

Java内部类:如何在内部类中返回外部类对象 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_Java_E5_86_85_E9_83_A8_c104_145427.htm

Nasted Class 的介绍本文讨论的不是内部类的概念，而是具体使用的一个场景 - 如何在内部类中返回外部对象看一段代码

```
import java.util.LinkedList;
import java.util.List;
public class OuterClass{
    private List listeners = new LinkedList();
    public void addListeners(Listener listener) {
        this.listeners.add(listener);
    }
    private class InnerClass {
        public void publish() { //将事件发布出去
        }
        for(int i=0; i < listeners.size(); i++) {
            Listener listener = (Listener) listeners.get(i);
            listener.receiveEvent(OuterClass.this);
        }
    }
    public void execute() {
        InnerClass in = new InnerClass();
        in.publish();
    }
}
public interface Listener{
    public void receiveEvent(OuterClass obj);
}
```

你可能觉得这个例子很别扭，在哪里让你觉得难受呢？其实问题的关键就在于接口Listener的定义，这里需要给receiveEvent方法传递的参数是外部对象！（别激动，下面我会说明需要传递的一个场景）场景在一个GUI系统中，我们要在画板Workspace（Workspace实现了Listener接口）上产生一颗树，但树中的每个节点的产生（绘图）是我们不知道的算法，系统只为我们提供了一些绘图的接口，并返回元素的句柄！看来我们需要"包装"一下这个绘图的句柄Brush（其实我把它叫做笔刷，因为它只知道如何"刷"出图像来，就这点本事！）并对外提供节点Node这样一个通用的类。此时Node与Brush的关系就很微妙了，不过我们可以抛开这些外表，看到Node与Brush其实就是外部类与内部类的关系！ - 第

一步完成了：确定了两者的关系然而，事情没有这么简单，Node类必须处理一些事件，而这些事件理所当然只有Brush能够看懂，而Node根本不知道这样的事件处理过程，现在有两个办法：办法一，让Node实现Brush所有的事件；办法二，把Brush返回回去，让它来处理自己的事件，看来办法二是个好主意，因为我可以不关心事件的种类！ - 第二步完成了：确定了事件处理的责任还没完呢，你肯定不希望画板WorkSpace面对的是绘图的句柄Brush这样的对象，相反你只希望WokSpace只知道Node的存在！ IListener接口中receiveEvent方法的参数定义为OuterClass 就由此而来！ - 第三步完成：接口的定义public interface IListener{public void receiveEvent(OuterClass obj).} 既然说清楚了这个问题（应该比较清楚了吧？）那改如何实现这样一个蹩脚而有无可奈何的设计呢？让我们回忆一下内部类，内部类拥有访问外部类的方法与属性的权限 private OuterClass outer = this. - 这个对外部类的引用就是为内部类的访问准备的 private class InnterClass { public void publish() { //将事件发布出去 for(int i=0;i { IListener listener = (IListener) listeners.get(i). listener.receiveEvent(outer). - 这里不可以返回this，因为this代表的是内部类自己 } } 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