

extends 继承或 implements 实现是多态性的前提

一个对象有多种形态就叫多态

父类引用指向子类对象: 父类名称 对象名 = new 子类名称();  
接口名称 对象名 = new 实现类名称();

访问成员变量: 1. 直接通过对象名称访问成员变量: 等号左边是谁, 优先用谁, 没有则向上找

2. 间接通过成员方法访问成员变量: 该方法属于谁, 优先用谁中的该成员变量, 没有则向上找

成员方法优先用子类中的 (new 谁用谁, 没有则向上找)

编译看左边, 运行看右边

实际上多态只能调用父类的方法和成员变量, 但若子类覆写了, 就先调用子类对象, 但由于无法覆写成员变量, 故不能调用子类的成员变量。

成员变量: 编译、运行都看左边

成员方法: 编译看左, 运行看右

对象的向上转型: 父类名称 对象名 = new 子类名称();

(一定是安全) (右侧创建的子类对象当作父类来看) (类似自动类型转换)

弊端: 对象一旦向上转型为父类, 则无法使用子类特有的方法。

对象的向下转型: 子类名称 对象名 = (子类名称) 父类对象;

(相当于将父类对象还原为本来的子类对象) (类似于强制类型转换)

必须是还原为原来的子类对象, 否则会报错 `ClassCastException`

判断前面的对象能不能当作后面类型的实例, 返回 boolean 值

对象 instanceof 类名

了解原来是什么子类, 向下转型时要先判断一下。

