



**Java课程设计报告**

**题目： 简易聊天室程序的实现**

**学院：计算机学院（国家示范性软件学院）**

**专业： 计算机科学与技术**

**班级： 2023211301**

**学号： 2023210896**

**姓名： 王书翰**

**2025年6月13号**

**目录**

[1. 程序功能要求 1](#_Toc22292)

[1.1 总要求 1](#_Toc21683)

[1.2 服务器端功能要求 1](#_Toc12492)

[1.3 客户端功能要求 1](#_Toc20752)

[2. 设计思路 3](#_Toc30795)

[2.1 总体设计思路 3](#_Toc14709)

[2.2 服务器类设计思路 3](#_Toc9366)

[2.3 客户端类设计思路 3](#_Toc7623)

[3. 类的功能 5](#_Toc28458)

[3.1 服务器类说明 5](#_Toc15888)

[3.1.1 服务器类的作用 5](#_Toc23138)

[3.1.2 服务器类成员变量 5](#_Toc13265)

[3.1.3 服务器类构造方法 5](#_Toc5)

[3.1.4 服务器类方法 6](#_Toc9943)

[3.1.5 内部类HandleClient说明 6](#_Toc4420)

[3.2 客户端类说明 8](#_Toc29119)

[3.2.1 客户端类的作用 8](#_Toc31670)

[3.2.2 客户端类成员变量 8](#_Toc31628)

[3.2.3 客户端类构造方法 8](#_Toc779)

[3.2.4 客户端类方法 8](#_Toc30460)

[3.2.5 内部类HandleServer说明 9](#_Toc23954)

[4. 类的关系 11](#_Toc32129)

[4.1 服务器类内部的关系 11](#_Toc4558)

[4.2 客户端类内部的关系 11](#_Toc27403)

[4.3 客户端类与服务器端之间的关系 11](#_Toc1436)

[5. 使用说明 13](#_Toc18686)

[5.1 初始化文件 13](#_Toc29639)

[5.2 启动服务器 13](#_Toc18960)

[5.3 启动客户端 15](#_Toc25337)

[5.3.1 发送信息 17](#_Toc9699)

[5.3.2 发送图片 18](#_Toc20836)

[5.3.3 系统命令 19](#_Toc23186)

[6. 课程设计总结 20](#_Toc8868)

# 程序功能要求

## 总要求

要求设计并实现一个基于图形界面的C/S结构的简易聊天室程序，程序分为服务器端和客户端两部分。

## 服务器端功能要求

1. 启动时首先读入一个用户文件，用户文件中保存的是全部用户（至少10个）的用户名及密码（明文保存，真实系统不会明文保存的）。然后显示一个提示符，等待输入命令。可接受的命令包括list：列出全部在线用户；listall：列出全部用户；quit：退出系统；
2. 在固定端口（8000以上）侦听，等待客户端连接；
3. 客户端连接成功后有一个验证用户名和密码的过程。如果用户名密码正确，要给该客户端发成功验证的提示信息，否则给客户端发送用户名或密码错误信息，并等待用户再次验证，直到验证成功或客户端断开；
4. 至少支持5个客户端同时聊天；
5. 转发客户端发送的聊天信息且支持匿名聊天；
6. 日志功能。用文本文件记录每个用户的行为。这些行为包括每次登录成功信息（要包含IP地址，时间等）、每次登录失败信息（要包含IP地址、时间等）、每次退出聊天室时间等。

## 客户端功能要求

1. 启动后首先要连接到服务器端；
2. 连接成功后提示用户输入用户名及密码，并将用户输入的用户名及密码发给服务器端验证。验证成功后则成功进入聊天室，否则提示用户重新输入用户名及密码，直到验证成功或用户结束程序；
3. 验证成功后进入聊天室，聊天室要有提示符；
4. 聊天室内要能显示所有用户的非私聊信息及给自己的私聊信息；
5. 在聊天室内用户可以输入三类字符串：①普通字符串，为广播类聊天信息，所有人都可以看到；②以@+用户名开头的字符串，为私聊信息，只有@到的那个用户可以看到；③@@+命令，为系统命令；
6. 系统命令有：list：列出当前在线用户；quit：退出系统；showanonymous：显示当前聊天方式是否为匿名；anonymous：切换聊天方式，即使用匿名聊天，还是实名聊天。

# 设计思路

## 总体设计思路

为实现一个服务器端多个客户端同时通信的功能，设计两个java类，分别为Server类和Client类，在类中封装各自的功能，如ui界面功能，显示聊天功能，建立通信功能，接收和发送信息功能等。

