## 回顾

1. 构造方法(特点:方法名和类名一样与无返回类型),有参与无参就是方法的重载
2. this关键字:①构造方法之间相互调用 this() this(参数1,参数2) ②代指当前这个类的当前对象
3. toString():返回对象信息 :打印:syso(对象名.toString());toString()**可以省略**,syso(对象名)

4. 继承：关键词extends

继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法. 继承就是子类继承父类的特征和行为(公共的特征和行为)，使得子类对象（实例）具有父类的实例域和方法，或子类从父类继承方法，使得子类具有父类相同的行为. 增强了程序的可维护性.

能清晰地体现出类与类之间的层次结构关系.

extends 继承,扩展

子类可以扩展父类 子类可以包含自己的特殊的部分

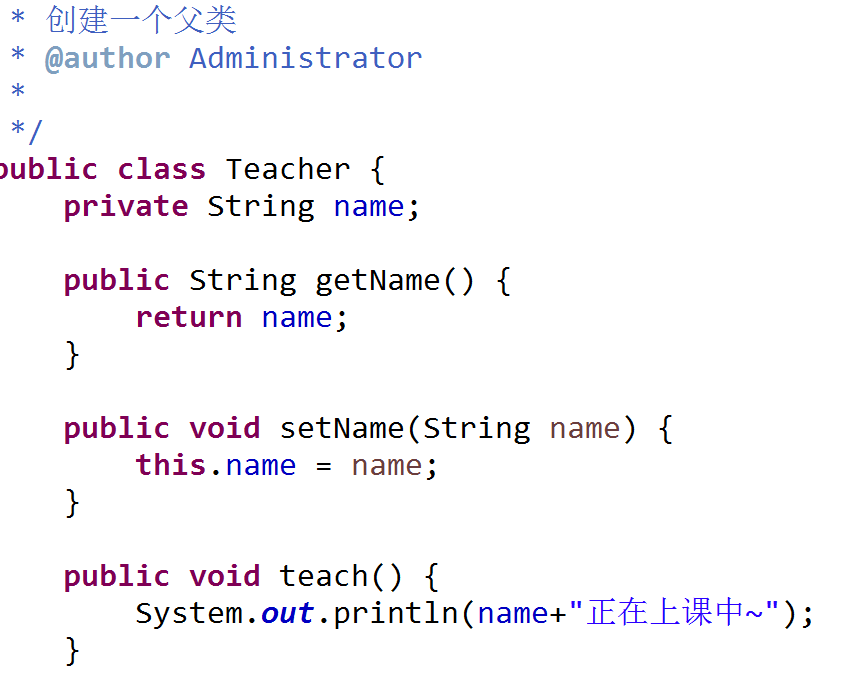
重写

含义:发生在继承关系的子类中,在子类的某个方法, 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(覆写).

