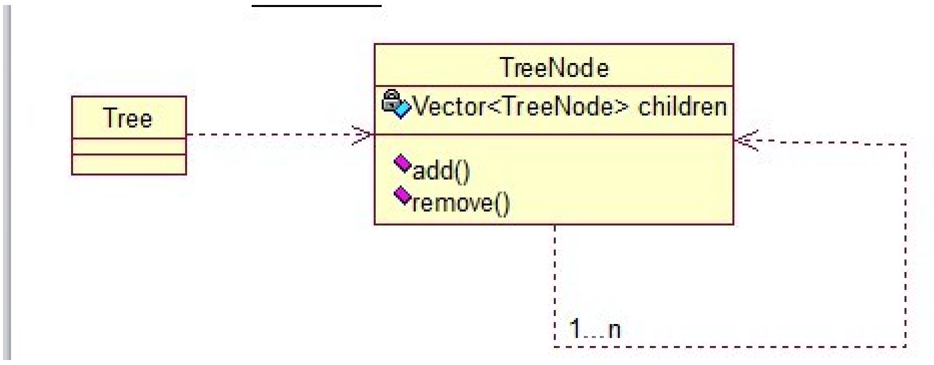
组合模式（Composite）

组合模式有时又叫**部分-整体**模式在处理类似树形结构的问题时比较方便，看看关系图：



直接来看代码：

[java] view plaincopy

```java
public class TreeNode {

    private String name;
    private TreeNode parent;
    private Vector<TreeNode> children = new Vector<TreeNode>();

    public TreeNode(String name){
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public TreeNode getParent() {
        return parent;
    }

    public void setParent(TreeNode parent) {
        this.parent = parent;
    }

    //添加孩子节点
    public void add(TreeNode node){
        children.add(node);
    }

    //删除孩子节点
    public void remove(TreeNode node){
        children.remove(node);
    }

    //取得孩子节点
    public Enumeration<TreeNode> getChildren(){
        return children.elements();
    }
}
```

[java] view plaincopy

```java
public class Tree {

    TreeNode root = null;

    public Tree(String name) {
        root = new TreeNode(name);
    }

    public static void main(String[] args) {
        Tree tree = new Tree("A");
        TreeNode nodeB = new TreeNode("B");
        TreeNode nodeC = new TreeNode("C");

        nodeB.add(nodeC);
        tree.root.add(nodeB);
        System.out.println("build the tree finished!");
     }
}
```

使用场景：将多个对象组合在一起进行操作，常用于表示树形结构中，例如二叉树，数等。