

学习笔记：设计模式树结构(合成模式 模板模式) PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/264/2021\\_2022\\_\\_E5\\_AD\\_A6](https://www.100test.com/kao_ti2020/264/2021_2022__E5_AD_A6)

[\\_E4\\_B9\\_A0\\_E7\\_AC\\_94\\_E8\\_c104\\_264316.htm](https://www.100test.com/kao_ti2020/264/2021_2022__E5_AD_A6_E4_B9_A0_E7_AC_94_E8_c104_264316.htm) 这是一次合成模式 模板模式的尝试这是一个树结构的设计。该组件只注重设计一个可扩展的树结构，并不提供创建树和遍历树的方式。用户可以针对这个树结构自己写实现。该组件使用了合成模式 模板模式来设计。树的结构是根据透明的合成模式来设计的，这样使得客户端可以把树叶和树枝对象当成同一类对象处理。而树叶、树枝的抽象和具体实现之间的关系是通过模板模式来设计的，这样使得客户端可以很简单地将结点的具体对象置换掉。结构说明：TreeComponent:树结点的高度抽象，定义树结构的一些方法，所有树叶结点和树枝结点都上转成TreeComponentTreeComposite:抽象树枝结点。TreeLeaf:抽象树叶结点。PermissionComposite：具体树枝结点

。PermissionLeaf:具体树叶结点。PermissionBean:结点的具体内容，也就是结点对象所包含的成员变量。就是一个简单的JavaBean注意：如果要为这个树结构设计一个遍历的方式，也许可以用迭代子模式为TreeComponent设计一个外禀子的迭代器。(有空再考虑^\_^)树结点的高度抽象，定义树结构的一些方法，所有树叶结点和树枝结点都上转

```
成TreeComponentpublic interface TreeComponent { /** * 添加子  
结点方法 * @param component * @return */ boolean  
add(TreeComponent component). /** * 删除子结点方法 *  
@param component * @return */ boolean remove(TreeComponent  
component). /** * 返回子结点方法 * @return */ List
```

```
components(). /** * 统计该结点(包括该结点)下, 所有结点的  
数量 * @return */ int count(). /** * 判断当前结点是否是树叶结  
点 * @return */ boolean isLeaf(). /** * 返回该结点 * @return */  
TreeComponent getComponent(). /** * 返回当前结点名 *  
@return */ String getName(). /** * 统计该结点(包括该结点)下,  
所有树叶结点的数量 * @return */ int leafCount().} 100Test 下载  
频道开通, 各类考试题目直接下载。详细请访问  
www.100test.com
```