

# 组合模式

## 概述

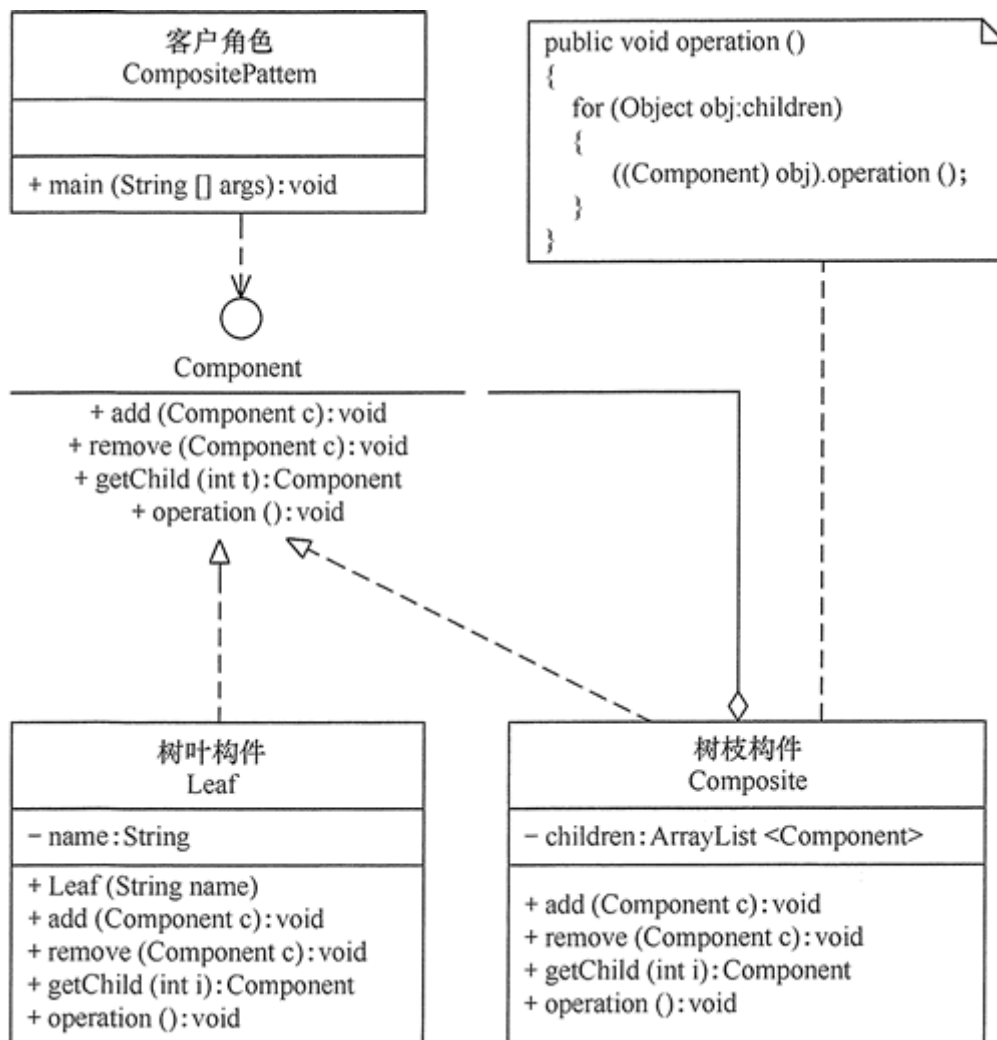
---

组合（Composite Pattern）模式的定义：有时又叫作整体-部分（Part-Whole）模式，它是一种将对象组合成树状的层次结构的模式，是类和对象的组织结构，用来表示“整体-部分”的关系，使用户对单个对象和组合对象具有一致的访问性，属于结构型**设计模式**。

## 图例

---

结构性设计模式：是介绍如何将对象和类组装成一个较大的结构



**Leaf:叶子节点，完成实际工作**

**Composite:组合体，它可以包含更多组合体和叶子节点，将工作委派给子元素**

## 案例代码

叶子节点和composite都要实现同样的功能，即实现component接口，这个接口定义它们要有的方法

```
1 interface Counter(){
2     //计数
3     //实现计数统计功能
4     int count();
```

```
5  }
6
7
8  //叶子结点
9
10 class City implements Counter(){
11     //实际统计普查它的人口数量
12     private int sum=0;
13
14     public City(int sum){
15         this.sum=sum;
16     }
17     @Override
18     public int count()
19         return sum;
20 }
21
22
23 //容器
24 //也要实现计数功能，交给下级处理
25 class Composite implements Counter {
26     private List<Counter> counterList= new
    ArrayList<>();
27
28     public void add (Counter counter){
29         counterList.add (counter);
30     }
31
32     public void delete(Counter counter){
```

```
33     counterList.remove(counter);
34 }
35
36 public List<Counter> getchild(){
37     return counterList;
38 }
39
40 @Override
41 public int count() {
42     int sum = 0;
43     for (Counter counter : counterList){
44         sum += counter.count();
45     }
46     return sum;
47 }
48 }
49
50
51
52 public class CompositePattern
53 public static void main(String[][] args)
54 {
55     Composite china = new Composite();
56     china.add (new City( sum: 1000) ); //北京
57     china.add (new City ( sum: 2000)); //上海
58     Composite HeNan = new Composite:
59     HeNan.add (new City( sum: 3000)); //焦作市
60     HeNan.add (new City( sum: 3000) ); //郑州市
61     china.add (HeNan)
62     System.out.println(china.count());
```

```
63  }
64  }
65  /*
66  China
67      北京
68      上海
69      河南
70          焦作
71          郑州
72  */
73
```