

重载、重写、覆盖：

重载：在一个类中，方法名相同，参数个数或类型不同

重写：在父类和子类之间，当父类的方法不满足于子类需求时，可以重写父类的非静态方法

覆盖：重写父类的静态方法

多态：

一个事物的多种形态

编译时多态：

方法重载

运行时多态：

父类对象可以接收子类对象

注：调用方法时，先从本类找，找不到时，再去父类中找，直到找到Object类

引用类型的数据转换：

向上转型：子类转父类，是安全的转换

父类对象名 = 子类对象名；

向下转型：父类转子类，是不安全的转换

```
if(父类对象名 instanceof 子类名){  
    子类对象 = (子类名)父类对象;  
}
```

抽象方法：

由abstract修饰的方法

特点：

1. 抽象方法没有方法体
2. 抽象方法必须放在抽象类中

抽象类：

由abstract修饰的类

特点：

1. 抽象程度：抽象类 > 类 > 对象

2. 抽象类不能实例化对象
3. 抽象类需要先转成普通类（继承抽象类）才能实例化对象
4. 抽象类中可以有抽象方法，也可以有非抽象方法
5. 一个类继承于抽象类，有两种选择
 - a. 重写抽象类中的抽象方法
 - b. 本身也是抽象类

接口：

把公共的属性和方法封装成一个特定的功能集合

访问修饰符 interface 接口名{

属性1

属性2

.....

方法1

方法2

}

特点：

1. 接口中的方法都是抽象方法 默认是public abstract
2. 接口中默认属性都是 public static final
3. 一个类实现了接口，有两种选择
 - a. 重写接口中的方法
 - b. 本身也是抽象类
4. 接口不能实例化对象

类和接口的关系

1. 类和类之间：单继承 extends、
2. 类和接口之间：多实现implements
3. 接口和接口间的关系： 多继承 extends