

java内部类的继承问题 (thinkinginjava3thp294) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/284/2021_2022_java_E5_86_85_E9_83_A8_c104_284360.htm

因为内部类的构造器要用到其外围类对象的引用，所以在你继承一个内部类的时候，事情变得有点复杂。问题在于，那个“秘密的”外围类对象的引用必须被初始化，而在被继承的类中并不存在要联接的缺省对象。要解决这个问题，需使用专门的语法来明确说清它们之间的关联：

```
class WithInner { class Inner { Inner(){
System.out.println("this is a constructor in WithInner.Inner"). } } }
public class InheritInner extends WithInner.Inner { // !
InheritInner() {} // Won ' t compile InheritInner(WithInner wi) {
wi.super(). System.out.println("this is a constructor in
InheritInner"). } public static void main(String[] args) { WithInner
wi = new WithInner(). InheritInner ii = new InheritInner(wi). } } 输出
结果为： this is a constructor in WithInner.Inner this is a
constructor in InheritInner 可以看到，InheritInner 只继承自内部
类，而不是外围类。但是当要生成一个构造器时，缺省的构造器并不
算好，而且你不能只是传递一个指向外围类对象的引用。此外，你必
须在构造器内使用如下语法：
enclosingClassReference.super(). 这样才提供了必要的引用，然后程
序才能编译通过。 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
```