Programmazione Orientata agli Oggetti

1 – La tecnologia orientata agli oggetti

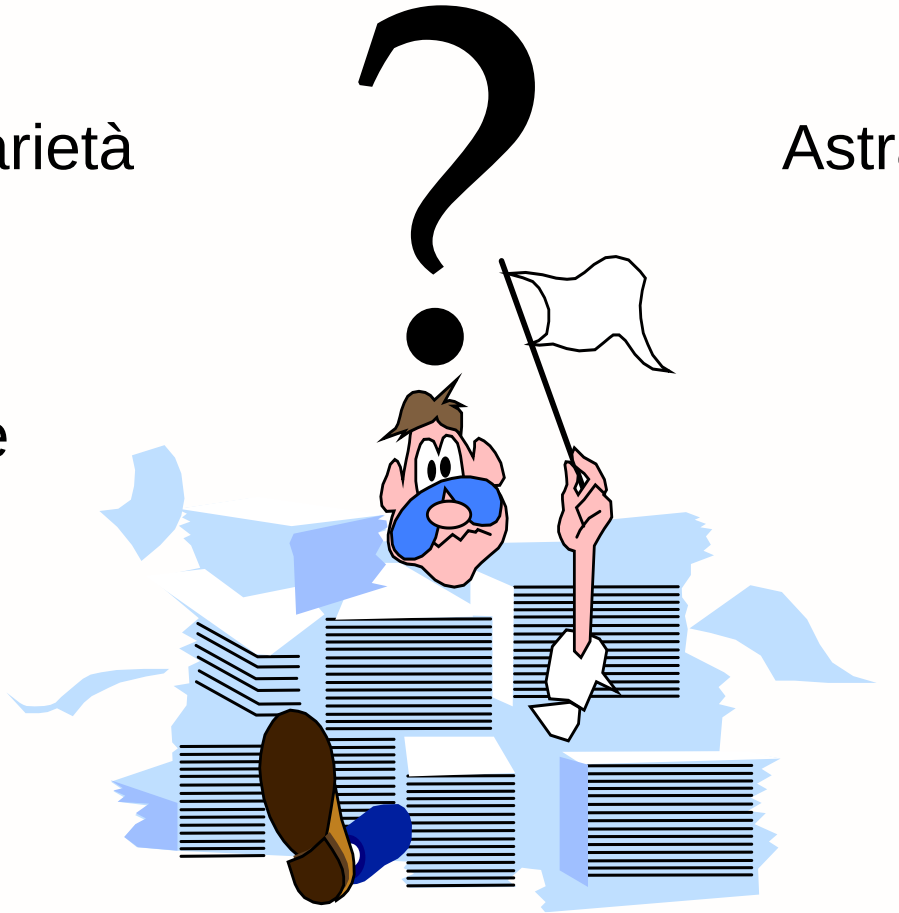
La Tecnologia Object Oriented

Oggetto

Polimorfismo

Ereditarietà

Astrazione



Classe

Istanza

Incapsulamento

Metodo

Cos'è un oggetto?

- In fase di analisi:
  - un' astrazione dal mondo reale (classe_+ astratta)

- Nella soluzione attuata:
  - una rappresentazione (classe concreta)

  costituita da:
    - dati
    - interfacce di comunicazione con il mondo esterno
    - operazioni sui dati

  - istanza di una data classe

Classi, Istanze ed Oggetti

- Oggetti ed Istanze sono spesso usati come sinonimi
- Si può distinguere:

  - Classe Oggetto (Object Class) come descrizione delle caratteristiche

  - Istanza Oggetto (Instance Object) entità fisica con tali caratteristiche

Tre tipi di classi

- **Entità**

- Incapsulano dati ed interfaccia con il mondo esterno

- **Oggetti di controllo**

- Nascono per soddisfare le esigenze di logica della data applicazione

- **Oggetti di presentazione**

- Controllano informazioni e comportamenti tipici di un ambiente (Es. S.O. + GUI)

Una visione naturale

Entità

- Identificano gli oggetti naturali esistenti nella nostra azienda
- Organizzano la visione globale del sistema

Gli ‘oggetti’ assumono comportamenti e caratteristiche umane

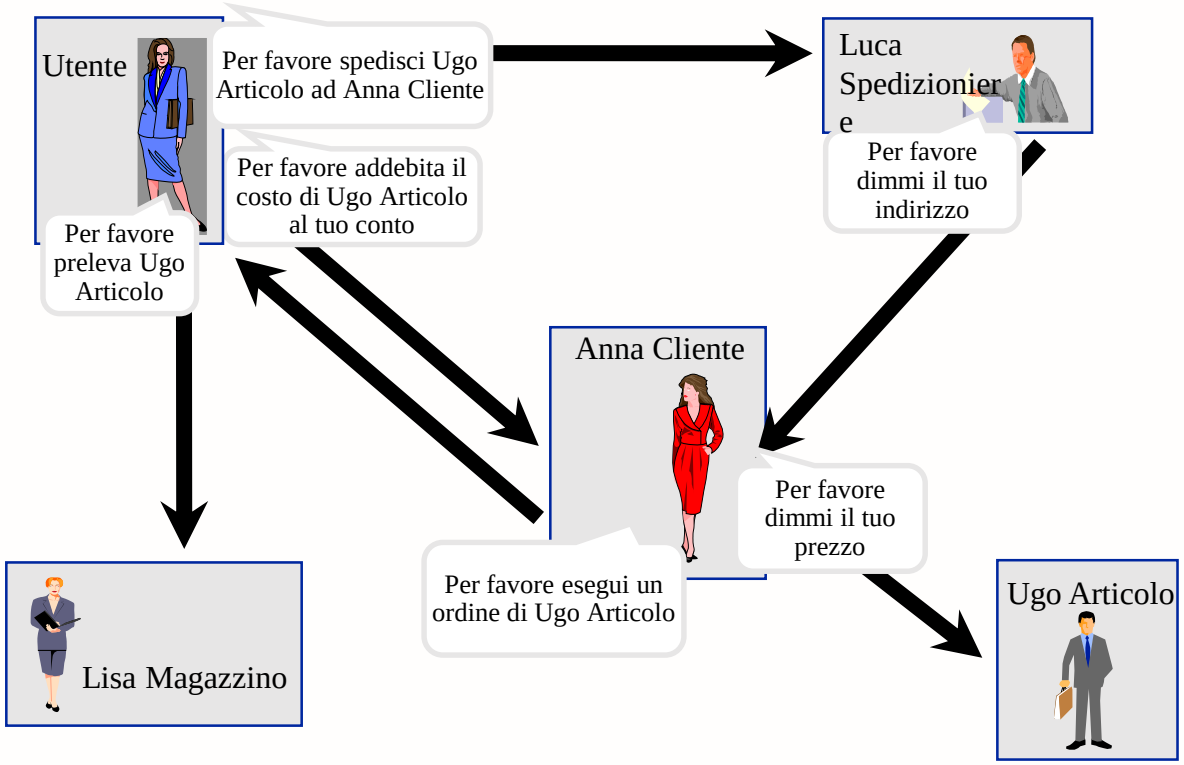
- Il Sig. *Articolo* dice:
  - “Se me lo chiedi, ti dirò il mio prezzo di acquisto"

- La Sig.ra *Cliente* dice:
  - " Se me lo chiedi, ti dirò il mio indirizzo"
  - " Se me lo chiedi, effettuerò un addebito al mio conto"

- La Sig.ra *Magazzino* dice:
  - " Se me lo chiedi, preleverò un articolo per te"

