就和上次一样的格式写吧。

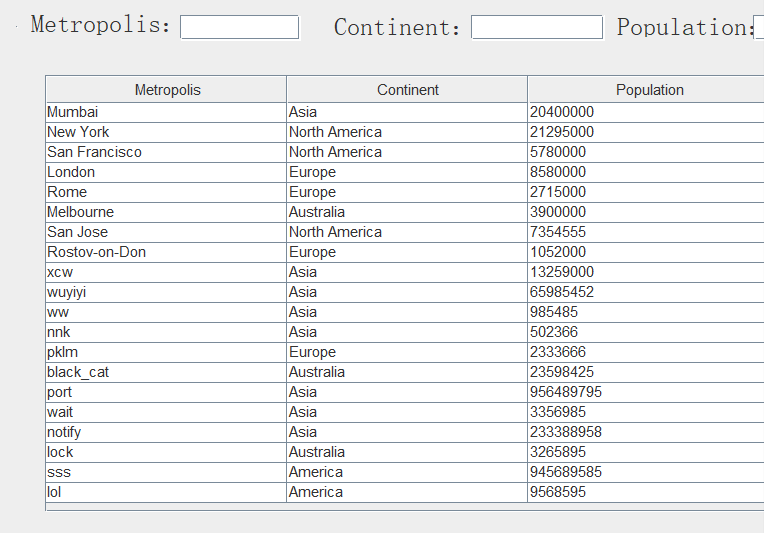
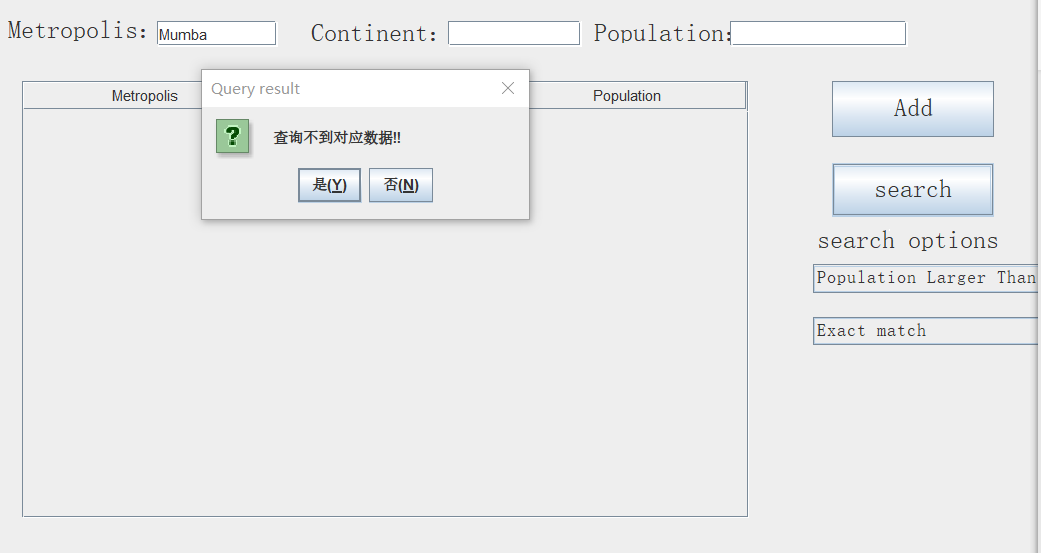
关于文件中的txt：一个



**这个是用来自动生成第五题的转账记录的东西，自己写一百条太麻烦了。**

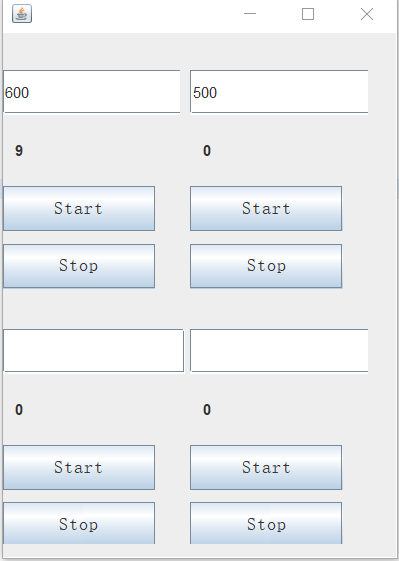
**第一题：**作为gui界面的第一场正式实践，这个题还是花了我不少时间的，从连接数据库，**因为不是自己的，而且每次连接之前，还要先打开数据库，用一下use当前数据库才行。**

