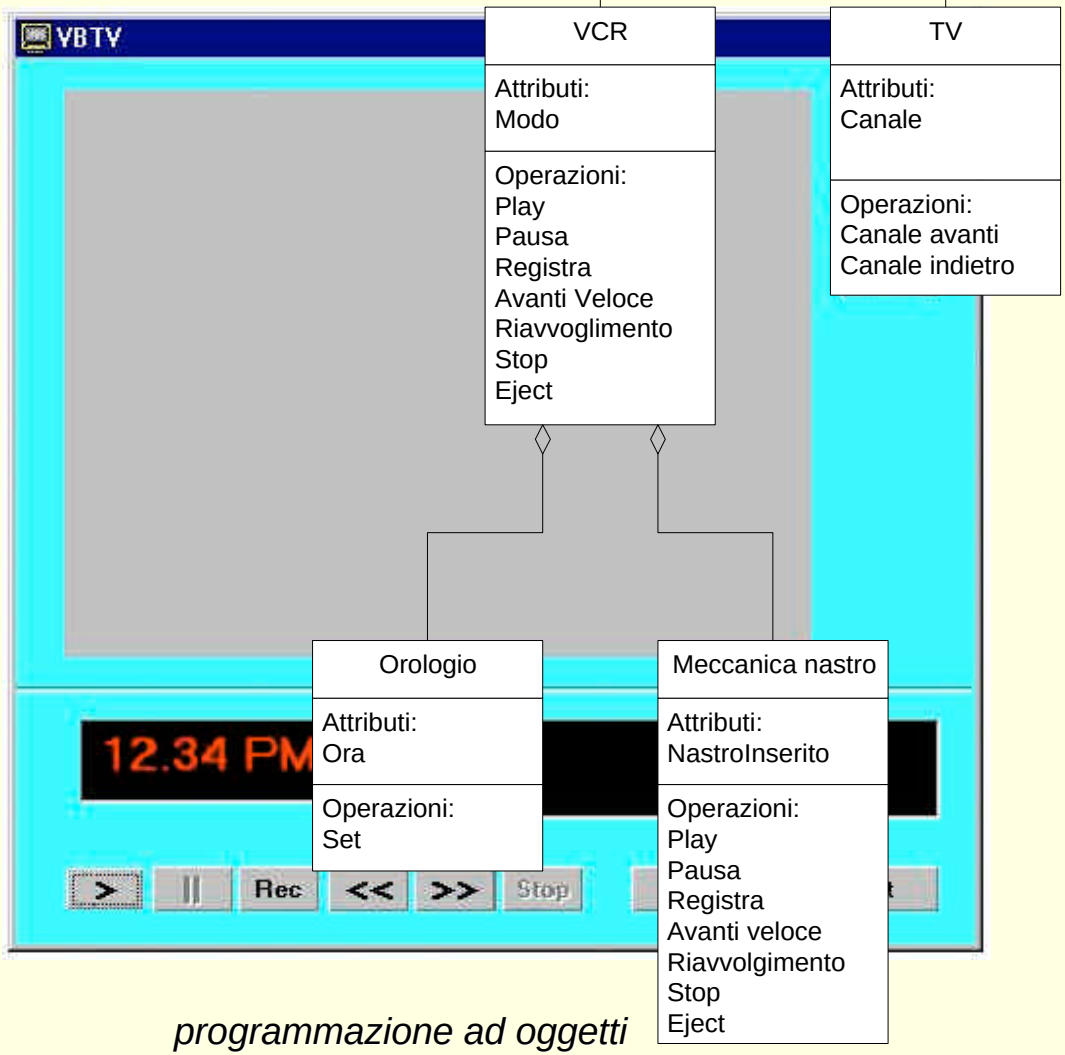
"Propagazione del messaggio"