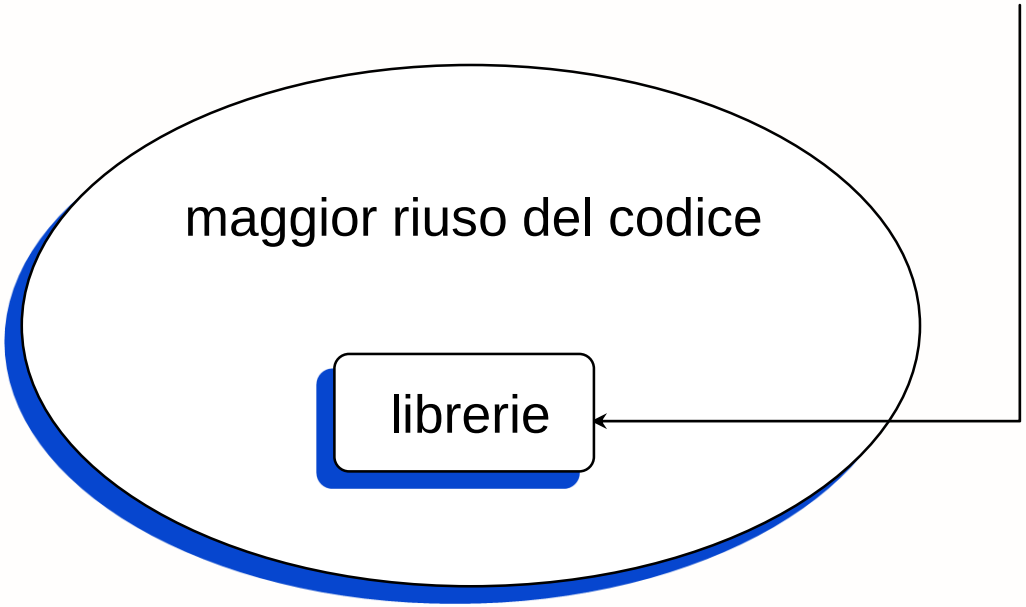
- Il Sig. *Spedizioniere* dice:
  - " Se me lo chiedi, ti spedirò un articolo all’indirizzo che mi indicherai"

Gli oggetti comunicheranno fra loro per completare il lavoro



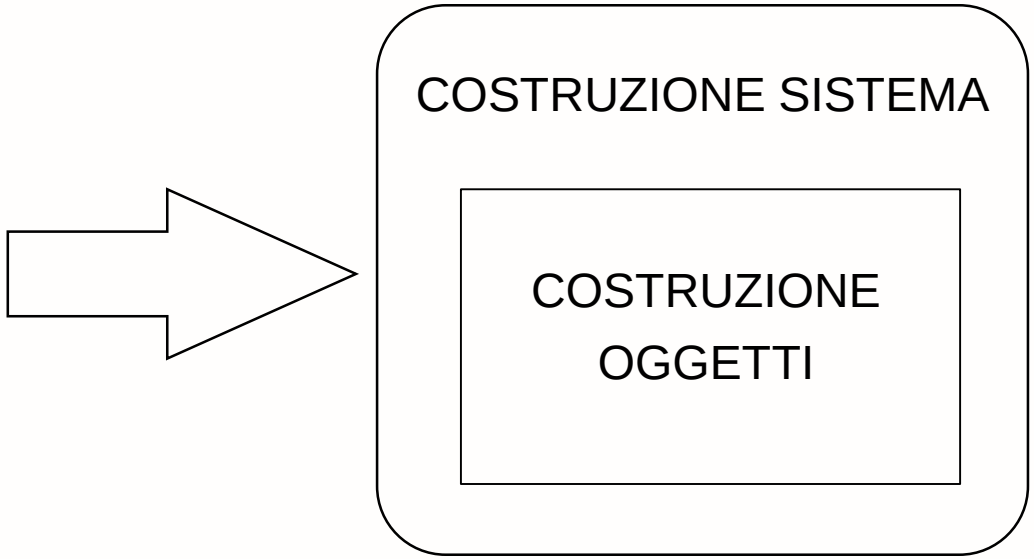
Maggiori benefici

- Migliore manutenibilità grazie all'incapsulamento
- Più facile il disegno grazie all'astrazione (risultato dell'analisi)
- Automatismi rivolti alla generazione di nuovi oggetti



Orientamento agli oggetti

Orientamento agli oggetti

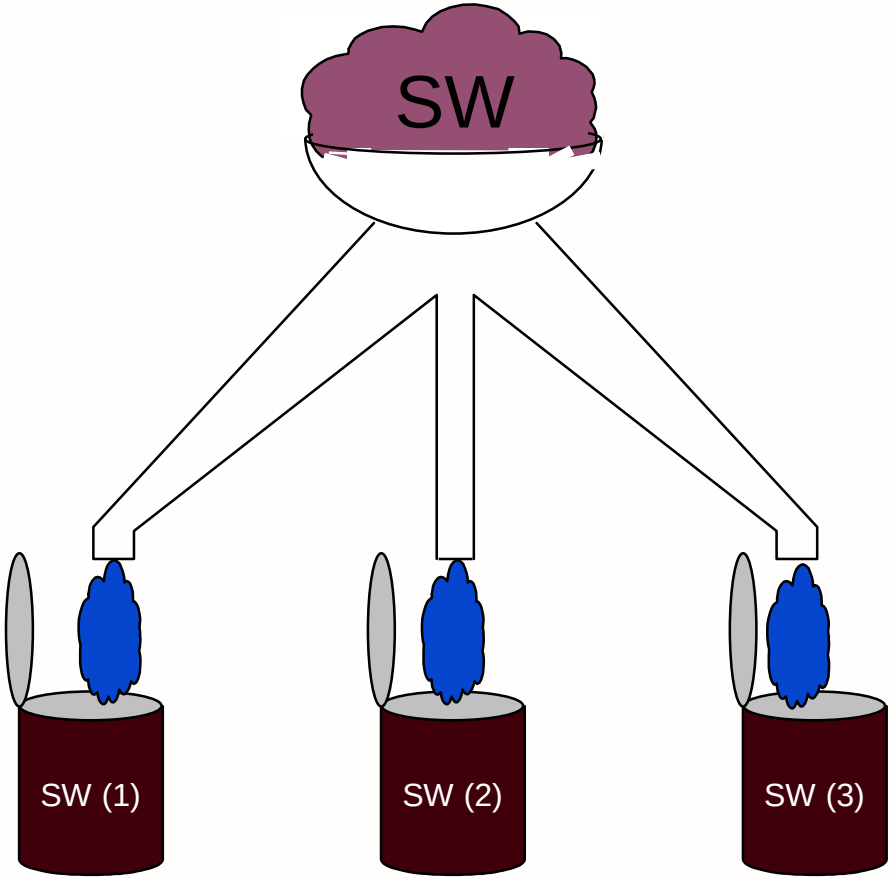


2 – I concetti fondamentali dell'Object Oriented

I concetti dell'Object Oriented

- Modularità
- Incapsulamento
- Astrazione
- Ereditarietà
- Gerarchia
- Polimorfismo
- Comunicazione fra gli oggetti (messaging)

Modularità



Modularità

- Proprietà che possiede un sistema che è stato scomposto in una serie di moduli
  - Coesivi (trattano interamente un problema)

  e con scarso accoppiamento (hanno pochi riferimenti l'uno all'altro)

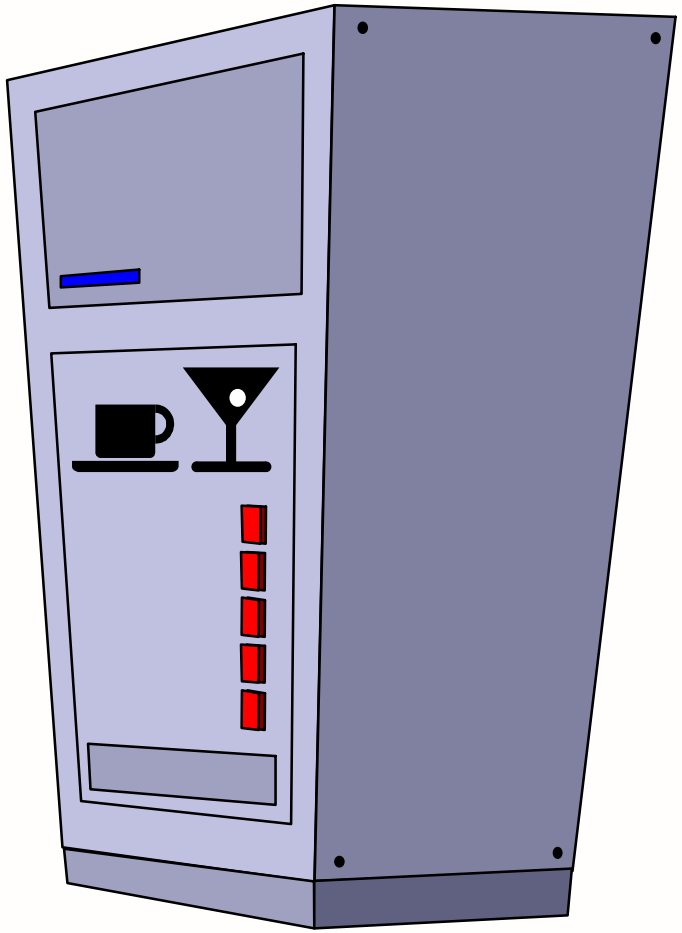
  ciascun modulo non sa cosa accade attorno a lui

