COMP2511

Object-Oriented Design and Programming

Design Patterns

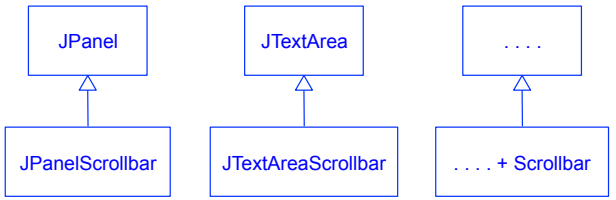
Wayne Wobcke

w.wobcke@unsw.edu.au

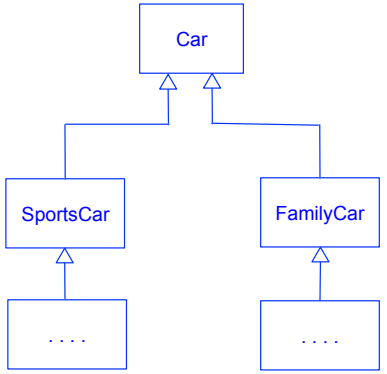
Today's Lecture

- Decorator Pattern
- Composite Pattern
- Combining Patterns

Motivation: Inflexible Hierarchies



Motivation: Complex Hierarchies



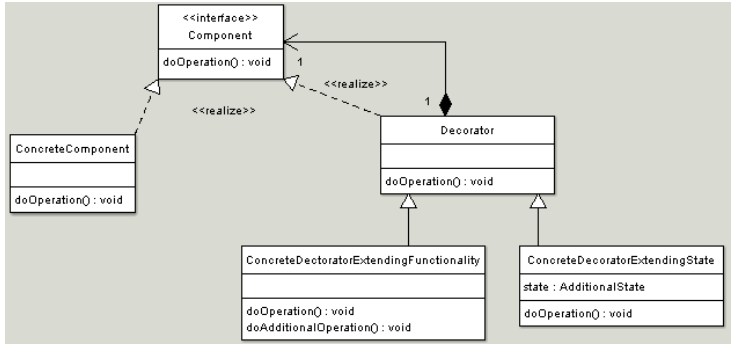
Decorator Pattern

- Motivation
  - Need a way to enhance an object's functionality dynamically at runtime, yet use the enhanced object the same way
- Intent
  - Have a new class whose objects manage other objects, whose methods provide modified functionality over the other object

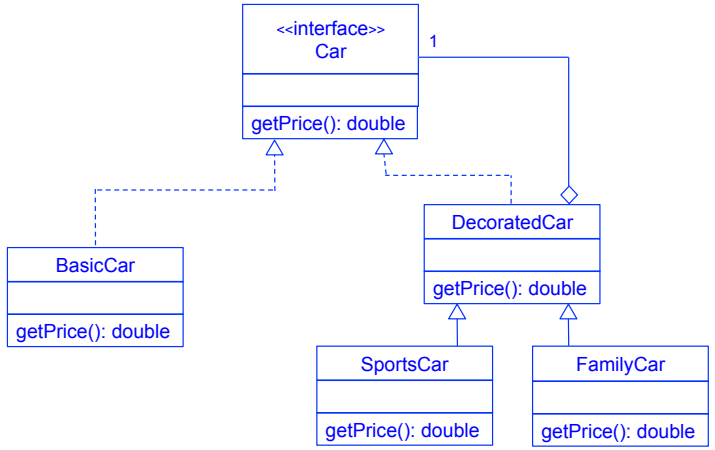
Decorator Pattern Examples

- Graphics windows
  - Adding a scrollbar or border
- File readers
  - Adding a buffer (allows reading lines)
- Can have multiple types of decorations
- Can decorate already decorated objects

