

软件体系结构

Zhenyan Ji

— Beijing Jiaotong University —

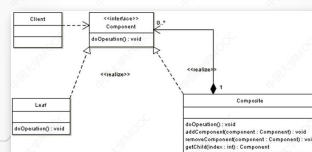
结构型模式--组合模式

组合模式

» 目的

- 将对象组织成树结构以表示“部分 - 整体”的层次结构。
- 组合模式让client程序以统一的方式对待单个对象和组合对象。

组合模式



组合模式

» 参与者

- **Client**
Client使用Component接口操作层次结构中的对象。
- **Component**
Component是Leaf节点和Composite节点的抽象，它定义了组合中对象必须实现的接口。

组合模式

» Leaf

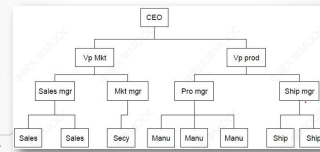
Leaf是没有子节点的对象，其实现由组件接口声明的服务。

组合模式

• Composite

Composite 存储子组件，实现 Component 接口定义的方法。除了实现 Component 接口定义的方法。此外，Composite 提供添加、删除以及获得组件 Component 的方法。

例子



例子

```
public class Employee {
    String name;
    float salary;
    Vector subordinates;
    //-----
    public Employee(String _name, float _salary) {
        name = _name;
        salary = _salary;
        subordinates = new Vector();
    }
}
```

例子

```
public float getSalary() {
    return salary;
}
public String getName() {
    return name;
}
public void add(Employee e) {
    subordinates.addElement(e);
}
```

例子

```
public void remove(Employee e) {
    subordinates.removeElement(e);
}
//get a list of employees of a given supervisor
public Enumeration elements() {
    return subordinates.elements();
}
```

例子: 通用接口

» 通用接口

```
• //returns a sum of salaries for the employee and his subordinates
public float getSalaries() {
    float sum = salary; //this one's salary
    //add in subordinates salaries
    for(int i = 0; i < subordinates.size(); i++) {
        sum +=
        ((Employee)subordinates.elementAt(i)).getSalaries();
    }
    return sum;
}
```

例子:构建一棵Employee树

» 构建一棵Employee树

```
boss = new Employee("CEO", 200000);
boss.add(new Employee("Marketing VP", 100000));
boss.add(new Employee("Production VP", 100000));
marketVP.add(new Employee("Sales Mgr", 50000));
marketVP.add(new Employee("Advt Mgr", 50000));
//add salesmen reporting to Sales Manager
for (int i=0; i<5; i++)
    salesMgr.add(new Employee("Sales "+new
Integer(i).toString(), 30000.0F+(float)(Math.random()-0.5)*10000));
```

例子

```
advMgr.add(new Employee("Secy", 20000));
prodVP.add(new Employee("Prod Mgr", 40000));
prodVP.add(new Employee("Ship Mgr", 35000));
//add manufacturing staff
for (int i = 0; i < 4; i++)
    prodMgr.add(new Employee("Manuf "+new
Integer(i).toString(),
25000.0F+(float)(Math.random()-0.5)*5000));
//add shipping clerks
for (int i = 0; i < 3; i++)
    shipMgr.add(new Employee("ShipClrk "+new
Integer(i).toString(),
20000.0F+(float)(Math.random()-0.5)*5000));
```