Modularità

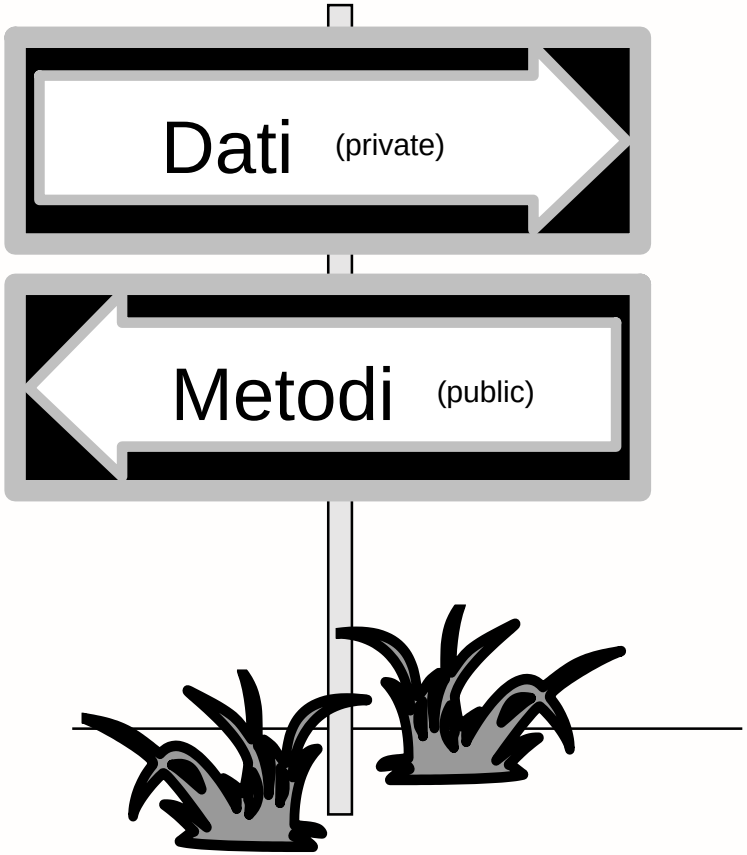
- Ogni modulo ha una visione parziale del mondo dove opera e non può agire al di fuori di tale visione

Modularità

I moduli possono interagire fra loro nel modo predisposto dal sistema stesso (Interfaccia/Messaggi)



Incapsulamento



Incapsulamento

Protegge l'oggetto nascondendo:

- lo stato dei dati
- l'implementazione delle sue funzioni (comportamenti)

Incapsulamento

- Esempio:

Oggetto Ordine Acquisti

attributi (dati)

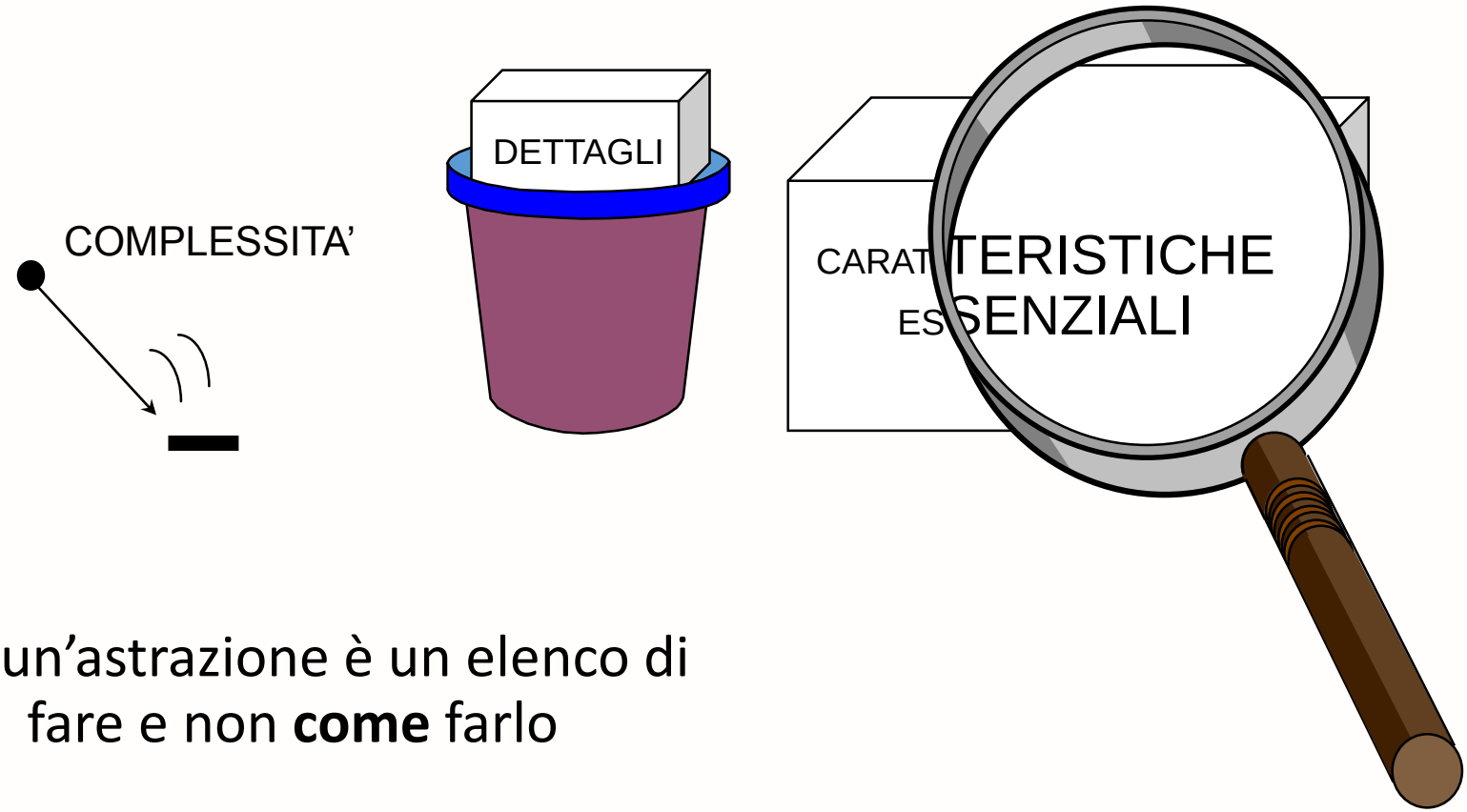
metodi (funzioni)

Astrazione

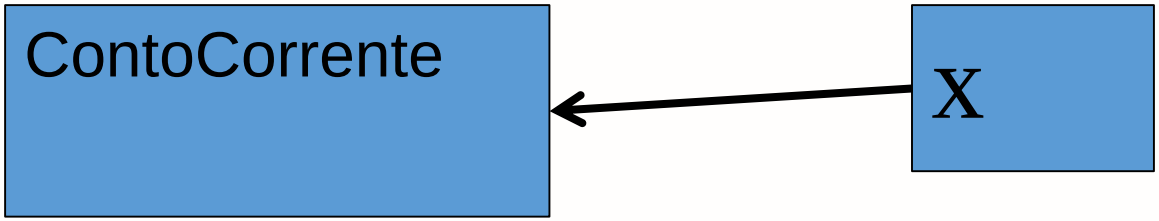
Astrazione

Procedimento tendente a sostituire con una formula la concreta molteplicità del reale

