浙江大学计算机学院

Java 程序设计课程报告

2017—2018 学年秋冬学期

题目	多人聊天室设计开发
学号	3150103457
学生姓名	邓墨琳
所在专业	计算机科学与技术
所在班级	1502 班

目　录

1　引言..... 1
1.1 设计目标..... 1
2　总体设计..... 1
2.1 功能模块设计..... 1
2.2 流程图设计..... 1
3　详细设计..... 3
3.1 UI 布局设计..... 3
3.2 用户信息类 User.java 设计..... 3
3.3 服务端 Server 类设计..... 3
3.4 用户端 Client 类设计..... 4
3.5 Server 类中内部类 ServerThread 设计..... 5
3.6 Server 类中内部类 ClientThread 设计..... 5
3.7 Client 类中内部类 MessageThread 设计..... 6
3.8 消息发送机制..... 7
4　测试与运行..... 8
5　总结..... 14

1 引言

本次编写的是一个简易的多人聊天室，这是一个综合性的题目，其中运用了 GUI 编程和网络编程，提高了自己的编程水平，为以后的学习工作打下了一定基础。

1.1 设计目标

(1) 使用 GUI 编程。用户界面通过 GUI 实现。

(2) 支持多组用户同时使用。

(3) 同时使用数据库(未实现)。

2 总体设计

2.1 功能模块设计

本程序需实现的主要功能有：

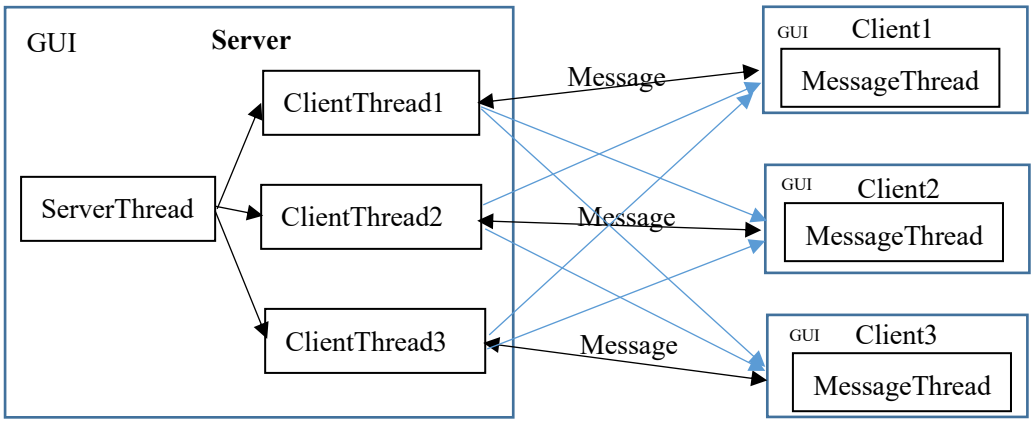
(1) 创建服务端(Server)和多个客户端(Client)；

(2) 在 Server 中实现多线程，每个线程负责于一个 Client 通信

(3) 通过 Server 实现 Client-To-Client 通信

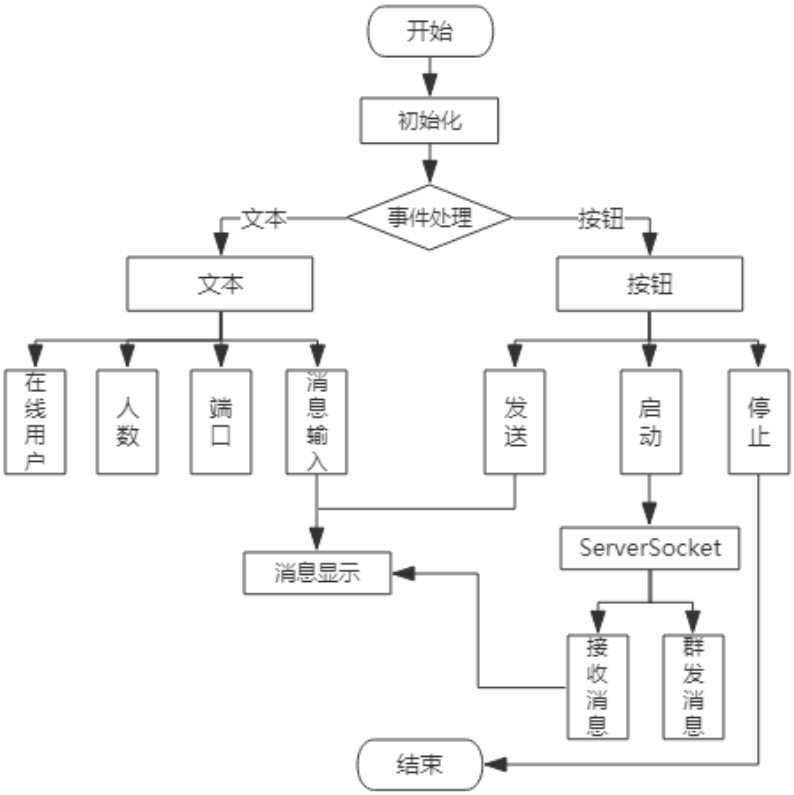
(4) 通过 Swing 设计实现图形窗口界面，响应响应事件。

程序的总体功能如图所示：

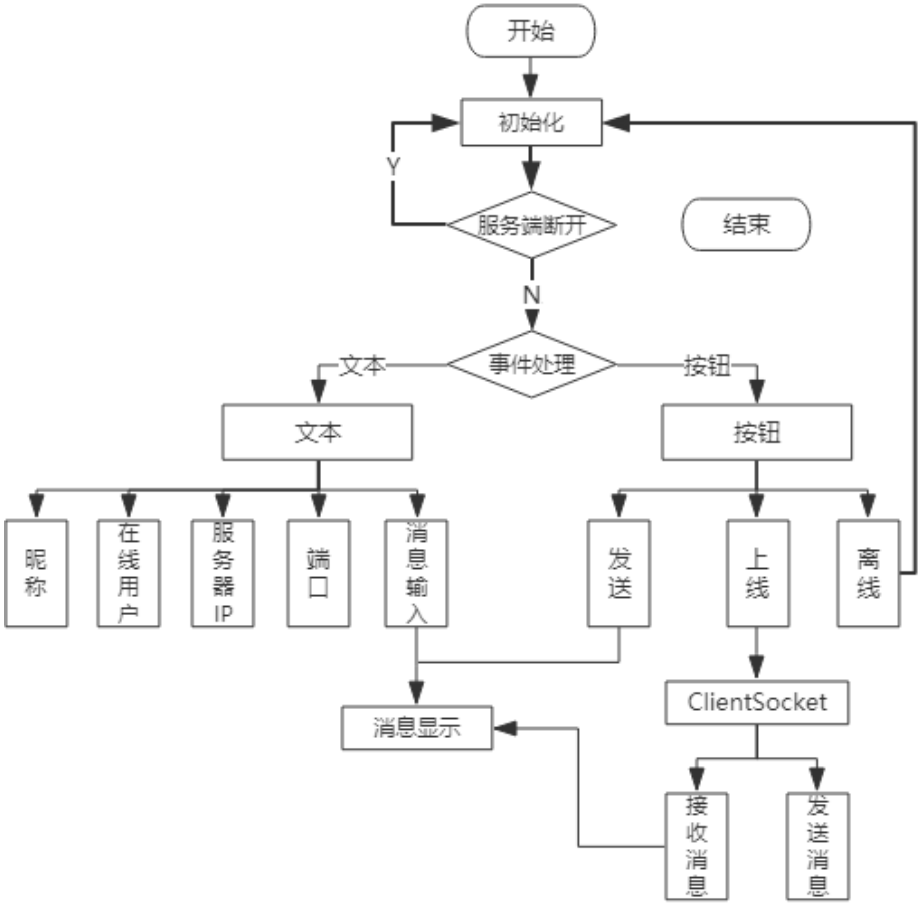


2.2 流程图设计

Server 服务端流程如下图所示：



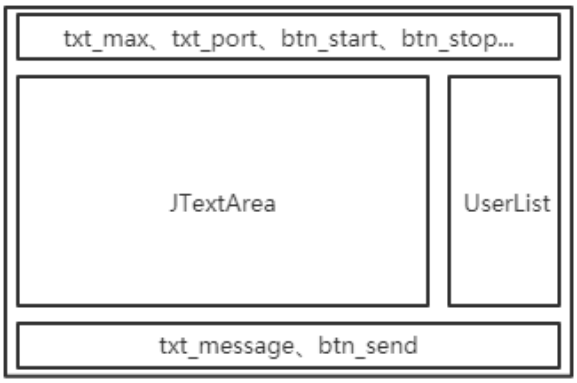
Client 客户端流程如下图所示



3　详细设计

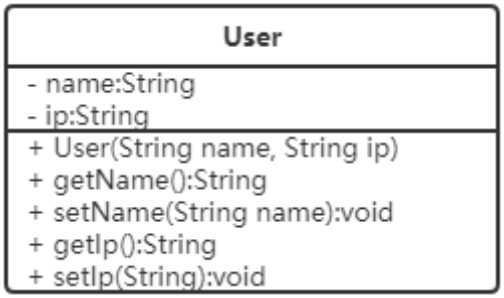
3.1 UI 布局设计

服务端、用户端窗口布局策略均采用边布局(BorderLayout)，将容器内的空间分为东、西、南、北 4 个区域，每个组件布局如下如图所示



