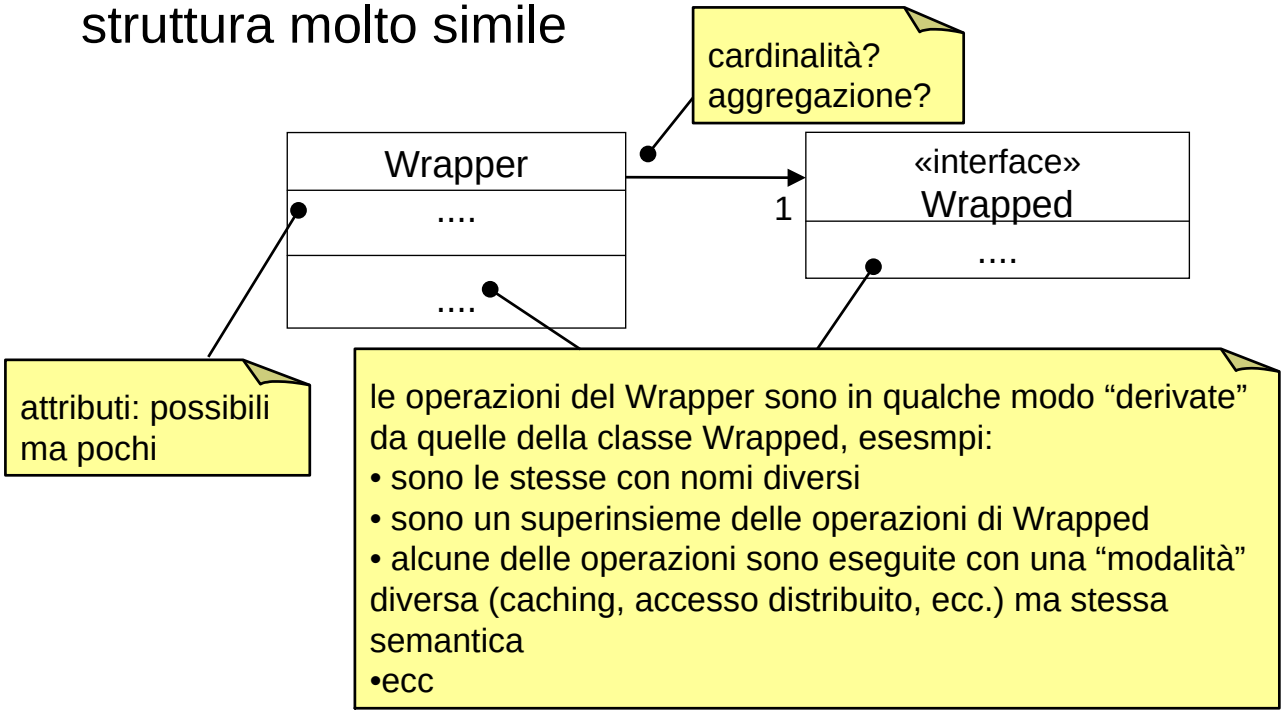
design patterns (GoF)

Wrappers, Adapter, Decorator,
Observer, MVC, esempi in java

wrappers

- sono una categoria di design patterns con una struttura molto simile



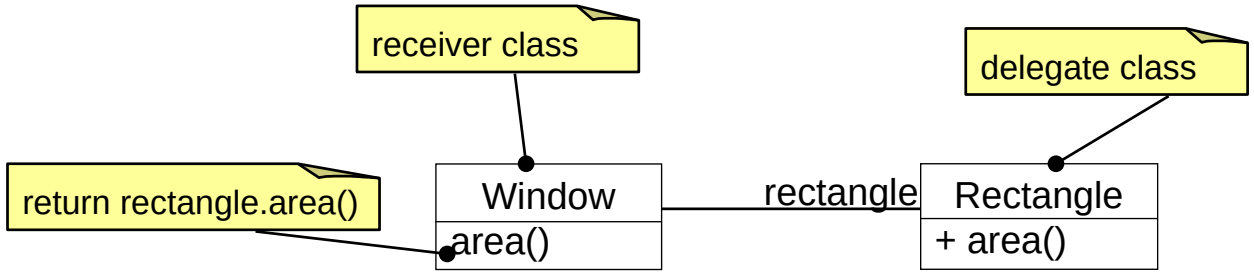
delegation o forwarding

- una classe ricevente (*receiver*) delega l'esecuzione di un metodo ad un altra classe detta delegata (*delegate*)