Astrazione



Risultato di un'astrazione è un elenco di **cosa** fare e non **come** farlo

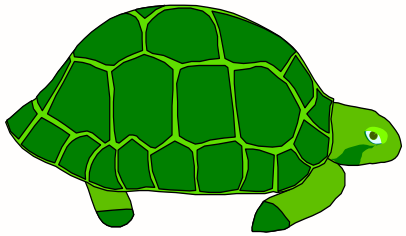



```
ContoCorrente x=new ContoCorrente();
```

Gli oggetti

Elementi d'insieme che, in una ipotesi di funzionamento basata su di una modellazione del mondo reale, si **comportino** conformemente alle attese del sistema

Comportamento

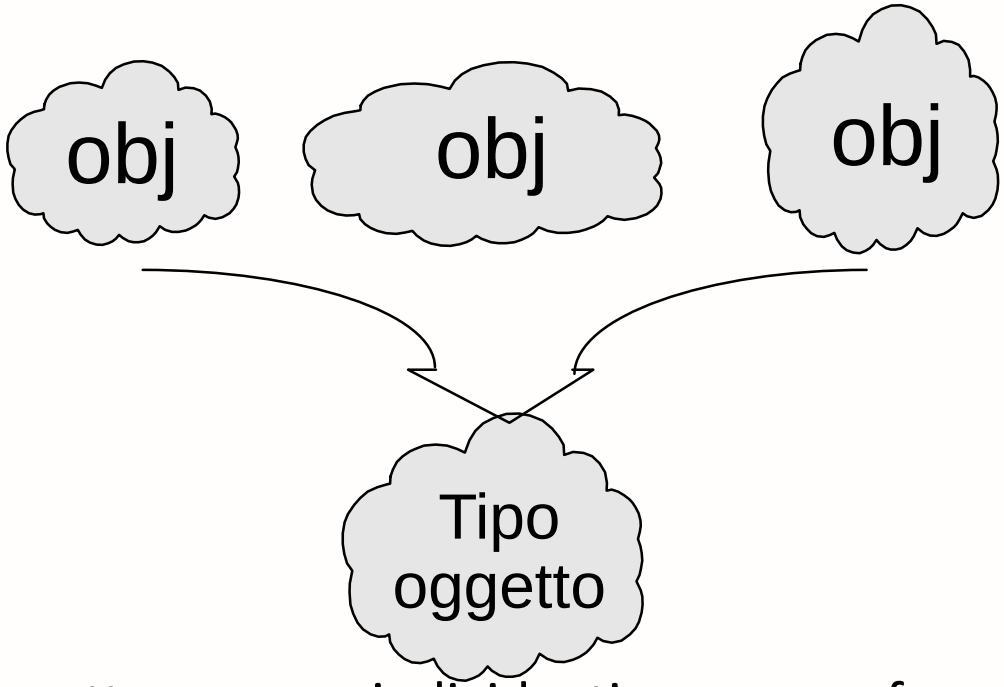


COMPORTAMENTI COMUNI

MANGIARE
MUOVERSI
BERE
...

Catalogare l'ambiente in oggetti tipo

- Non bisogna avere un tipo oggetto per ciascuna entità da rappresentare



- Meno tipi oggetto saranno individuati, meno confuso sarà lo scenario.

Puntare sull'astrazione

Esempio

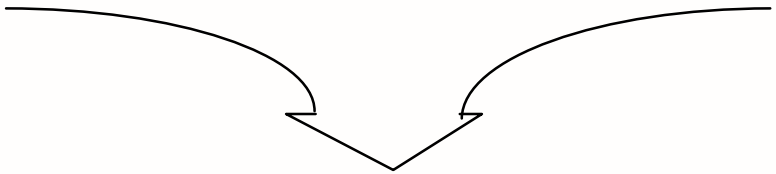
- In una grossa banca sono state contate 1800 tipi di transazioni diverse.
- Dopo una approfondita analisi O.O. sono stati individuati 49 tipi base di transazioni.
- Nel tempo potranno aumentare

Esempio

Otteniamo:

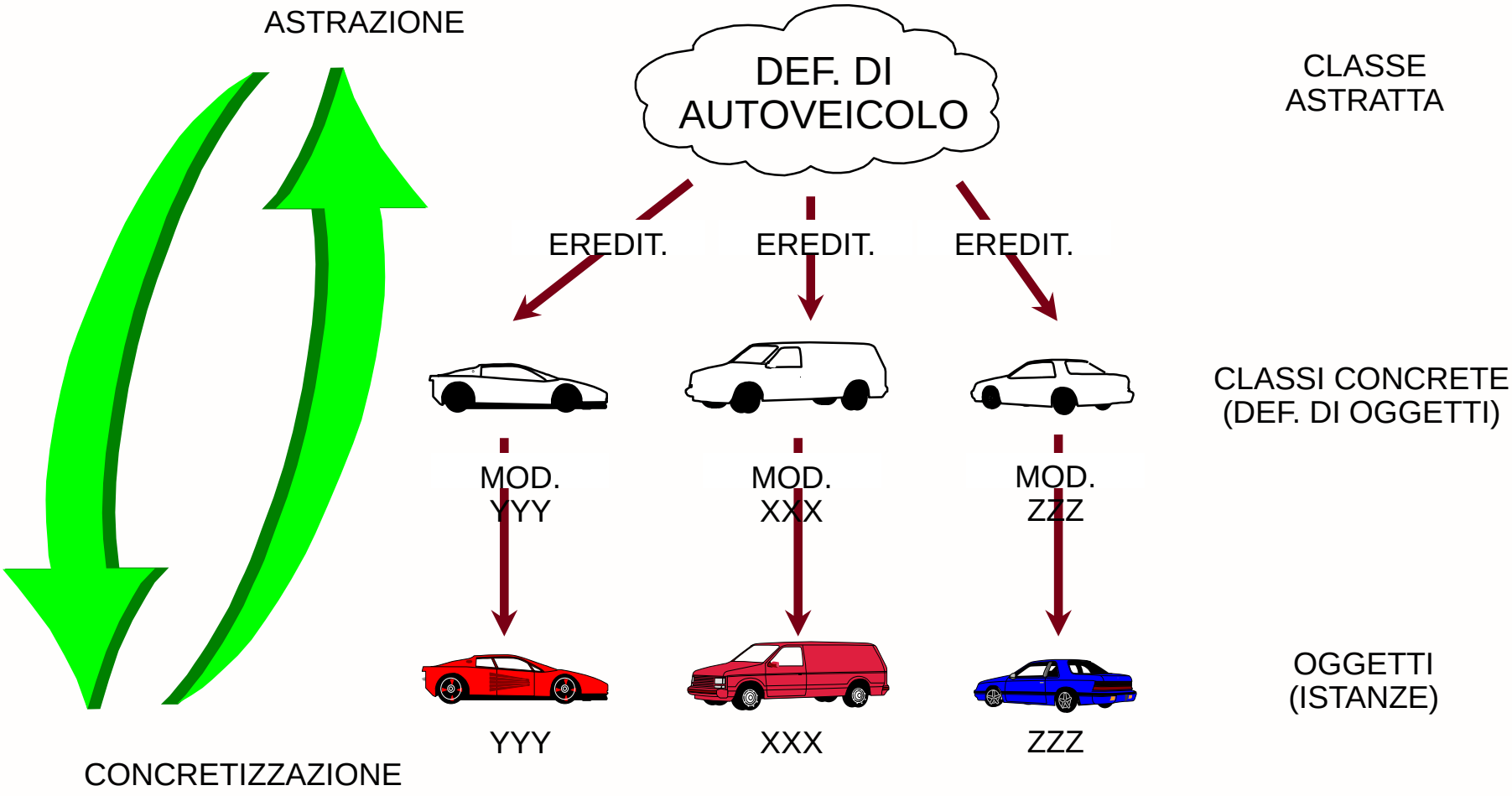
- La diminuzione della complessità dei singoli oggetti
- La diminuzione della varietà di oggetti

