

Java Swing 树状组件JTree的使用方法【图】

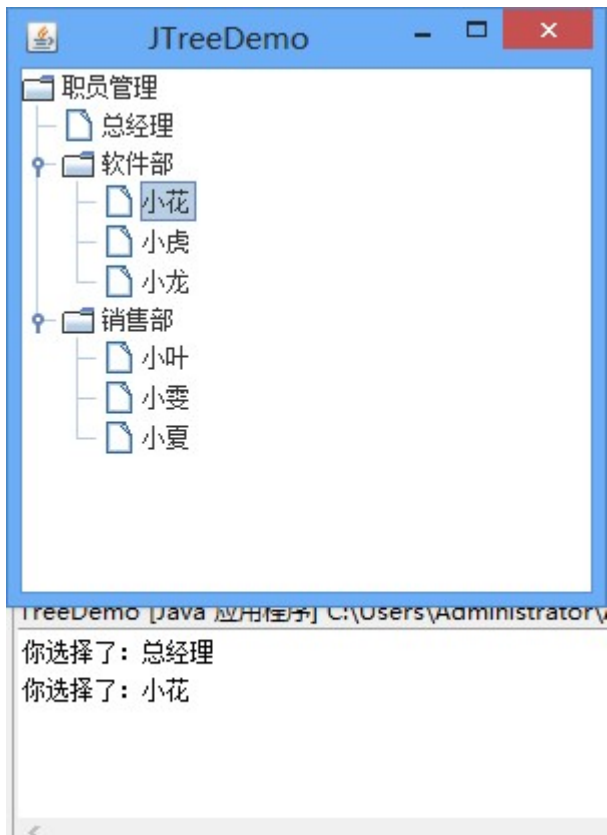
树中特定的节点可以由 `TreePath`（封装节点及其所有祖先的对象）标识，或由其显示行（其中显示区域中的每一行都显示一个节点）标识。*展开* 节点是一个非叶节点（由返回 `false` 的 `TreeModel.isLeaf(node)` 标识），当*展开* 其所有祖先时，该节点将显示其子节点。*折叠* 节点是隐藏它们的节点。*隐藏* 节点是位于折叠祖先下面的节点。所有*可查看* 节点的父节点都是可以展开的，但是可以显示它们，也可以不显示它们。*显示* 节点是可查看的并且位于可以看到它的显示区域。

构造方法摘要
<div><u>JTree()</u> 返回带有示例模型的 JTree。</div>
<div><u>JTree(Hashtable<?,?> value)</u> 返回从 Hashtable 创建的 JTree，它不显示根。</div>
<div><u>JTree(Object[] value)</u> 返回 JTree，指定数组的每个元素作为不被显示的新根节点的子节点。</div>
<div><u>JTree(TreeModel newModel)</u> 返回 JTree 的一个实例，它显示根节点 – 使用指定的数据模型创建树。</div>
<div><u>JTree(TreeNode root)</u> 返回 JTree，指定的 TreeNode 作为其根，它显示根节点。</div>
<div><u>JTree(TreeNode root, boolean asksAllowsChildren)</u> 返回 JTree，指定的 TreeNode 作为其根，它用指定的方式显示根节点，并确定节点是否为叶节点。</div>
<div><u>JTree(Vector<?> value)</u> 返回 JTree，指定 Vector 的每个元素作为不被显示的新根节点的子节点。</div>

```
public JTree(TreeNode root)
```

返回 `JTree`，指定的 `TreeNode` 作为其根，它显示根节点。默认情况下，树可以将叶节点定义为不带子节点的任何节点。

下面用构造方法 `public JTree(TreeNode root)` 创建实例



```
import javax.swing.JFrame;
import javax.swing.JTree;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.DefaultMutableTreeNode;

public class TreeDemo {
    public static void main(String[] args) {

        // 创建没有父节点和子节点、但允许有子节点的树节点，并使用指定的用户对象对它进行初始化。
        // public DefaultMutableTreeNode(Object userObject)
        DefaultMutableTreeNode node1 = new DefaultMutableTreeNode("软件部");
        node1.add(new DefaultMutableTreeNode(new User("小花")));
        node1.add(new DefaultMutableTreeNode(new User("小虎")));
        node1.add(new DefaultMutableTreeNode(new User("小龙")));

        DefaultMutableTreeNode node2 = new DefaultMutableTreeNode("销售部");
        node2.add(new DefaultMutableTreeNode(new User("小叶")));
        node2.add(new DefaultMutableTreeNode(new User("小雯")));
        node2.add(new DefaultMutableTreeNode(new User("小夏")));

        DefaultMutableTreeNode top = new DefaultMutableTreeNode("职员管理");

        top.add(new DefaultMutableTreeNode(new User("总经理")));
        top.add(node1);
        top.add(node2);
```

```
final JTree tree = new JTree(top);

JFrame f = new JFrame("JTreeDemo");

f.add(tree);

f.setSize(300, 300);

f.setVisible(true);

f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// 添加选择事件
tree.addTreeSelectionListener(new TreeSelectionListener() {

    @Override
    public void valueChanged(TreeSelectionEvent e) {
        DefaultMutableTreeNode node = (DefaultMutableTreeNode) tree
            .getLastSelectedPathComponent();

        if (node == null)
            return;

        Object object = node.getUserObject();
        if (node.isLeaf()) {
            User user = (User) object;
            System.out.println("你选择了: " + user.toString());
        }
    }
});

}

class User {
    private String name;

    public User(String n) {
        name = n;
    }

    // 重点在toString, 节点的显示文本就是toString
    public String toString() {
        return name;
    }
}
```

分类: Java

标签: JTree

好文要顶

关注我

收藏该文





ImWiki

粉丝 - 148 关注 - 17

3

0

+加关注

« 上一篇: Myeclipse 实现java 可视化开发【图】

» 下一篇: Java Swing 绝对布局管理方法, null布局【图】

posted @ 2013-02-08 12:10 ImWiki 阅读(56337) 评论(1) 编辑 收藏 举报