  il receiver

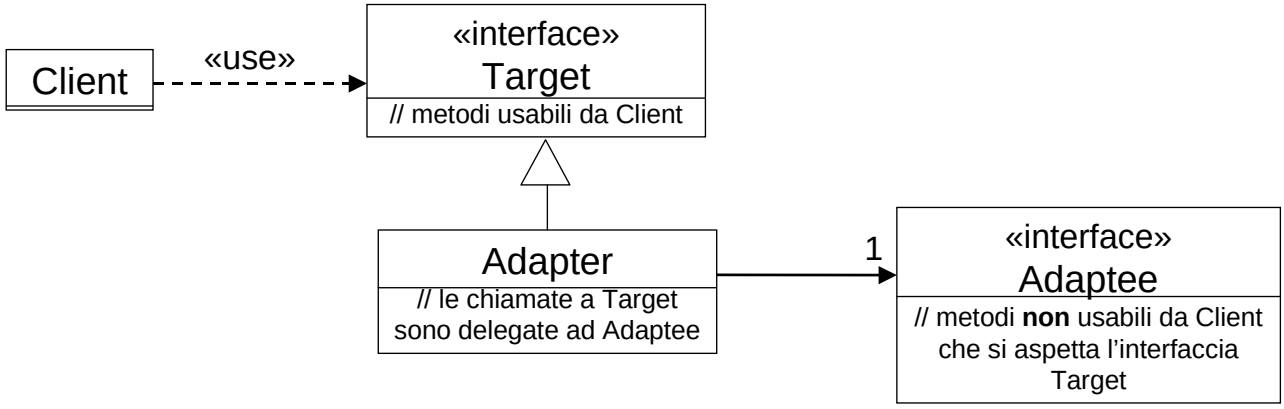
  - non aggiunge alcun comportamento
  - aggiunge poche operazioni prima o dopo l'esecuzione del metodo delegato

- esempio



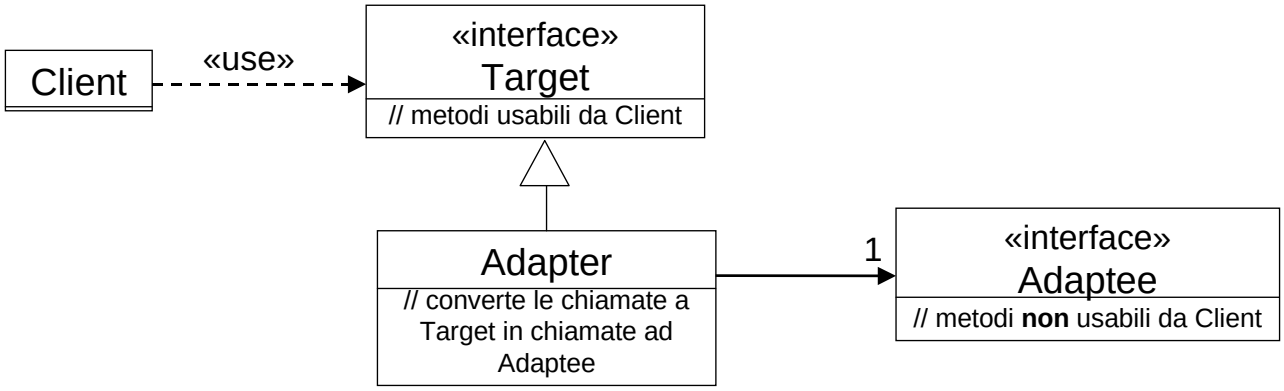
Adapter

- intento
  - convertire l'interfaccia di una classe (esistente) in un'altra interfaccia in modo che sia compatibile con un certo client (esistente).
- struttura