Attraverso



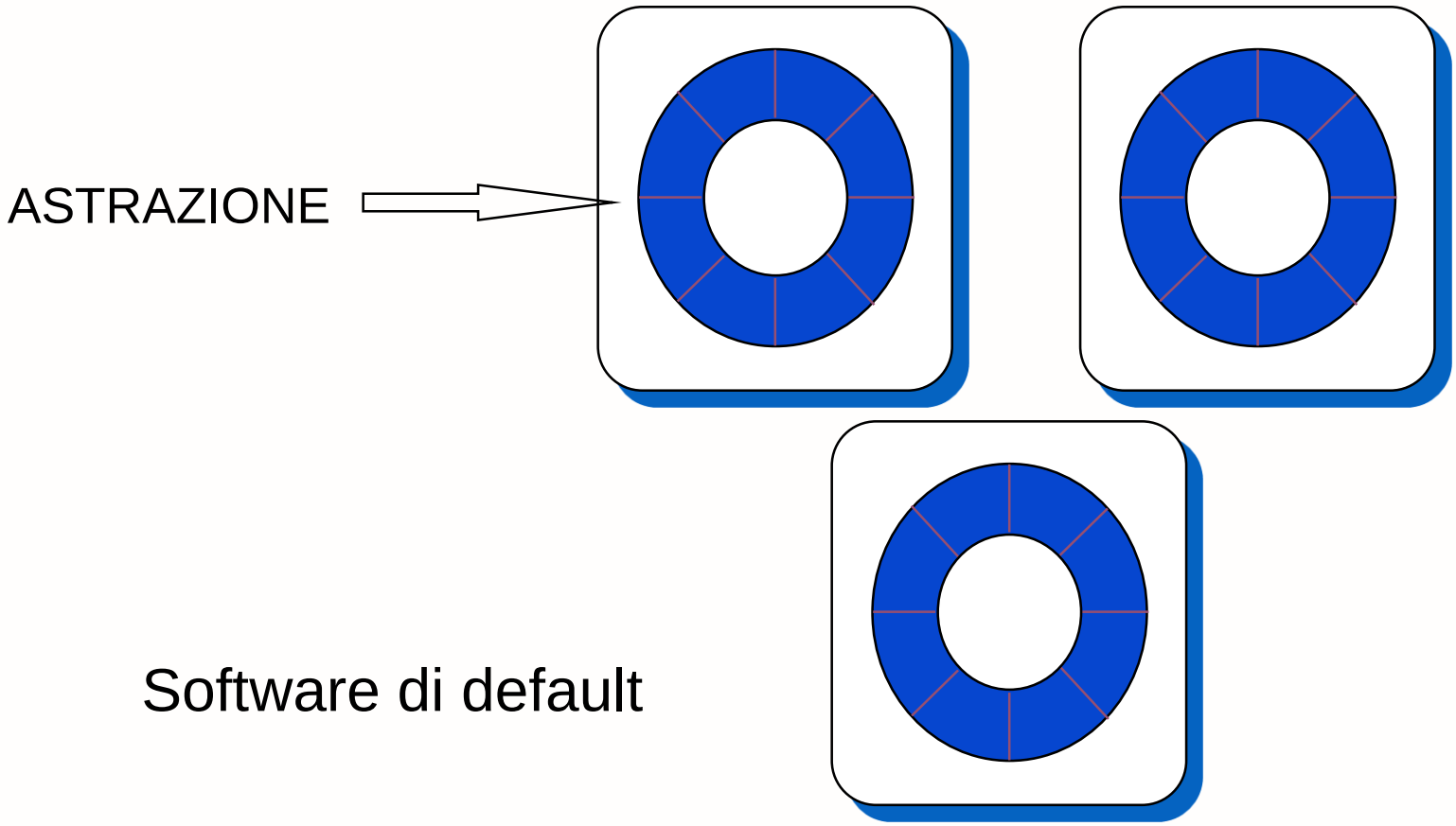
- La scrittura di regole ad un livello superiore
- L'elenco delle caratteristiche comuni degli oggetti derivati

