抽象类

抽象类体现的就是一种模板模式的设计，抽象类作为多个子类的通用模板，子类在抽象类的基础上进行扩展、改造，但子类总体上会保留抽象类的行为方式。

接口

接口就是给出一些没有实现的方法,封装到一起,到某个类要使用的时候,在根据具体情况把这些方法写出来。语法:

```
interface 接口名{
  //属性
  //抽象方法
}
class 类名 implements 接口{
        自己属性;
        自己方法;
        必须实现的接口的抽象方法
}
```

小结:接口是更加抽象的抽象的类，抽象类里的方法可以有方法体，接口里的所有方法都没有方法体【jdk7.0】。接口体现了程序设计的多态和高内聚低偶合的设计思想。

特别说明：Jdk8.0后接口类可以有静态方法，默认方法，也就是说接口中可以有方法的具体实现

注意细节

1) 接口不能被实例化
2) 接口中所有的方法是 public方法，接口中抽象方法，可以不用abstract 修饰 图示:

```
void aaa();
实际上是  abstract void aa();
```

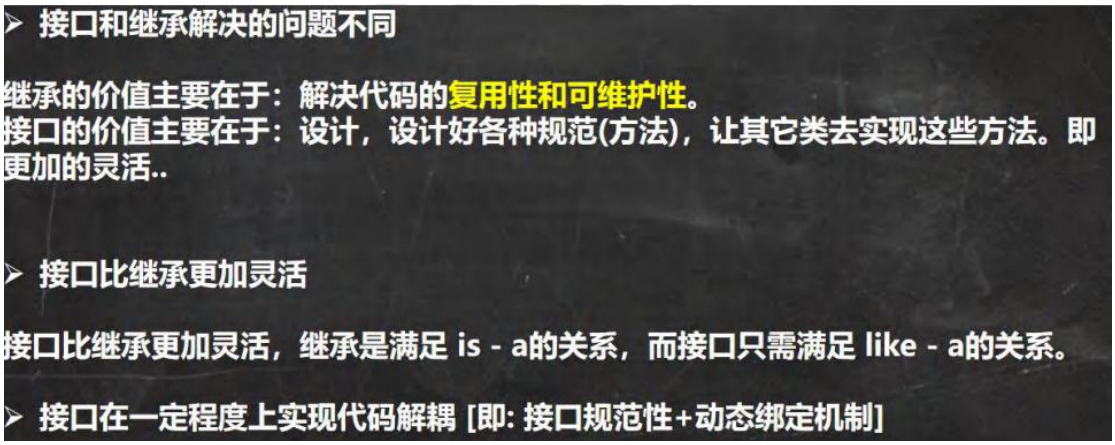
```
void aaa(){ };
```

3) 一个普通类实现接口,就必须将该接口的所有方法都实现。
4) 抽象类实现接口，可以不用实现接口的方法。
5) 一个类同时可以实现多个接口 [举例]
6) 接口中的属性,只能是final的，而且是 public static final 修饰符。比如: int a=1; 实际上是 public static final int a=1; (必须初始化)
7) 接口中属性的访问形式: 接口名.属性名
8) 接口不能继承其它的类,但是可以继承多个别的接口 [举例]

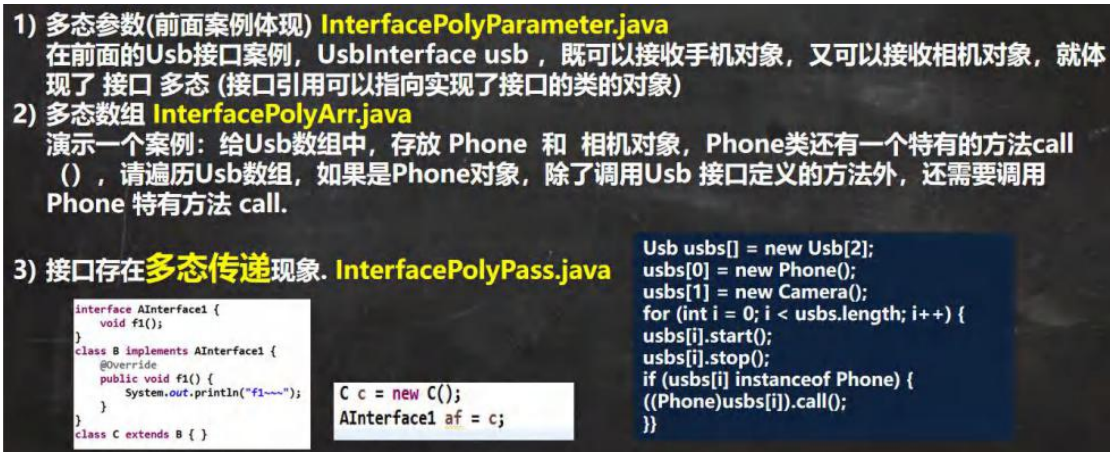
```
interface  A extends B,C{}
```

9) 接口的修饰符 只能是 public 和默认，这点和类的修饰符是一样的。

接口和继承的区别



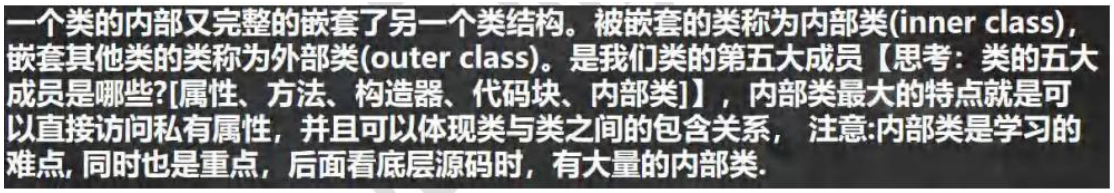
接口的多态特性



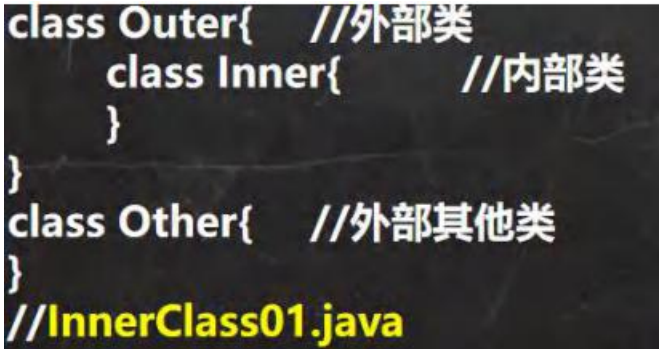
内部类

如果定义类在局部位置(方法中/代码块) :(1) 局部内部类 (2) 匿名内部类

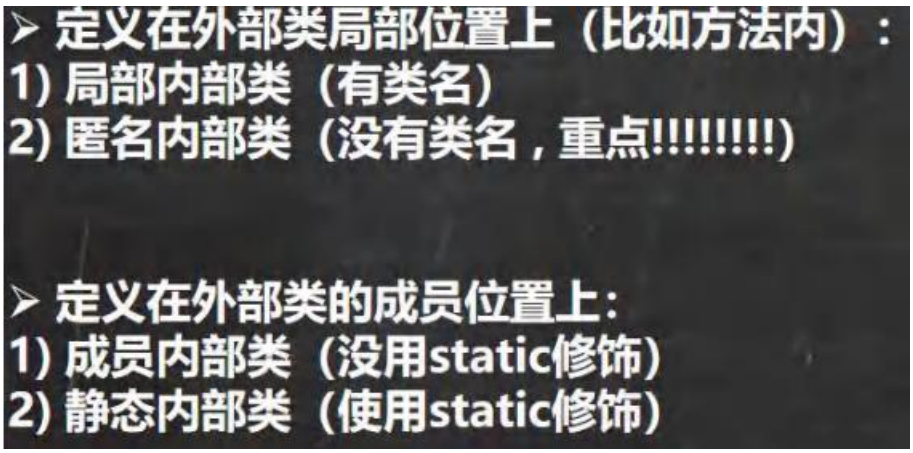
定义在成员位置 (1) 成员内部类 (2) 静态内部类



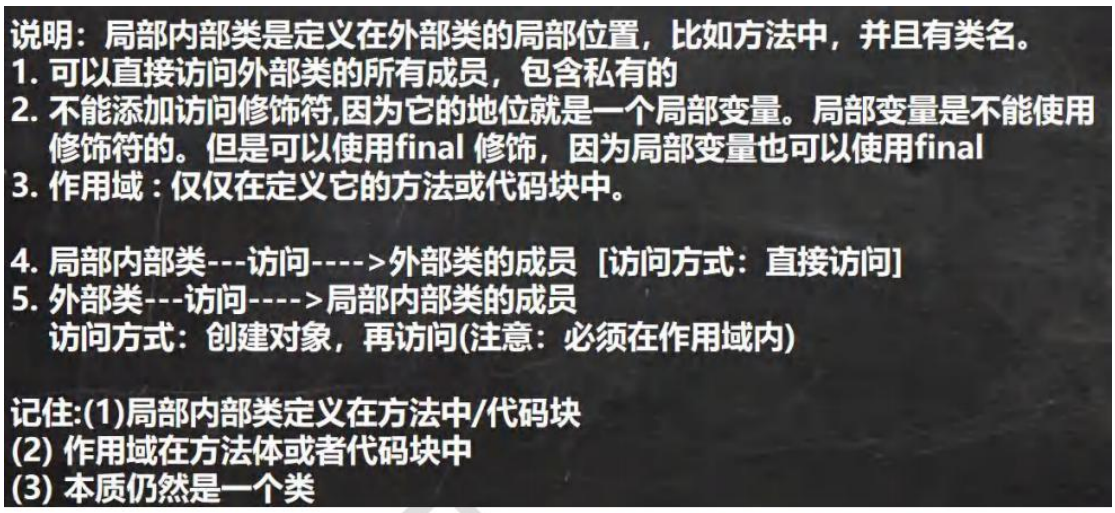
语法

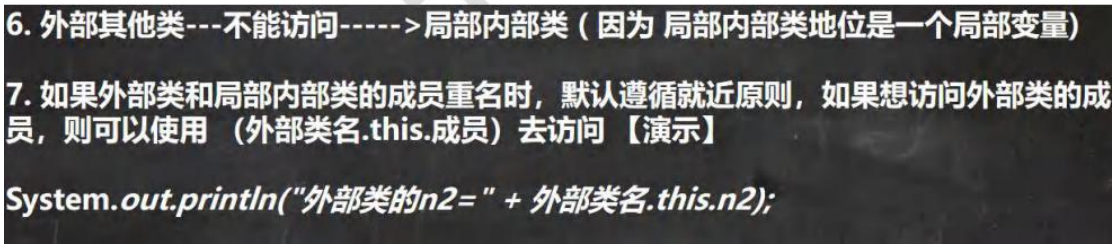


内部类分类

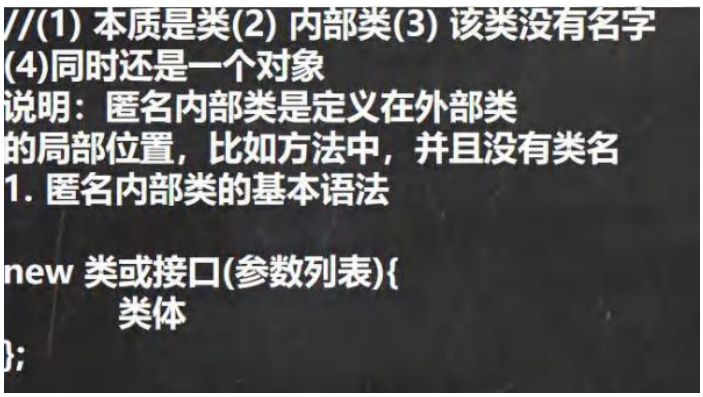


局部内部类的使用





匿名内部类的使用（重要）



2. 匿名内部类的语法比较奇特，请大家注意，因为匿名内部类既是一个类的定义，同时它本身也是一个对象，因此从语法上看，它既有*定义类的特征，也有创建对象的特*征，对前面代码分析可以看出这个特点，因此可以调用匿名内部类方法。

3. 可以直接访问外部类的所有成员，包含私有的 [案例演示]
4. 不能添加访问修饰符,因为它的地位就是一个局部变量。 [过]

5. 作用域：仅仅在定义它的方法或代码块中。 [过]
6. 匿名内部类---访问---->外部类成员 [访问方式：直接访问]

7. 外部其他类---不能访问----->匿名内部类（因为 匿名内部类地位是一个局部变量)
8. 如果外部类和匿名内部类的成员重名时，匿名内部类访问的话，默认遵循就近原则如果想访问外部类的成员，则可以使用 （外部类名.this.成员）去访问