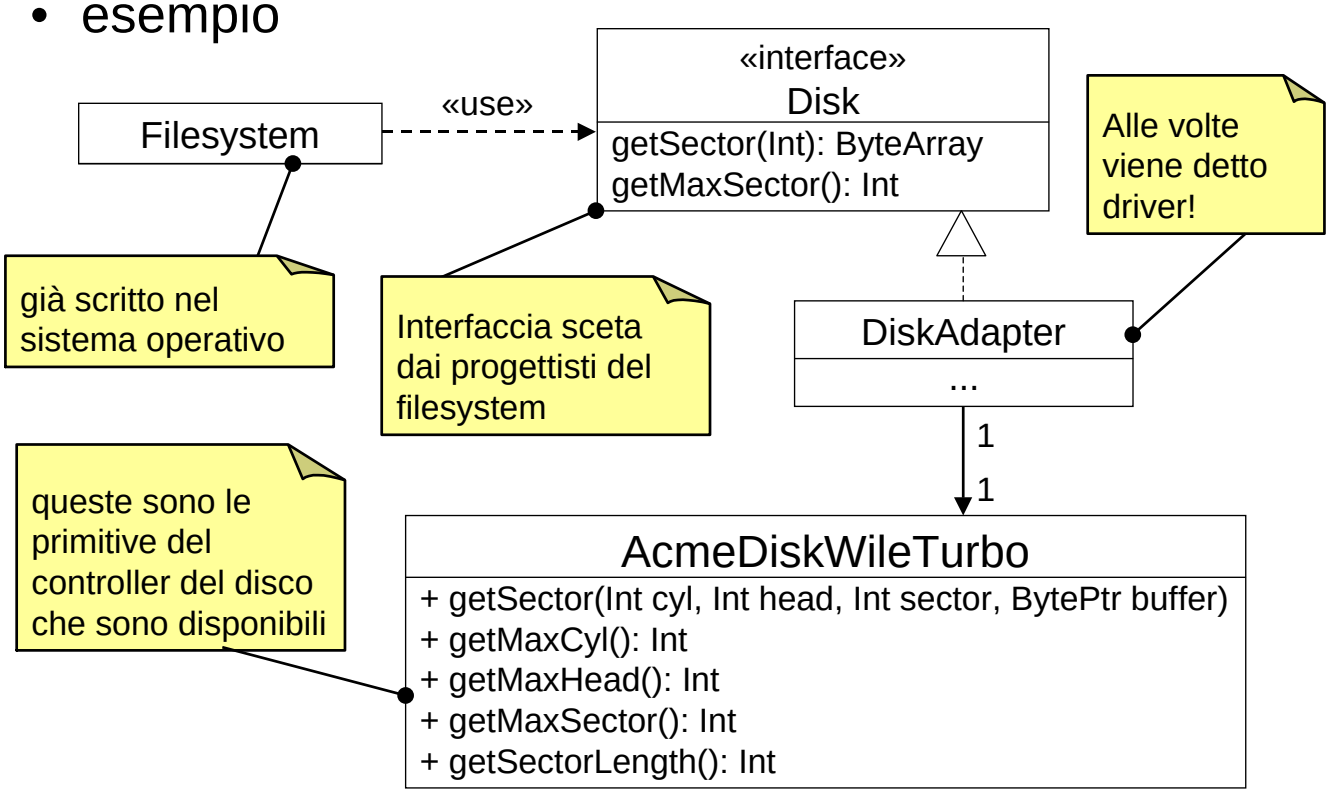
Adapter

- un adapter è un wrapper che...
  - non modifica il comportamento della classe adattata ma solo la sua interfaccia
    - nessun comportamento aggiunto che non sia relativo all'adattamento dell'interfaccia
- client e server esistono e devono comunicare ma non lo possono fare perché il client si aspetta una interfaccia diversa



