# 三大特性

面向对象 OOP Object Oriented Programming

继承性，封装性，多态性

继承 两个类之间的关系。一个类继承另一个类中的属性和方法

一个类 属性 方法

代码复用

继承之后 子类和父类

只能是单继承

A extend b

A extend c X

A extend b

b extend c 连续继承√

默认有一个类Object

默认继承它的

Object这个类有哪些方法

构造函数是不能被继承的

默认情况，子类new之后，默认会调用父类的一个无参的构造函数

但是我可以自己去调用

Jvm加载类，不是一次性把所有的类都加载的，是按需加载的，懒加载机制

一个项目1000个类

规范，每个类都要自己定义一个无参的构造函数

可以定义一个抽象类。可以没有抽象方法，定义的抽象类，即使没有抽象方法，也不能被实例化，只能通过子类去实例化

抽象类的出现，就是一个标准

我们要求每个子类都有show1 show2 show3 方法

父类可以引用子类，和抽象类没有关系

Int a  
long b =a

Byte c=(int)a

父类引用子类

向上转型 自动转换

向下转型 强制转换

Final定义的变量 不能改变

重写

重载 方法名一致，参数个数，类型应该不一样。

重写 在继承之后，子类的方法名字和参数完全一致。

针对方法的

属性？可以理解重写 方法被重写，属性隐藏

多态 针对方法 属性？

# 接口

配合反射 机制 实现功能

工厂模式

它就是一个标准

接口和抽象类一样，不能被实例化

类是单继承的

结构简单，但是功能弱了一点

接口 实现 多 弥补 这个弱点

父类可以引用子类

接口可以引用实现类