3.2 用户信息类 User.java 设计

User 类用来储存用户的 IP 地址和用户名，UML 图如下所示



成员变量 name、ip 类型均为 String，方法 getName()、setName()分别用来获取用户名和设置用户名；方法 getIp()、setIp()分别用来获取、设置 IP 地址。

3.3 服务端 Server 类设计

Server 类中实现了窗口的创建与布局，事件的监听利用匿名内部类实现多个内部类响应不同组件产生的各种事件，Server 中创建内部类 ServerThread 处理服务器与用户的连接，内部类 ClientThread 处理服务器与用户之间的通信，一个 ClientThread 对应一个用户，在 Server 类中定义类型为 Client Thread 的数组，实现一个服务器与多组用户进行交互，进而实现了用户间的通信。

(1)成员变量(Swing UI 组件在此省略)

①-serverSocket:ServerSocket 服务器 Socket

②-serverThread:ServerThread 服务器线程，ServerThread 为内部类。线程不停等待用户的连接，接受用户的基本信息，反馈连接成功的信息，一旦成功连接，开启一个用户服务线程 ClientThread，并更新用户数组，更新在线用户列表。

③-clients:ArrayList<ClientThread> 用户组，每一个元素是一个用户服务线程，类型为内部类 ClientThread,用于接受用户端的消息，向用户下达下线命令，同时将该用户发送的消息转发给其他用户，实现聊天功能。

④-isStart:boolean 服务端是否启动。

(2)方法

①+CreadWindows():void 窗口初始化，实现布局设计

②+send():void 处理消息发送异常，调用 sendServerMessage()群发服务端发送的消息

③+Server() 构造方法。实现事件监听匿名内部类；设置房间人数、端口，调用 serverStart()启动服务端。

④+serverStart():void 创建用户数组，创建 ServerSocket，创建并运行服务器线程 ServerThread

⑤+closeServer():void 关闭服务器，清空用户列表

⑥+sendServerMessage():void 将从服务器输入的消息，群发给用户

3.4 用户端 Client 类设计

Client 类实现了窗口的创建与布局，事件的监听利用匿名内部类实现多个内部类响应不同组件产生的各种事件，同时定义内部类 MessageThread 接受来自服务器的指令，进行更新上线列表、接收服务端发送的消息和转发其他用户的消息。