Adapter

- esempio

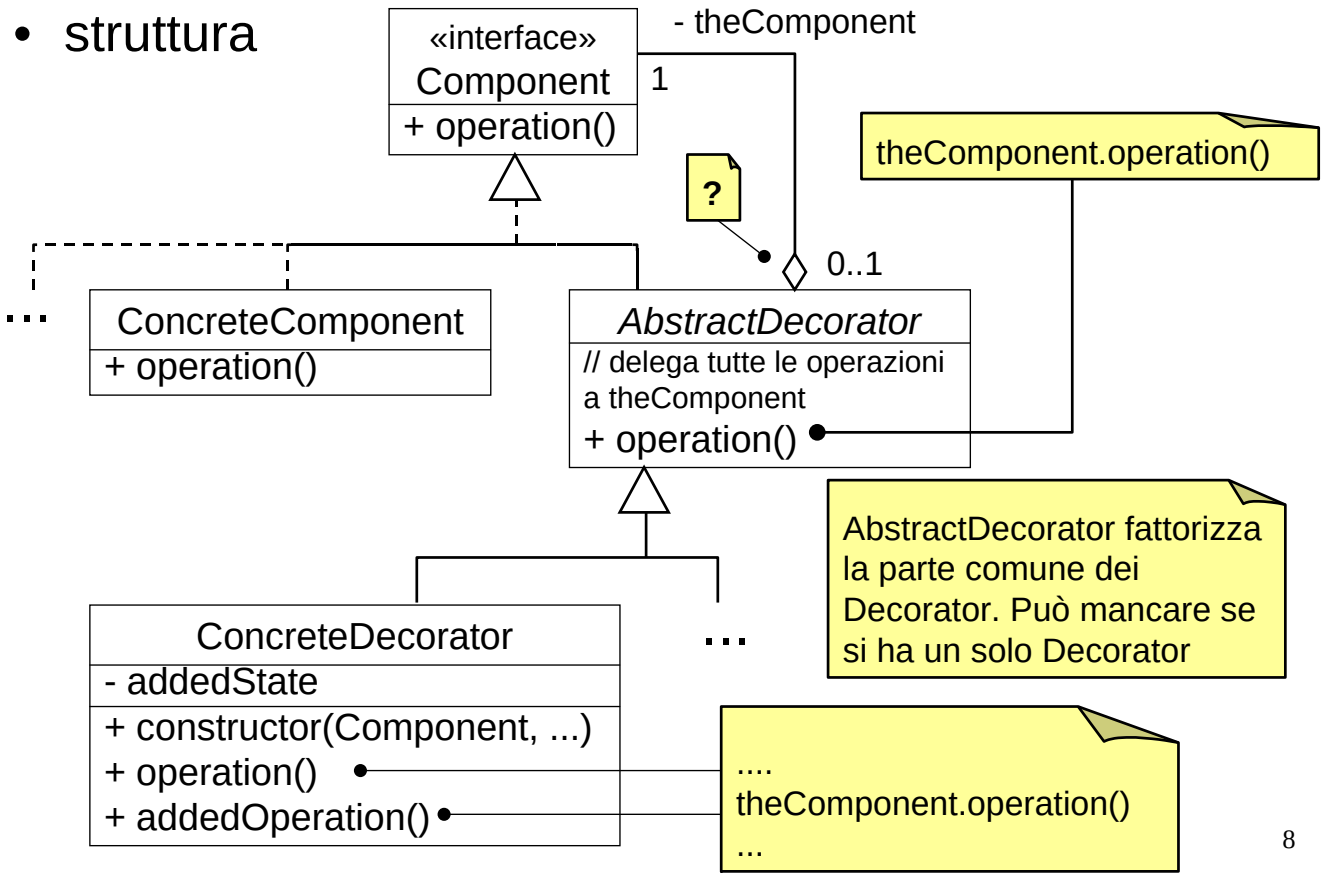


Decorator

- intento
  - aggiungere dinamicamente responsabilità ad un oggetto
  - alternativa flessibile al subcassing per estendere le funzionalità di un oggetto
  - risolve i seguenti problemi di modifica a run-time:
    - un oggetto non può estendere la sua interfaccia
    - un oggetto non può estendere il suo stato interno
    - un oggetto non può modificare il comportamento dei suoi metodi
    - tutto ciò è vero nella maggio parte dei linguaggi e in tutti quelli fortemente tipati come ad es. java e c++
  - decorator è un caso particolare di Wrapper