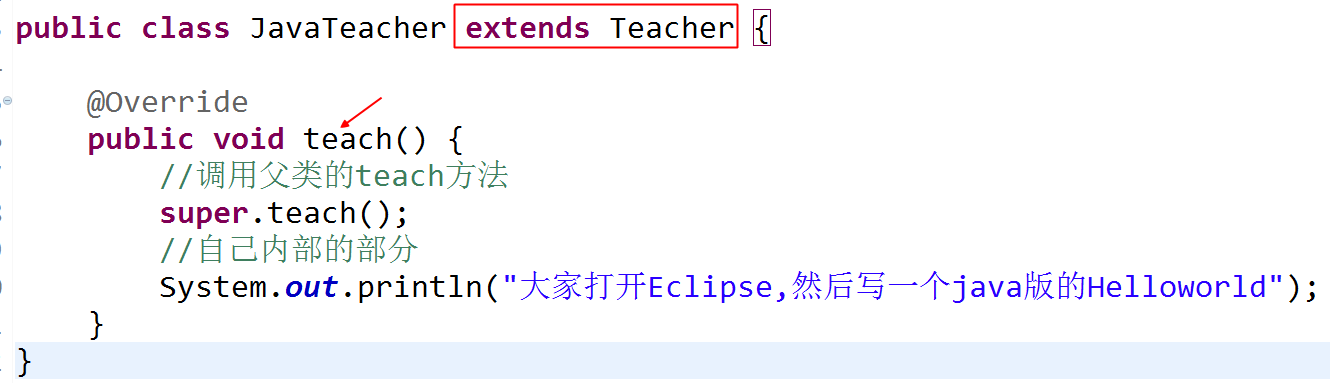
## 作业

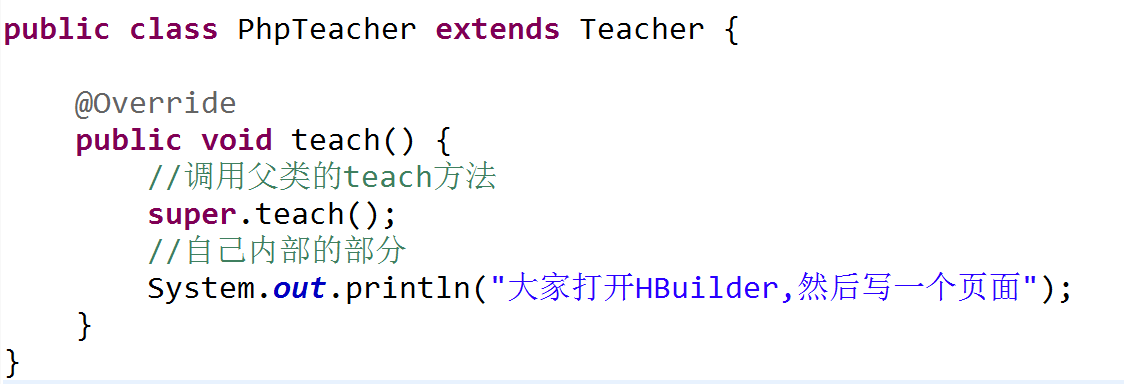
1. 模拟课堂上课:zq(JavaTeacher)来了,我们开始上正经的java课程;林珑(Driver)来了,我们就开始开车;玉莹(OldDriver)来了,我们就上职业素养课,飙车.

