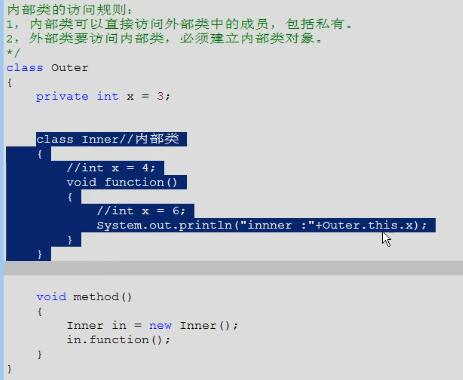
**P108 面向对象（内部类访问规则）**



想访问局部变量x：直接x；

想访问内部类的成员变量x：this.x；

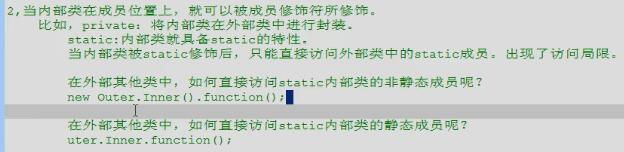
想访问外部类的成员变量x:外部类名.this.x；

这也解决了内部类为什么可以直接访问外部类的成员（当该成员的名称是唯一时）：x前面能自动补上省略的外部类名.this.

直接访问内部类的格式：

Outer.Inner in = new Outer().new Inner()

**P109 面向对象（静态内部类）**



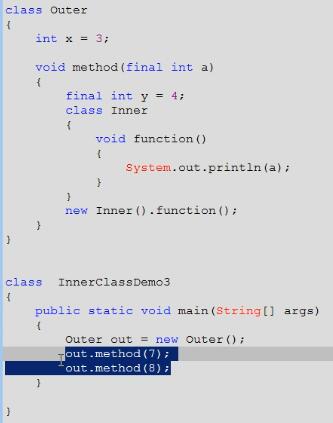


**P110 面向对象（内部类定义原则）**

//

**P111 面向对象（匿名内部类）**

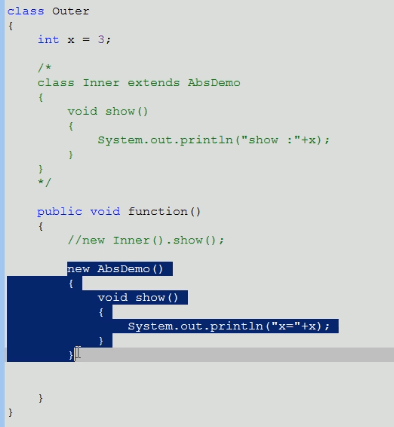
内部类定义在局部时，不可以被成员修饰符修饰，包括private和static



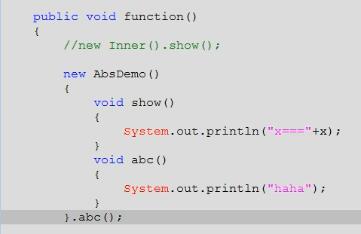
注意理解这里：为什么a明明被final修饰，编译还是能通过？

对象调用成员函数，是在栈内存中，当执行第一个method时，它进栈，然后a=7并且一直被锁住无法修改，直到整个method方法运行完毕，这时第一个method出栈，第二个method进栈。然后同样的a=8并且一直被锁住无法修改，知道这俄格method方法运行完毕。

匿名内部类的实例



蓝色部份就相当于绿色部份中的建立类并创建一个该类的对象



而且看这个我们发现，除了复写了父类的show方法，还定义了一个全新的abc方法，当然，这很正常，因为它是继承了父类的子类，但是，它还能立刻调用这个父类没有定义过的abc方法。