(1)获得方法句柄：

MethodHandles.Lookup lookup=MethodHandles.lookup();

(2)通过lookup，可以查找到你想要的方法句柄

MethodHandle handle=lookup.bind(xxx,xxx,xxxx,xxxx)

(3)通过调用handle的invokeExact()方法，可以实现真正的方法调用

参数可变的方法(可变的参数，在其内部一个对象数组)

将可变参数转化为数组：

Handle.asController()---🡪不是放数组

Handle.asSpreader()---🡪放数组

参数绑定：

Handle.bindTo()在进行invoke()时，不需要传入参数。

方法的查找：

(1)查找类的构造方法：handle.findConstructor()

(2)查找类的实例方法：handle.findVirtual()

(3)查找类的静态方法：handle.findStatic()

(4)查找类的特殊方法：handle.findSpecial()--🡪参数注意

Reflect与方法句柄一起使用：

(1)通过反射获得方法

(2)调用handle.unrefiect()方法将其转化为句柄

(3)属性的获取与方法的获取相同

交换点。进行方法句柄是否可用的判断(一次性操作)

动态指令：invokeDynamic指令

(1)动态调用点 callsit，分为三类(1)ConstabtCallSit(2)MutableCallSit(3)VolatileCallSit(区别：三者包含的方法句柄性质不同)。。。。。启动方法：bootstrap方法：参数;lookup,name,MethodType 返回值：callsit对相

(2)调用callsit的dynamicInvoker方法=执行了Invokdynamic指令(测试用)

(3) 使用asm进行Java字节码的生成工作，