## 服务器类设计思路

1. 使用java系统库中的awt库和swing库实现ui界面的设计
2. 使用java系统库中的net.ServerSocket库中的ServerSocket类设计服务器，实现和客户端建立通信，接收和发送信息的操作
3. 使用java系统库中的net.Socket库中的Socket类设计客户端，实现和服务器建立通信，接收和发送信息的操作
4. 使用java系统库中的io库实现从文件中读取用户信息和向文件中存储信息
5. 使用java系统库中的util.Date库实现获取当前系统时间的功能
6. 使用java系统库中的util.HashMap库实现用哈希表对实现存储用户账号密码信息的功能
7. 增加一个HandleClient内部类，实现时刻接收和管理连接服务器的客户端的信息

## 客户端类设计思路

1. 使用java系统库中的awt库和swing库实现ui界面的设计
2. 使用java系统库中的net.Socket库中的Socket类设计客户端，实现和服务器建立通信，接收和发送信息的操作
3. 使用java系统库中的imageio.ImageIO库实现对图片进行读取和显示操作
4. 使用java系统库中的util.Date库实现获取当前系统时间的功能
5. 使用java系统库中的util.Base64库实现将图片信息进行编码和解码
6. 增加一个HandleServer内部类，实现时刻接收来自服务器端的消息

# 类的功能

## 服务器类说明

### 服务器类的作用

负责处理一切客户端的事务

### 服务器类成员变量

private static final int PORT = 8080; // 定义服务器监听端口号为 8080

private int peopleOnline = 0; // 在线人数计数器，记录当前在线的人数

private boolean isStart = false; // 标识服务器是否启动

private ArrayList<HandleClient> client = new ArrayList<>(); // 用于存储客户端连接的列表

private Map<String, String> users = new HashMap<>(); // 存储用户名和密码的哈希表

JTextArea jta; // 定义用于显示服务器日志的文本区域组件

private JPanel south; // 定义在窗口中放在最下面的面板

private JLabel jb; // 定义标签用于显示在线人数

private JButton stopBtn; // 定义停止服务器的按钮

private JButton startBtn; // 定义启动服务器的按钮

private ServerSocket serverSocket = null; // 定义服务器套接字

### 服务器类构造方法

public Server()

构造图形化界面布局，显示图形化界面

从user.txt中读取用户信息

将图形化界面中的三个按钮（发送控制台命令按钮，启动服务器按钮和关闭服务器按钮）添加监听器

### 服务器类方法

private void loadUsers()//从文件加载用户数据

private synchronized void saveUser()//将新注册的用户信息存入文件

private void listOnlineUsers()//列出在线用户

private void listAllUsers()//列出所有用户

private void appendToLog(String message)//将消息在服务器窗口中显示

public void startServer()//启动服务器，等待客户端连接

public static void main()//主函数

### 内部类HandleClient说明

#### 内部类HandleClient的作用

处理每个客户端连接

验证用户登录

处理客户端通信

切换匿名和实名聊天模式

处理私聊消息

发送和接收消息

处理分块图片信息

发送图片信息

#### 内部类HandleClient成员变量

Socket socket = null; // 客户端套接字

private String name; // 用户名

private String anonymousName; // 匿名用户名

private boolean isAnonymous = false; // 是否匿名标识

DataInputStream dis; // 数据输入流

DataOutputStream dos; // 数据输出流

private static final int MAX\_MESSAGE\_SIZE = 1024 \* 1024 \* 1024; // 10MB最大消息限制

#### 内部类HandleClient构造方法

public HandleClient()