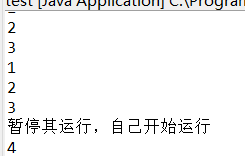
然后就是每个按钮和各样的不符合预期标准输入的异常抛出。



功能都算是办到了，也没什么要说的了，**就是大洲和校名需要是字符串，纯字符串，人口需要是数字。**

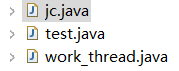
**第二题：**作为第一个线程的题，还是很麻烦的，最麻烦的就是让其他线程知道别的线程是否在运行，**不过我没有用题目中提到的SwingUtilities.invokeLater 而是用了一个 timer类用来计时**，但是这个类没有停止的方法，只有stop，但是线程中的stop被废弃了，我也不知道这个类是不是还能用，不过现在用下来没有什么问题





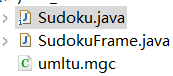
这里我是先运行了第一个，然后暂停，然后运行第二个，然后再运行第一个，受到第一个线程开始的信息之后，就会把第二个暂停且置0。

因为暂停的时候，用的是jc中的timer的stop，而不是把线程stop，其实相当于把计数器停止了，而没有线程被停止。

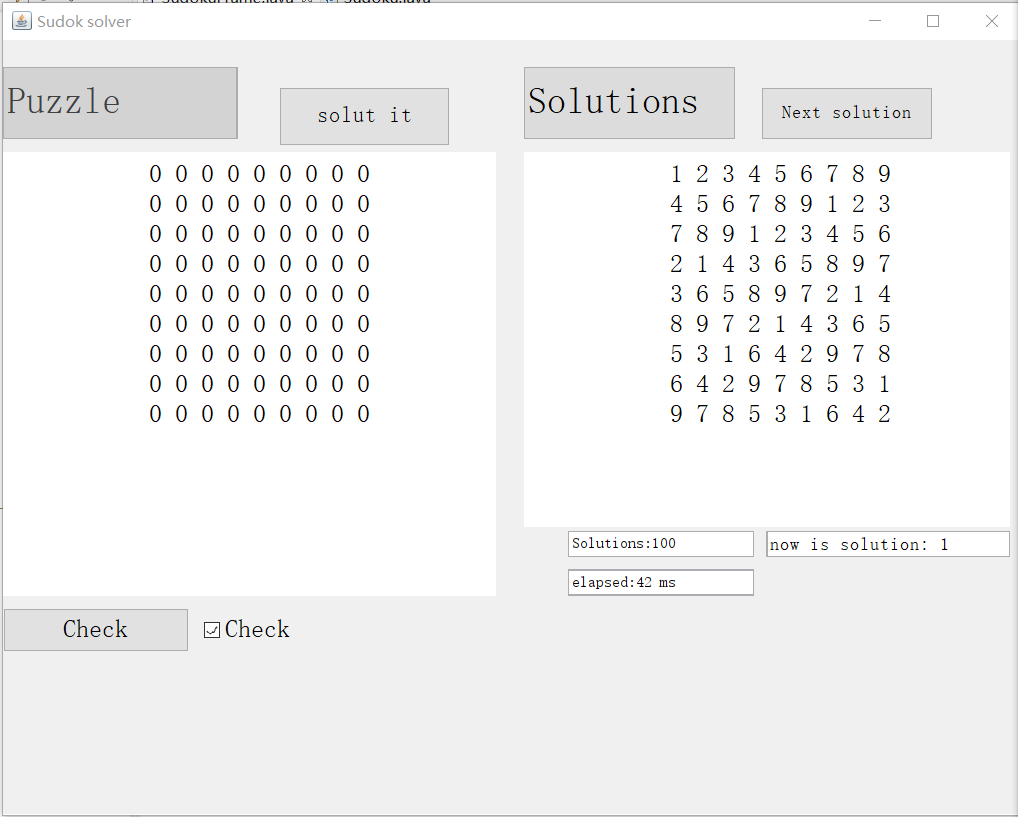
这个题我写了三个类

第一个是jpanel的子类，实现了ActionListener的接口，第二个是主类，是jfame的子类，运行的时候用这个，第三个是用来睡眠的类，因为每数到1000就会睡眠，因为10000太大我自己改成了一千。当然也有检测是否是数字来抛出异常。没了。

**第三题：这个题我写错地方了，我写在了我的jdbc\_4里，那个本来应该放第四题的地方。**



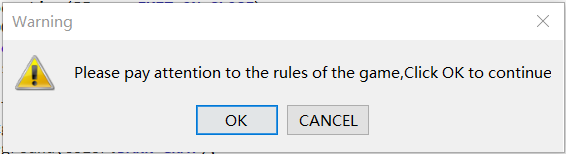
第一个是实现数独方法的类，第二个是显示的类，第三个是用来画uml图的，但是我是真的不会画。



这是我的界面，我介绍一下，一开始的初始文本是在外面设计的，我写了注释，在哪里，然后是按钮，我多了一个next solution，**因为题目说，如果解法超过100，那就找出100种，但是找出来了不显示就很奇怪，所以只要点next solution就可以查看下一种解法长什么样子。**