(1)成员变量(Swing UI 组件在此省略)

①-isConnected:Boolean 判断是否与服务器连接

②-socket:Socket 用户端 socket

③-messageThread:MessageThread 消息线程，接收服务器发送的命令，处理上线列表的更新、接收服务端发送的消息和转发其他用户的消息。

④-onLineUsers:Map<String, User> 储存在线的用户信息

⑤-writer:PrintWriter

⑥-reader:BufferedReader

(2)方法

①+CreadWindows():void 窗口初始化，实现布局设计

②+send():void 处理消息发送异常，调用 sendMessage()向服务端发送消息

③+Client() 构造方法。实现事件监听匿名内部类；设置房间 IP、端口，调用 connectServer()启动用户端

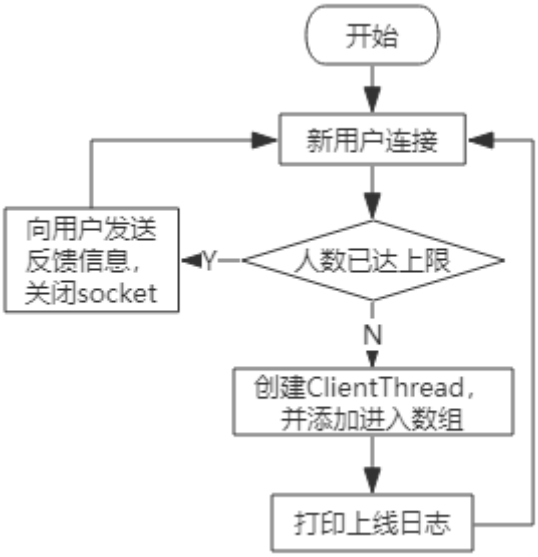
④+connectServer():boolean 连接服务器，启动消息线程 MessageThread。成功连接返回 ture