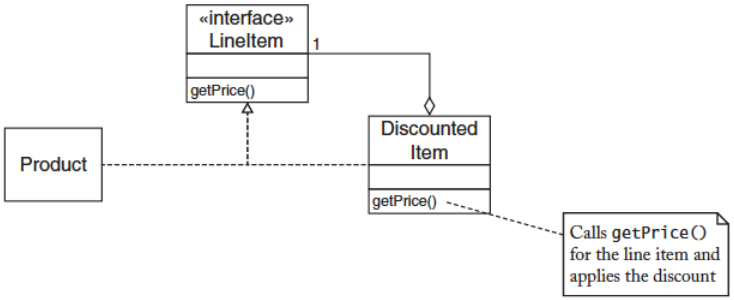
Decorator Pattern Implementation



Decorator Pattern Example



Decorator Pattern Example



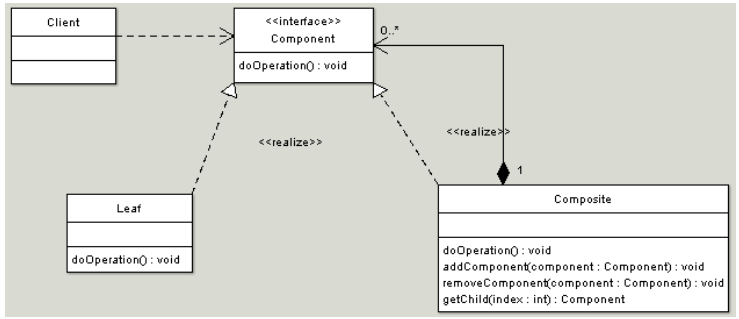
Composite Pattern

- Motivation
  - Need a way to treat collections of objects as if they were a single (combined) whole
- Intent
  - Have a new class whose objects manage a collection of other objects, where to apply a method to the whole requires applying it to each of the objects

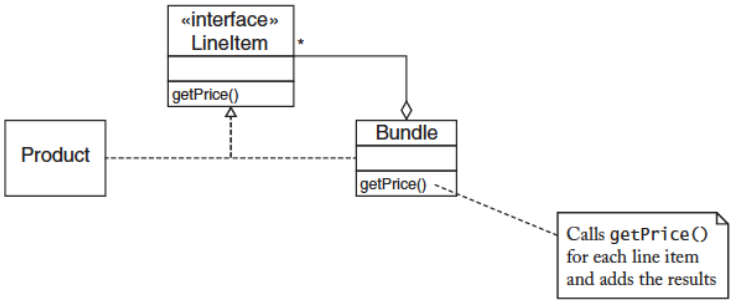
Composite Pattern Examples

- File hierarchies
  - Directories aggregate files and directories
  - Printing a directory requires printing each subdirectory
- Graphics elements
  - Components aggregate (or are composed of?) containers and components
  - Painting a component requires painting each sub-component

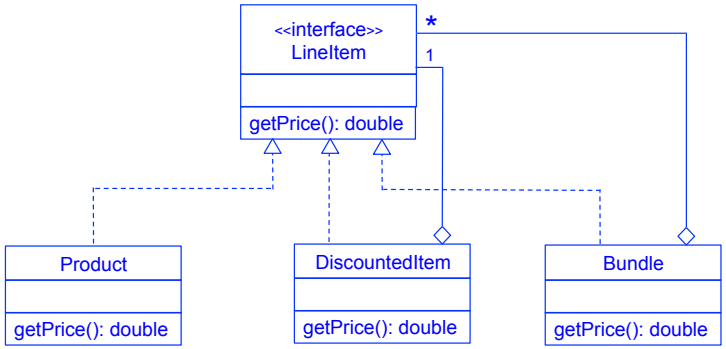
Composite Pattern Implementation



Composite Pattern Example



Combining Patterns



Implementation

```
// DiscountedItem (Decorator Pattern)
private LineItem item;

public double getPrice()  {
    return item.getPrice() * (1 - discount/100);
  }

// Bundle (Composite Pattern)
private ArrayList<LineItem> items;

public double getPrice() {
    double price = 0;
    for (LineItem item : items)
      price += item.getPrice();
    return price;
  }
```


Next Week

- Concurrency
  - Synchronization and Locks