# 内部类

## 内部类概述

类A内部声明的类B：A外嵌类；B为内部类

在一个类中定义另一个类叫做内部类，是一个独立的类，在编译时内部类会被编译成独立的.class文件； 只不过在文件名前面冠以外部类的类名和$符号

在A类中定义一个类B --> A$B.class； 内部类是外部类的一个成员，因此内部类可以自由的访问外部类的成员，包括私有成员

## 内部类访问特点

1.内部类可以直接访问外部类的成员，包括私有。（可以看做本类中的一个成员）；内部类之所以能获取到外部类的成员,是因为他能获取到外部类的引用外部类名.this

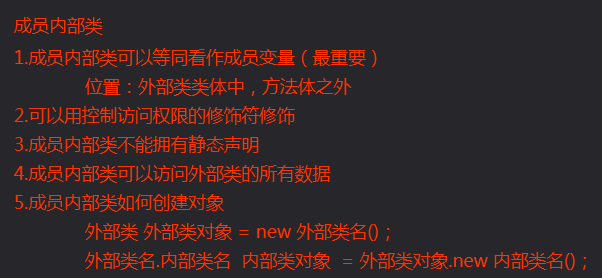
2.内部类中类体不能是类变量，类方法

# 访问内部类属性

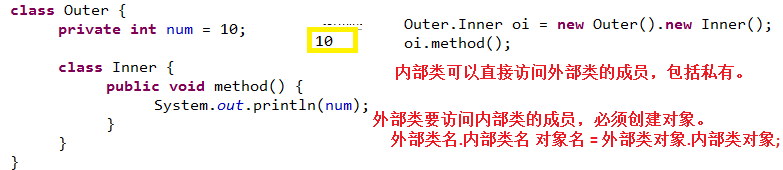


## 成员内部类（类中）

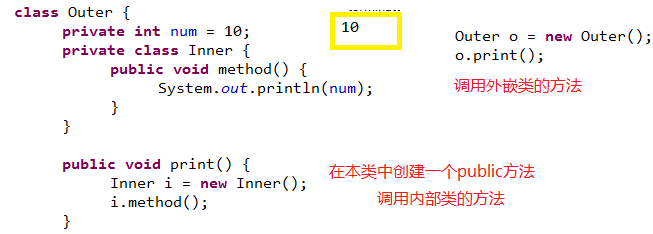
### 概述



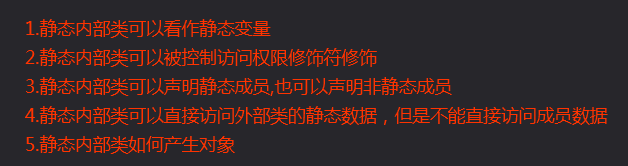
### 成员内部类公有

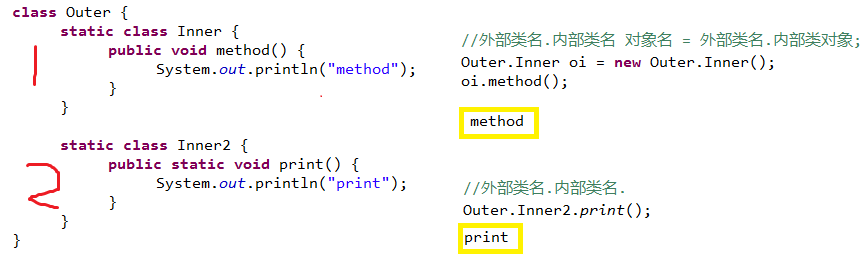


### 成员内部类私有

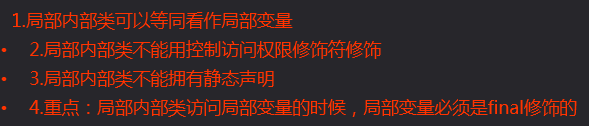


