方法调用阶段主要做什么事情？

方法调用阶段唯一的任务就是确定被调用方法的版本（即调用哪一个方法）

怎么确定被调用方法的版本？

编译期间，一切方法调用在Class文件里面存储的都只是符号引用 。

在类加载期间或运行期间符号引用会被解析成确定目标方法的直接引用。

总结：确定被调用方法版本的过程，其实就是方法符号引用解析成直接引用的过程。

什么是解析调用？

在编译期间通过符号引用就能确定调用的是哪一个方法，在类加载的解析阶段把符号引用转换成目标方法的直接引用。

“编译期可知，运行期不可变”。

这样的方法有：

invokestatic 静态方法

invokespecial 调用实例构造器<init>方法、私有方法和父类方法

final方法（使用invokevirtual指令调用）

这些方法可以称之为非虚方法。

什么是分派调用？

解析和分派这两者的关系并不是二选一的排他关系，

可以分为静态分派和动态分派。

静态分派也是在编译期间确定方法的版本。只不过

理解静态类型、实际类型及重载原理。

Number num = new Integer(1);

num = new Double(2D);

sayHello(num); //invokevirtual Method inc(Ljava.lang.Number;)

pbulic void inc(Number num){} \*\*\*

public void inc(Integer int){}

public void inc(Double d){}

Human是变量man的静态类型，有编译期可知；

Man是变量man的实际类型，在运行期间确认。

在编译期间，javac编译器会根据参数的静态类型决定使用哪个重载版本。

什么是静态分派？

所有依赖静态类型来定位方法执行版本的分派动作称为静态分派。静态分派的典型应用就是重载。