左边的文本是可以随意的改变的，改变的时候就会自动检测是否符合条件和数字数量。

还有一个问题，这个自动的check确实是没问题，但是它反应有点慢，我也不清楚他为什么那么慢，如果嫌弃它的话，可以手动点击check。



还有一开始我就提醒了使用者注意游戏的规则问题，这个点哪个都无所谓。

这题一开始的时候被坑到一样的感觉，因为题目的要求是找到所有的解，但是偏偏那三个例子都是唯一的解，让我寻找了好久我得不出全部解的问题，最后我自己设立一个全0的，果然有多个解。

因为我在寻找的解的过程中，是采用的remove的方法，设立一个1-9的arrylist，每当他在的行，列，正方形有相同的，就remove那个数字，这样剩下来的，就是这个位置可以填的数字，再于递归之中遍历这个不为空数组，得到的理所当然是全部的解。

还有确定正方形的时候，我是一个一个写的，后面看见了part=3才想到应该可以

For(i=0;i<3;i++)

For(j=0;j<3;j++)

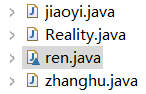
{

X=i\*part,y=j\*part;

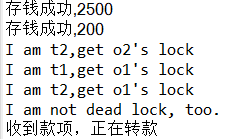
}

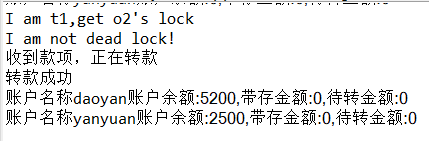
这样的实现，不过这样虽然可能会简洁的多，但是这样循环的次数也会变多，我的虽然空间效率低，但是时间效率高。

**第四题：这个写在了jdbc\_3的第四题，**这个题挺简单的，就是为第五题做准备吧



**运行这个reality就可以了，人是用来让他们死锁或者不死锁的，现在我设定的是不死锁，把wait和notify删了，就死锁了**





1号的余额是5000，代转2500给2,2的余额是200，待转200给1

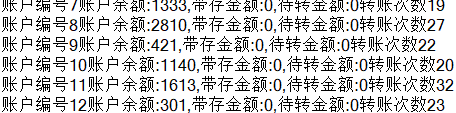
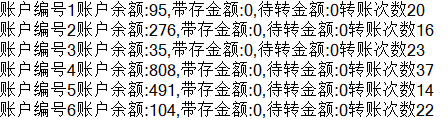
其他没什么要说的了，都写在了注释里。

