Composite模式及其在JSF中的应用 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/167/2021_2022_Composite_E6_c104_167340.htm 一 学习背景 在学习关于JSF组件时涉及到了composite模式，于是就查看一些资料，以下是自己对这种模式的理解。二 自己整理的一些资料（见参考资料）1.composite模式意在组成任意复杂度的整体--部分组件层次结构，同时将单个组件或复合组件视为统一的接口。树形组织结构就是其中一种表现形式。树形结构中有叶子结点和非叶子结点（根结点是特例），非叶子结点可以添加，删除（add(),0delete()）子结点，获取子结点（getChild()），叶子结点没有；此外树结构的所有节点还有共同的操作（operator()）.用户界面通常由两种基本类型的组件构造：基本组件和容器组件，容器组件可以在其内部嵌套任意数目的组件，而基本组件则不行。使用这两种组件类型，开发者可以建立更强大的组件，进而创建多姿多彩的用户界面。但是在与复杂的组件层次结构打交道时，必须在容器组件和基本组件之间进行区分，比较麻烦，composite提供了一种解决方案。适用它的情况：a. 要表现“ 部分-整体 ”的层次结构时b. 希望在事件组件层次中，同等对待复合组件与单个组件。2. 通过下面的示例来理解示例1： 基类shape 类有两个派生类Circle和Square（相当于叶子结点或者是单个组件），第三个派生类CompositeShape是个组合体（相当于非叶子结点或者是容器组件），它持有一个含有多个shape实例的列表，当调用CompositeShape中的draw()时，它就把这个方法委托给列表中的每一个实例。 对于系统而言，一个CompositeShape实

例就像是一个独立的shape,可以把它传给使用shape的方法或者对象。实际上，它只是一组shape实例的proxy.程序：Shape.java: Public interface Shape { Public void draw(). } CompositeShape.java: [code]Public class CompositeShape implements Shape { private Vector Comshape = new Vector(). public void add(Shape shape) { Comshape.add(shape). } Public void draw() { for( int i = 0. i Shape shape = (Shape) comshape.elementAt(i). Shape.draw(). } } } 示例2： 抽象类Equipment就是Component定义，代表着组合体类的对象们,Equipment中定义几个共同的方法。package com.interf. public abstract class Equipment { private String name. private double netPrice. private double discountPrice. public Equipment(String name) { this.name = name. } public abstract double netPrice(). public abstract double discountPrice(). } Disk是组合体内的一个对象，或称一个部件，这个部件是个单独元素( Primitive)。Disk.java:package implEquip. import com.interf.Equipment. public class Disk extends Equipment { public Disk(String name) { super(name). // TODO Auto-generated constructor stub } //定义Disk实价为1 public double netPrice() { return 1.. } //定义了disk折扣价格是0.5 对折。 public double discountPrice() { return .5. } } 还有一种可能是，一个部件也是一个组合体，就是说这个部件下面还有儿子，这是树形结构中通常的情况，应该比较容易理解。现在我们先要定义这个组合体：CompsiteEquipment.java:package implEquip. import java.util.ArrayList. import java.util.Iterator. import java.util.List. import java.util.NoSuchElementException. import

com.interf.Equipment. public class CompositeEquipment extends Equipment { private int i=0. // 定义一个Vector 用来存放儿子 private List equipment = new ArrayList(). public CompositeEquipment(String name) { super(name). // TODO Auto-generated constructor stub } public boolean add(Equipment equipment) { this.equipment.add(equipment). return true. } public double netPrice() { double netPrice=0.. Iterator iter=equipment.iterator(). while(iter.hasNext()) netPrice =((Equipment)iter.next()).netPrice(). return netPrice. } public double discountPrice() { double discountPrice=0.. Iterator iter=equipment.iterator(). while(iter.hasNext()) discountPrice =((Equipment)iter.next()).discountPrice(). return discountPrice. } // 注意这里，这里就提供用于访问自己组合体内的部件方法。 // 上面dIsk 之所以没有，是因为Disk是个单独(Primitive)的元素. public Iterator iter() { return equipment.iterator() . } // 重载Iterator方法 public boolean hasNext() { return i// 重载Iterator方法 public Object next(){ if(hasNext()) return equipment.get(i ). else throw new NoSuchElementException(). } } 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