子类覆盖父类的方法

父类的变量指向子类对象，在调用方法时，实际调用的仍然是

子类的方法，这就是类的多态性。

一个子类对象在一定条件下可以转换成继承链中的另一个类的

对象：

1. 一个子类对象的类型一般可以向上转换成它的父类对象，这种转换

是安全的，java编译器能自动进行

1. 一个父类对象的类型一般不能向下转成它的子类。但有一种情况例外，如果一个父类对象引用连接的实际上是一个子类对象，就可以使用强制类型转换将这个父类对象转换成子类类型

将一个父类对象的类型转换成它的子类类型时，必须确保其医用连接的是该子类的一个对象，这可以利用运算符instanceof来完成。运算符instanceof的作用是检查某个对象的类型是否是某个指定类或其子类，如果是，则该对象不为null，则返回true，否则返回false

注意：

1. 方法覆盖（重写overide）时，必须注意子类中的方法应该与父类中被覆盖的方法有相同的方法名、返回类型和参数列表。
2. 私有方法不能被覆盖
3. final修饰的方法不能被覆盖
4. 方法覆盖时，不能降低其访问权限（public>protected>default>private）
5. 静态方法也能被继承，但不能被覆盖

方法重载（overload）！=方法覆盖（override）

方法重载是在同类中，方法名相同，方法参数列表不同（参数个数，参数顺序，参数类型）

方法重写（覆盖）是解决子类继承父类之后，父类的某个方法不满足子类的具体特征，此时需要重写方法体

抽象类（abstract主要注意点）

1. 抽象类不能实例化
2. 可以定义抽象类的变量， 但不能创建对象

接口（interface主要注意点）

1. 接口中的成员均是public访问类型的，接口里的变量默认是用public static final共同标识的。方法默认public abstract修饰
2. 一个类可以继承一个父类的同时，实现一个或多个接口，extends关键字必须位于implements关键字之前