

有时候将一个父类设计得非常抽象，以至于它没有具体的实例，这样的类称为抽象类  
abstract: 抽象的，可以用来修饰类和方法

1, abstract修饰方法: 抽象方法 (**public abstract 返回值类型 方法名 (参数列表);**)

①格式: 没有方法体, 参数括号后面使用” , “;

②抽象方法只保留方法的功能, 不包含方法的具体实现, 具体的执行, 交给继承该类的子类来实

现, 由子类重写该抽象方法。

③若子类继承抽象类, 并重写了所有的抽象方法, 则此类是一个实体类, 即可以进行实例化

④若子类继承抽象类, 没有重写所有的抽象方法, 意味着此类还有抽象方法, 即此类还是一个抽

象类, 即不能被实例化

2, abstract修饰类: 抽象类 (**abstract class 类名 {}**)

①抽象类不能被实例化

②抽象类可以有构造器, 属性 (凡是类都有构造器)

③抽象方法所在的类, 一定是一个抽象类

④抽象类中可以没有抽象方法

3, ①abstract不能用来修饰属性、构造器、private、final、static (构造器不能被重写) (子类不能覆盖 (或重写) 声明为private的方法的)

模板方法设计模式: 抽象类作为子类的通用模板, 子类在抽象类的基础上进行扩展和改造, 但子类总体上会保留着抽象类的行为方式