

面向对象与面向过程的区别？

面向对象的编程是一组对象互相配合通过沟通完成特定功能。

类

成员变量初始化时，整型和浮点型数据都为**0**；布尔为**false**；引用为**Null**；字符为空字符。方法只有执行时才占用内存 构造方法与类同名，不用写返回值类型；每个类默认有一个无参数的构造方法，但是如果自定义了构造方法就不会有默认

方法的重载和重写

- 方法的重载：

在同一个类里，方法名相同，参数列表不同，就称为方法的重载；不能用返回值类型来判断区分重载的方法，如果有方法名相同，返回值不同的情况，编译时会报方法名重复的错；java识别方法的方法是：1.看方法名；2.看参数列表。

- 方法的重写：

在子类中，包含与父类相同名的方法称为重写；遵循两同两小一大，两同指：方法名相同，参数列表相同；两小指：返回值类型和抛出的异常子类方法要比父类小或者相同；一大指：访问权限子类比父类大或者相同。

super和this

都不可以与static共用；this是调用自己类的实例变量和方法，super是调用父类的实例变量和方法，因为都要先找到调用的对象本身，所以不能与static通用，因为静态方法和变量是属于类而不是对象。

四个概念：源文件，包，类，project；一个包可以包含多个源文件，源文件中可以包含多个类。

访问控制：private、default、protected、public

继承：extends 继承中的构造方法：

子类的构造方法必须调用父类的构造方法，所以如果父类没有无参数构造方法，那么子类对象必须显示调用父类构造方法，不然编译报错。

- 规范：不要让子类随意访问、修改父类的方法，父类中辅助其他的方法引用Private修饰；不希望被重写的可以用final public修饰。不要在父类构造器中使用将被子类重写的方法。

多态：

引用变量有两种类型，一个是编译时类型，另一个是运行时类型，前者是指声明定义的，后者是实际指向的。多态：把子类对象赋给父类引用，使相同类型的变量、调用同一个方法时呈现出多种不同的行为特征。注意：1.对象的实例变量不具备多态性，调用时，还是使用父类的实例变量，多态的是方法；2.编译阶段只能使用编译时类型所拥有的方法，即编写时，只能调用父类的方法名。

引用变量的强制类型转换

只能在有继承关系的两个类型之间进行转换，不然会在编译时报错；还有一种运行时会报错（ClassCastException）的情况，父类引用指向子类对象时（多态），如：Object obj = new

`Integer(5); String str = (String) obj;` 上述情况会报错，因为转换的类型与变量的运行时类型不是继承关系！ 所以引用类型转换时要考虑变量的编译时类型和运行时类型。

instanceof

为了防止引用类型转换时出错，可以先用instanceof进行判断。格式：引用变量 instanceof 类名； 判断前者是否是后者的实例。注意，instanceof前面的引用变量的编译时类型要么与后面的类相同或者有继承关系，否则会有编译错误。如： `//报错的，string和math没有继承关系 String s = "hello"; s instanceof Math;` `//正确的 Object d = "hello"; d instanceof Math;` `//虽然变量d的运行时类型为string，但是编译时类型object是math的父类。`