

Design pattern sample

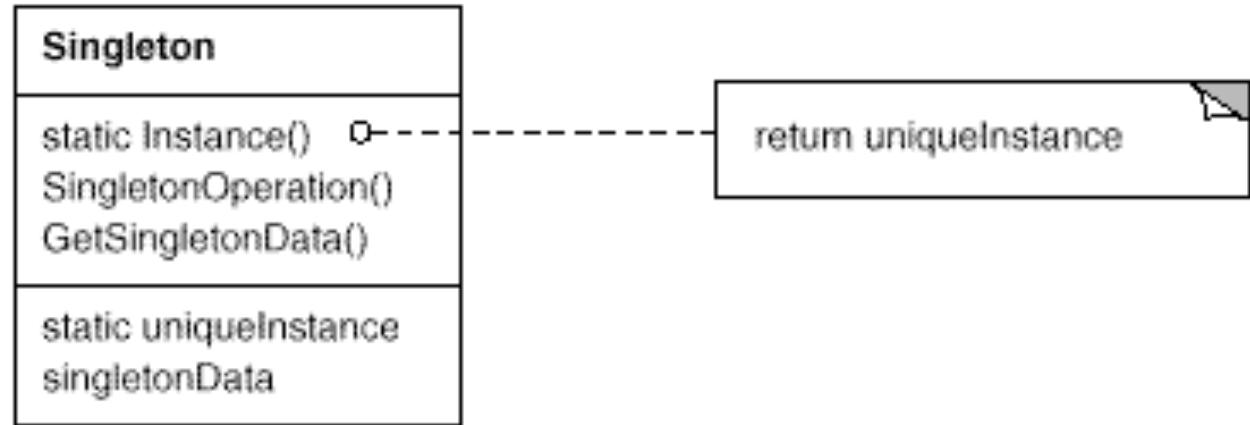
# Case study

- Case 1: Global variables, constants
- Case 2: Switching between DB and Files ?
- Case 3: n-ary tree
- Case 4: Decorator
- Case 5: Simplify the (module) interface
- Case 6: Visitor

# Case 1: Global variables, constants

# Case 1: Global variables, constants

- Singleton



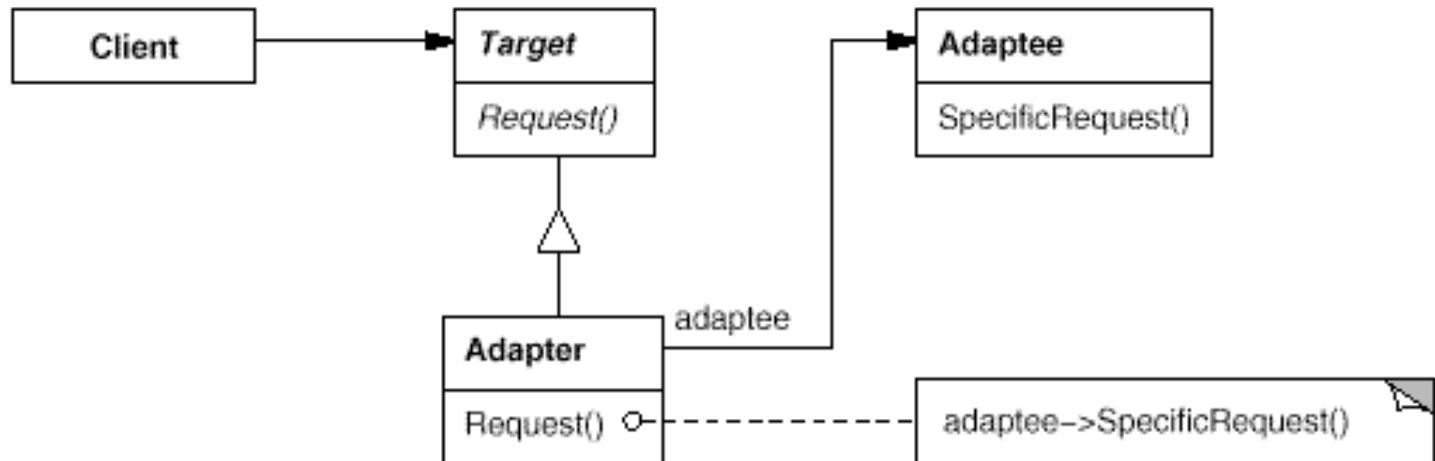
```
public class Singleton {
    private Singleton() {...}
    final private static Singleton instance
        = new Singleton();
    public Singleton getInstance() { return instance; }
}
```

