课程设计报告二：聊天室系统

1. **课程设计要求与目的**

目的：编写一个小型Java聊天室系统，掌握Java网络通信、多线程、IO文件操作等高级应用编程技能。

要求：以课本第15章 Java网络通信例15.3、15.4的源代码为基础，编写一个小型Java聊天室系统。

完成如下功能：

1、多客户端模式下，实现客户与客户的单独通信，要求信息通过服务器中转

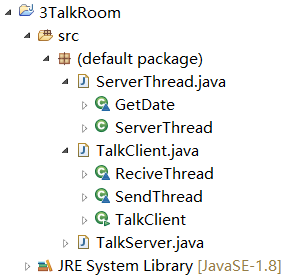
2、端到端的通信，实现并行通信模式（不再是你说一句，我说一句，一端的信息发送不受另一端的影响）

3、实现端到端的文件传输

4、添加图形界面（选做）

1. **系统设计**

**I.**



TalkClient.java：

每启动一个客户端，将会新建三个线程，分别用来发送文件，发送消息和接收消息。

其中，在发送消息和接收消息的run()实现里，分别用死循环接收和发送消息，即若客户端socket套接字的输入流里有数据则接收消息并进入下一个循环，没有就阻塞等待；发送消息同理。

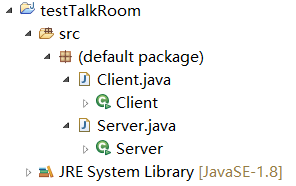
TalkServer.java:

其工作是给要连接的两个客户端分配线程。其中多客户段的存储用List列表实现。

ServerThread.java：

实现服务端接收客户端文件的功能以及服务端作为消息中转站的功能。

**II.**



Client.java：

实现客户端的界面，以及消息的收发。

Server.java:

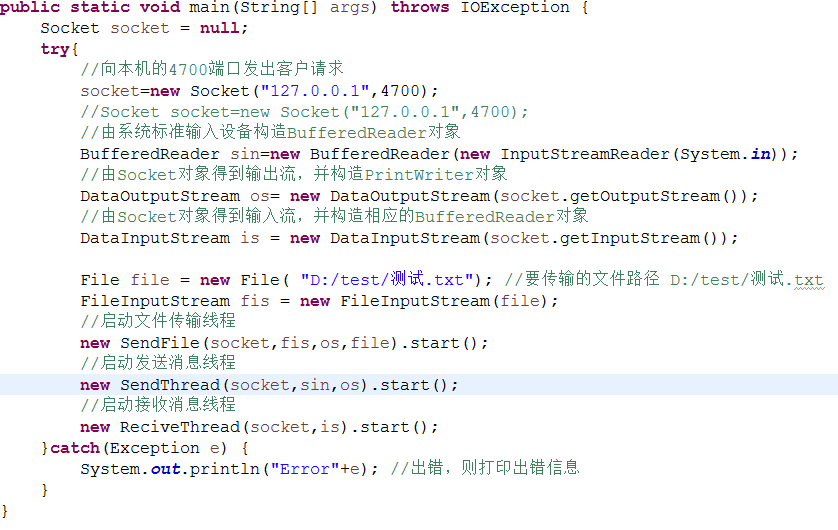
实现服务端的界面，以及消息的接收和转发。

1. **系统实现**

**Ⅰ.3TalkRoom:**

**TalkClient.java:**

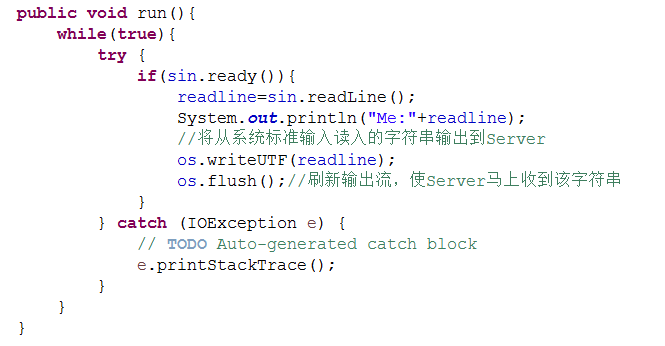
main()方法主线程里：



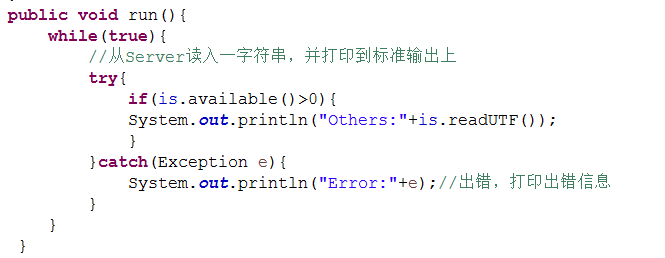
SendFile线程的run方法：



SendThread线程里的run方法：

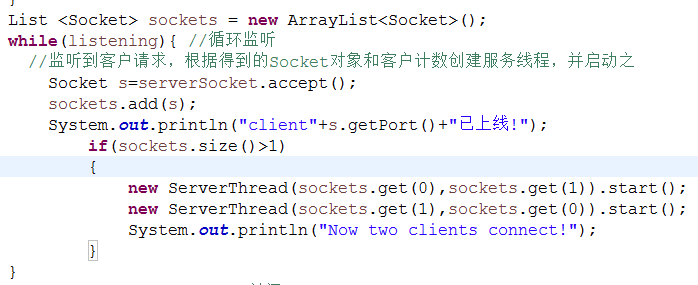


ReciveThread线程里的run方法：



**TalkServer.java:**

其余代码类似书上的例题，只给出关键代码如下（只实现了两个客户登录立即自动链接的功能）：

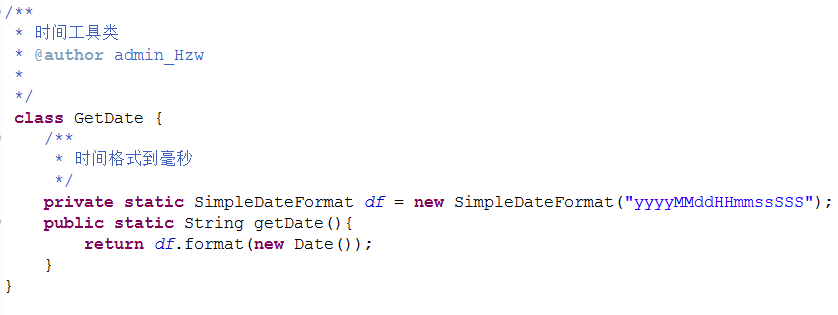


**ServerThread.java:**

客户端接收文件的存储路径：

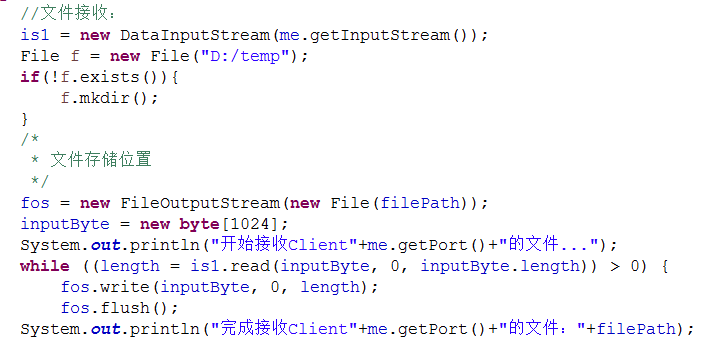
H_}{H9)RKD6L0N7@X[6%}$F

其中GetDate类如下：（其作用是为了更清楚地命名接收到的文件）

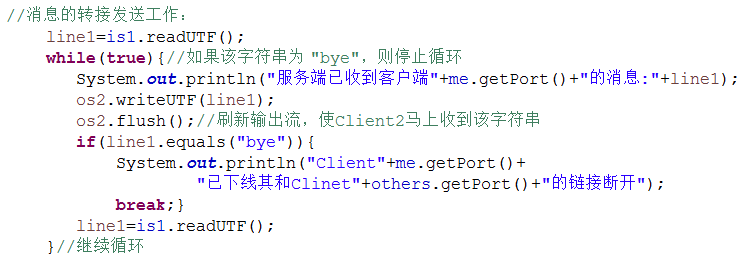


下面给出run函数中的关键代码：

文件接收功能实现：



接收客户端的消息并转发给另一个客户端的功能实现：