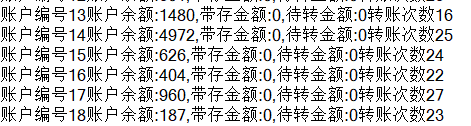
**第五题：这个题是真的挺麻烦的，**明明上一题还在想尽办法让他死锁，现在死锁自然而然的就出现了，还很难解决。

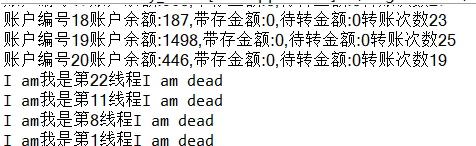


一开始是确定解决问题的线程的数量，因为我是自动生成的消费记录，所以我随便定了个范围（100,5100）

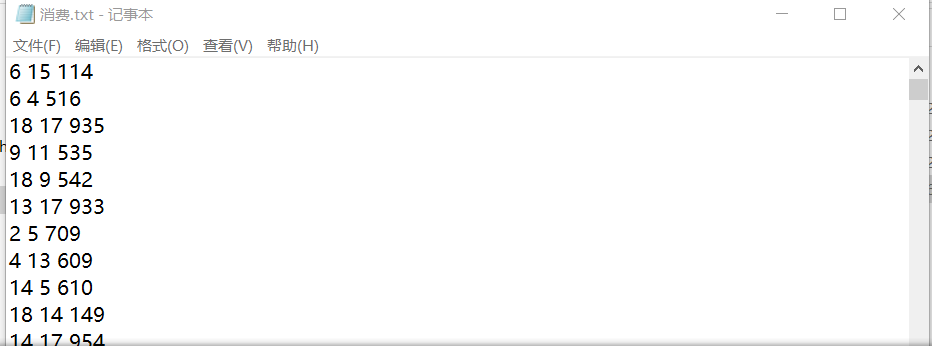
所以运行会比较慢







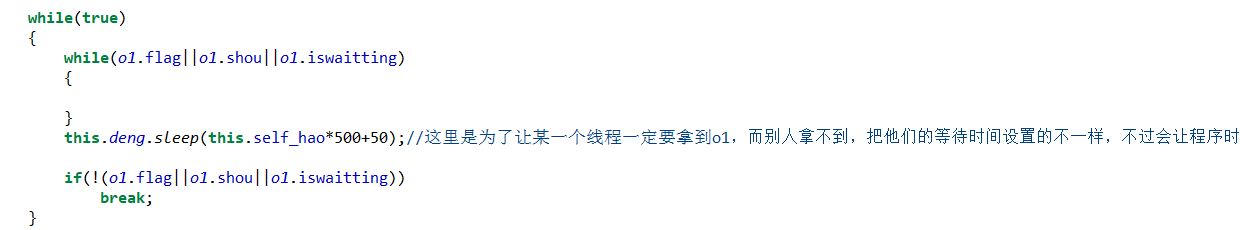
最后线程结束还会输出死亡，不是每一条转账记录都能成功转账，有些时候余额不足就会失败，所以所有账户的转账次数加起来小于等于转账记录数。



**这是生成的转账记录表。**

**算是记录在了主函数之中了，因为如果不重新运行，这个txt文件的数据就可以读出来。虽然我没有做，劳烦老师自己点开txt看了。**

这里我发现用wait和notify很麻烦，因为你notify的不一定是那个在等待你的线程，而一旦唤醒错误，死锁就会出现，更不用说使用notifyAll了，所以我手动写了一个wait