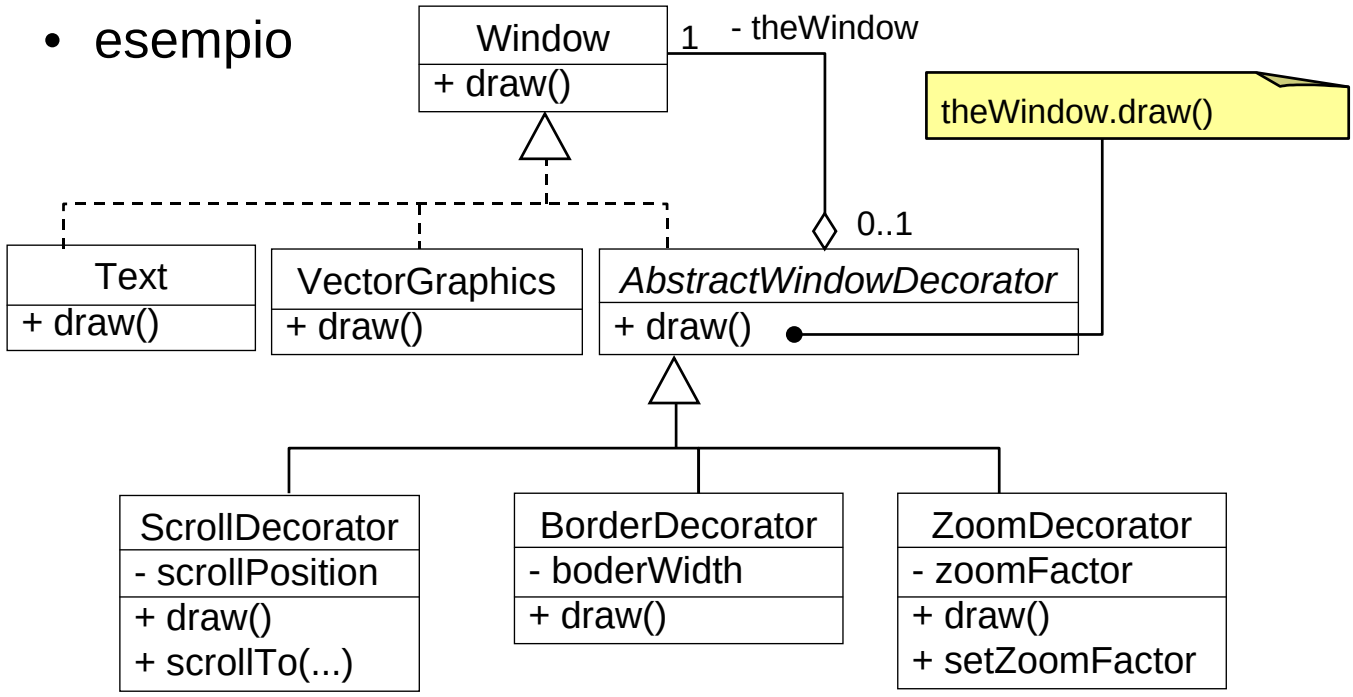
Decorator

- struttura



Decorator

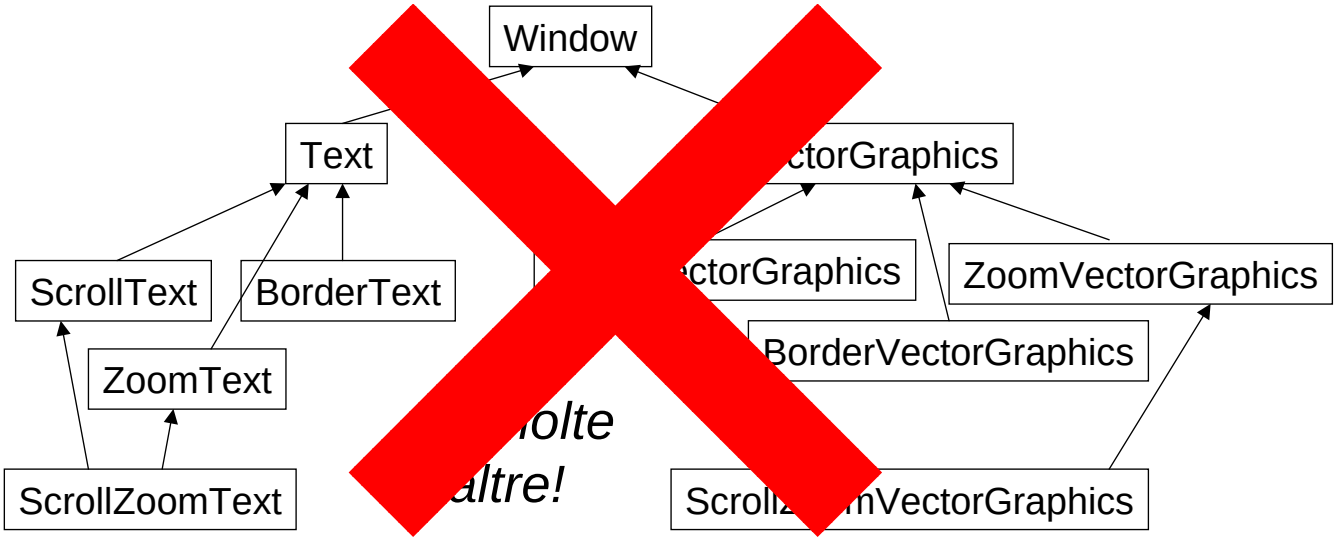
- esempio



- alcuni dettagli sono stati omessi
  - es. Window.draw() deve supportare il disegno di parti di finestra su un dispositivo definibile dall'utente