## 静态内部类（类中）



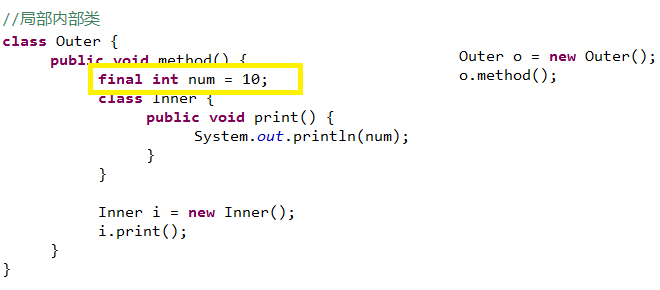


## 局部内部类（方法中）



局部内部类访问他所在方法中的局部变量必须用final修饰，为什么?

因为当调用这个方法时,局部变量如果没有用final修饰,他的生命周期和方法的生命周期是一样的,当方法弹栈,这个局部变量也会消失,那么如果局部内部类对象还没有马上消失想用这个局部变量,就没有了,如果用final修饰会在类加载的时候进入常量池,即使方法弹栈,常量池的常量还在,也可以继续使用



# 匿名内部类

## 匿名内部类概述

就是内部类的简化写法。局部内部类的一种

本质是一个继承了该类或者实现了该接口的子类匿名对象。

## 前提：

存在一个类或者接口

这里的类可以是具体类也可以是抽象类。

## 格式：

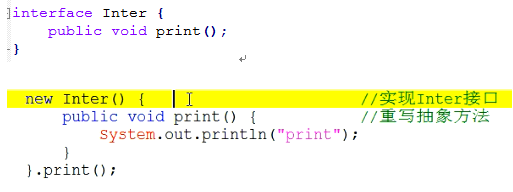
new 类名或者接口名(){

重写方法;

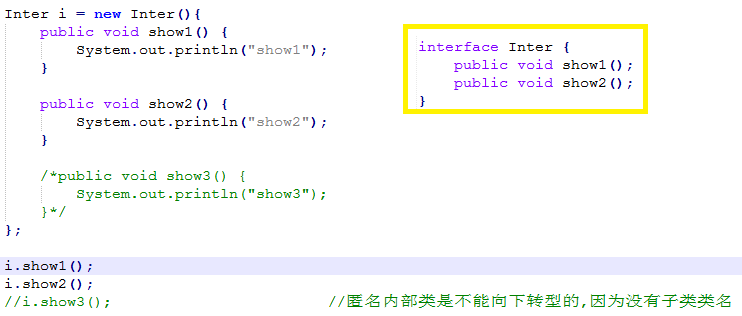
}

## 方法调用

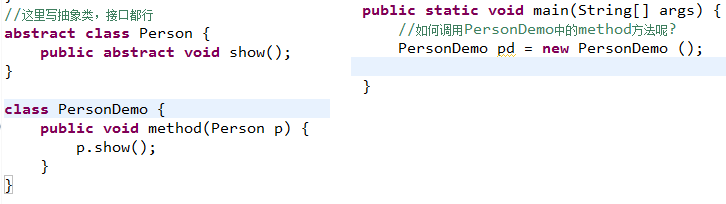
### 一个方法



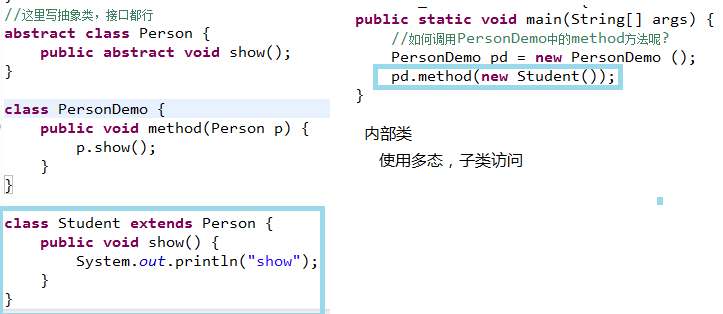
### 多个方法



## 参数传递



### 一般局部内部类



### 匿名内部类

