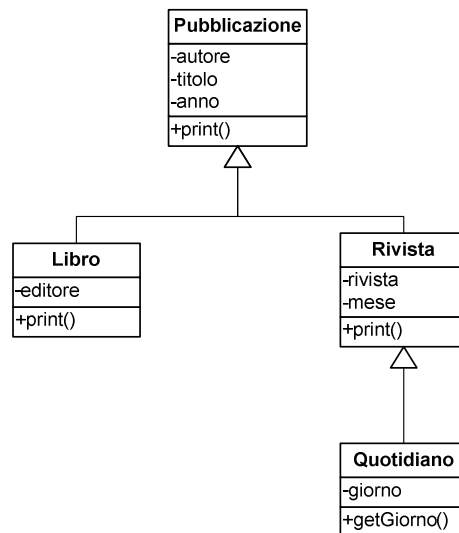


Programmazione ad Oggetti

Ereditarietà

V 1.1 © Marco Torchiano 2005

Esempio – pubblicazioni



Polimorfismo & Up-cast

- Polimorfismo
 - ♦ È sempre possibile vedere un oggetto attraverso un riferimento ad una delle sue classi di base
 - Up-cast
 - ♦ Cast da una classe derivata ad una classe di base
 - ♦ E' eseguito automaticamente dal compilatore
-

Esempio

```
Quotidiano q = new Quotidiano
                ("Buongiorno", "2005",
                "Gramellini", "La Stampa",
                "Ottobre", "12");

q.getGiorno(); // OK

Rivista q_come_rivista = q;
q_come_rivista.getGiorno(); // NO

Pubblicazione q_come_pubb = q;
q_come_pubb.getGiorno(); // NO
```

Dynamic binding

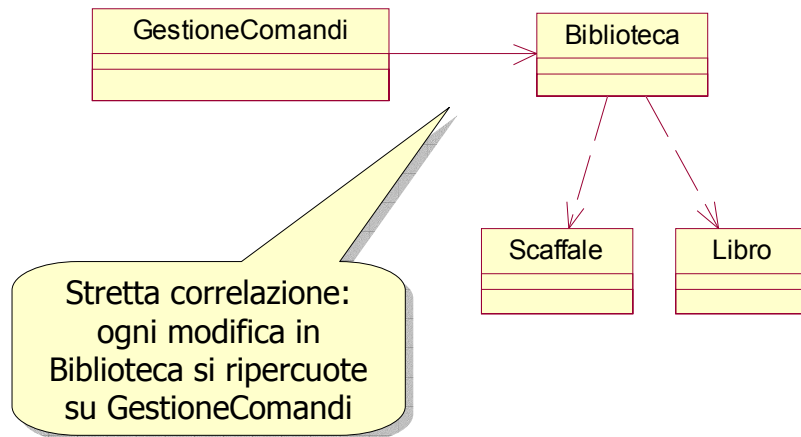
- All'invocazione di un metodo, la JVM invoca il metodo definito nella classe effettiva dell'oggetto
 - ♦ Indipendentemente dal tipo del riferimento
 - ♦ Se un metodo non è definito viene risalita la gerarchia fino a trovare il metodo
-