Decorator

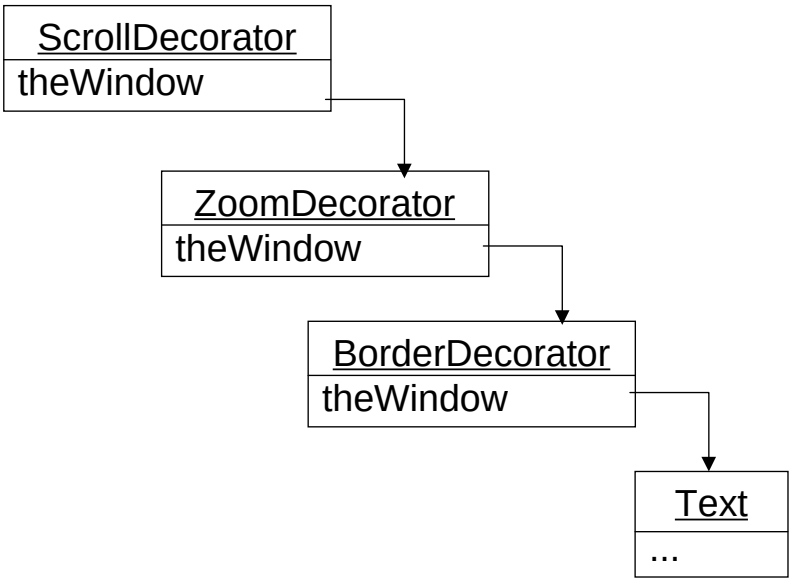
- l'uso dell'ereditarietà può portare ad una esplosione combinatoria del numero delle classi



- decorator permette di combinare funzionalità di vari decoratori mantenendo la possibilità di aggiungerne di ulteriori

