

java大型实验

**2018/2019(1)**



实验题目：Java实现的聊天系统

学生姓名：高源

学生学号：201706062013

学生班级：软件工程1704

任课教师：闵勇

完成日期：2019年1月20日星期日

**计算机科学与技术学院**

**Java实现的聊天系统报告**

# 实验题目和要求

利用基于TCP的Socket编程、多线程等技术实现一个聊天系统，包括服务器端程序和客户端程序。

(1)服务器端程序要求如下：

①能等待用户联机，并建立和客户端通信的I/O通道；

②当有用户要求加入聊天室，先验证其帐号和密码的合法性，验证成功后允许联机并将登陆信息写入日志文件或数据库中；

③能将用户发来的信息存入文件或数据库保存，以便查询统计，同时能将消息广播给所有聊天室成员；

④能清除不正常终端的联机；

⑤系统管理：能设置加入聊天系统的人数限制，能查看用户登陆信息和聊天信息。

(2)客户端程序要求如下：

①能注册和登陆帐号；

②能与服务器程序联机并且建立输入/输出通道，通道建立完成后，其工作便是接收由客户端传来的消息，然后根据消息来作出适当的处理；

③列出在线的所有用户登陆名或昵称。

# 设计思路

## 系统总体设计

服务器端新建服务器线程进行侦听，对连接上的客户端新建线程进行管理，获取其输入输出流，实现信息的接收和转发。

客户端打开先进入登陆界面，只有账号密码正确才能进入对话窗口，（另外，也可以通过注册界面进入系统），进入后，新建客户端线程，管理向服务器的输入输出

完成客户端和服务器的UI界面，为其添加监听器完成各种功能

## 系统功能设计

### 客户端

#### 登录窗口



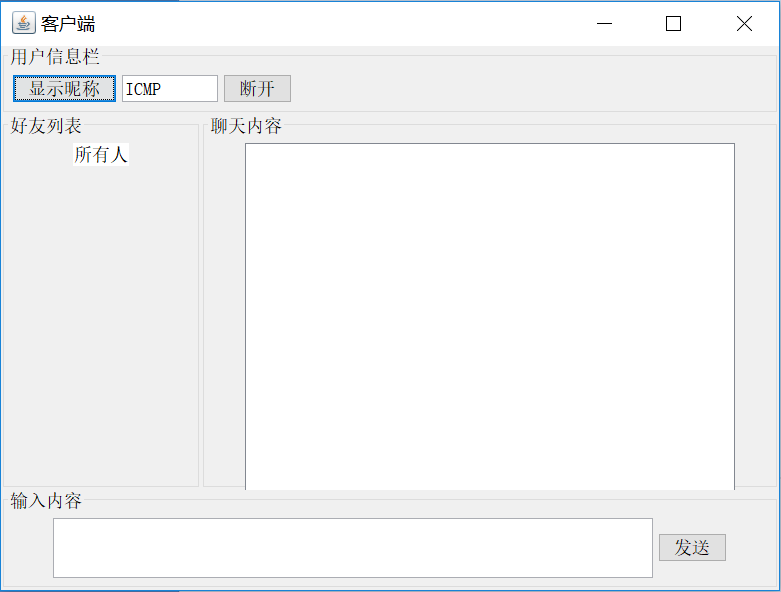
|  |  |
| --- | --- |
| 控件 | 功能 |
| 登录 | 读取用户名，密码，端口，ip向服务器发送登录信息，成功则进入对话窗口，否则报错 |
| 取消 | 将用户名，密码JtextFileld置空 |
| 退出 | 退出程序，关闭窗口 |
| 注册 | 关闭原窗口，跳转至注册窗口 |

