

用composite模式写的一个二叉树的例子 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E7_94_A8co
mposi_c104_144648.htm 1，Component 是抽象组件Tree 和Leaf

继承Component private String name. //树或叶子的名称
addChild(Component leftChild,Component rightChild). //给一个树上加上一个左孩子，一个右孩子
getName(){return name.}
getTreeInfo(){ //得到树或叶子的详细信息
getLength(). //得到树的高度
2，Tree 二叉树，一个左孩子，一个右孩子
3，Leaf 是叶子节点4，Test 是测试节点

```
/** Component.java  
*****  
/package binarytree.  
public abstract class Component  
{private String name.  
public abstract Component  
addChild(Component leftChild,Component rightChild).  
public String getName(){return name.}  
public void getTreeInfo(){  
public abstract int getLength().}  
/** Leaf.java *****  
/package  
binarytree.  
public class Leaf extends Component  
{private String name.  
private Component leaf=null.  
public Leaf(String name)  
{this.name=name.}  
public Component addChild(Component  
leftChild,Component rightChild){return this.}  
public String  
getName(){return name.}  
public int getLength() {return 1.}  
public  
static void main(String[] args) {}  
/** Tree.java  
*****  
/package binarytree.  
public class Tree extends  
Component {private String name.  
private Component  
leftChild.  
private Component rightChild.  
public Tree(String  
name,Component leftChild,Component rightChild)  
{this.name=name.  
this.leftChild=leftChild.  
this.rightChild=rightChild
```

```

}public Tree(String name)
{this.name=name.this.leftChild=null.this.rightChild=null.}public
Component addChild(Component leftChild,Component
rightChild){this.leftChild=leftChild.this.rightChild=rightChild.retur
n this.}public String getName(){return name.}public void
getTreeInfo() //得到树或叶子的详细信息//先打印自己的名字
, 再遍历左孩子 , 再遍历右孩子//如果左孩子或右孩子是树
, 递归调用{System.out.println(" this trees name is "
getName()).if(this.leftChild instanceof
Leaf){System.out.println(getName() "s left child is "
this.leftChild.getName() ",it is a Leaf").}if(this.leftChild instanceof
Tree){System.out.println(getName() "s left child is "
this.leftChild.getName() ",it is a
Tree").this.leftChild.getTreeInfo().}if(this.leftChild==null){System.
out.println(getName() "s left child is a null").}if(this.rightChild
instanceof Leaf){System.out.println(getName() "s right child is "
this.rightChild.getName() ",it is a Leaf").}if(this.rightChild
instanceof Tree) {System.out.println(getName() "s right child is "
this.rightChild.getName() ",it is a
Tree").this.rightChild.getTreeInfo().}if(this.rightChild==null){Syste
m.out.println(getName() "s right child is a
null").}//System.out.println(getName() "s 高度是 "
getLength()).}public int getLength() {//比较左孩子或右孩子的高
度 , 谁大 , 1 返回// 空孩子的处理if(this.leftChild==null)
{if(this.rightChild==null)return 1.elsereturn
this.rightChild.getLength() 1.}else {if(this.rightChild==null) {return

```

```
this.leftChild.getLength() 1.)else
{if((this.leftChild.getLength())>=(this.rightChild.getLength()))return
this.leftChild.getLength() 1.elsereturn this.rightChild.getLength()
1.}}public static void main(String[] args) {} /** Test.java 测试类
***** */package binarytree.public class Test { public Test()
{}public static void main(String[] args) {Component tree=new
Tree("luopeng").Component leaf_child=new
Leaf("luopeng1").Component right_child=new
Leaf("luopeng2").tree=tree.addChild(leaf_child,right_child)//tree=t
ree.addRightChild(right_child).tree.getTreeInfo().Component
tree1=new
Tree("luopeng2").tree1.addChild(tree,leaf_child).tree1.getTreeInfo(
).Component tree2=new
Tree("luopeng3").tree2.addChild(tree,null).tree2.getTreeInfo().Co
mponent tree4=new
Tree("luopeng4").tree4.addChild(null,tree).tree4.getTreeInfo().Syst
em.out.println(tree4.getName() "的高度是 " tree4.getLength()).}
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
```