重写：**发生在两个类中，一般是子父类，子类中的方法名和参数列表相同，方法体不同，称之为方法的重写**。

1. 子类中出现和父类中一模一样的方法（和**返回值有关**）
2. 重写的应用：当子类需要父类的功能，而对于功能主体子类有自己特有的实现方式（方法体不同），可以重写父类中的方法
3. 如果子类中重写父类中的方法，在使用向上造型创建对象时，虽然是父类类型的引用，但是在用该引用变量调用的重写方法是子类中重写后的方法

注意事项：

1. 父类中的私有方法不能被重写（父类中私有方法不能被继承）
2. 子类重写父类中的方法时，访问权限要大于等于父类的权限修饰符（建议等于）
   1. Public > protected > default > private
3. 父类中的静态方法，子类也必须通过静态方法重写

重载和重写的区别：

1. overload：重载🡪一个类中，方法名相同，参数列表不同（类型，个数，位置），与返回值无关
2. override：重写🡪发生在两个类中（可以是子父类），方法名和参数列表都相同，与返回值有关

（父类默认大于子类）

泛化：由子类提取公共的属性和方法，形成一个新的类，这个过程称之为泛化

1. 向上造型 🡪 父类（类型）引用指向子类的对象
2. Animal animal = new Dog();
3. 注意：使用引用变量能点出什么，看类型（多态的体现）

特化：由一个公共的类型，衍生其诸多子类的过程，称之为特化

1. 向下造型 🡪 强制类型转换 instanceof(防止类型转换异常)
2. Java.lang.ClassCastException
3. Instanceof：用于判断某个引用变量是否属于某一类，返回值为boolean。
4. Dog dog = (Dog) animal;
5. Animal t = new Dog();
6. Animal c = new Cat();
7. If(c instanceof Dog){Dog dc = (Dog)c;}

Public : 都可以用（本类、同包的类、跨包子类、跨包类）

Protected ： 本类、同包的类、跨包子类（有限定，不能直接跨包的父类创建对象调用，只能使用跨包子类创建对象调用）🡪 通过子类类型对象访问  
默认（default）：本类，同包的类

Private： 本类

一般实际开发中，常用的为public和private

Static修饰词：

1. 随着类加载而加载，类的成员变量（类变量）
2. 优先于对象存在
3. 被类的所有对象共享（共享的用静态static，特性是非静态），无论创建多少个对象，存在于方法的只有一份只会加载一次

成员变量：

1. 实例变量：属于对象。
2. 静态变量（类变量）：属于类的，所有对象共享一份。

注意：

1. 静态方法没有This关键字
   1. This随着对象创建而存在，静态比对象优先存在
2. 静态方法只能访问静态变量和静态方法
3. 非静态方法既能调用静态方法也能调用非静态方法，能访问实例变量，也能访问类变量

静态变量和实例变量的区别

1. 所属不同
   1. 静态变量属于类 🡪 类变量
   2. 实例变量属于对象 🡪 对象的变量
2. 内存中存储的位置不同
   1. 静态变量存储在方法区的静态区
   2. 实例变量存储在堆中
3. 在内存中出现的时间不同
   1. 静态变量随着类加载而加载，随着类消失而消失
   2. 实例变量随着对象的创建而存在，随着对象的消失而消失
4. 调用不同
   1. 静态变量可以通过类名调用（非私有的），也可以通过对象调用（不推荐）
   2. 实例变量是只能通过对象调用

Static代码块：

1. java中{}括起来的叫做代码块
2. 静态代码块：在类中，方法外，加上static修饰，用于给类进行初始化，在加载类的时候就执行，并且只执行一次一般用于加载驱动（JDBC）（声音，视频，音频，图片。。。）

Static优点：调用时简单，快捷，方便

Final：

1. 修饰成员变量：在声明时必须初始化，不可更改
   1. 声明的时候初始化
   2. 构造方法初始化
   3. 也可以修饰局部变量
2. 修饰方法：用final修饰的方法，该方法不能被重写（但是能被继承）
   1. 意义：防止子类在定义方法时造成“不经意的”的重写
3. 修饰类：用final修饰的不能被继承，但是可以继承别的类
   1. 意义：可以保护类不能被继承，可以控制滥用继承，对程序系统造成的伤害
4. Final和static搭配使用，用来修饰常量，常量名（大写）用以区别其他变量
   1. 例如：static final int NUM = 50;
   2. Static final 修饰常量，必须在声明同事初始化，不可更改，所有对象共有的
   3. Static final 在编译时会被直接替换具体的常量值
   4. 好处：效率高，方便维护