⑤+sendMessage():void 向服务端发送消息

⑥+closeConnection():boolean 向服务器发送主动关闭请求，停止消息线程，成功关闭返回 ture

3.5 Server 类中内部类 ServerThread 设计

ServerThread 继承了 Thread 类，不停等待用户端的连接，流程图如下所示



3.6 Server 类中内部类 ClientThread 设计

ClientThread 继承了 Thread 类

(1)成员变量

①-socket:Socket

②-reader:BufferedReader

③-writer:PrintWriter

④-user:User

(2)方法

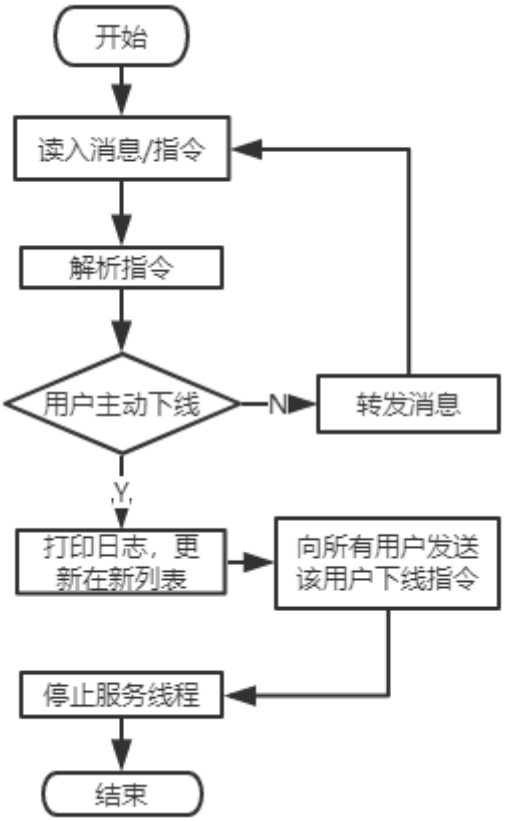
①+getReader():BufferedReader 返回 reader