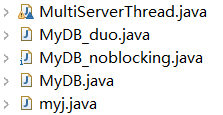


这样就解决了死锁的问题，不过又因为转账记录很多，确实让程序的时间长了不少。

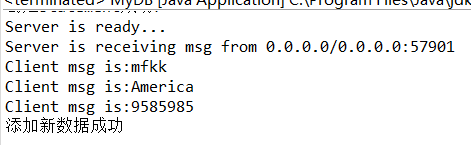
**对了忘记说，上面的题，我大都重新过tostring了，如果确实用得到的话**

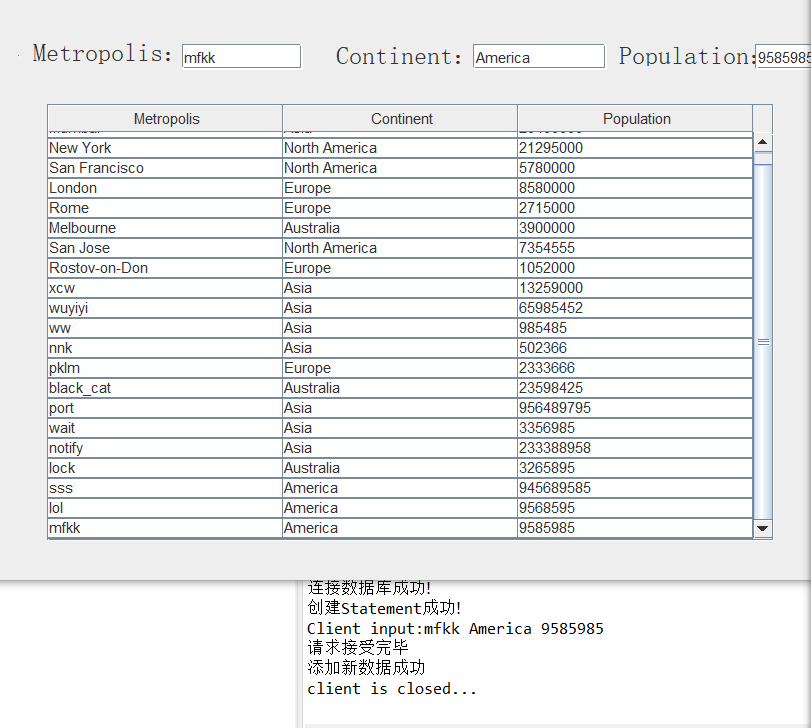
**第六题：这个题很有意思，很大的加深了我对网络编程的印象，我有一点困惑，对于端口，一个端口是不是只能由一个客户使用呢，这样绑定端口是不是达不到第二问的标准呢，还是说这个端口只是个工具，大家都可以在这里用呢。我找了好久，网上都只有解答接口的问题，没有端口的，还得劳烦老师了。**

第一个问，我感觉很简单，就是让他们接到同一个接口上，然后用阻塞的accept方法接受来自客户的信息，在检测信息正确后，就添加到数据库中就可以了，但是后来我发现可以省去检测这一步，因为数据库自己会报错，不过对于输出数据的类型，我还是抛出了几个数据类的异常。

