JVM执行方法调用

1. 重载与重写：对于父类与子类

重载：子类拥有父类非私有的同名方法，传入参数不同

重写：子类拥有父类非私有的同名方法，传入参数相同。如果这两个方法都是静态的，那么子类中的方法隐藏了父类中的方法。如果这两个方法都不是静态的，且都不是私有的，那么子类的方法重写了父类中的方法。

2. 静态绑定、动态绑定

JVM定位目标方法的关键是类名+方法名+方法参数类型+方法返回值类型，于是就出现了两种JVM找目标方法的方式，静态绑定、动态绑定

静态绑定
在解析时JVM便知道该调用那个目标方法

动态绑定
在运行时JVM需要根据对应的类类型来具体定位应该调用那个目标方法。对于方法重写，对应的类会拥有一个方法表（一个二维数组表，给方法标上序号，重写的方法序号一致）

3．虚方法调用

Java 里所有非私有实例方法调用都会被编译成 invokevirtual 指令，而接口方法调用都会被编译成 invokeinterface 指令。这两种指令，均属于 Java 虚拟机中的虚方法调用。

在绝大多数情况下，Java 虚拟机需要根据调用者的动态类型，来确定虚方法调用的目标方法。这个过程我们称之为动态绑定。那么，相对于静态绑定的非虚方法调用来说，虚方法调用更加耗时。

4．方法表

类加载的准备阶段，它除了为静态字段分配内存之外，还会构造与该类相关联的方法表。这个数据结构，便是 Java 虚拟机实现动态绑定的关键所在

方法表满足两个特质：其一，子类方法表中包含父类方法表中的所有方法；其二，子类方法在方法表中的索引值，与它所重写的父类方法的索引值相同。