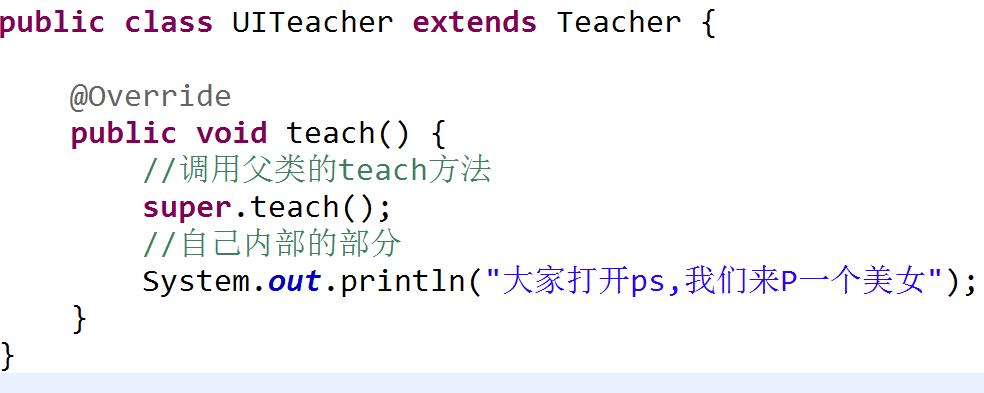
步骤1:先创建父类Student



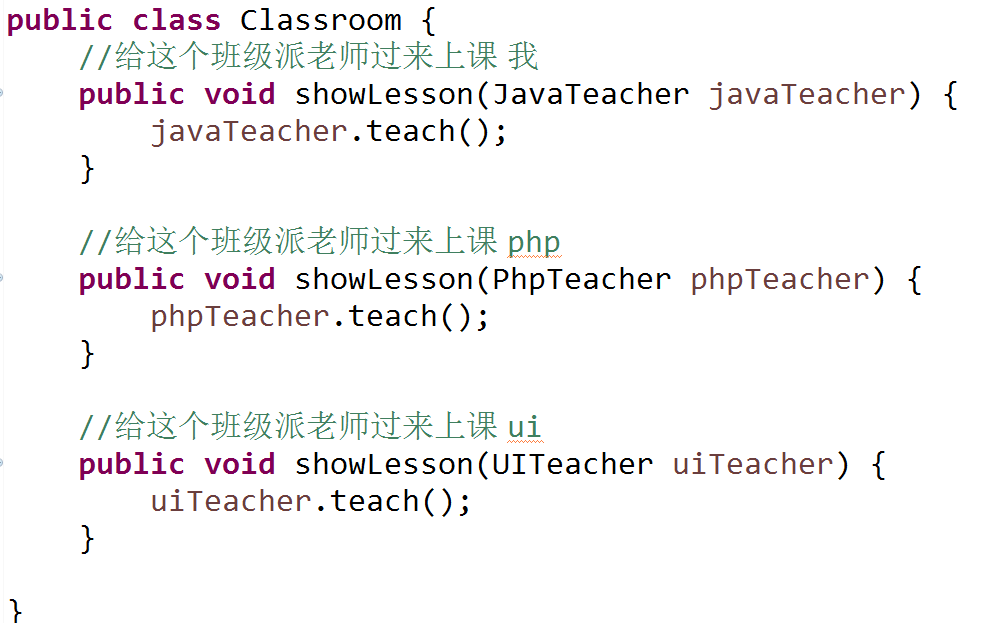
步骤2:再创建Teacher的各个子类



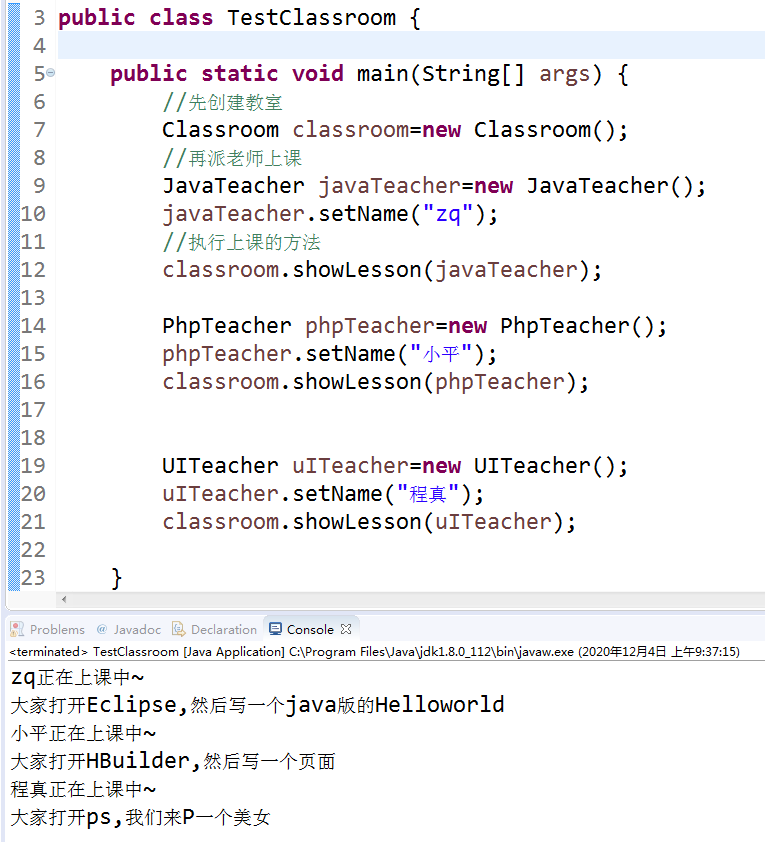




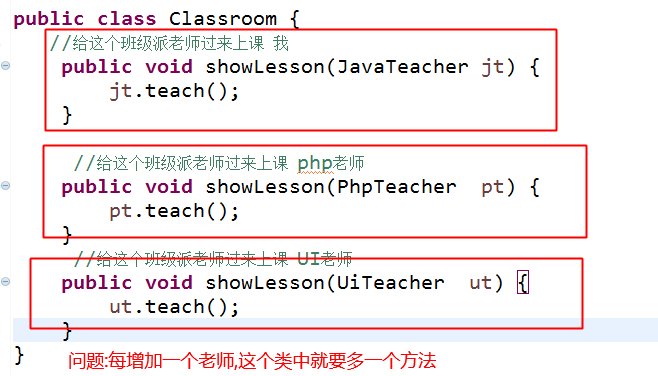
步骤3:创建一个教室,声明一个上课的方法,这个方法指定特定的老师来上课



步骤4:测试:

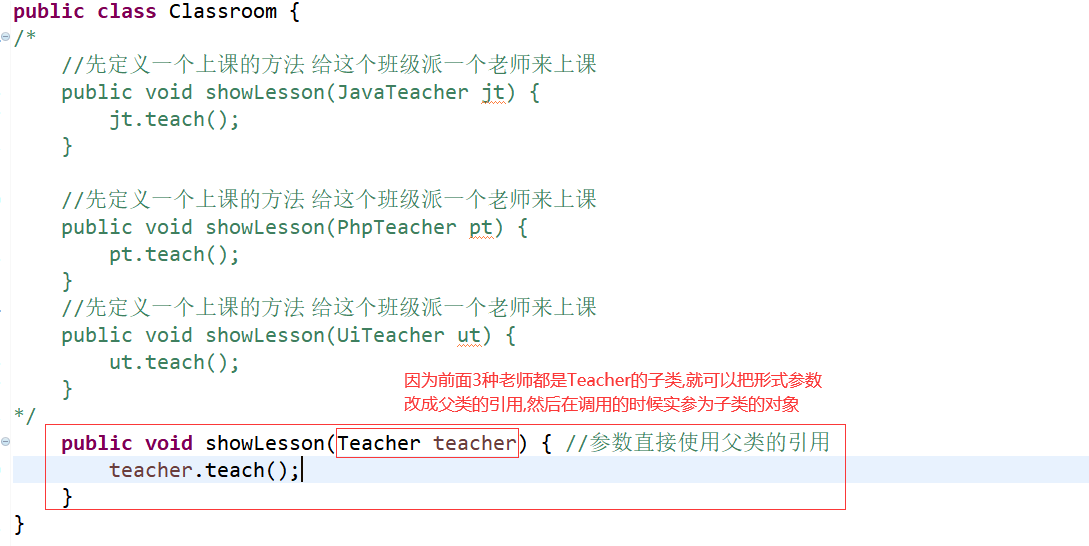


问题:



改进:

将父类作为一个方法的参数,定义方法的时候,并不确定这个参数的实际对象;jt.teach();看起来是父引用的那个方法名



## 多态实现方式1-使用继承

也是我们面向对象的特征之一

**概念：多态是同一个方法具有多个不同的表现形态**。

多态其实就是同一个父类(接口)的引用，使用不同的子类(实现类)而执行不同的操作。

对于不同的对象,javaTeacher的对象和PhppTeacher的对象,UiTeacher的对象;执行的过程和结果可能是不同的.;多态

### 多态的实现方式:使用继承

***操作步骤***:创建父类, 创建子类, 重写父类的方法;特定的场景中,将父类作为形式参数,实际参数为子类,执行的过程和结果和子类对象有关;

不同的对象,对于同一个方法执行过程和响应是不同的.

### 多态的练习

乐器（Instrument）分为：钢琴（Piano）,小提琴（Violin）

各种乐器的弹奏（play）方法各不相同。

编写一个测试类InstrumentTest,要求：

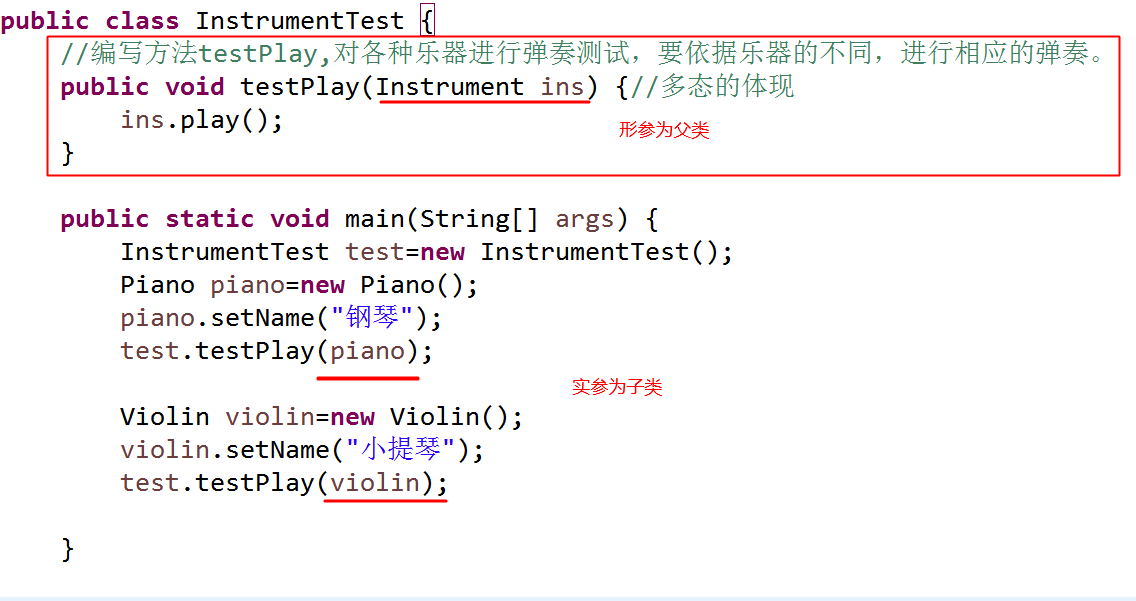
编写方法testPlay,对各种乐器进行弹奏测试，要依据乐器的不同，进行相应的弹奏。