# Case 2: Adaptation

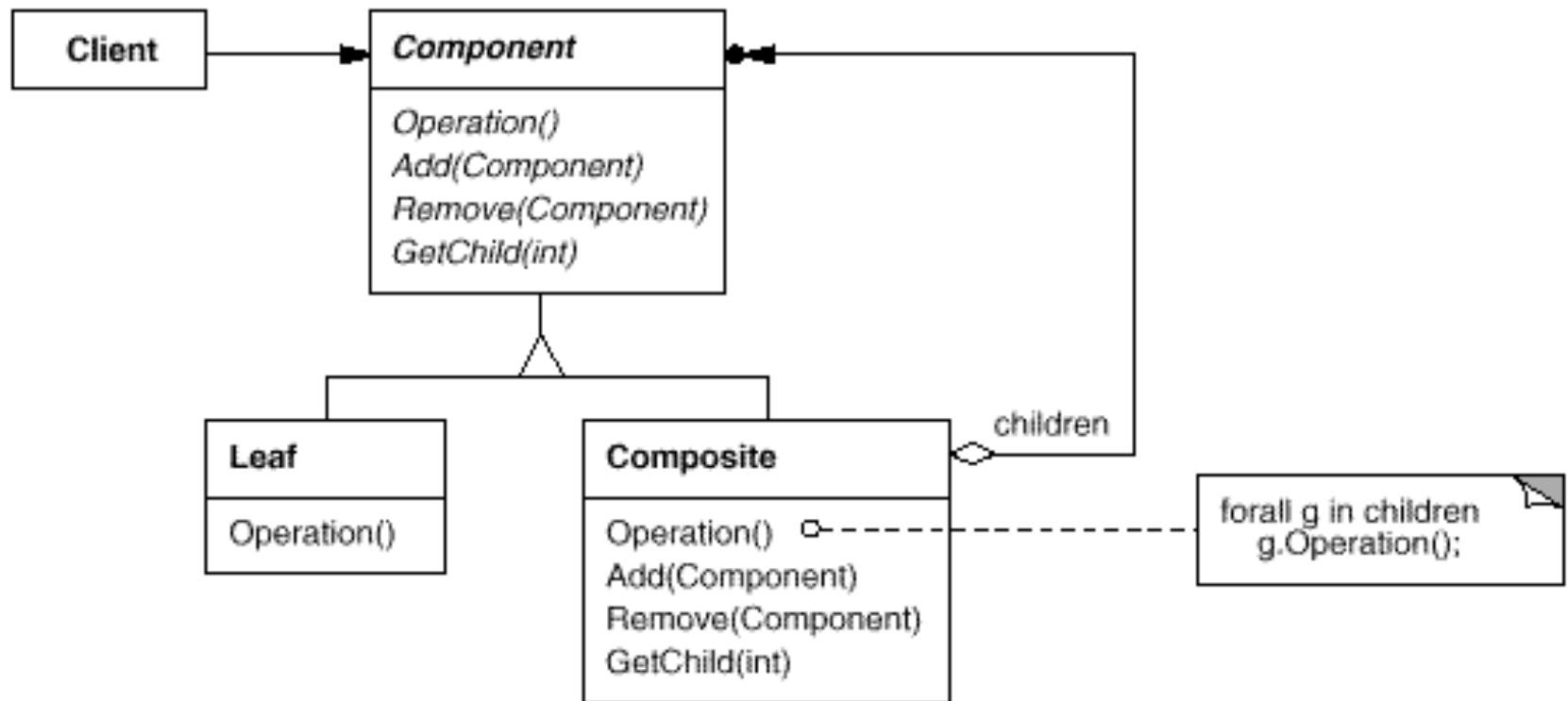
- Situation:
  - You have some code you want to use for a program
  - You can't incorporate the code directly (e.g. you just have the .class file, say as part of a library)
  - The code does not have the interface you want
    - Different method names
    - More or fewer methods than you need
- To use this code, you must adapt it to your situation