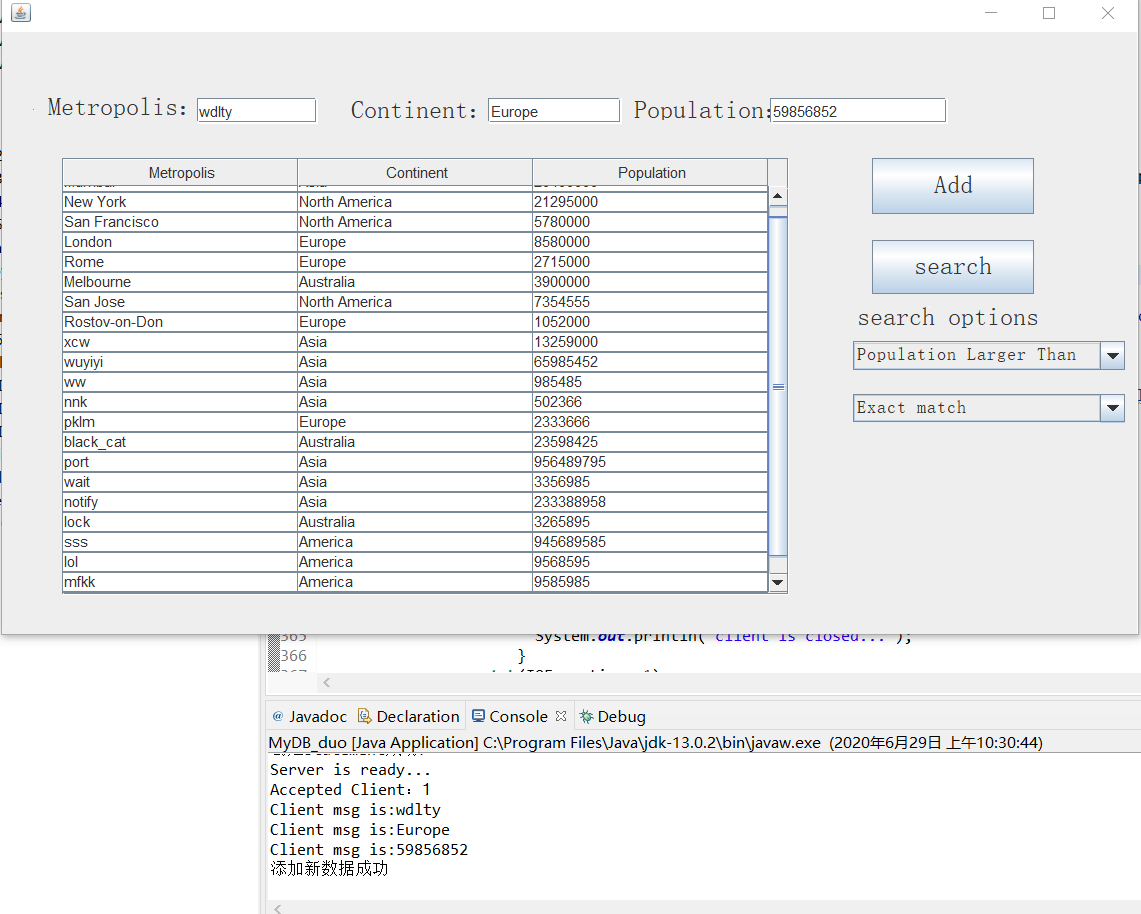


第一个是用来实现1对多的线程，第二个是一对多的主线程，第三个是noblacking，第四个是一对一的，第五个是客户端方的。

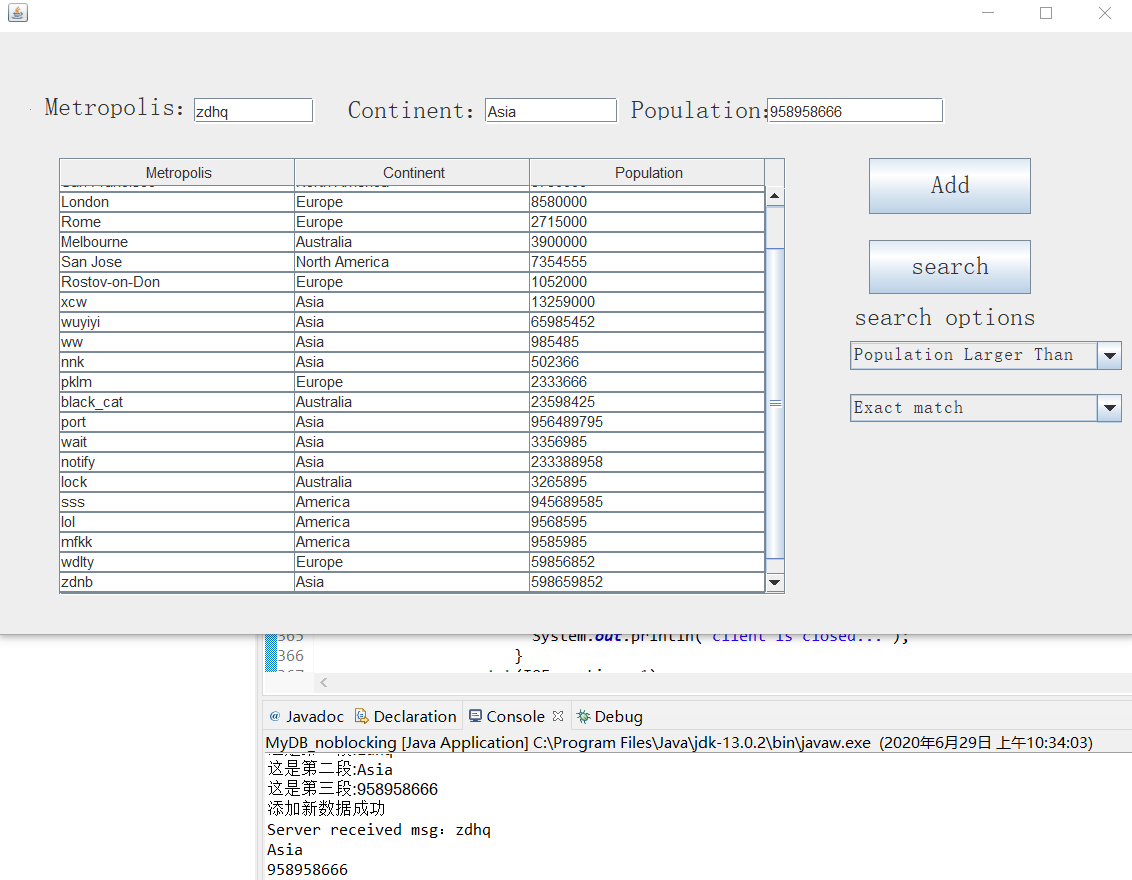




这就是第一题的。第二题也差不多，就是看了视频之后，学习到了那个在结束的时候new一个线程的方法，也就是主线程只负责接收，而其他线程负责处理，即可以一对多。



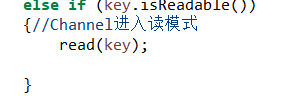
这里添加的时候也可以看见第一题的那个内容已经添加成功了。

下来是第三题的

这个题很神奇，他的写给客户的回复很难自定义，怎么都要带上客户的输入，所以在我的myj的353行，对运行第三个题目有一个要求，运行前两个用一个while，而第三个应该是另外一个while

其次就是多线程，我看了好几遍ppt和视频，还有网上查了很久，我觉得这个程序直接运行就是1对多的，因为他可以同时读和写多个信息

我一开始想尝试像2中一样，在他的