在main方法中进行测试









## 抽象类

存在一个类,这个类不太好描述,但是确实是一个类;通常一个类中会有属性和方法 ;而这种类内部的方法又不太好实现;

概念：在面向对象的概念中，所有的对象都是通过类来描绘，但是反过来，并不是所有的类都是用来描绘对象的。

如果一个类中没有包含足够的信息来描绘一个具体的对象，这样的类就是抽象类。

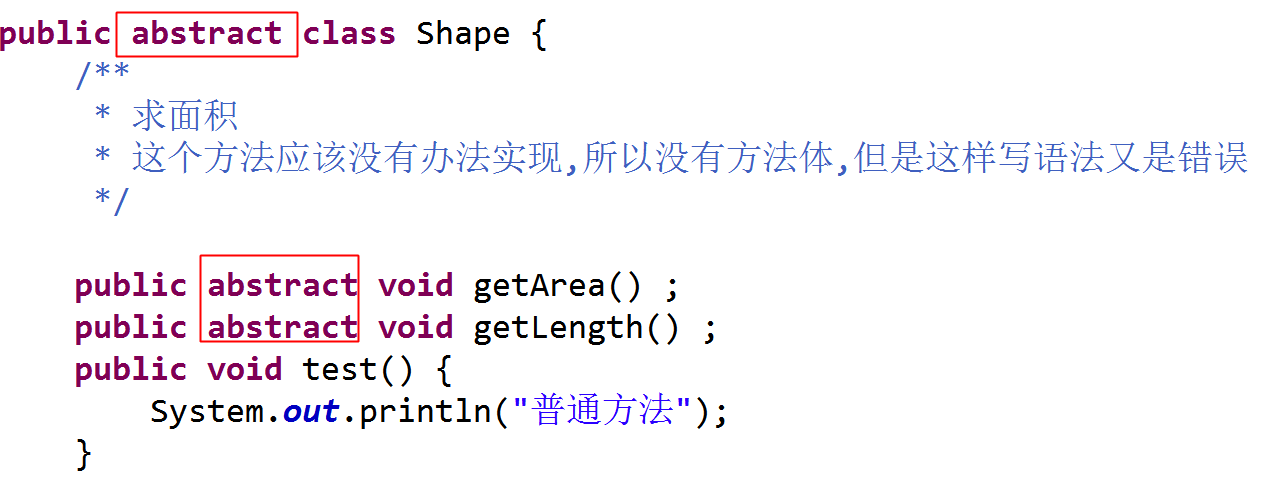
**引入例子:**形状是否能定义成一个类,如果可以请问该类可以拥有什么属性和方法?边长?求面积的方法? 子类:圆形:R 矩形:w h

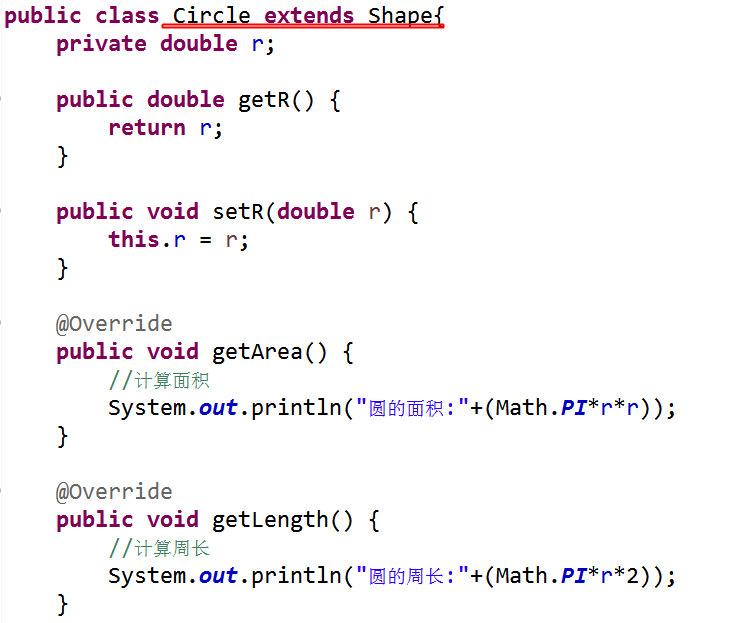
**出现问题:**如果将形状设置为父类,此时其实没有共同的属性.

