BIO是阻塞式的IO
1)等待客户端连接
2)等待客户端输入

把上面BIO两个步骤分开
把上面第二个步骤放到线程中去处理
即服务端接受了一个客户端连接，然后创建一个线程去等待客户端输入，然后继续去等待客户端连接

把上面BIO两个步骤分开
把上面第二个步骤放到线程中去处理
首先创建一个线程池
即服务端接受了一个客户端连接，然后从线程池中获取一个线程去处理等待客户端的输入，然后继续去等待客户端连接

存在的问题：
不能同时处理多个请求，不能解决高并发的问题

解决方案：使用线程

存在的问题：
1.利用线程可以同时接受多个连接，但是过多的线程需要OS切换，会影响性能
2.线程很耗性能，吃内存

解决方案：使用线程池
优点：
1.控制线程数据量
2.提前创建好线程

存在的问题：
如果并发量过大，而线程池中线程处理时间过长，将导致没有线程去处理过多的请求，即请求的响应时间过长，甚至请求遭到拒绝

归根结底 都是阻塞导致的

NIO替换
BIO