**Ⅱ.testTalkRoom（代码来源网络，作了改动及注释）:**

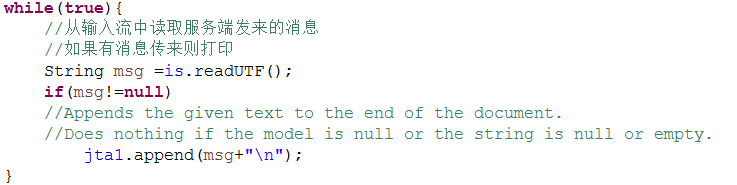
**Client.java：**

下面给出部分关键代码：

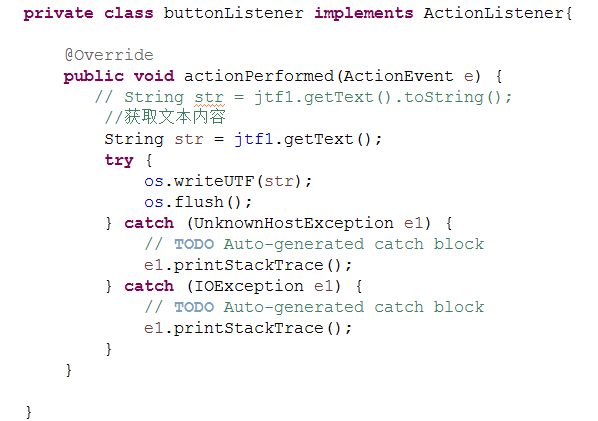
界面的实现（包括界面的大小，界面按钮，以及标题栏设置）：



客户端的消息接收：

****

若客户端界面的文本输入框有输入且点击了发送按钮，则向socket输出中发送消息：

****

**Server.java:**

具体说明见代码文件中注释：

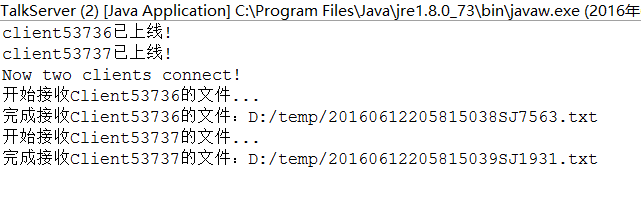
****

****

1. **系统测试**

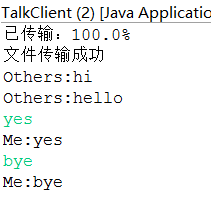
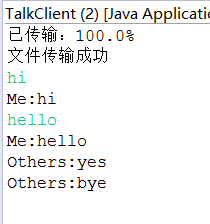
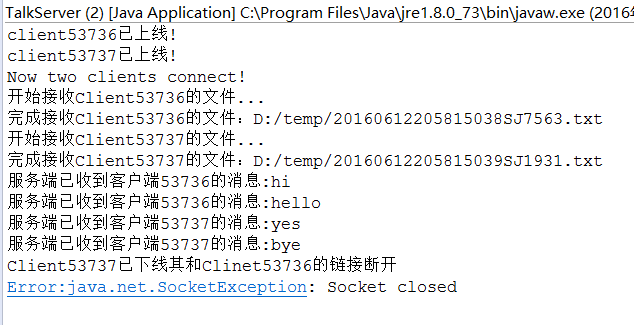
1)3TalkRoom:

运行TalkServer和两个Client后显示为：



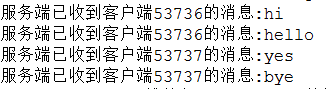
![U%T](GMLVROFPMWE1_AUNGT](data:image/png;base64,)（两个客户端的显示相同）

在发送消息后：

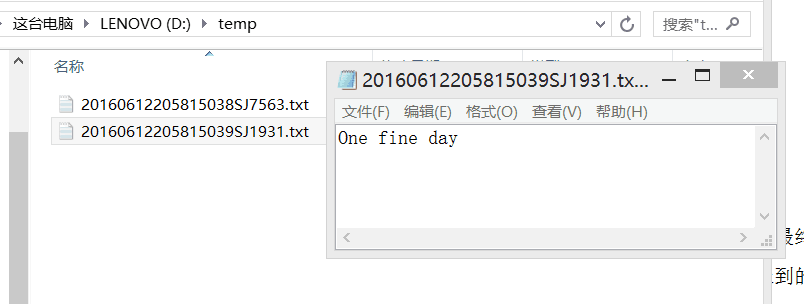
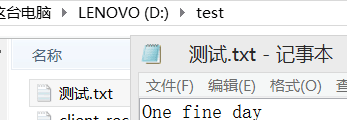


分析：1.和4FQ~`T[PYUTTY$UWA@84XZY证明实现了并行通信模式。

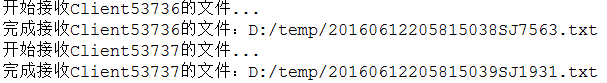
1. 在一方说出bye后，两个客户端之间的socket链接关闭，客户端给出显示OE]_7OUY1LW{P`G200F{[1D
2. 服务端可以监控每个客户端的发言，两个客户端之间的通信消息通过客户端中转：



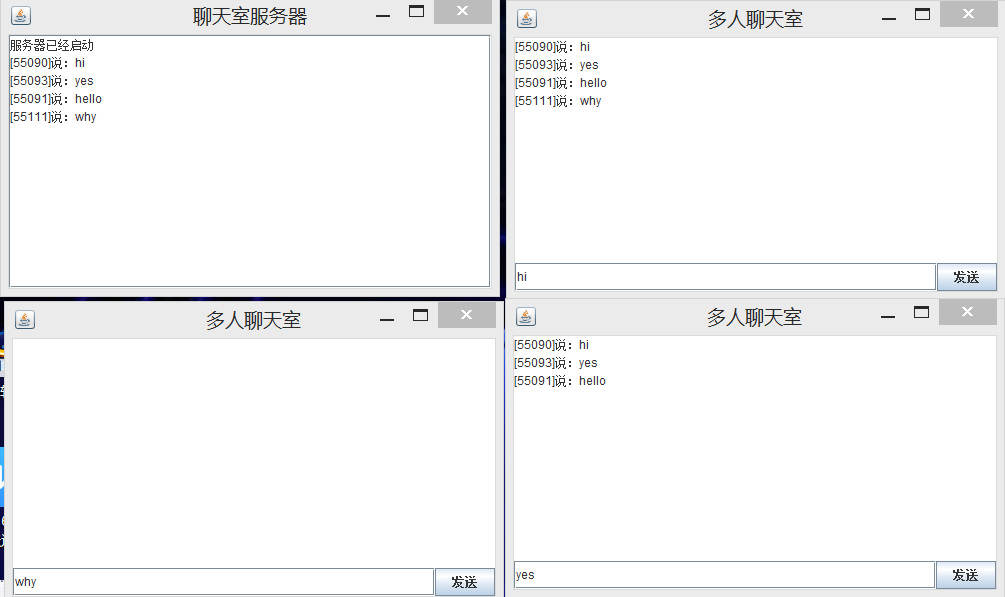
4.传输文件功能的实现（将D:/test/测试.txt发送至D：/temp）：



服务端有显示最终接收到的文件路径及名称：



1. testTalkRoom:



分析：该程序实现了一个多人聊天室的功能，一个客户端发出的消息可以被多个客户端接收到，中途有客户端下线或者登录都不影响聊天室消息接收发送的稳定性。

1. **课程设计总结**

Ⅰ.最终实现的系统功能：

3TalkRoom:

1. 两个客户间可以单独通信，信息通过服务器中转（局限性仅限两个用户）
2. 端到端的通信，实现并行通信模式（不再是你说一句，我说一句， 一端的信息发送不受另一端的影响）

3、实现客户端到服务端的文件传输

testTalkRoom（代码来源网络，自己作了改动及注释）:

1、多客户端模式下，实现聊天室的功能，多客户能一起交流，信息通 过服务器中转

2、端到端的通信，实现并行通信模式（不再是你说一句，我说一句， 一端的信息发送不受另一端的影响）

3、实现了图形界面

Ⅱ.遇到的问题、问题的解决以及收获：

1. 用List列表存储多个客户端的套接字。本次实验中用到的方法有：

实例化：List[<数据类型>] list = new ArrayList[<数据类型>]();

获得集合内元素个数：list.size()；

添加元素：默认添加：list.add(e)；

取出元素：list.get(index)；

1. String类型不能用==或!=比较和目标内容是否相当。应当调用String.equals(“目标内容”)方法。原因：==或!=比较的是String类型的引用，而equals()方法比较的才是String的内容。
2. BufferedReader的时候用readline()方法会将线程阻塞，换用DataInputStream和DataOutputStream就可以实现了。
3. 调用String line=ins.readline()方法，若未读取到内容，其判断条件应是if(line!=””)，而不能用if(line!=null)。
4. 判断socket套接字的输入流中是否有可接收的数据可以用is.available()>0；判断系统标准输入流中是否有数据等待发送可以用sin.ready()。