Modello ad Oggetti di VCR - TV

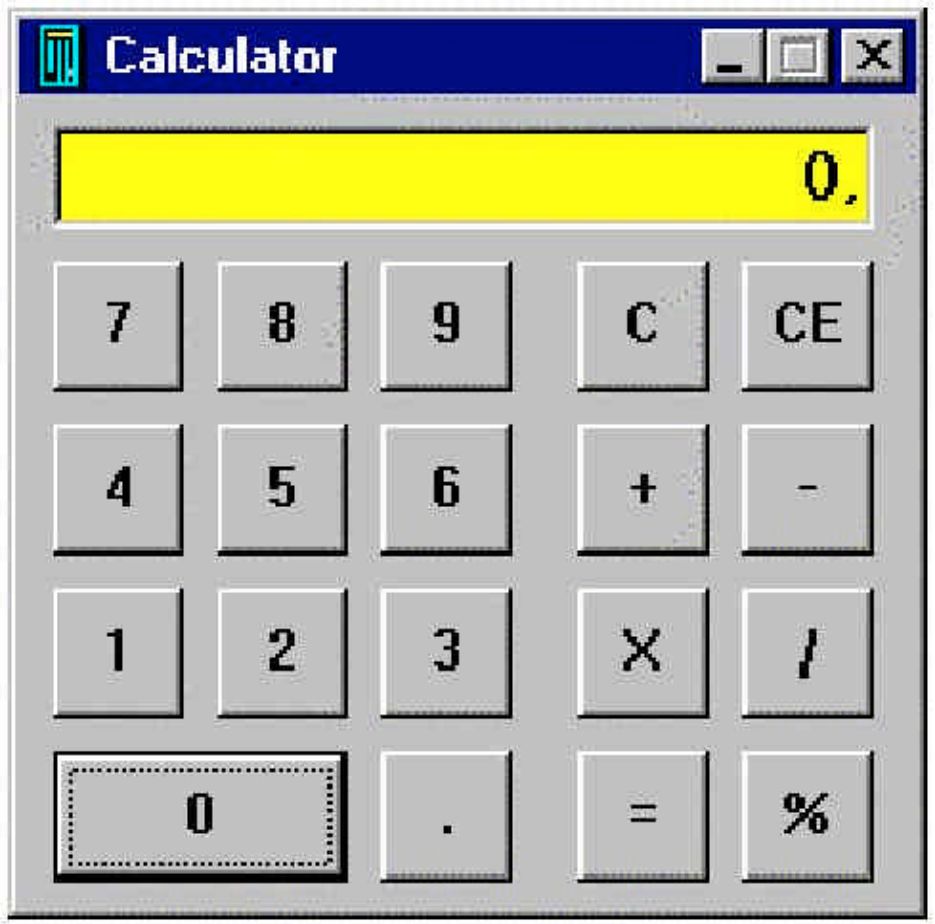


Modello ad Oggetti di VCR - TV

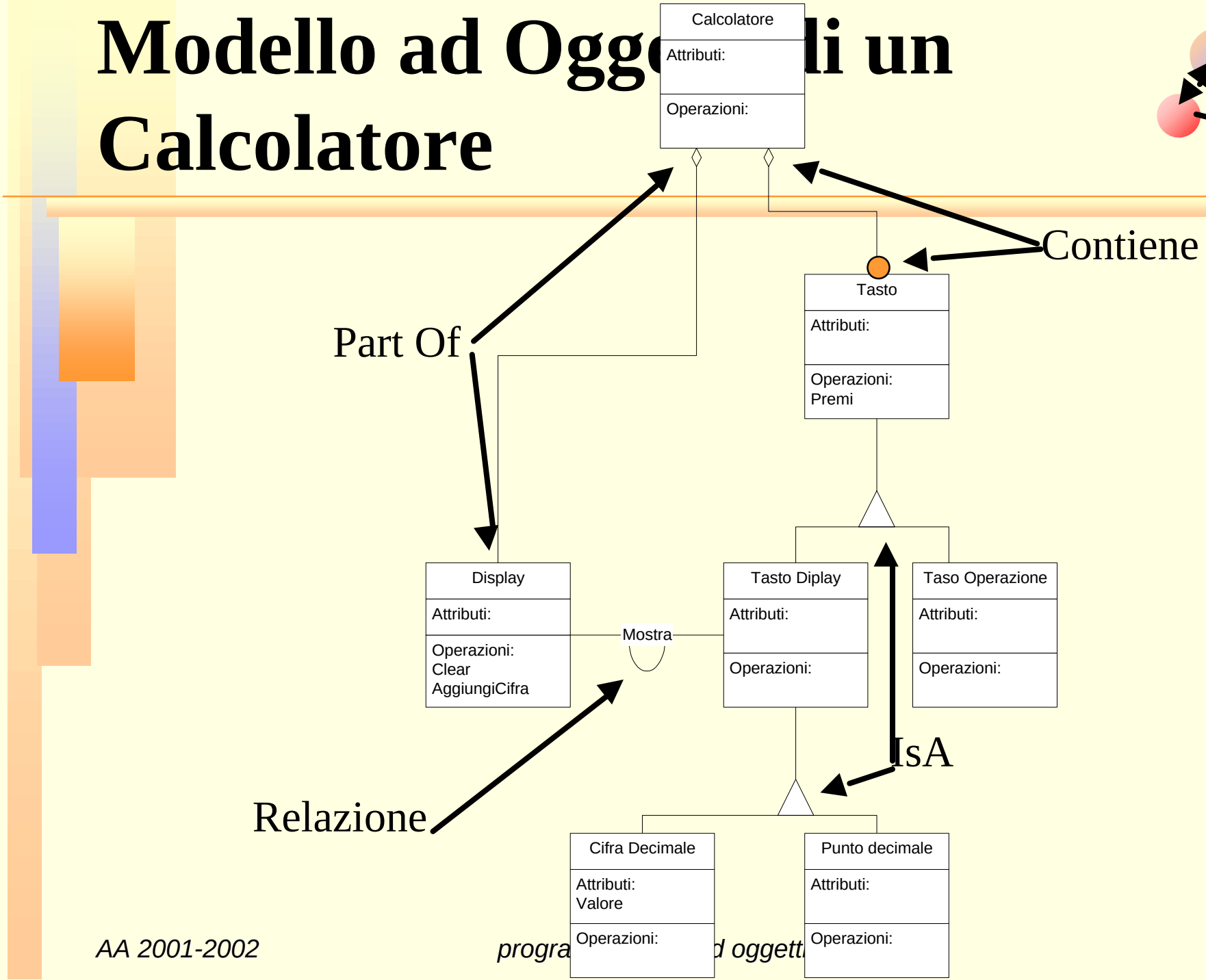




Modello ad Oggetti di un Calcolatore

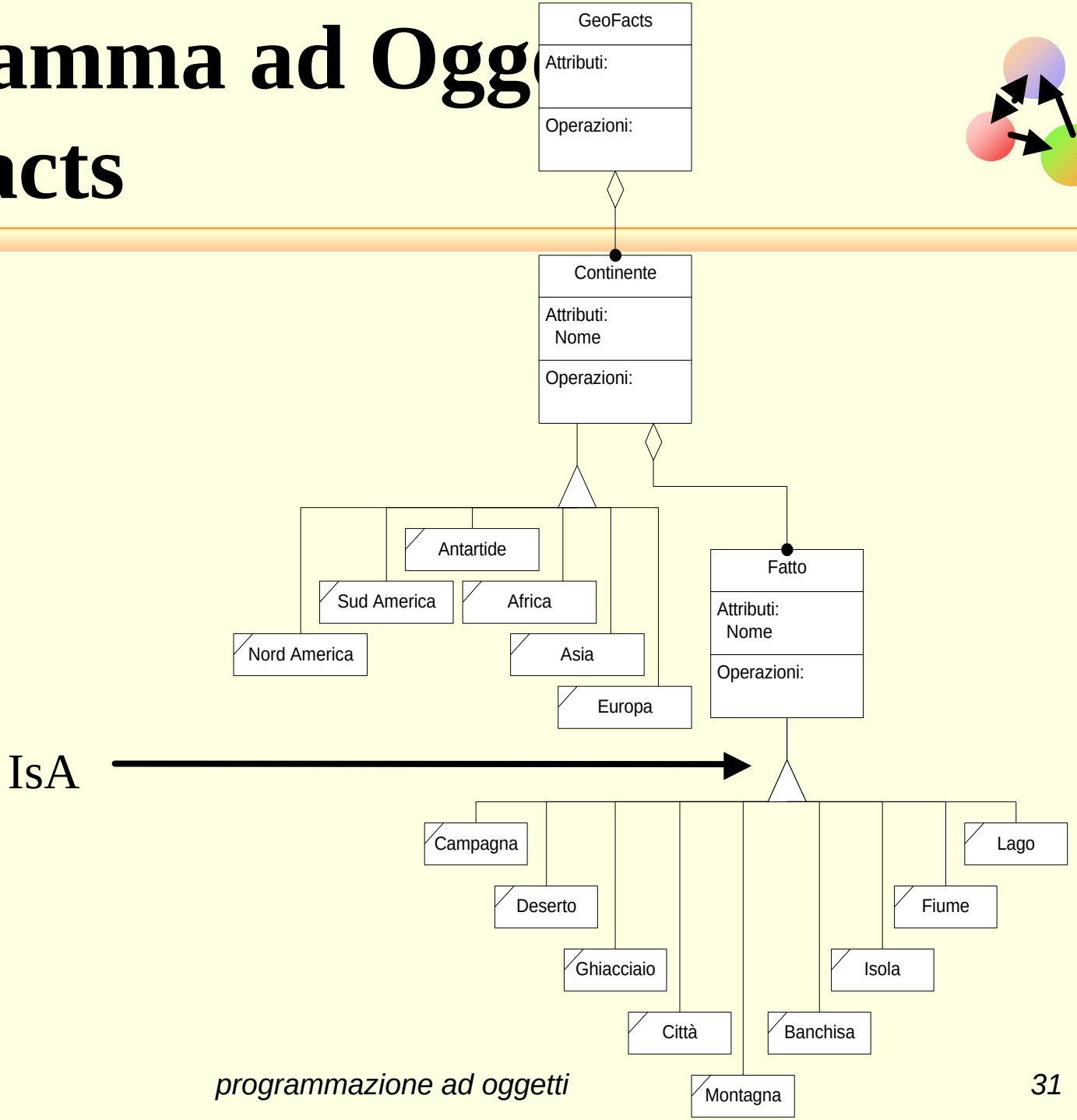


Modello ad Oggetti di un Calcolatore

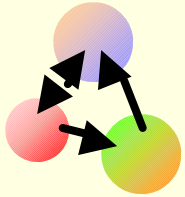


GeoFacts -- Un piccolo database turistico

Diagramma ad Ogg
GeoFacts



Esercizi



- Costruire un modello ad oggetti che descriva il funzionamento di uno dei programmi grafici che state utilizzando (es. finestra di login…)

- Costruire un modello ad oggetti che descriva le relazioni tra utenti, processi, directory, file in un sistema operativo tipo Unix