②+getWrite():PrintWriter

③+getUser():User

④+dispatcherMessage(String message):void 将该线程服务的用户所发送的消息转发给其他用户，实现多人聊天

⑤+run():void 解析用户请求，调用 dispatcherMessage()转发消息，流程图如下所示



3.7 Client 类中内部类 MessageThread 设计

MessageThread 继承了 Thread 类

(1)成员变量

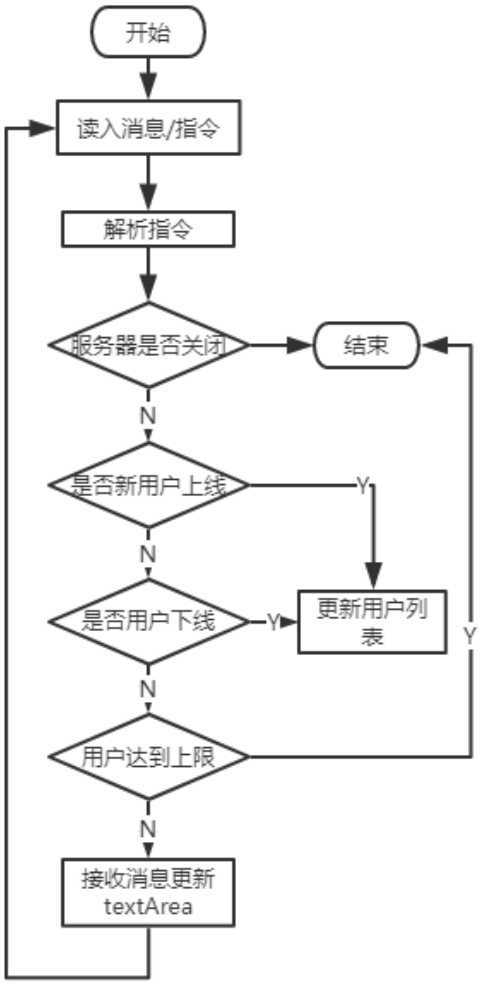
①-reader:BufferedReader

②-textArea:JtextArea

(2)方法

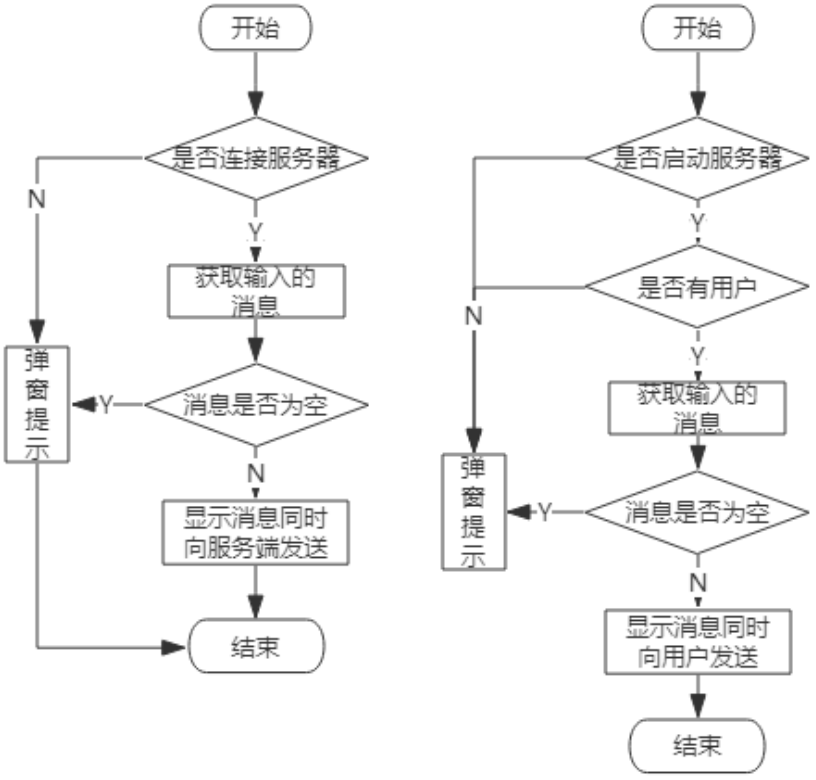