Astrazione e Concretizzazione



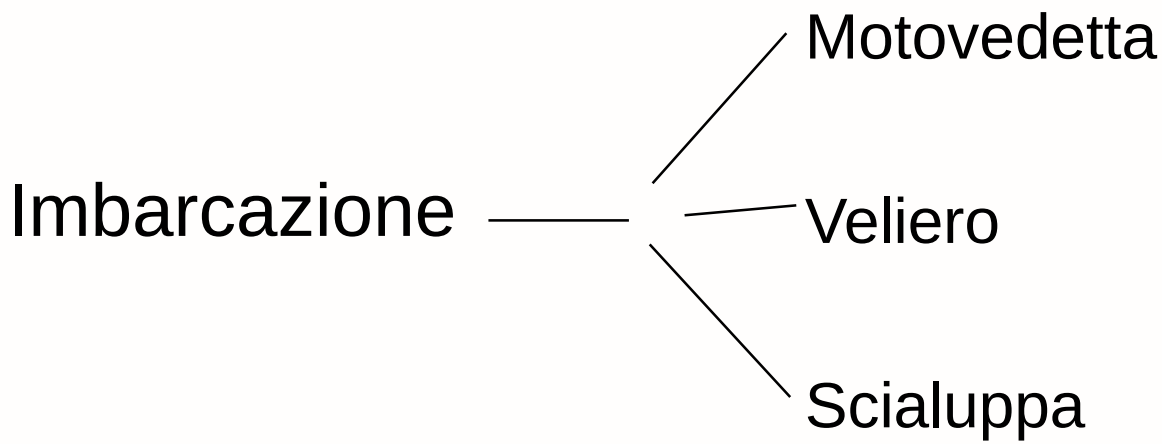
Astrazione

Astrazione = oggetti (astratti) dal carattere forte

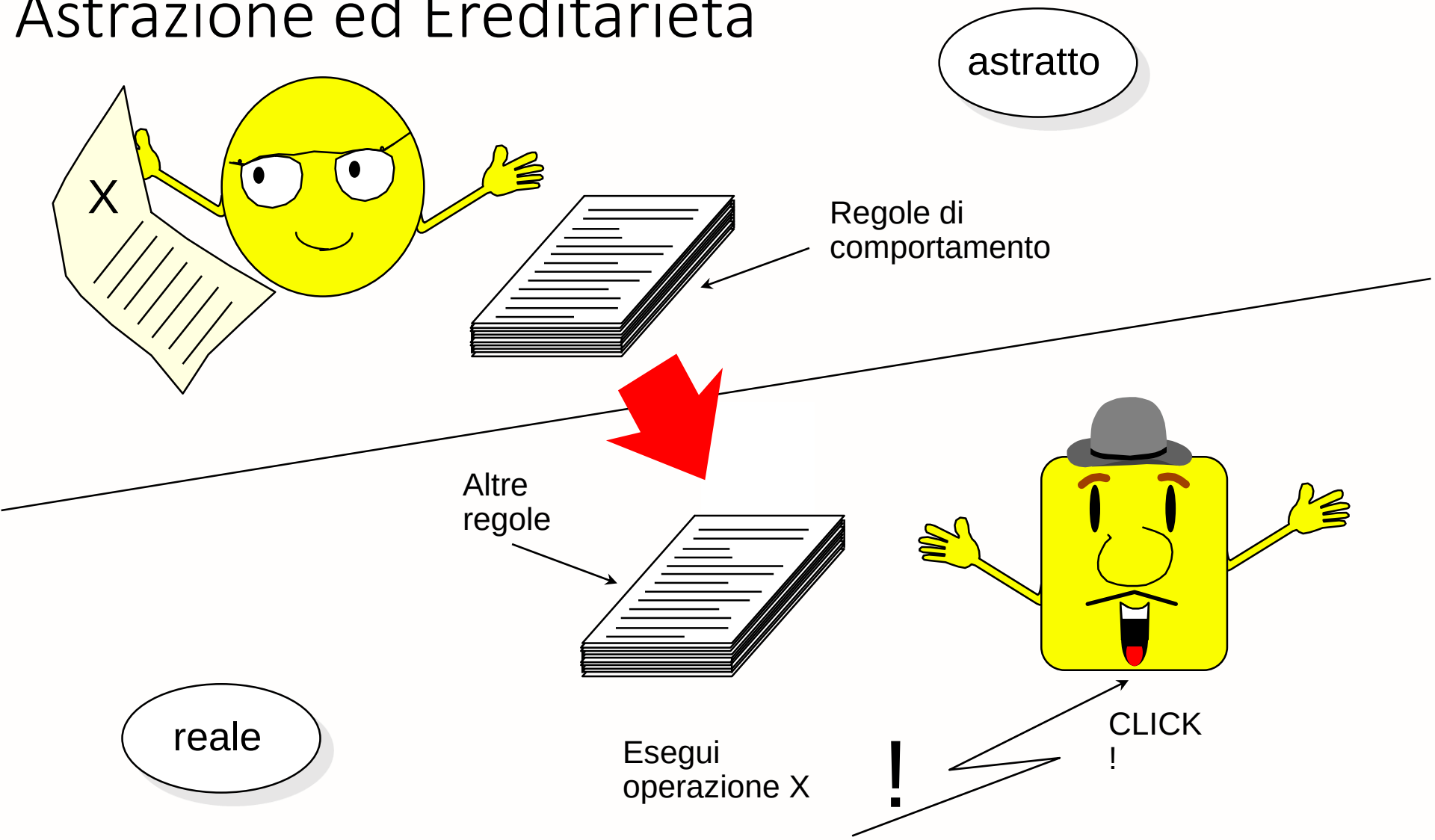


Ereditarietà

Per ottenere il comportamento di default diremo che il nostro oggetto discende dall'oggetto X



Astrazione ed Ereditarietà



Gerarchia

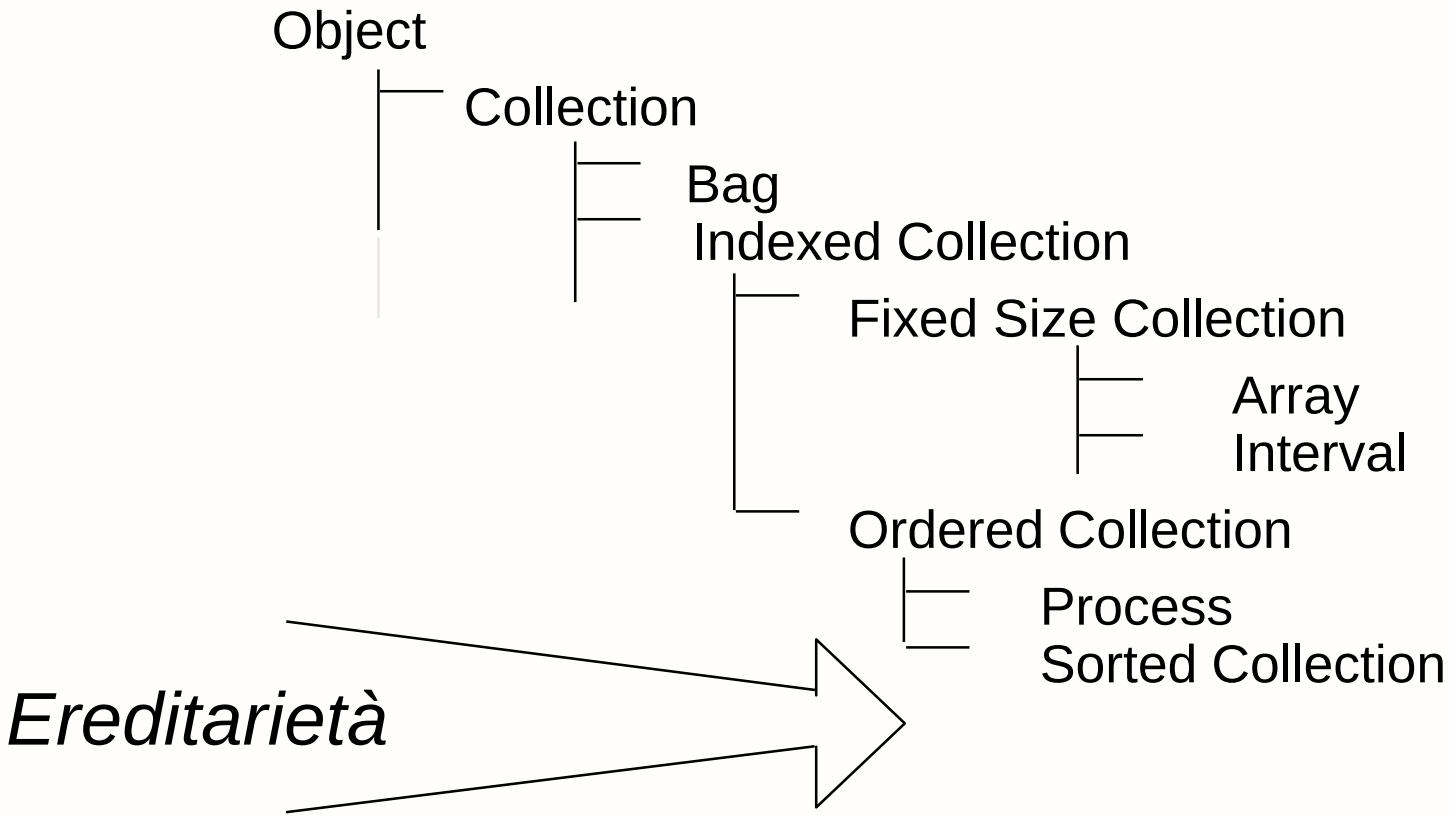
Struttura dei collegamenti tra le classi

Distinguiamo due tipi di gerarchie

- “un tipo di”
- “una parte di”

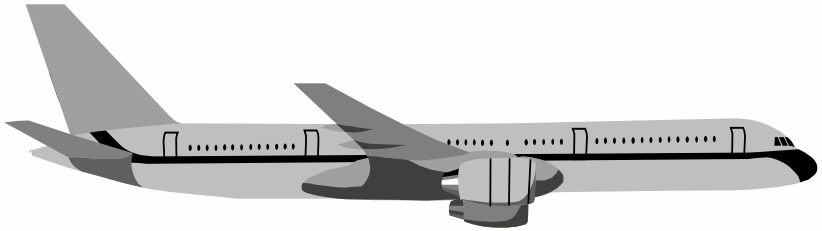
Gerarchia - un Tipo di

Ogni classe eredita comportamento e informazioni contenute nella classe dalla quale dipende gerarchicamente

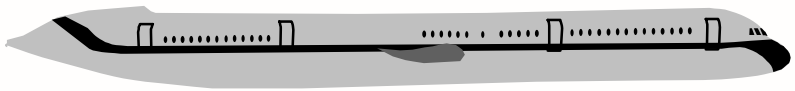


Gerarchia - una Parte di

- Una classe contiene istanze di altri oggetti

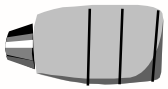


Aereo

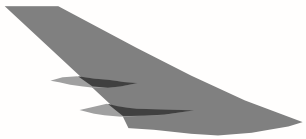


Fusoliera

Coda



Motore



Ala

Polimorfismo

Polimorfo:

dal greco polymorphos “dalle molte forme” in mineralogia, botanica e biologia che ha o può assumere forme diverse

Devoto, Oli

→ in O.O. il concetto è esteso ai comportamenti (azioni)