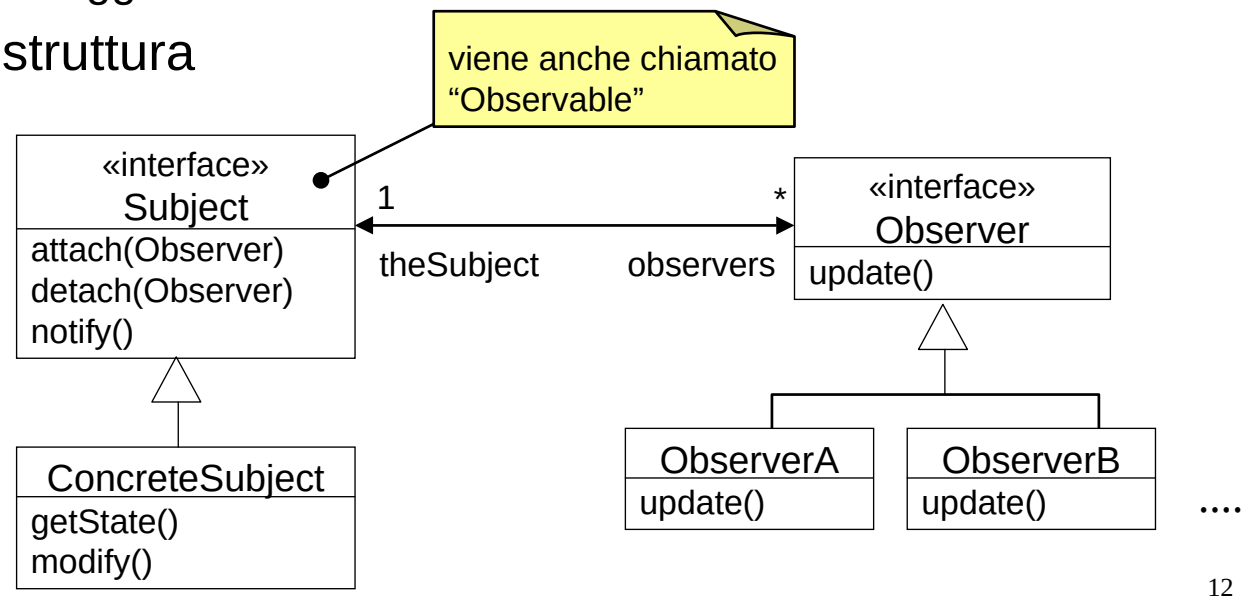
Decorator

- object diagram per una finestra di testo dotata di bordo, il testo e il bordo sono zoommati e su tale finestra zoommata si può fare scrolling



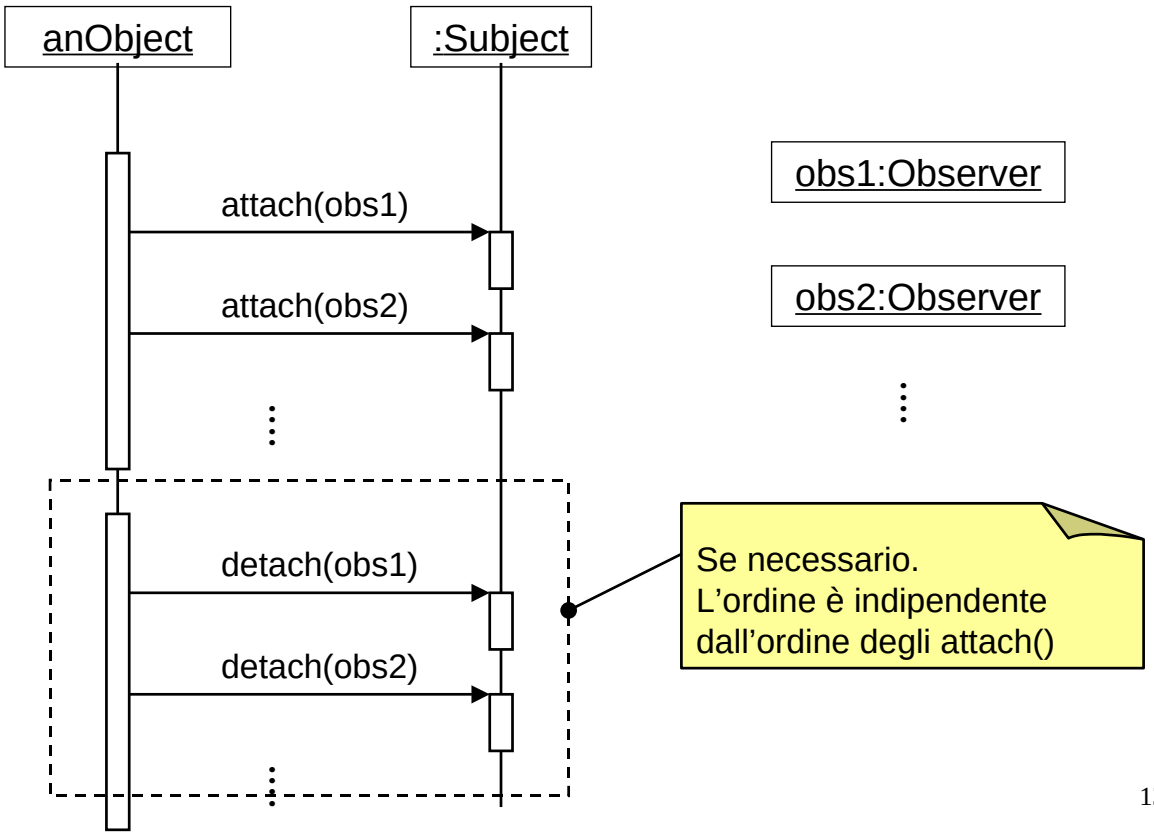
Observer

- intento
  - quando un oggetto cambia stato un certo numero di altri oggetti da esso dipendenti vengono notificati perchè si aggiornino
- struttura