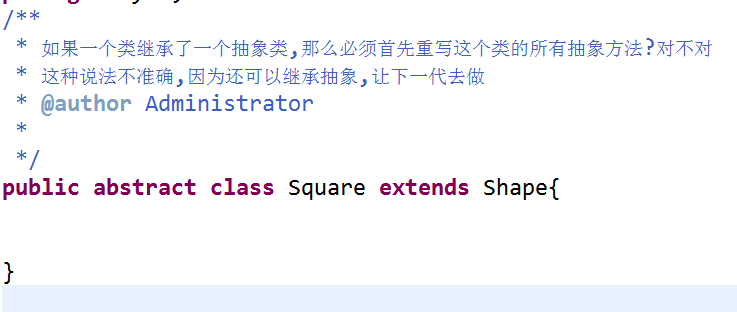
**解决问题:**可以将这个类变成抽象

基本结构:

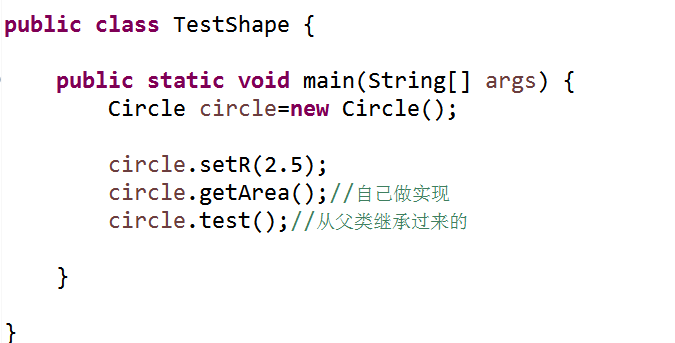
abstract class 类名{}







测试类:



提问:

那抽象类能否包含普通方法? 可以 反之不行;如果某个类中包含抽象方法,那么这个类就必须定义成抽象类

抽象类的普通方法,子类能不能继承? 可以

### 抽象特点

①抽象方法：被abstract修饰的方法就是抽象方法，没有方法体。{}

②抽象类：一个类中如果有至少1个抽象方法，那么这么类必须也是抽象类，抽象类中也可以包含普通方法。

③抽象类不能直接实例化，只可以被继承，如果一个类继承了抽象类，需要重写抽象类中的所有抽象方法(如果不想重写可以把这个子类设置成抽象类)。

④抽象类中有构造方法,但是不能被abstract修饰，因为子类构造的时候需要调用父类的构造方法。

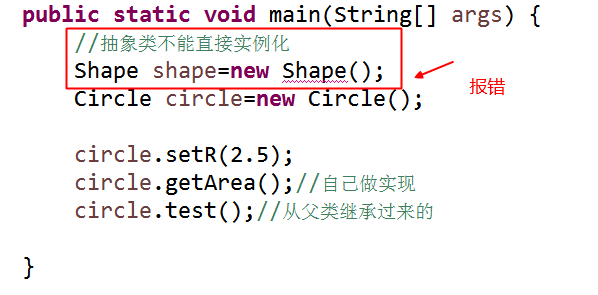
### 课堂练习:

自定义抽象类形状,子类: 圆形,正方形,梯形; 在子类中父类的抽象方法做实现(面积 周长);测试类来测试;

问题:当我定义一个抽象类形状,再去定义它的子类的时候,子类一定需要抽象吗?

答:子类如果实现了父类的方法,就不用抽象,如果子类不想实现父类的方法,就抽象.