# Case 2: Adaptation

- Adapter



# Case 3: n-ary tree

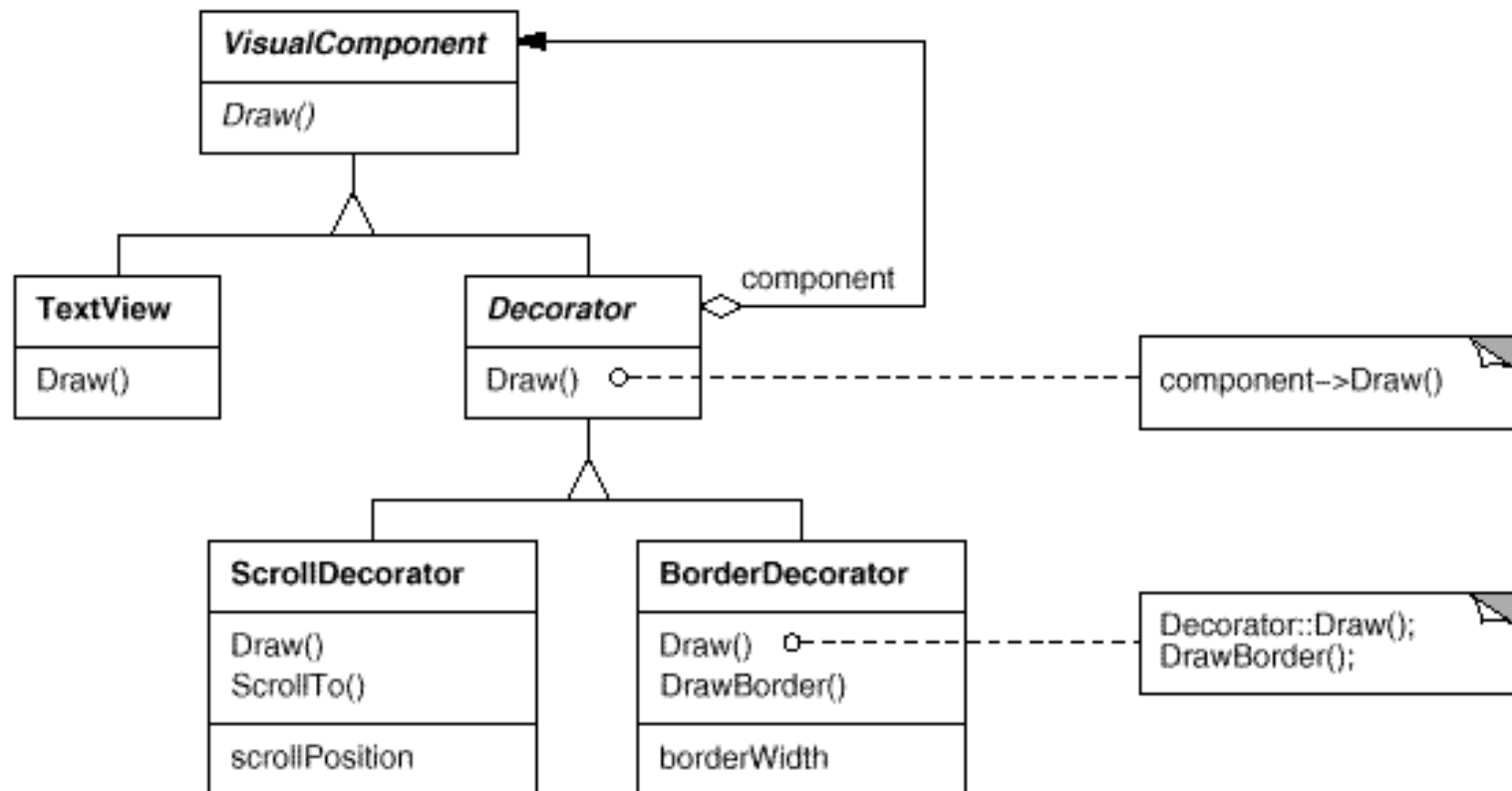


# Case 3: n-ary tree

- class Node {
  - String fullName;
  - void travel();
- }
- class FileNode extends Node {
  - void Travel() {System.out.println(fullName); }
- }
- class FolderNode extends {
  - List<Node> children;
  - ...
  - void Travel() {
    - System.out.println(fullName);
    - for (int i=0; i<children.size(); i++)
      - (children.get(i)).travel();
  - }
- }