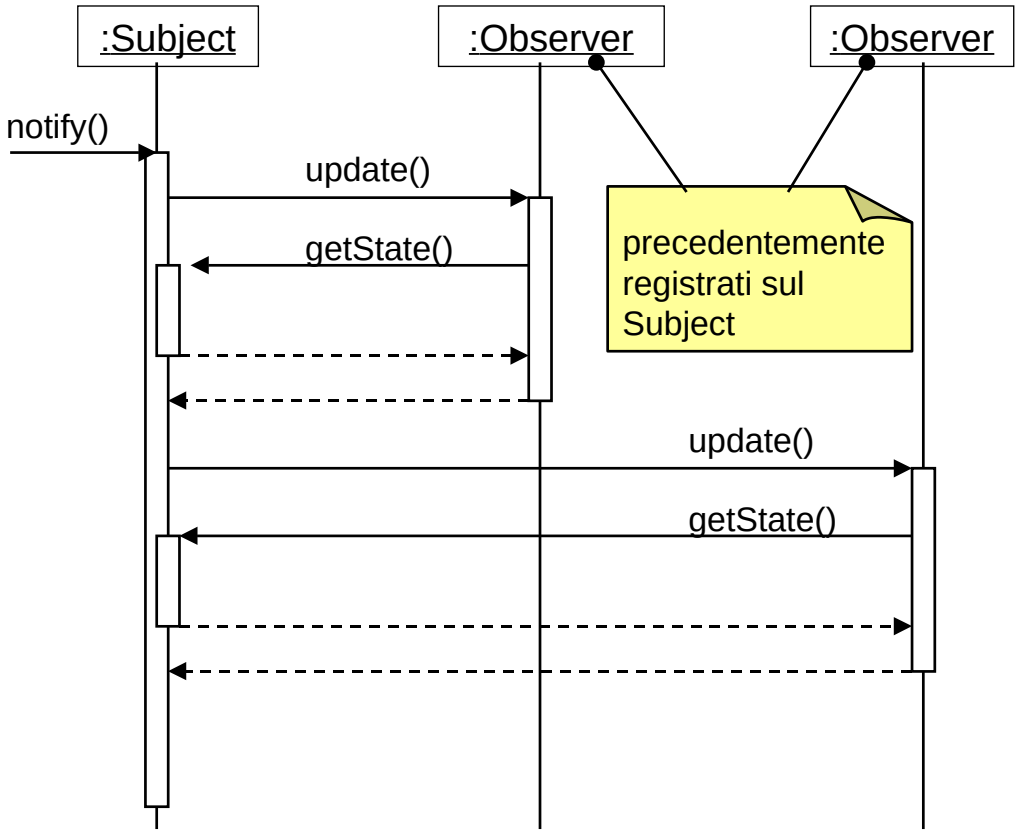
Observer

- interazione: setup e teardown degli observer



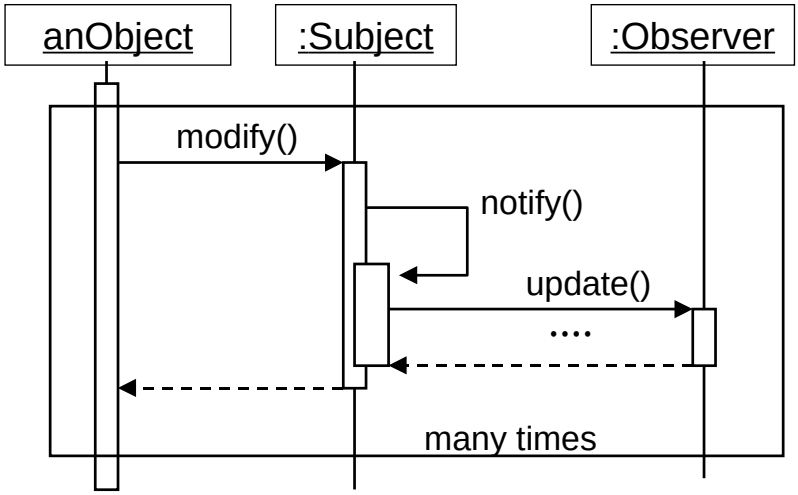
Observer

- interazione: notify



Observer

- interazione
  - notify() chiamato ad ogni modifica del Subject



Observer

- interazione
  - notify() chiamato dopo una sequenza di modifiche del Subject