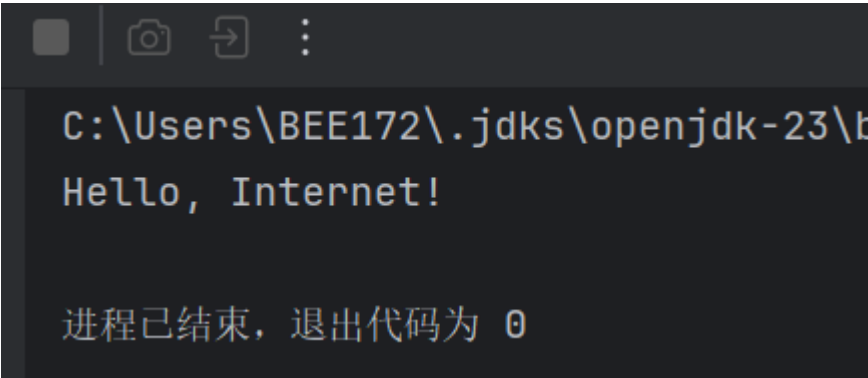


建立一个服务器和客户端

ServerSocketChannel类定义在java.nio.channels包中，用法和ServerSocket类似 客户端的go方法就是调用setUpNetworking()方法外加一个发送消息，接下来就是用printwriter写一个输出流，再加一个发送信息，很多代码在前面的题目都有用到，这次进行复用，同时把ip设成本机地址127.0.0.1. ServerSocketChannel不能使用socket.outputstream，在实际编写时有点被束手束脚的感觉，还不太理解使用channel和直接使用套接字的区别，优劣性（查资料发现**可以不用使每个mesocket连接使用一个线程，也避免了管理大量线程所需的上下文交换开销。借助新的 NIO 类，一个或几个线程就可以管理成百上千的活动 socket 连接了并且只有很少甚至可能没有性能损失**【<https://cloud.tencent.com/developer/article/1915356>】）最开始往setNetworking中加了一句sendMessage("Hello, Internet!");，运行客户端和服务端时发现没有反应，梳理代码中发现选的socket对象的输出流错了（我设了一个serversocket和socket（都是对象名），想让socket=socketChannel.socket() ;时却令serversocket=socketChannel.socket()。。。索性不创建了，并把两个都删了。随后发现服务器里的BufferedReader应该用inputstream，而我用了output。。。之后把sendmessage放在go里，再修改一会之后就能输出了 控制类与线程有关，进阶时才要用到线程，所以在第一阶段的代码里都删掉了

运行截图



改进你的服务器

进程与线程对比 进程是:一个应用程序（1个进程是一个软件）。线程是：一个进程中的执行场景/执行单元 **堆内存和方法区**内存共享。但是**栈内存**独立，一个线程一个栈 服务器控制类和run函数的编写参考了csdn的博客 <https://blog.csdn.net/u012225398/article/details/17150861> 想要多线程输出时****List clientWriters = new ArrayList<>();****就显得很重要了于是我改写了tellEveryone();使之向每个客户端输出。为了让客户端接收信息，我写了一个receiveMessages()，测试效果成立了，只展示客户端输出内容