①+MessageThread() 构造方法

②+closeCon():void 被动关闭连接
③+run():void 解析服务器发送的指令，更新线上用户列表，接收消息，流程图如下图所示



3.8 消息发送机制

发送消息时服务端和用户端处理流程如下图所示



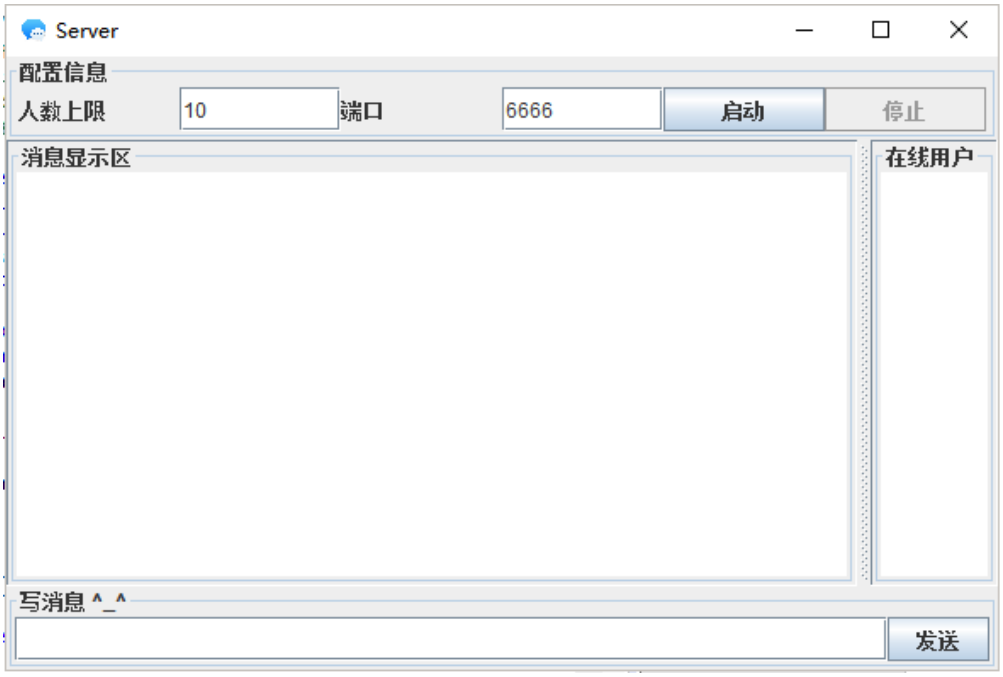
Client　　Server

4　测试与运行

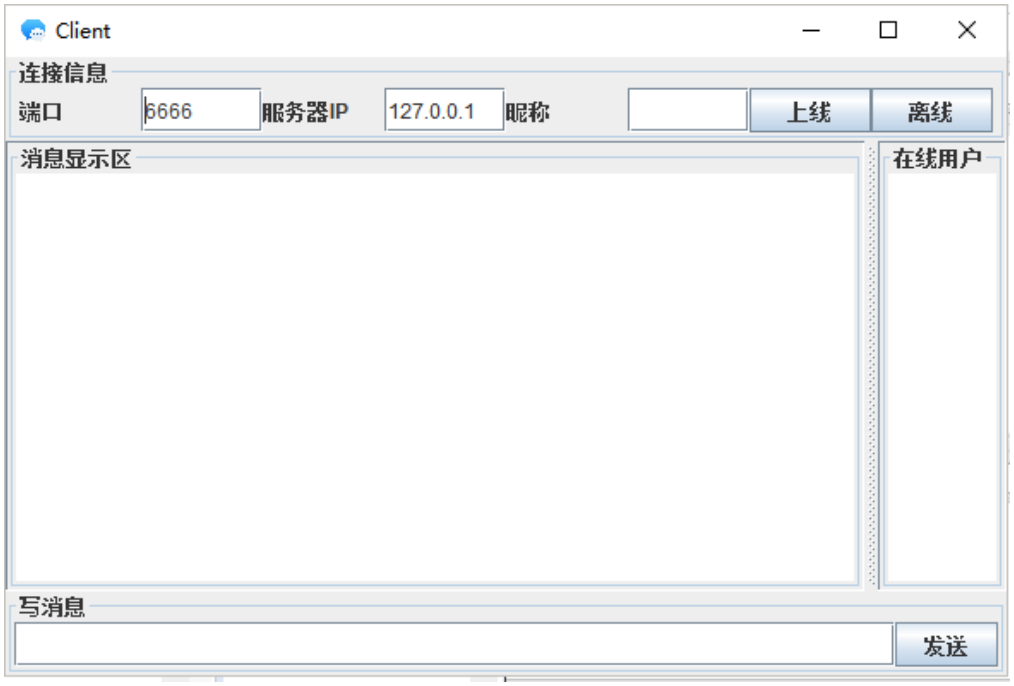
操作说明：

运行 Client.java 创建一个用户、运行 Server.java 创建一个服务器

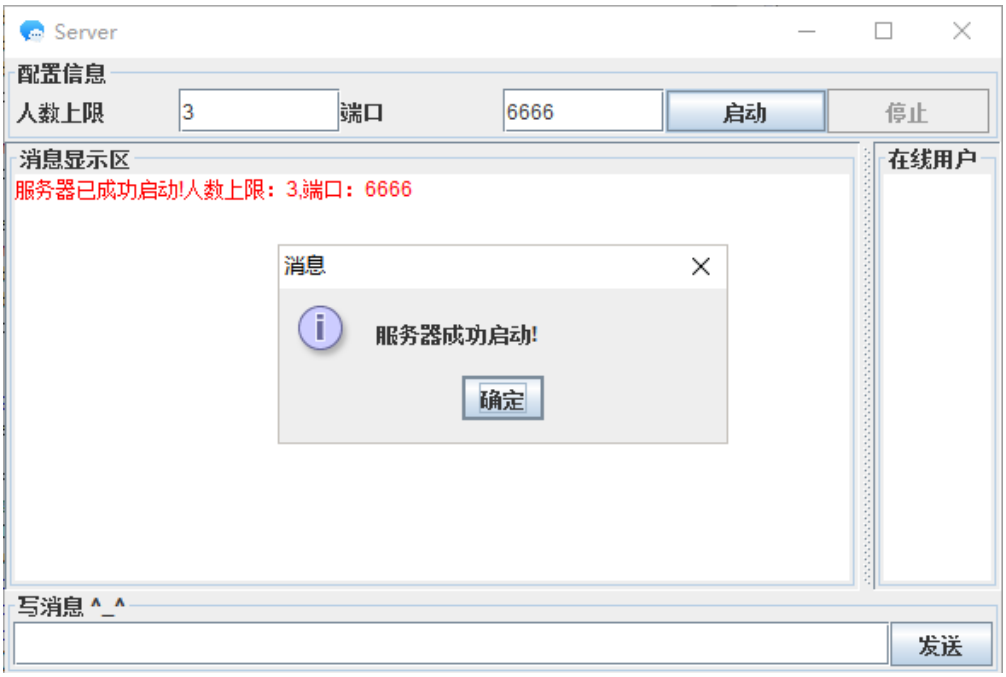
服务端运行界面如下图所示



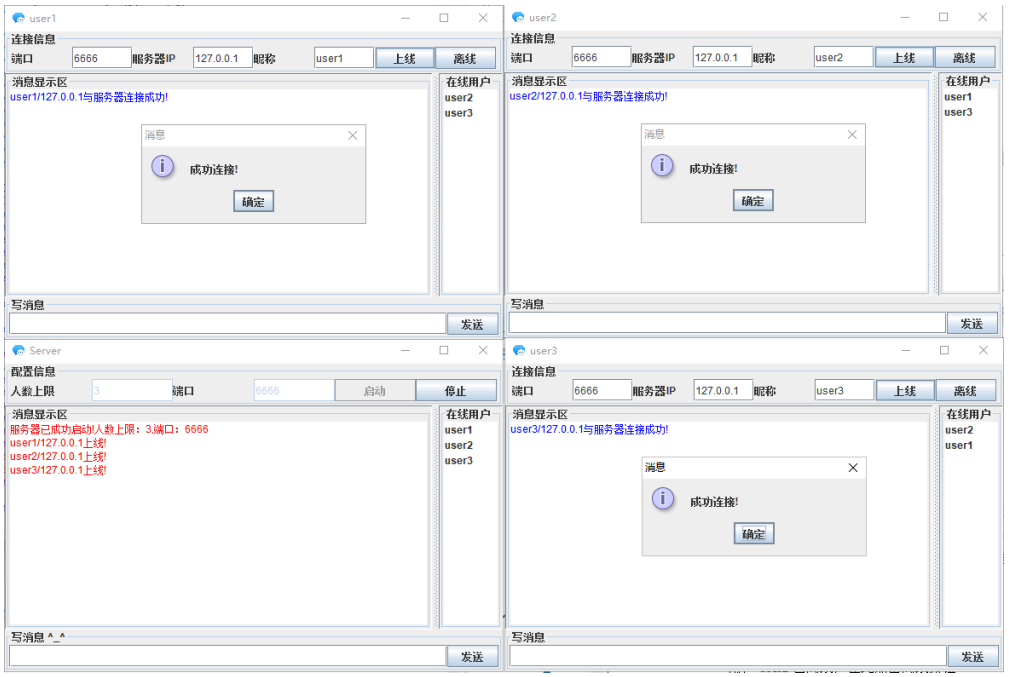