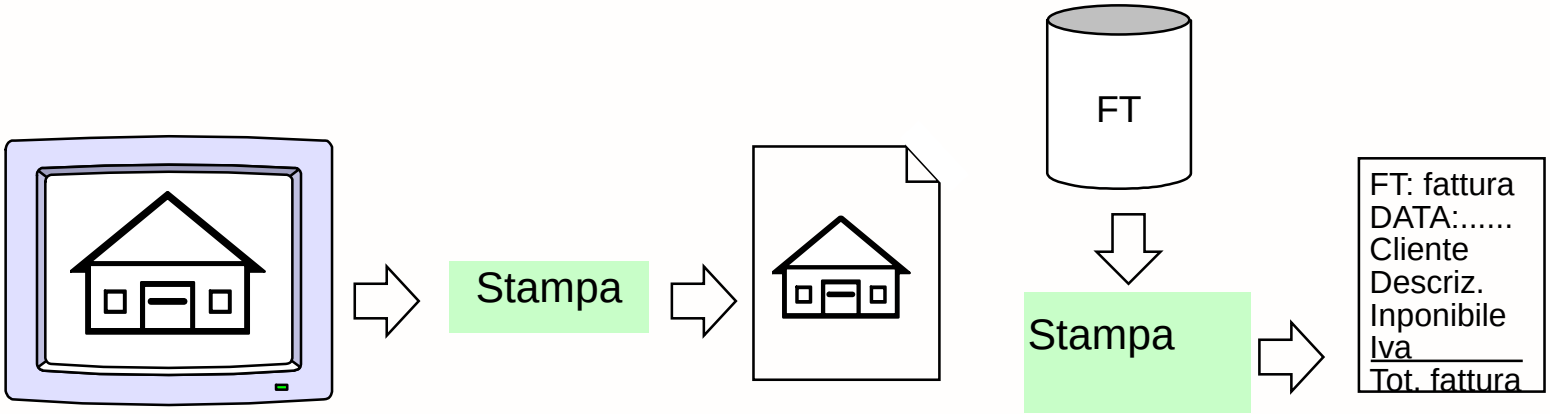
Polimorfismo

Es.

Neve cristallizzata

Es.

Stampa oggetto



Polimorfismo

- Ottimo strumento per rendere flessibili i sistemi

- Derivando una nuova classe da una esistente si possono apportare modifiche al comportamento
  - minimali o
  - importanti

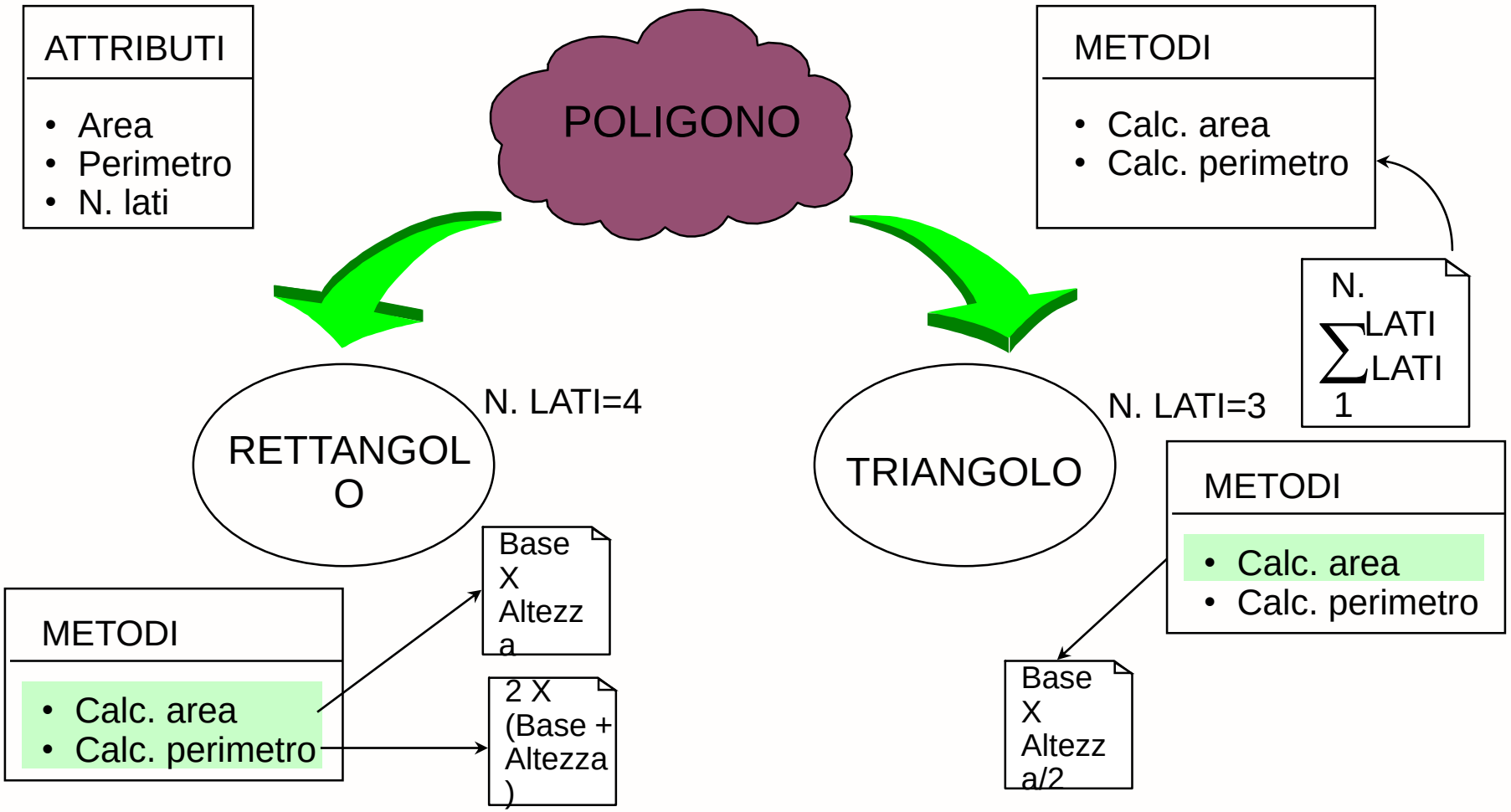
  senza ripercussioni sull'intero sistema dell'invio dei messaggi