Dynamic binding

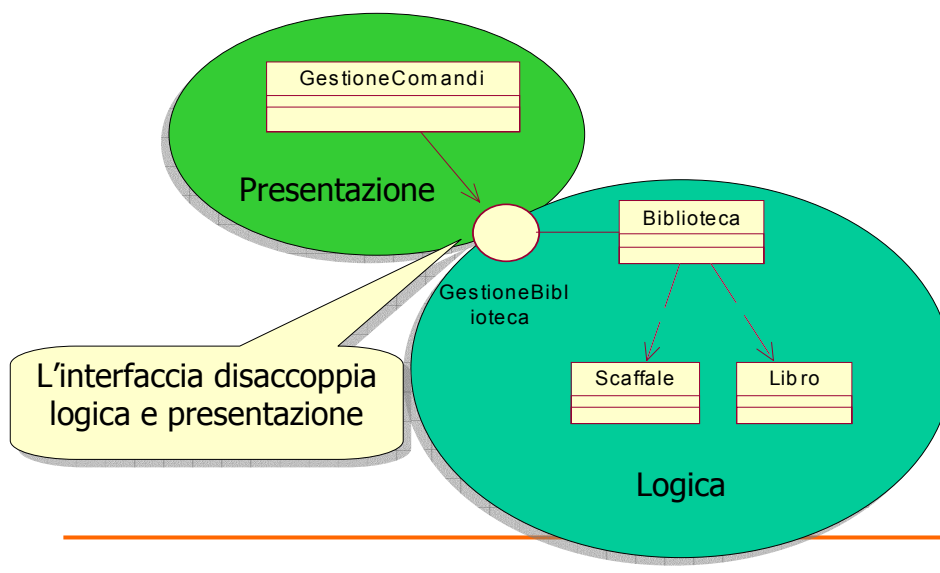
```
Rivista r = new Rivista("Rubrica",  
                        "2005", "Pippo",  
                        "Panorama", "Settembre");  
  
stampa(r);  
stampa(q);  
...  
stampa(Pubblicazione pubb){  
    pubb.print();  
}
```

Quale variante di print() viene richiamata?

Architettura Biblioteca



Interfaccia



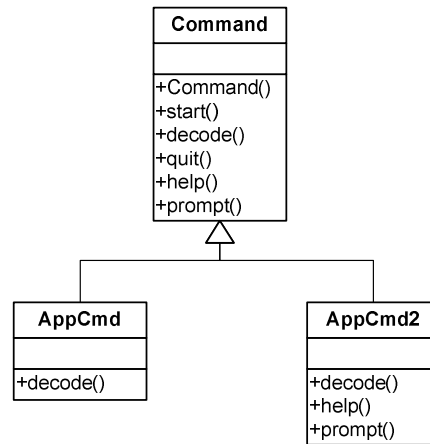
Problema

- Nelle interfacce utente a carattere, ci sono una serie di operazioni ricorrenti
 - ♦ Prompt
 - ♦ Lettura comando
 - ♦ Quit
 - ♦ Help
 - Come faccio a riutilizzare il piu' possibile?
-

Soluzione: Fattorizzazione

- Si raccoglie in una classe tutto quello che è in comune
 - Si definisce un metodo che esegue l'algoritmo comune
 - ♦ Richiama altri metodi che possono essere ridefiniti nelle classi derivate
-

Esempio



Pattern: Template Method

- Il metodo `start()` definisce
 - ♦ Comportamento comune delle interfacce a linea di comando (CLI)
 - ♦ Definisce un algoritmo basato su operazioni atomiche
 - Le operazioni atomiche
 - ♦ Sono implementate da metodi
 - ♦ Si possono ridefinire per personalizzare
-

Esercizio

Espressioni aritmetiche

Esercizio: espressioni

- Rappresentare espressioni aritmetiche su variabili e costanti.
 - Il valore delle variabili può essere modificato in qualsiasi istante.
 - Un'espressione può essere ricalcolata in qualsiasi istante utilizzando il valore attuale delle variabili.
-

Soluzione: AST Abstract Syntax Tree

- Espressione
 - ◆ $3 * (x + 2)$
- Rappresentazione:


```

graph TD
    A["*"] --- B["3"]
    A --- C["+"]
    C --- D["x"]
    C --- E["2"]
            
```
- Il valore di un'espressione può essere calcolato con una visita dell'albero.

Ipotesi di soluzione

left →

Node

- type: {opnd, opr}
- value: int
- operator: {+, -, *, /}
- name: String

- evaluate()

← right

Distingue operandi e operatori

Valore delle costanti e variabili

Tipo di operatore

Nome delle variabili

• Presenza di campi tipo

• Attributi utili in casi diversi

} Ereditarietà

Classi per Espressioni