抽象类不能直接实例化

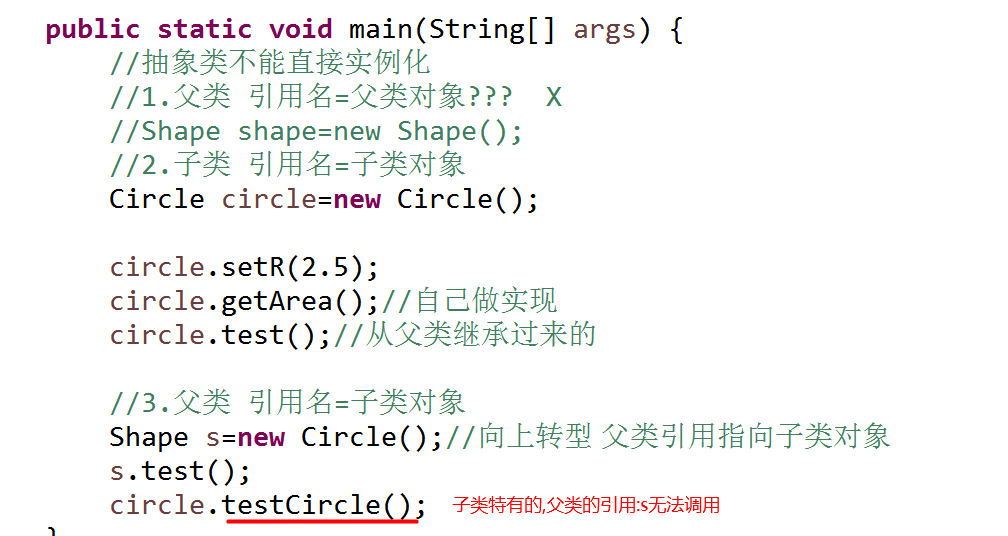


向上转型:父类的引用指向子类对象

父类 引用名=new 子类()

注意:

父类的所有属性和方法+子类和父类共有的属性和方法,对于只存在子类中的方法或属性无法调用



为什么要用抽象类:

分析,设计,定规则规范;A(分配 告诉大家要做什么 设计, 不做具体实现) B(要做具体实现) C(要做具体实现).实现之前做一个标准->定规范->抽象类

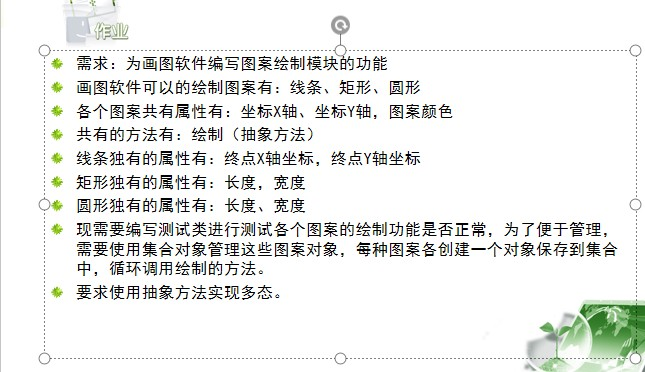
## 【面试题】普通类和抽象类的区别

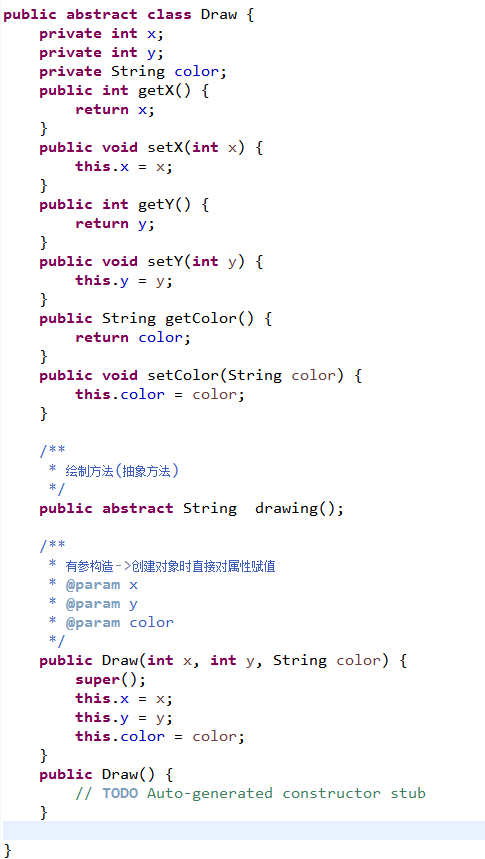
　　1）普通类中不能包含抽象方法，如果一个类中有了抽象方法这个类就是抽象类。抽象方法必须为public或者protected（因为如果为private，则不能被子类继承，子类便无法实现该方法）。