#### 注册窗口



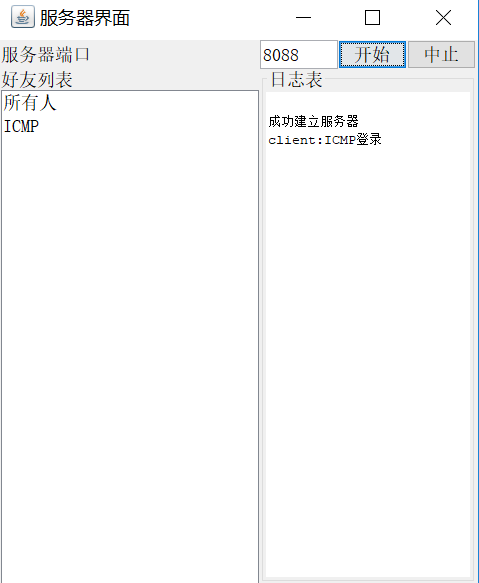
|  |  |
| --- | --- |
| 控件 | 功能 |
| 注册并登录 | 读取输入的ip,端口,用户名,昵称，密码，并向服务器发送注册消息，服务器端判断若没有重复用户名,则返回登录窗口，并将其信息写入文件 |
| 取消 | 将用户名，昵称，密码JtextFileld置空 |
| 退出 | 退出程序，关闭窗口 |

#### 对话窗口



|  |  |
| --- | --- |
| 控件 | 功能 |
| 显示昵称 | 显示自己的昵称 |
| 断开 | 向服务器发送下线信息，断开与服务器的连接 |
| 好友列表 | 1. 当登录成功后，服务器会向客户端发送在线人员列表，客户端将在线人员添加至好友列表， 2. 当之后有人上线下线时，服务器会将消息发至客户端，客户端依据添加删除在线人员。 3. 同时点击好友，会显示与目标的过往聊天记录。 |
| 聊天内容 | 显示聊天记录和用户上线下线消息 |
| 输入内容 | 输入想要发送的内容 |
| 发送 | 将输入内容文本框中的内容发送至目标 |

### 服务器



|  |  |
| --- | --- |
| 控件 | 功能 |
| 开始 | 获取端口，并在服务器上侦听这个端口 |
| 中止 | 关闭服务器 |
| 好友列表 | 显示在线人员，有人上线会向其中添加人员，下线则从好友列表中移除他 |
| 日志表 | 显示服务器操作记录 |

### 服务器和客户端之间的通信协议

|  |  |  |
| --- | --- | --- |
|  | 消息格式 | 表示内容 |
| 客户端发至服务器 | Login|UserAccount|Password | 登录，用账户登录 |
| Register|Useraccount|UserNickName|Userpassword | 注册 |
| 服务器发至客户端 | RegisterAnswer|true或者  RegisterAnswer|false | 注册成功或注册失败 |
| LoginAnswer|true或者  LoginAnswer|false | 登录成功或登录失败 |
| Login|UserNickName | 用户UserNickName上线 |
| 客户端服务器皆可收到和发送 | Logout|UserNickName | 用户UserNickName下线 |
| Message| ToTargetName|MyNickName|message | 发送消息，发送者ToTargetName（昵称）,接受者是MyNickName（昵称）， |

## 类的设计

### 客户端类的主要函数及其子类

|  |  |  |  |
| --- | --- | --- | --- |
| Client(主类) | | | |
| 权限 | 分类 | 名称 | 说明 |
| public | 成员函数 | Client() | 调用CreatFramLogin() |
| ConnectServer() | 尝试用登录窗的信息进行登录 |
| ConnectServerRegister() | 尝试用注册窗的信息进行注册和登录 |
| CreateFormLogin() | 创建登录界面 |
| CreateFormRegister() | 创建注册界面 |
| CreateForm() | 创建对话窗口 |
| ClientThread | Client的线程类，详见下方 |
| private | AddActionListenerLogin() | 登录界面添加事件监听 |
| AddActionListenerRegister() | 注册界面的监听器 |
| AddActionListenerDialog() | 对话窗口监听器 |
| saveword() | 将聊天记录保存至本地 |
| readwords(String target) | 从文件里读取聊天记录到对话框 |
| RemoveElement(String target) | 移除已经下线的用户 |
| Log(String message) | 将消息显示到消息框 |

|  |  |  |
| --- | --- | --- |
|  | ClientTherard（内部类）implements Runnable | |
| 分类 | 名称 | 说明 |
| 成员函数 | public ClientThread() | 线程start |
| public void run() | 建立客户端线程实现对服务器接受过来的消息 |
| 成员变量 | boolean isRunning | 判断线程是否在运行 |

|  |  |  |  |
| --- | --- | --- | --- |
| PartitionString（用来将从服务器或客户端发出的消息中提取出有用的信息） | | | |
| 分类 | 名称 | 说明 | |
| 成员变量 | String sum[] | | 存储将消息分割后的数据 |
| int index | | 已经读取到sum的哪一部分 |
| 成员函数 | public PartitionString(String target,String splittarget) | | 构造函数，令sum=target.split(splittarget); |
| public String nextpart() | | index++;  return sum[index-1]; |