Polimorfismo

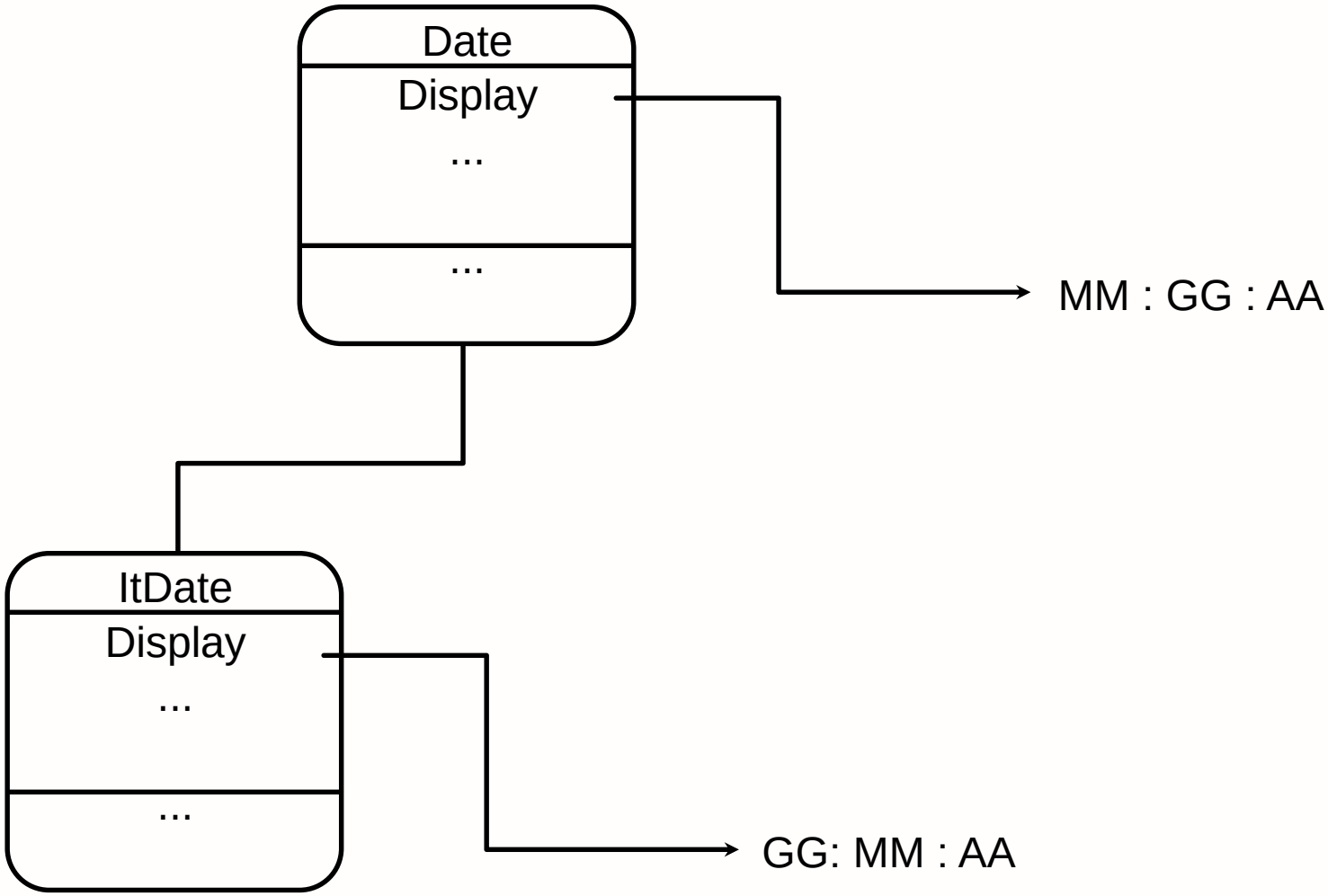
Meccanismo che permette di riscrivere **solo** quella parte di codice inadeguato e **solo** per gli oggetti che ne hanno bisogno

Polimorfismo

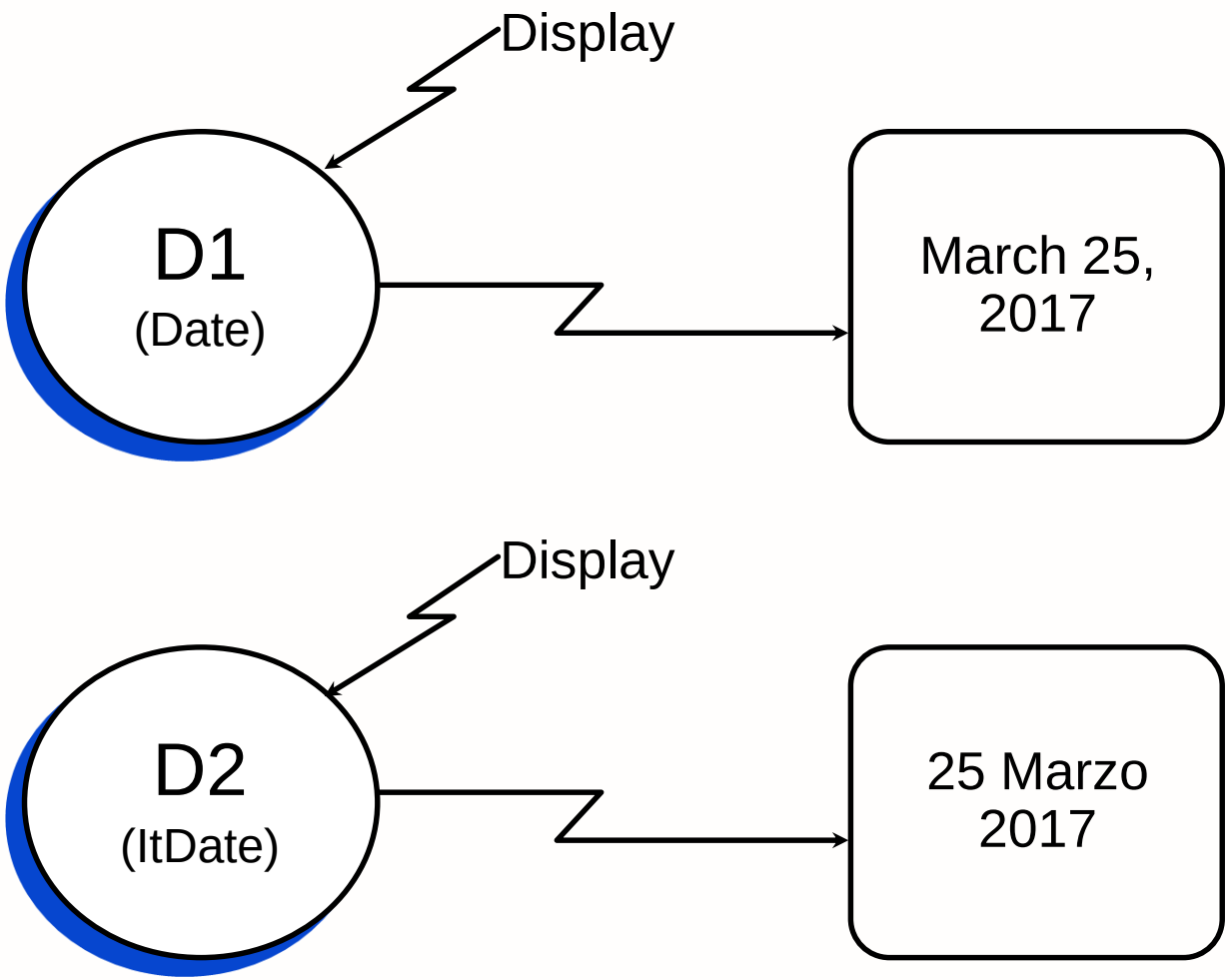
Nella uniformità di comportamento bisogna quindi poter riconoscere le differenze che esistono tra oggetti diversi



Polimorfismo

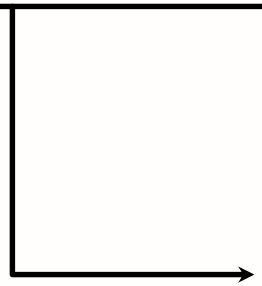


Polimorfismo



Polimorfismo

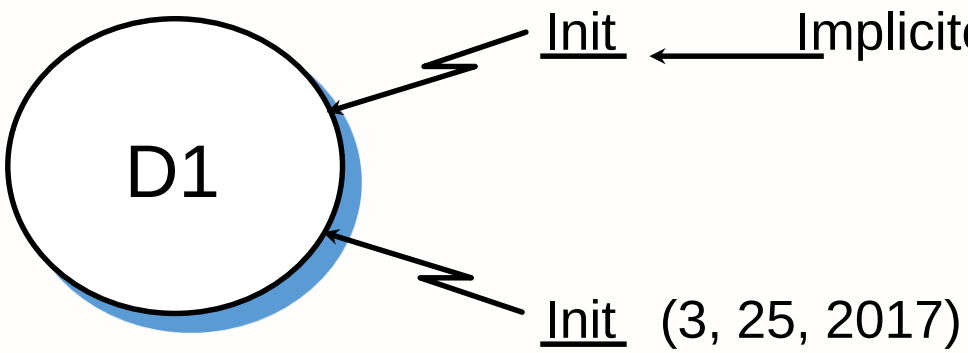
OVERRIDING



Il compilatore riconosce la classe di appartenenza e quindi il metodo da applicare

Esistono più metodi che hanno lo stesso nome e che hanno uguale modalità di scambio messaggi, ma appartengono a tipi (classi) diversi

Polimorfismo



OVERLOADING

Il compilatore riconosce fra le diverse modalità di richiamo e applica il metodo appropriato

Esistono più metodi con diverse modalità di scambio messaggi, tutti attivabili dalla medesima istanza di oggetto.