用户端运行界面如下图所示

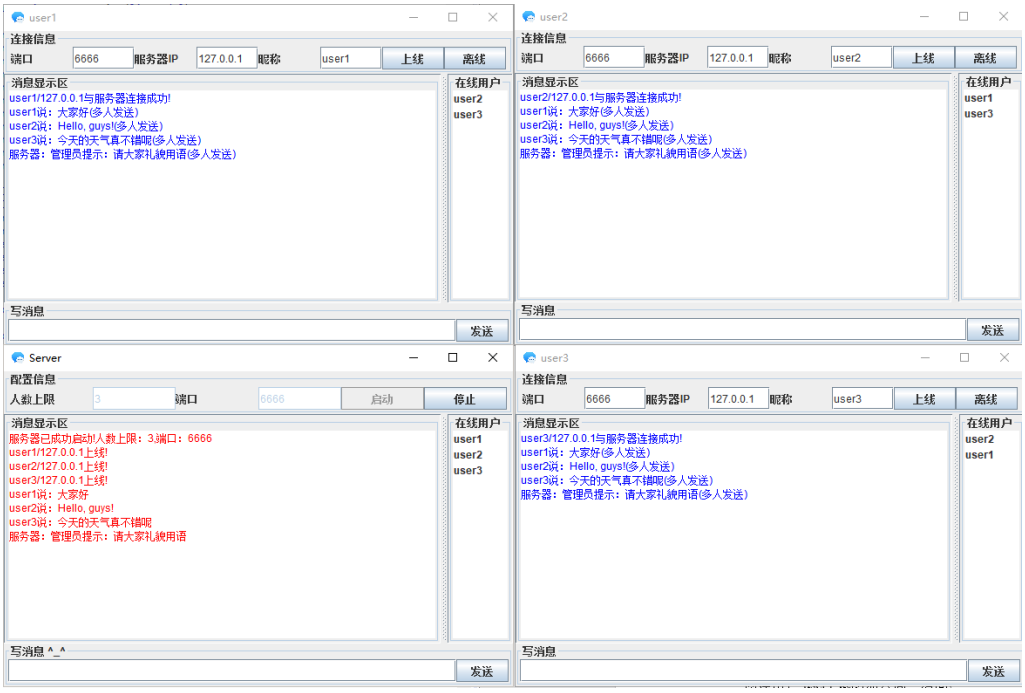


启动服务器，端口设为 6666，设置房间人数为 3 人

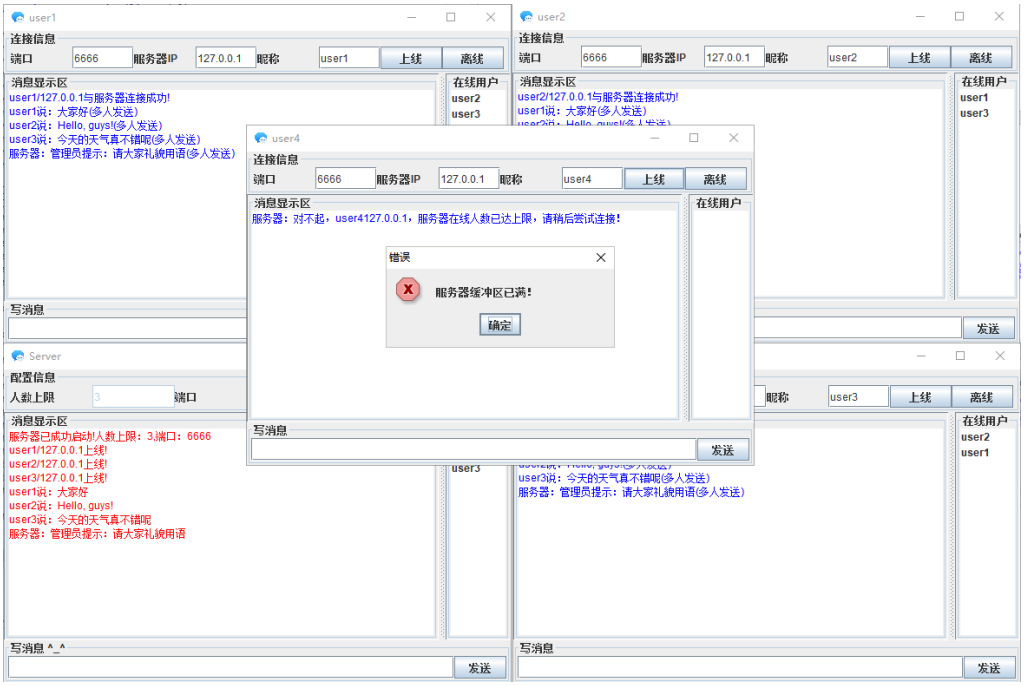


创建用户 user1、user2、user3 加入房间，开始聊天

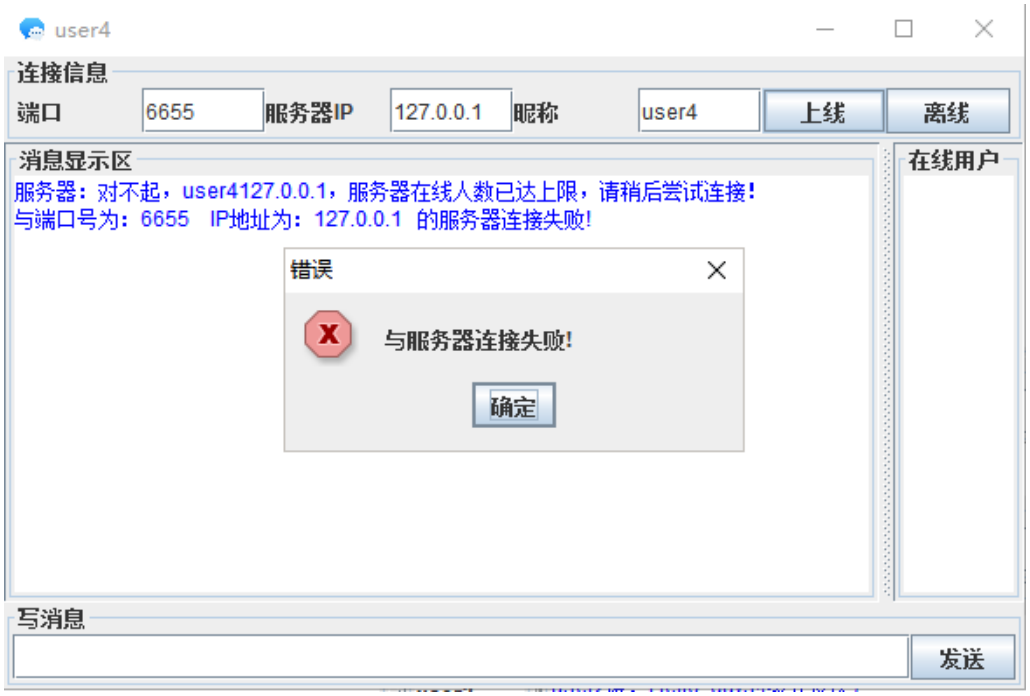




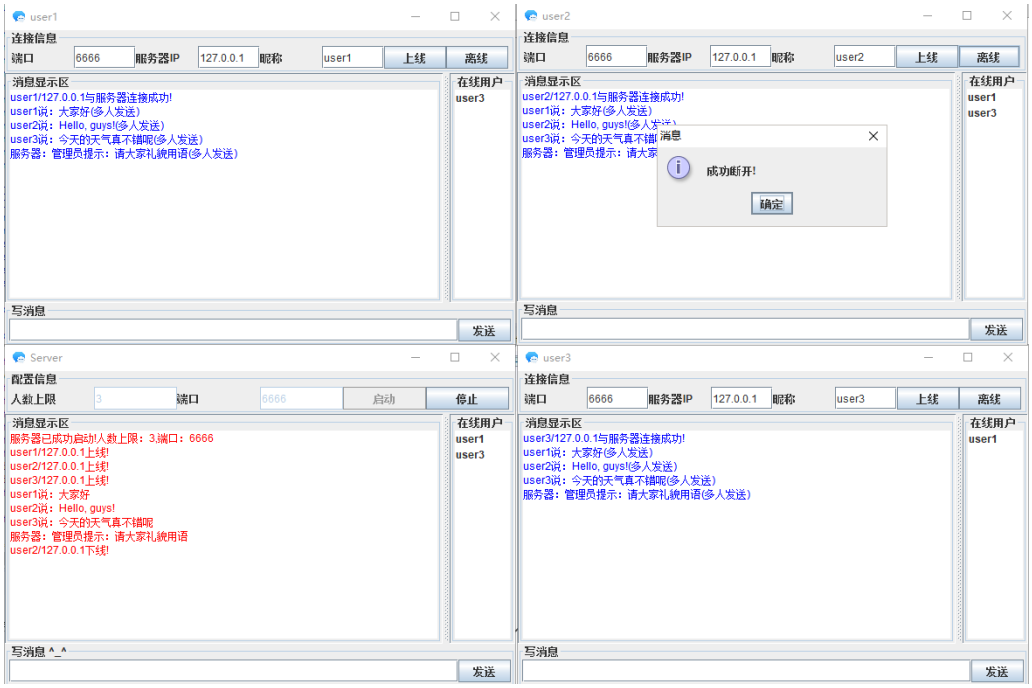
创建用户 user4 试图加入同一房间



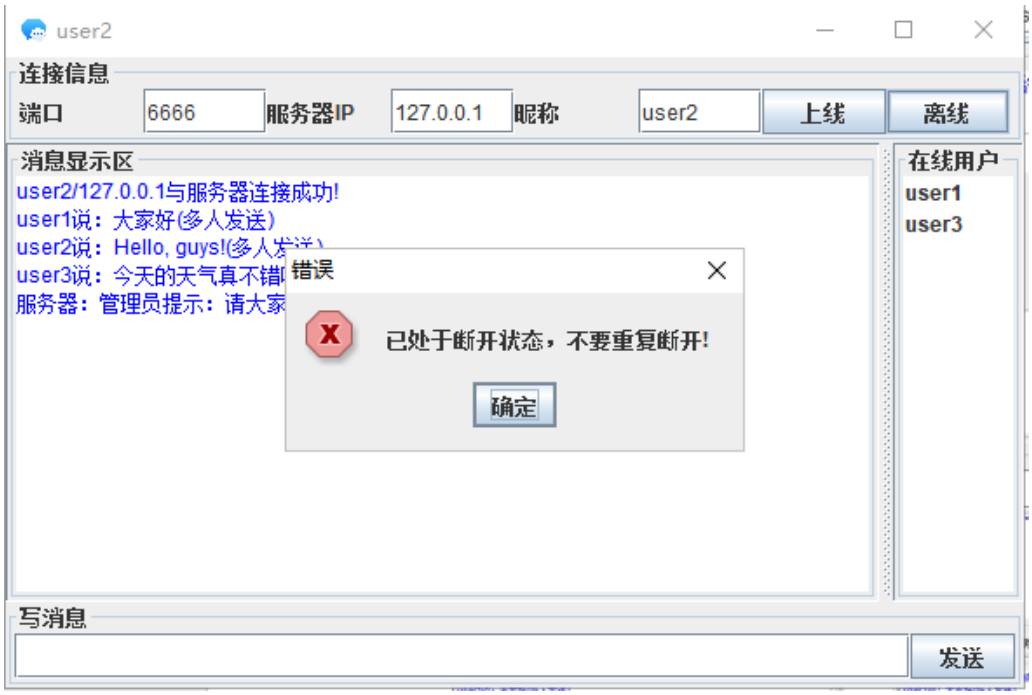
用户 user4 试图加入端口为 6655 的服务器（未创建）



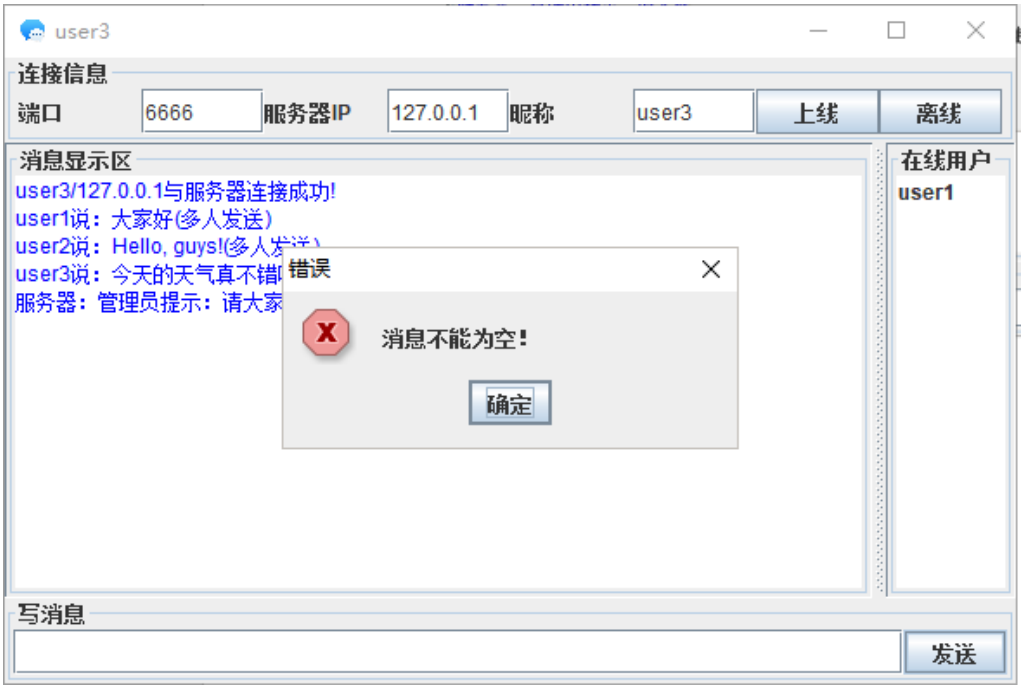