这里新建线程来帮助他处理内容，但是我发现不行，是因为在读取的过程中，在key处于isacceptable的时候。他也经常会处于isreadable，这样会导致一个客户的请求让他新建多个线程去处理，而有些线程没有收到信息，就会报错。

所以我决定让他自己来read

由于没有像第二题一样的让他新建别的线程来处理事件，所以我不确定他是否可以一对多的处理信息。

所以我新建了几个别的线程给他发消息试一试，我新建了三个线程，加上请求处理数据库的是四个，他处理的顺序却并不是1234，但是很可惜当时的代码我已经删掉了，不过这样的情况，我就认为他确实是已经属于1对多了。

当然如果不是的话，也希望老师给我批评指正。

**这就是这次的作业了，不得不说还是很快乐的，gui界面第一次让我觉得我活到现代社会了，而不是以前一直对着个黑框，很难受，数独也很有趣，这个界面和这样的方法为我提供了一些用java来制作一些小游戏的想法，放假了有时间我一定试一试，然后就是线程，对于这个经常在别人口里听到的词，现在也是真正明白了许多，最后的网络编程nio也是，明白了各个不同的进程，不同的客户端之间是如何联系的，收获不比前两次作业小，最后祝老师假期快乐吧，希望北京的疫情能尽早控制下来，我们也能回去上课。**