# Case 4: Decorator



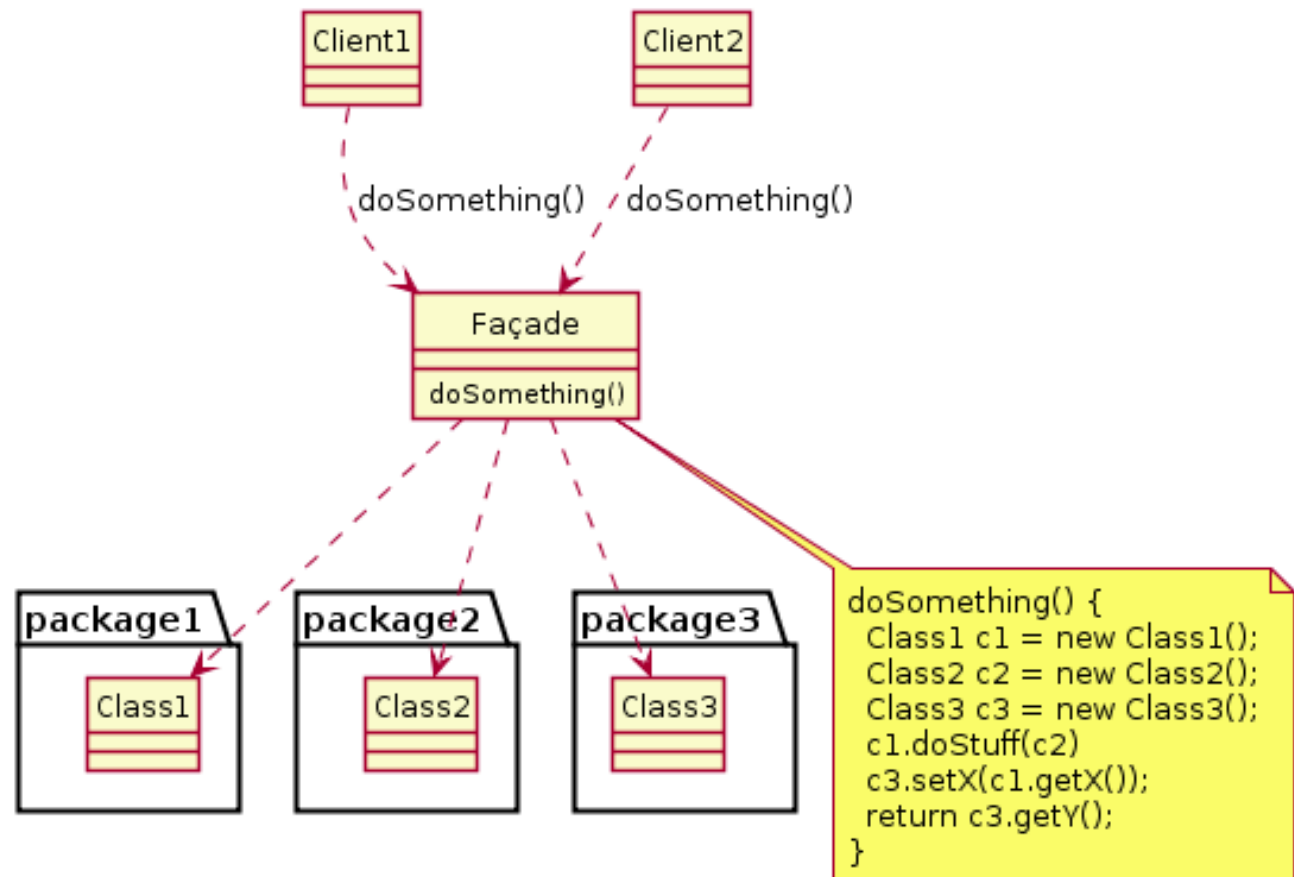
# Case 4: Decorator

- Décor a car example

# Case 5: Simplify the interface

# Case 5: Simplify the interface

- Façade

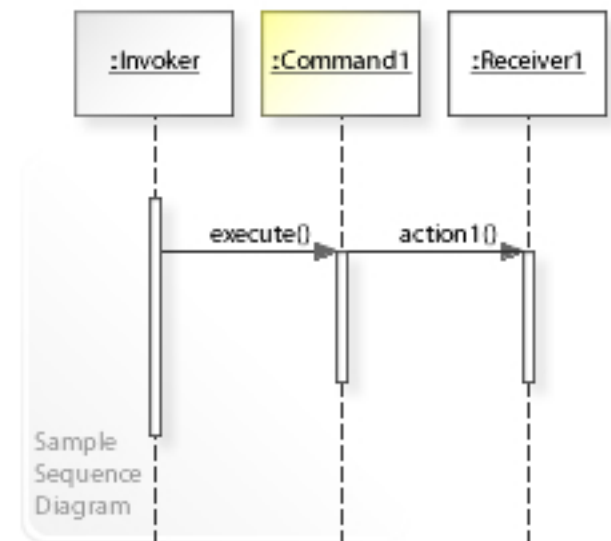
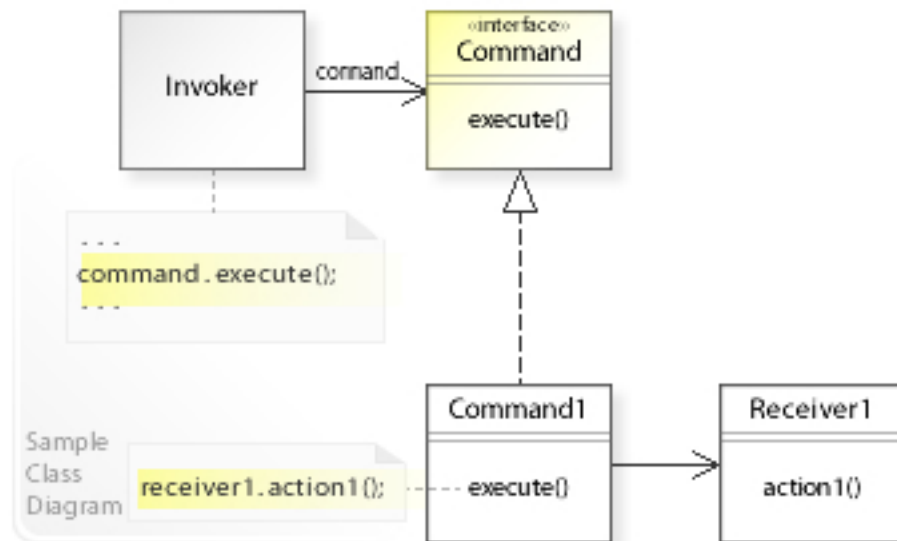


# Case 5: Simplify the interface

- Façade
  - turnAllLights() {
    - for ... {
      - Light aBulb = getBulb...
      - aBulb.turnOn();
    - }
  - }

# Case 5: Simplify the interface

- Command



# Case 5: Simplify the interface

- Command
  - void undo() {
    - Command cmd = the\_stack.pop();
    - cmd.execute();
  - }

# Case 6: Visitor

- Travel a folder
  - Show name of all files/folders, recursively
  - Check viruses on all files, recursively



# Case 6: Visitor

- interface Visitor {
  - public void visit(Node x);
- }
- class Node {
  - String fullName;
  - void travel(Visitor v);
- }
- Visitor 1:
  - visit(Node x) {
    - System.out.println(x.fullName);
  - }
- Visitor 2:
  - visit (Node x) {
    - if (x.isFile())
      - Scan\_for\_virus(x.fullName);
  - }