用户 user2 离开房间



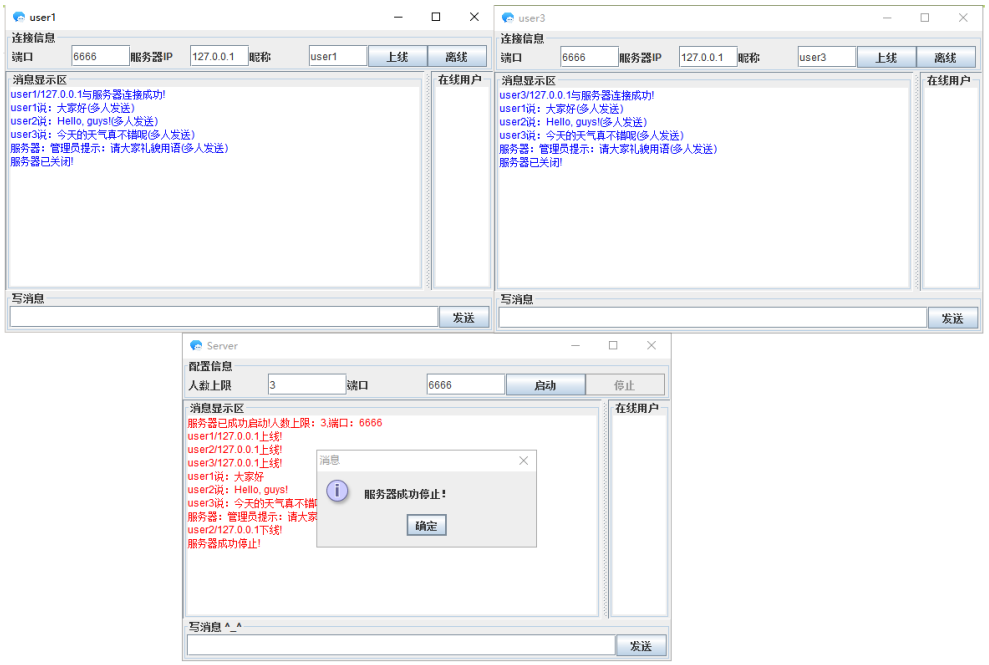
用户 user2 已离线，再次点击离线按钮



用户 user3 发送空消息



服务端终止



5．总结

通过简易多人聊天室的设计实现，我进一步了解了内部类的相关知识，了解了 Java 多线程原理，成功利用多线程实现了服务器与多组用户通信；进一步巩固 java 异常处理机制；了解了网络编程与 Socket 的基础应用；使用 Swing 组件创建了建议的图形界面，同时设计匿名内部类实现事件的监听，提高了编程技巧。通过该课程设计，全面系统地理解了程序构造的一般原理和基本实现方法。把死板的课本知识变得生动有趣，激发了学习的积极性。把学过的编程思想的知识强化，能够把课堂上学的知识通过自己设计的程序表示出来，加深了对理论知识的理解。