　　2）普通类可以创建类的对象，抽象类不能直接用来创建对象，只能由子类创建；

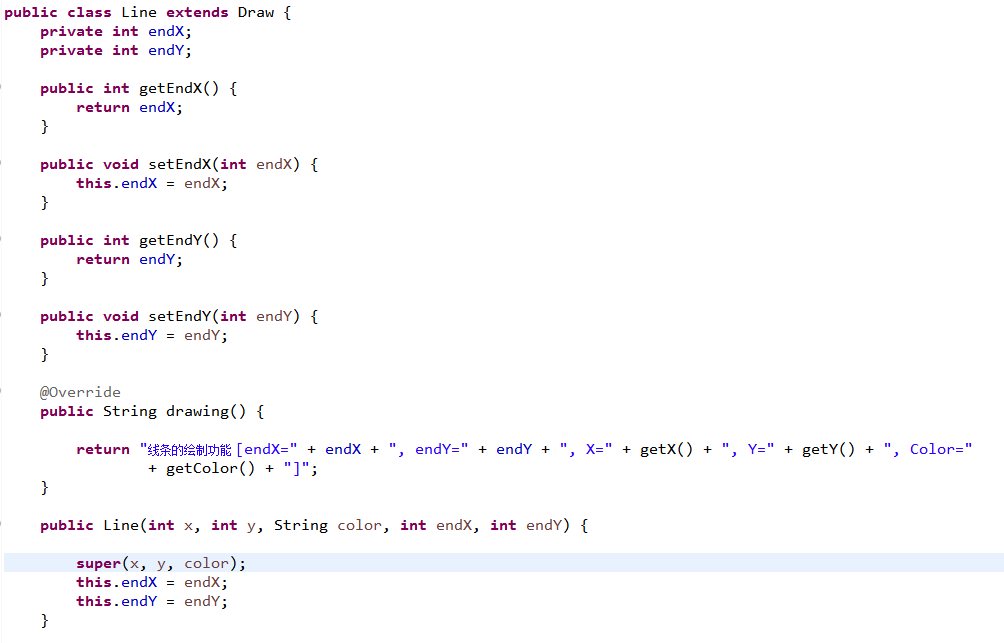
3）如果一个类继承于一个抽象类，则子类需要实现父类的抽象方法。如果子类没有实现父类的抽象方法，则必须将子类也定义为abstract类。

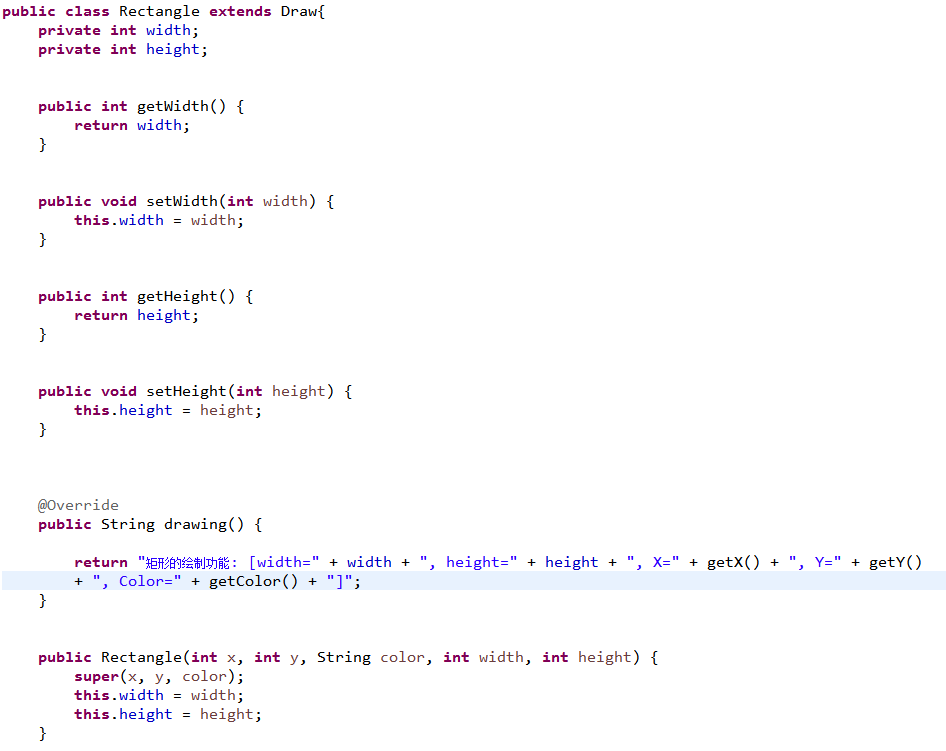
## 课堂练习





子类:





测试类:

