当子类和父类（接口和实现类）存在同一个方法时，子类重写父类（接口）方法时，程序在运行时调用的方法时，是调用父类（接口）的方法呢？还是调用子类的方法呢？我们将确定这种调用何种方法的操作称之为绑定。

 绑定又分为静态绑定和动态绑定。

静态绑定是在程序执行前就已经被绑定了(也就是在程序编译过程中就已经知道这个方法是哪个类中的方法)。

对于方法调用语句：引用变量.方法();     系统编译时采用的是什么绑定？**静态绑定**

对于方法调用语句：引用变量.方法();     系统运行时采用的是什么绑定？**动态绑定（private方法  static方法  构造方法 final方法除外）**

静态绑定首先在什么类型中寻找要调用的方法？ **引用变量所在的类型**

动态绑定首先在什么类型中寻找要调用的方法？**引用变量所指向对象的类型**

注：Java中只有private、static和final修饰的方法以及构造方法是静态绑定。

a、private方法的特点是不能被继承，也就是不存在调用其子类的对象，只能调用对象自身，因此private方法和定义该方法的类绑定在一起。

b、static方法又称类方法，类方法属于类文件。它不依赖对象而存在，在调用的时候就已经知道是哪个类的，所以是类方法是属于静态绑定。

c、final方法：final方法可以被继承，但是不能被重写，所以也就是说final方法是属于静态绑定的，因为调用的方法是一样的。

 总结：如果一个方法不可被继承或者继承后不可被覆盖，那么这个方法就采用的静态绑定。

动态绑定 编译器在每次调用方法时都要进行搜索，时间开销相当大。因此虚拟机会预先为每个类创建一个方法表（method table），其中列出了所有方法的签名和实际调用的方法。

动态绑定过程：

 <1>虚拟机提取对象的实际类型的方法表。

 <2>虚拟机搜索方法签名，此时虚拟机已经知道应该调用哪种方法。（PS：方法的签名包括了：1.方法名 2.参数的数量和类型~~~~返回类型不是签名的一部分。）

 <3>虚拟机调用方法

PS:由于动态绑定需要在运行时确定执行哪个版本的方法实现或者变量，比起静态绑定起来要耗时。所以在不影响整体设计，我们可以考虑将方法或者变量使用private，static或者final进行修饰。