### 服务器类中主要函数及其子类

|  |  |  |
| --- | --- | --- |
|  | ServerProgram（主类） | |
| 分类 | 名称 | 说明 |
| 主要函数或其内部类 | public ServerProgram() | 1. 调用CreatForm() 2. 从文件中读取全部用户信息，建立TreeMap<String,UserPass>类用户信息树 |
| CreateForm() | 建立窗口 |
| UserPass | 内部类，数据成员为昵称和密码，详见下 |
| CreateServer() | 建立服务器线程类，监听目标端口，等待客户端链接 |
| ServerTherad | 内部类，客户端线程类详见下 |
| ClientTherad | 客户端线程类，详见下 |
| public void writeLog(String infomation) | 在日志表上输出日志 |
| private void addactionlistenr() | 为各个按钮添加监听器 |
| public void RemoveElement(String target) | 将退出用户从好友列表里清除 |
| 主要成员变量 | TreeMap<String, UserPass> users; | 运用文件里的账户+昵称+密码建立TreeMap |
| TreeMap<String, ClientTherad> ClientTherads | 管理已经连接上服务器的客户端线程 |

|  |  |  |
| --- | --- | --- |
| ServerTherad implements Runnable(内部类) | | |
| 分类 | 名称 | 说明 |
| 成员函数 | public ServerTherad() | isRunning=true;  new Thread(this).start(); |
| public void run() | 1. 启用线程监听端口 2. 若是有客户端连接，新建客户端线程，如果新建成功，将其推入ClientTherads |
| public void Closeserver() | 关闭服务器 |
| 成员变量 | boolean isRunning | 判断服务器线程是否在运行 |

|  |  |  |
| --- | --- | --- |
| ClientTherad implements Runnable（内部类） | | |
| 分类 | 名称 | 说明 |
| 成员函数 | public ClientTherad(Socket target) | 1. isrunning=false 2. 调用initial() 3. 若isrunning==true,调用start（） |
| public synchronized void initial() | 处理客户端发送多来的登录或注册信息，如果正确，则  1.发送正确消息回复至客户端  2.isrunning=true  3.将用户上线消息发送给全体人员 |
| public synchronized void run() | 接受客户端发送的各种消息并进行处理 |
| public void sendMessage(String target) | 给客户端发送信息 |
| public void Broadcast(String Message) | 向全体成员广播消息 |
| public String JudgeNick(String Nickname3) | 查找是否有NickName对应的账号，有则返回账号，无则返回NULL |
| public String getUseraccount() | 返回当前客户端线程的账号 |
| public void AddUserToFile(String account4,String NickName4,String Password4) | 将注册用户添加入文件 |
| 主要成员变量 | boolean isRunning | 判断客户端线程是否在运行 |

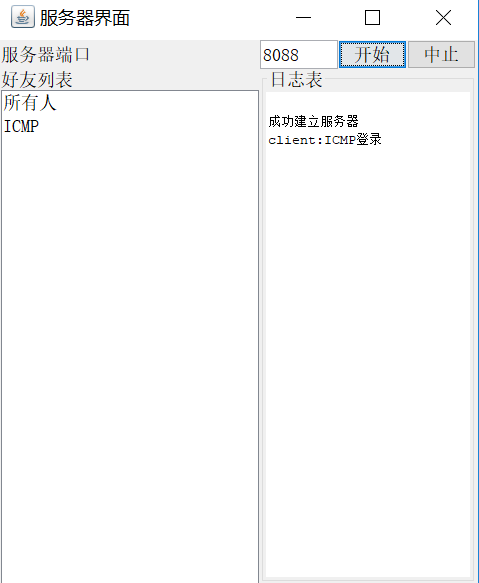
|  |  |  |
| --- | --- | --- |
| UserPass | | |
| 成员变量 | String NickName; | 昵称 |
| String Password; | 密码 |
| 成员函数 | public UserPass(String NickName,String Password) | this.NickName=NickName; this.Password=Password; |
| public String getNickName() | return NickName; |
| public String getPassword() | return Password; |

