

面向对象的特征 多态:

在Java中的两种体现:

① 方法的重载和重写

② 对象的多态性

\* Java引用变量有两个类型, 编译时类型和运行时类型。  
编译时类型由声明该变量时使用的类型决定, 运行时类型由实际引用该变量的对象决定。

student a = new Student();

student是编译时类型, a实际引用的堆内存中的对象是运行时类型  
当两者不一致时, 就会出现多态。

- 一个变量只能有一种确定的数据类型
- 一个引用类型变量可以指向多种不同类型的引用对象

Person p = new Person();

Student s = new Student();

Person e = new Student(); // 变量引用指向类实例, 但e不能访问 Student中定义的属性和方法  
但是可以用重写的  
子类是父类的扩展, 可以看作特殊的父类 (向上兼容)

\* 虚拟方法调用:

Person和Student都有一个 getInfo()

Person p = new Person();

p.getInfo();

Person s = new Student();

s.getInfo(); // 调用的是Student的getInfo(), 因为方法调用是  
// 在运行时决定的, 由运行时类型决定。

instanceof 操作符:

x instanceof A

检测x是否为 A或 A的子类

Object类:

① Object类是所有Java类的根父类

② 类的声明未使用 extends 关键字指明其父类时, 默认父类为 Object.

③ 将 Object 类型作为方法的参数, 这表示, 该方法可以接收所有类型的对象为参数.

Object 类中的主要方法:

public Object() 构造方法

public boolean equals(Object obj) 对象比较

public int hashCode() 取得 Hash 码

public String toString() 对象打印输出