# 回顾:

1. 封装:把细节隐藏,提高安全性

步骤:①把属性私有化 ②提供2个公共的方法设置/获取

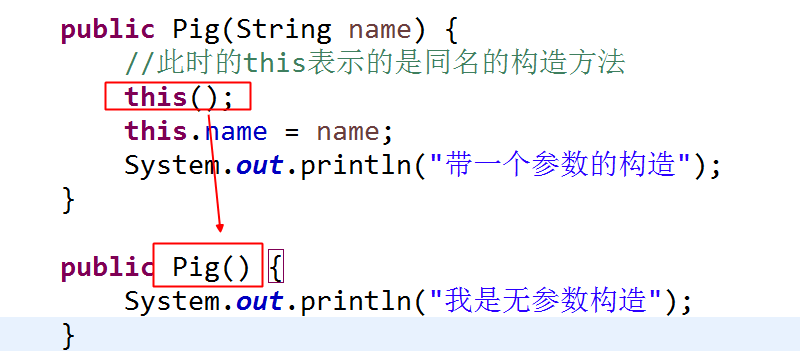
1. 重载:在同一个类方法名相同,参数类型不同(个数,数据类型,顺序不同),与返回值无关
2. 构造方法:方法名与类名相同,无返回类型,

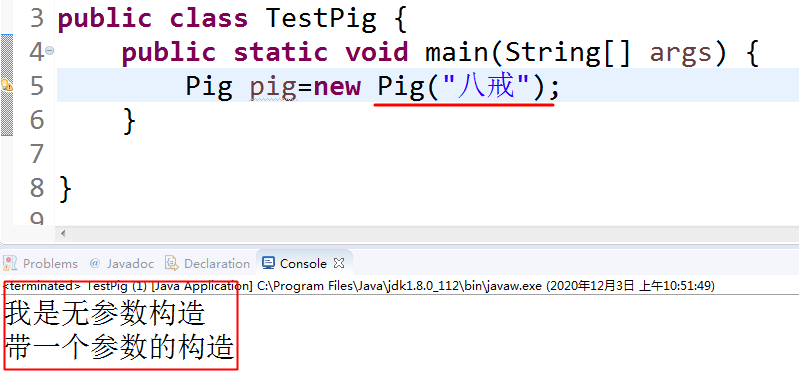
作用:创建对象/创建对象的同时需要对属性直接赋值->带参数构造

# 3.this关键字:

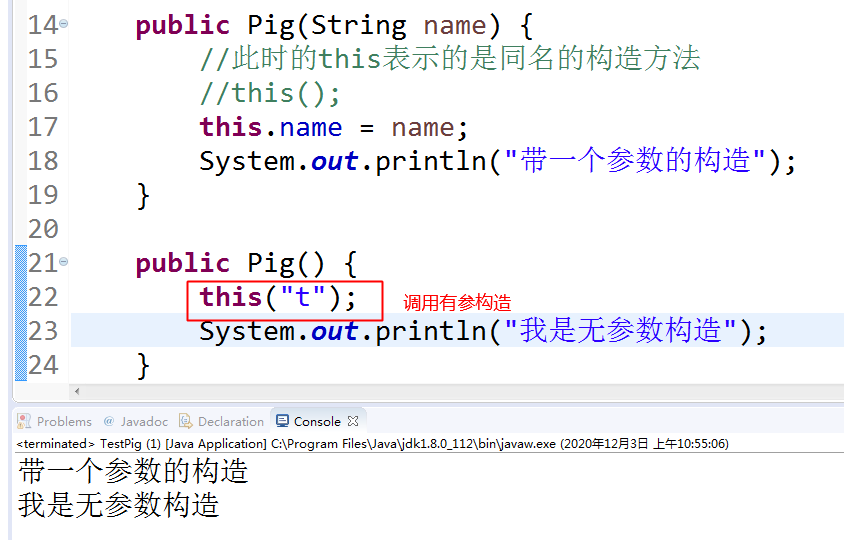
## 构造方法之间相互调用:

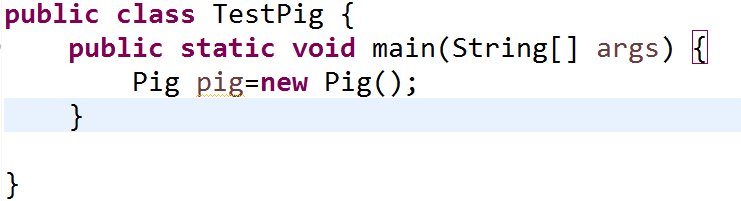
有参调用无参:



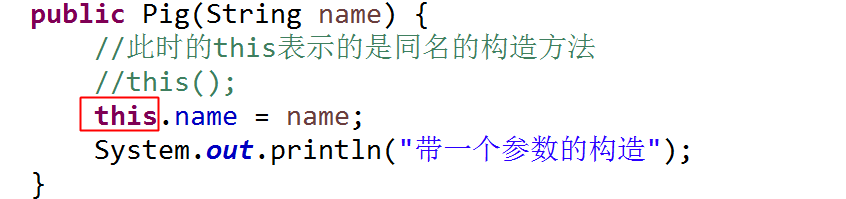


无参调用有参:



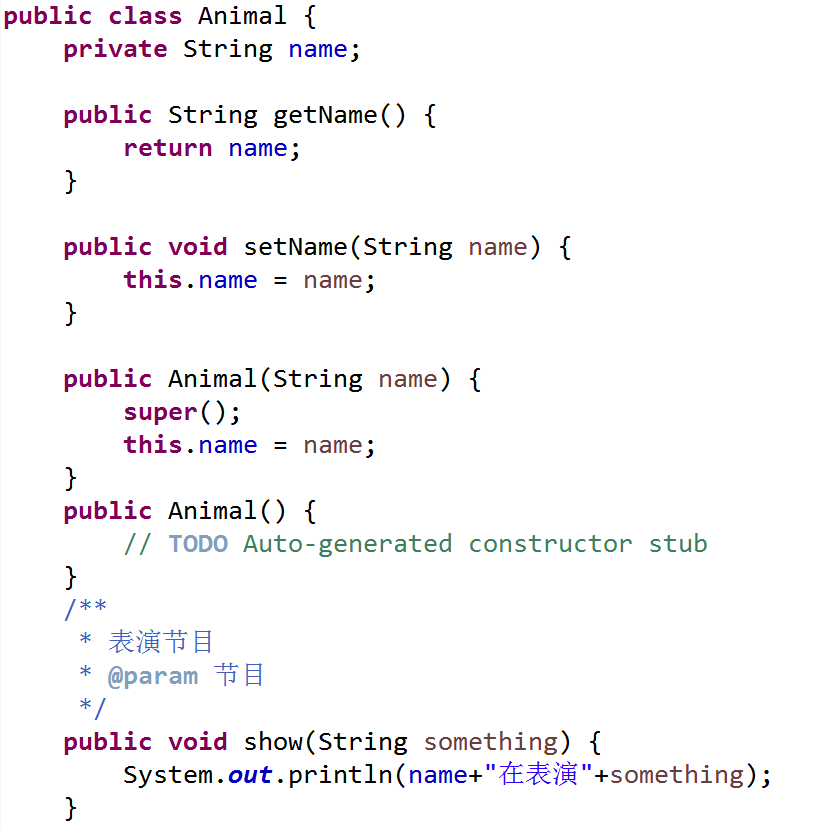


## 代指当前对象:



## 课堂练习:

1.定义一个**动物类Animal**，创建几个动物对象，模拟动物园Zoo有一群动物在表演节目.

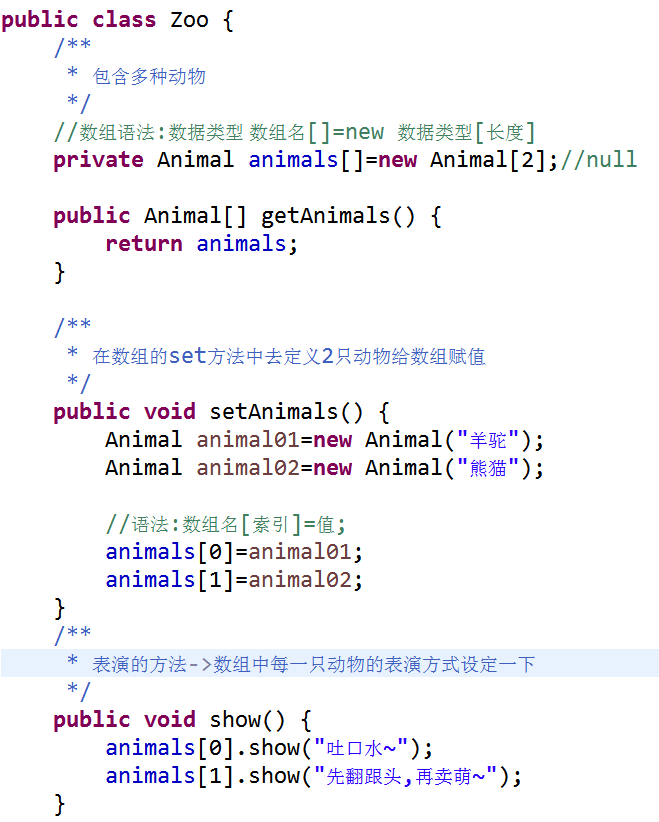


动物园类:1.定义数组去存放多只动物

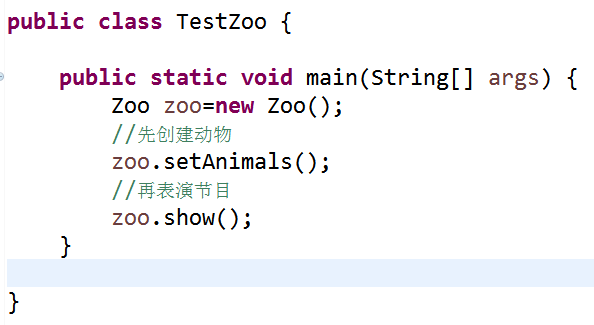
2.将属性私有化,提供公开的去访问

3.在数组的set方法中去定义2只动物给数组赋值

4.表演的方法->数组中每一只动物的表演方式设定一下



测试类:





# 2继承

也是面向对象特征之一

**引入例子**:定义学生类 幼儿园 小学生 初中生 大学生 会有公共的东西,也有不同的东西

**问题**:太多相同代码

**解决问题**:使用继承.为什么引入继承?刚才的案例分析?定义的幼儿园,小学生,大学生都是学生,他们有共同的属性和方法,如果按照我们刚刚的编码,代码有很多冗余(相同的代码)

可以这么去处理:定义一个学生类Student,,,如果一个类(BigStudent)是另外一个类(Student),?可以用BigStudent **extends** Student,就是大学生是学生,所以大学生继承学生类,我们称BigStudent叫做子类(派生类) ,Student是父类(基类)

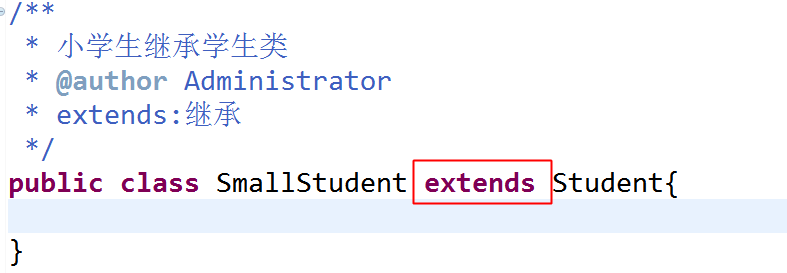
## **继承的特点**:

子类能够继承父类的所有的公共的部分(公共的属性和方法),但是构造方法除外

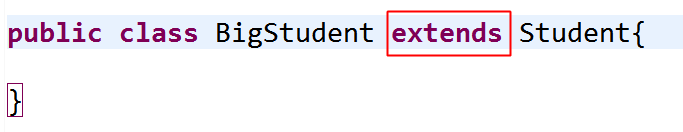
miniStudent.setStuName("小红帽"); miniStudent.setStuNo("202001");这2个公开的方法其实是从Student类继承而来的.

**继承的优点**:减少代码的冗余,方便代码的复用(父类定义后,子类继承父类就可以使用公共部分)

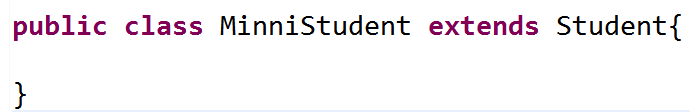
幼儿园类:



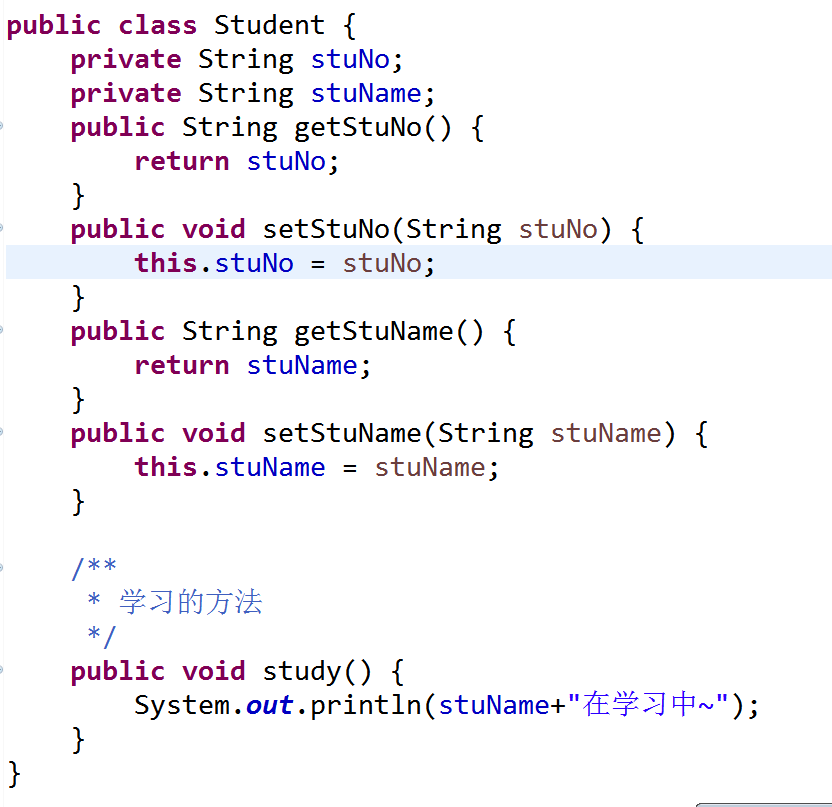
大学生类:



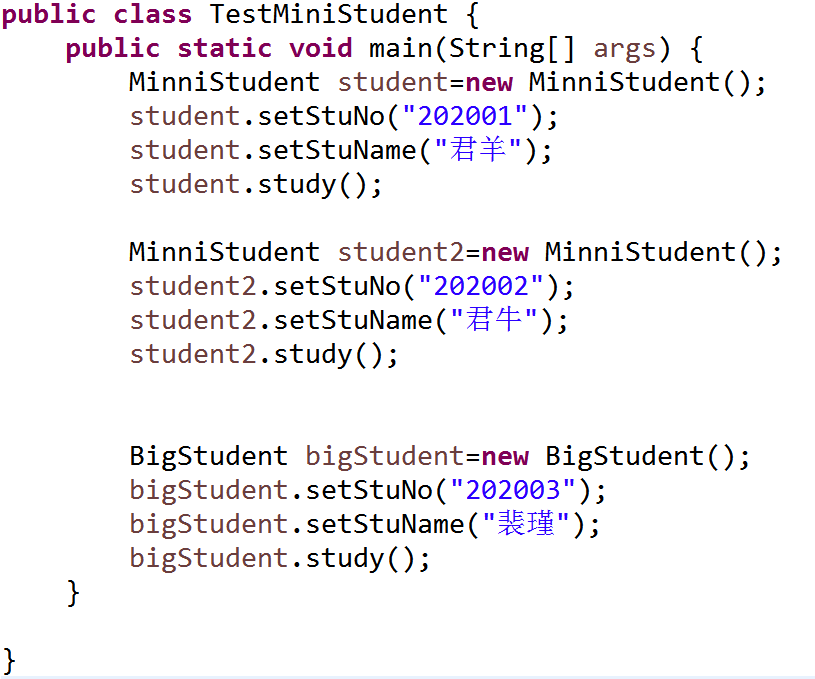
小学生类:



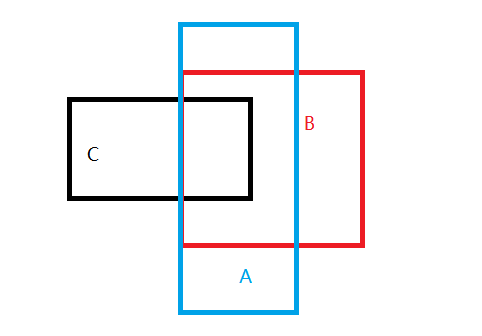
Student类:



测试类:



**Java中的类之间只能支持单继承:**BigStudent继承了Student,而Student**默认**继承了object,这种关系叫做多层继承.(A继承B,B继承了C,A可能也从C这边继承了一些公共的属性和方法)



extends除了继承的含义,还有一个含义:扩展

子类可以扩展父类,子类也可以包含自己的特有部分

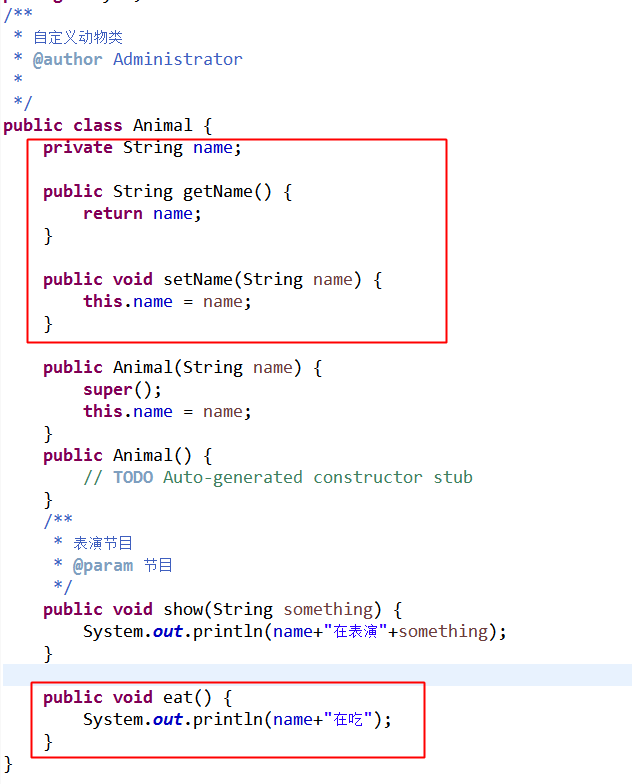
## 方法的重写(覆写)@Override[面试题]

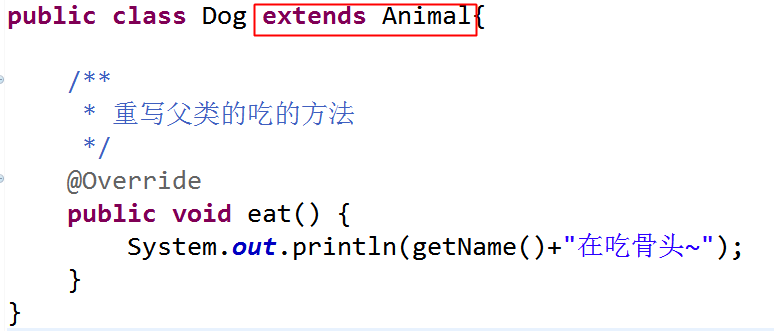
概念:发生在继承关系的子类中,在子类中的某一个方法,方法的修饰符,返回值类型,方法名,参数列表和父类的某一个方法完全一样,称为方法的重写

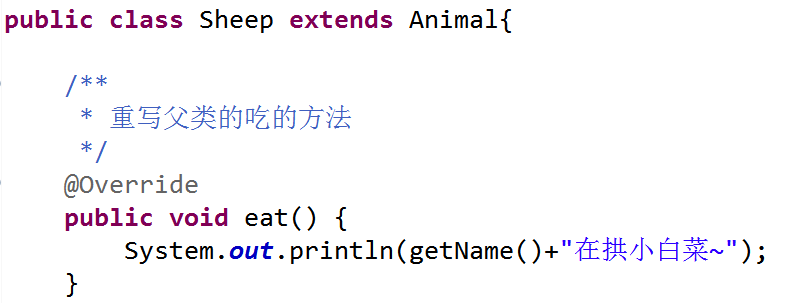
是jdk1.5新特性:@Override,会帮我们做检查:下面的方法是否是重写,如果不是重写,那么就会报错

## 课堂练习

定义一个动物类,动物都会吃,狗类(Dog),继承动物类,定义一个猪类,继承动物类,分别重写dog类和pig类的吃的方法.定义测试类 ,分别测试dog对象和pig对象,测试其吃的过程







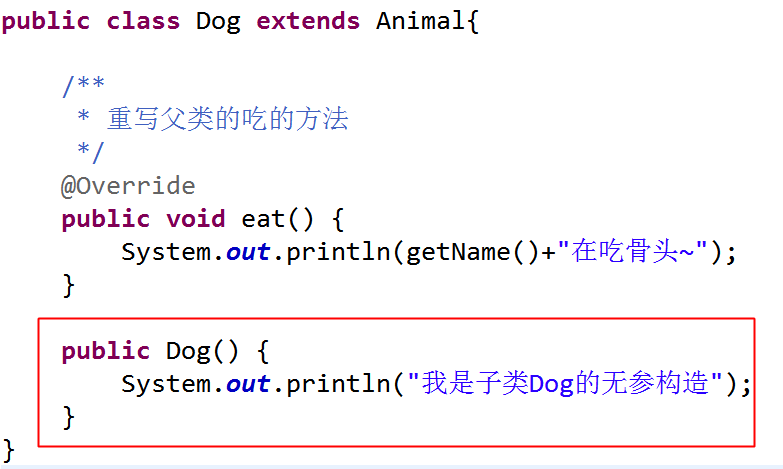
测试类:

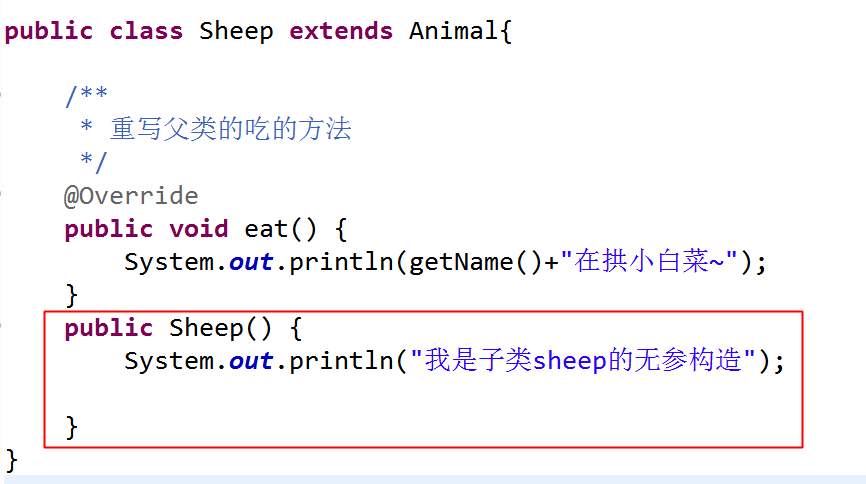


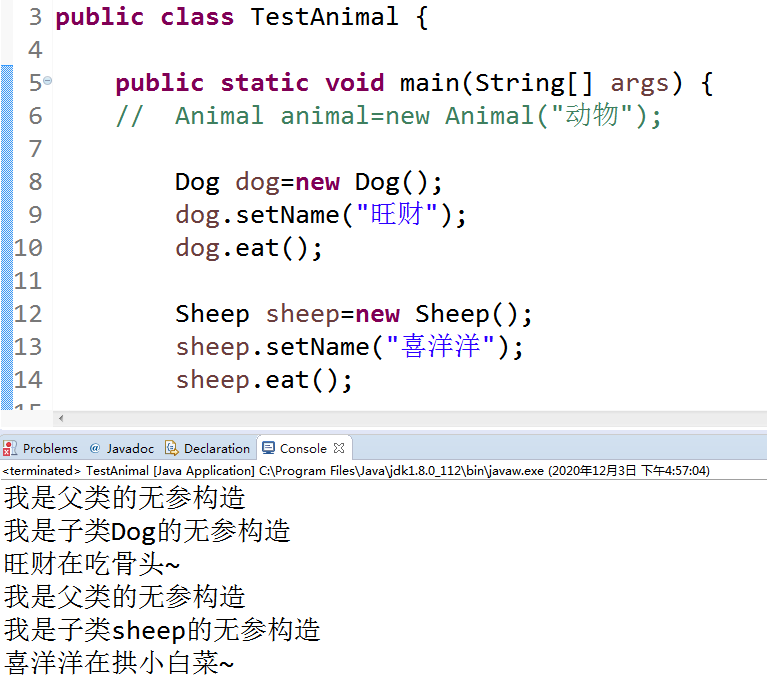
## 继承的特征

父类:









当创建一个子类对象的时候,默认会调用子类的无参数构造,但是因为存在继承关系,所以实例化子类的对象时,父类的无参构造先执行,然后再执行子类的构造方法.

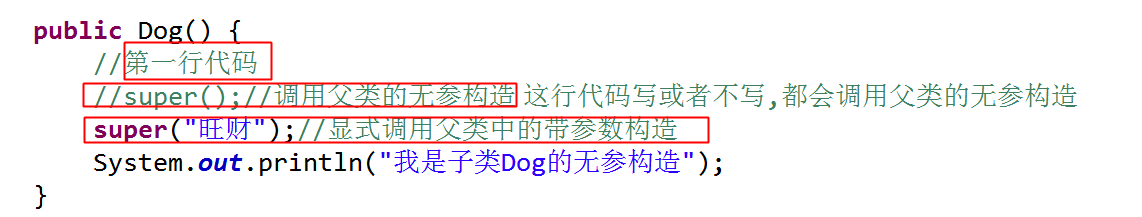
解决带参数报错:2种方法

1. 子类默认调用父类的无参,那就添加个无参
2. 在子类中指定调用父类的某个构造方法,就要用到super

## super

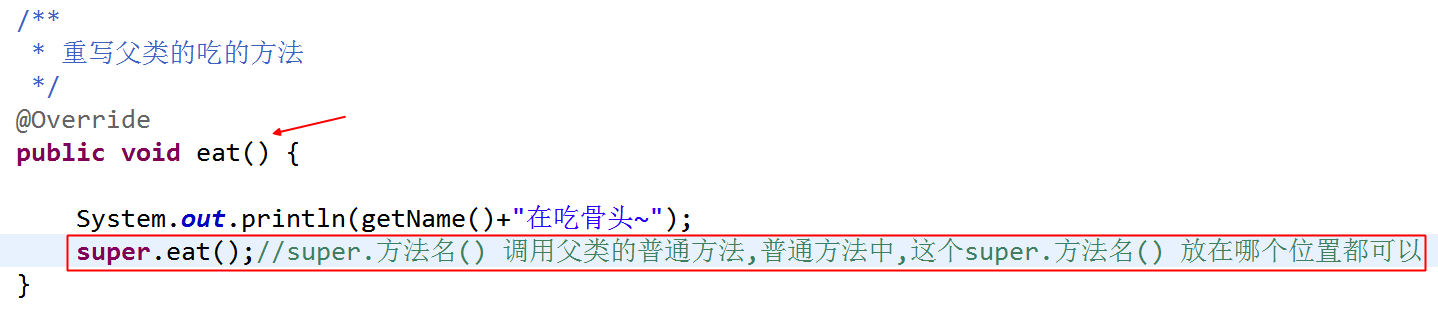
可以使用super关键字来调用父类的构造方法以及普通方法

Dog类:



调用父类构造方法的时候,必须把super放在构造方法中的第一行

调用普通方法:



提问:super关键字的作用是什么?

# 3总结:

1. this关键字:代指当前对象 构造方法间的相互调用
2. super关键字:子类调用父类的普通方法和构造方法
3. 继承:为什么要用继承,特点,方法的重写 特征

# 作业