|  |
| --- |
| PartitionString类同客户端一样，不再赘述 |

# 测试结果分析

## 服务器

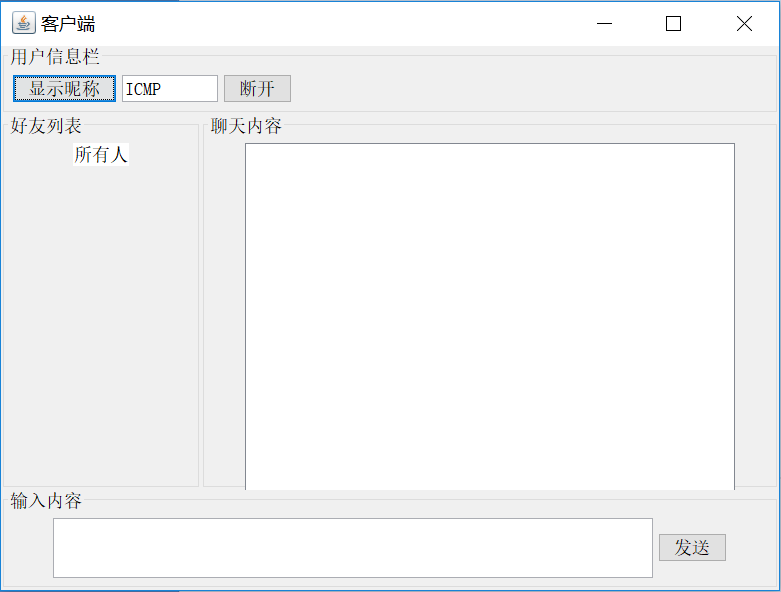
### 打开服务器



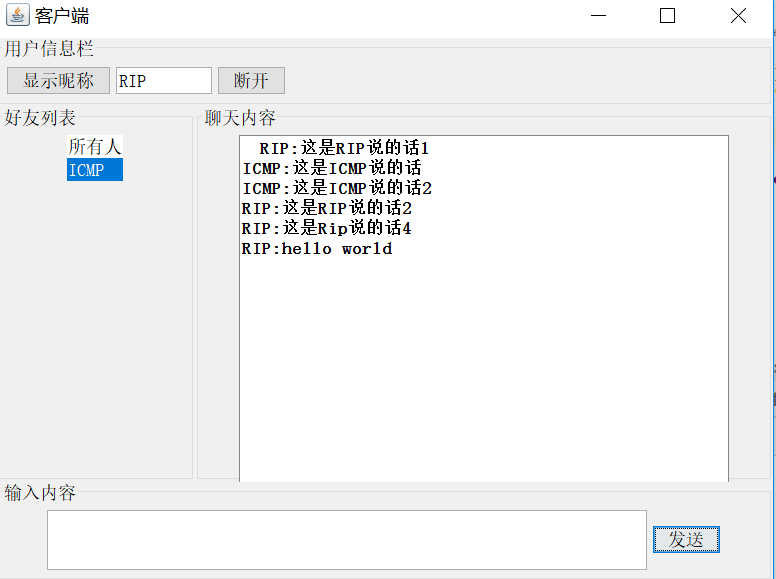
## 登录



### 登录成功后跳转至对话窗口



### 发送消息，群发和私聊



## 注册



# 实验体会

通过这次聊天系统的制作我对java图形化界面和、，网络编程，线程有了更深的了解，并且在制作的过程中知道很多书上所没有的知识，比如调用JFram.dispose()方法关闭一个旧窗口后，创建一个新窗口，实现界面的跳转。比如设置JtextArea中的字体使字体更清晰等等。又或者说是Jframe背景图片的设置。

同时在程序运行中出现的错误，也使我对Java的理解更上一层楼比如有一次我将JtextArea添加至JscrollPane后进度条却不能正常显示，经过学习后才发现，我预先设置了文本框的大小导致了这个问题。

# 源代码附录

见Client和server文件夹中的源文件。同时若是想要运行，Client中的javaLogin.jpg和javaloginScheme.jpg必须与字节码放在一起 。同时server中userdata.txt必须放在out\production\Java课程设计聊天系统\server\userdata.txt的相对路径中或直接修改ServerProgram.java中第39行的路径