初始化客户端套接字，初始化数据输入流和输出流

#### 内部类HandleClient方法

private String handleImageChunks()//处理分块图片信息

public void run()//重写run()函数，实现监听功能

private void handleClientCommunication()//处理客户端通信

private void listOnlineUsersForClient()// 列出在线用户给客户端

public void send()//给客户端发送信息

public void broadcast()//给所有客户端发送广播

public void sendImageData()//发送图片数据

private void handlePrivateMessage()//处理私聊信息

private void logToFile()//将日志写入文件

private String getCurrentTime()//获取当前时间的方法

## 客户端类说明

### 客户端类的作用

负责处理一切客户端的事务

### 客户端类成员变量

private boolean isConn = false; // 标记是否连接到服务器

private String name; // 客户端用户名

private boolean isAnonymous = false; // 是否选择匿名聊天模式

private JTextPane jta; // 用于显示聊天信息的文本区域

private JScrollPane jsp; // 滚动面板，用于显示文本区域

private JPanel jp; // 面板，用于放置输入文本框和发送按钮

private JTextField jtf; // 输入消息的文本框

private JTextField modeSelector;//选择发送信息的类型

private JButton jb1; // 发送消息的按钮

private JButton jbImage; // 发送图片按钮

private Socket socket = null; // 套接字对象

private static final String CONNSTR = "127.0.0.1"; // 连接服务器的IP地址

private static final int CONNPORT = 8080; // 连接服务器的端口号

private DataOutputStream dos = null; // 数据输出流

### 客户端类构造方法

public Client()

构造图形化界面布局，显示图形化界面

### 客户端类方法

private void init()//初始化图形界面，与服务器建立连接

private void appendMessage() //在聊天记录框中显示信息

private void appendImage() //在聊天记录框中显示图片

private void sendMessage()//发送消息到服务器

private void handleSystemCommand()//处理系统命令

public void keyPressed()//重写按下键盘按键的处理

private void handleReceivedMessage()//处理从服务器端接收到的信息

private void sendImage()//将图片信息发送至服务器端

private BufferedImage compressImage()//压缩从本地上传的图片

### 内部类HandleServer说明

#### **内部类HandleServer的作用**

与服务器端通信

接收服务器发来的信息

#### **内部类HandleServer成员变量**

private StringBuilder imageBuffer;//文件编码的缓冲区

private boolean isReceivingImage;//接收图片信息的标识

private int expectedChunks;//期望收到的块数

private int receivedChunks;//实际收到的块数

#### **内部类HandleServer构造方法**

public HandleClient()

初始化成员变量参数

#### **内部类HandleServer方法**

public void run()//重写run()函数，实现监听功能

# 类的关系

## 服务器类内部的关系

Server

性质: 服务器主类，继承 JFrame，包含 main 方法。

作用: 负责服务端的图形用户界面，负责启动服务器，接收客户端连接，并为每个客户端启动一个新的处理线程。

包含的类:

HandleClient: 内部类，用于处理客户端通信。

HandleClient

关系: 服务器主类的静态内部类，实现 Runnable 接口。

作用: 处理与单个客户端的通信，包括接收消息、发送消息和用户管理。

关键方法: run()，负责处理客户端的消息和连接。

## 客户端类内部的关系

Client

关系: 主类，继承 JFrame 并实现 KeyListener 接口，包含main方法

作用: 负责客户端的图形用户界面、与服务器的连接及消息发送和接收。

包含的类:

HandleServer: 内部类，用于接收服务器发送的消息。

HandleServer

关系: Client类的非静态内部类，实现 Runnable 接口。

作用: 负责从服务器接收消息并显示在客户端的文本区域中。

关键方法: run()，循环接收并显示服务器发送的消息。

## 客户端类与服务器端之间的关系

Socket 连接:

Client类在 init 方法中通过 Socket 连接到服务器 ServerChar。

Server类接受客户端连接，并为每个连接创建一个新的HandleClient类线程处理。

数据流:

Client和HandleClient都使用 DataInputStream 和 DataOutputStream 来进行消息的读写。

客户端通过 DataOutputStream 发送消息到服务器，服务器通过 DataInputStream 接收消息，并根据逻辑处理后再通过 DataOutputStream 发送消息给客户端，客户端通过 DataInputStream 接收。

消息处理:

Client负责发送消息和处理系统命令，并通过HandleServer类接收服务器的消息。

Server的HandleClient负责处理客户端发送的消息，并转发给其他连接的客户端。

# 使用说明

## 初始化文件

在运行目录下创建存放用户信息的user.txt文件，存放格式为：用户名,密码，如admin,admin。同时创建journal.txt文件用于储存系统日志

## 启动服务器

运行Server.java文件，弹出可视化窗口



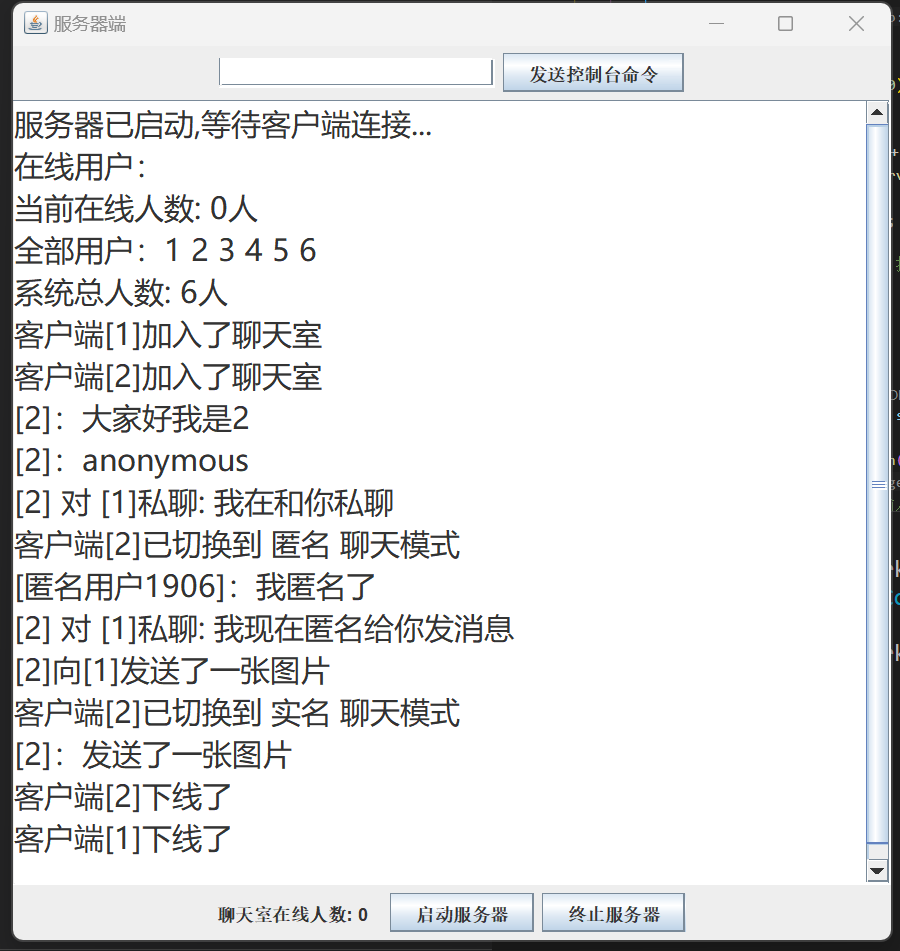
点击“启动服务器”按钮启动服务器，点击“终止服务器”按钮终止服务器



在文字输入框中输入“list”指令列出全部在线用户

在文字输入框中输入“listall”指令列出全部在线用户

在文字输入框中输入“quit”指令退出系统

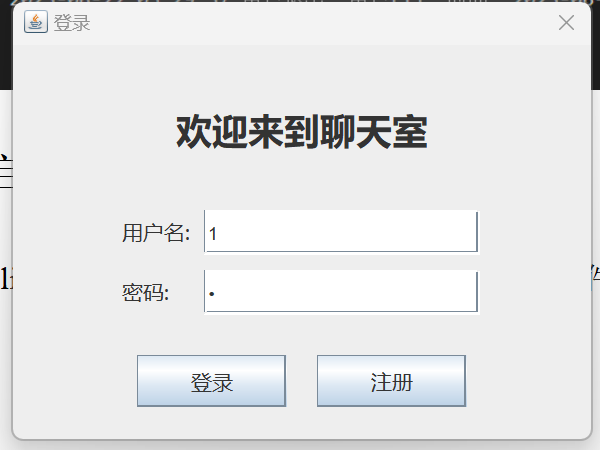
服务器窗口显示用户的行为

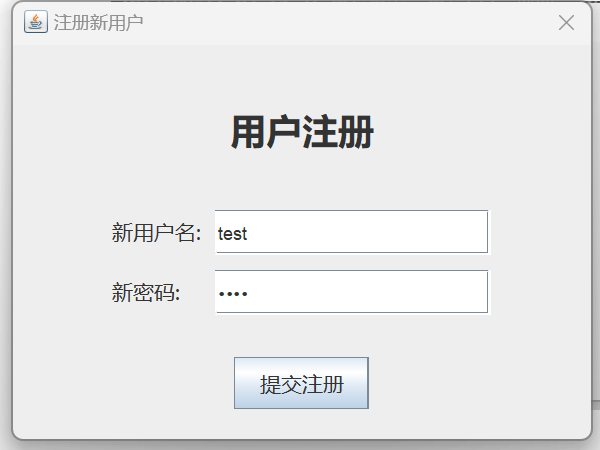
journal.txt中记录用户的行为



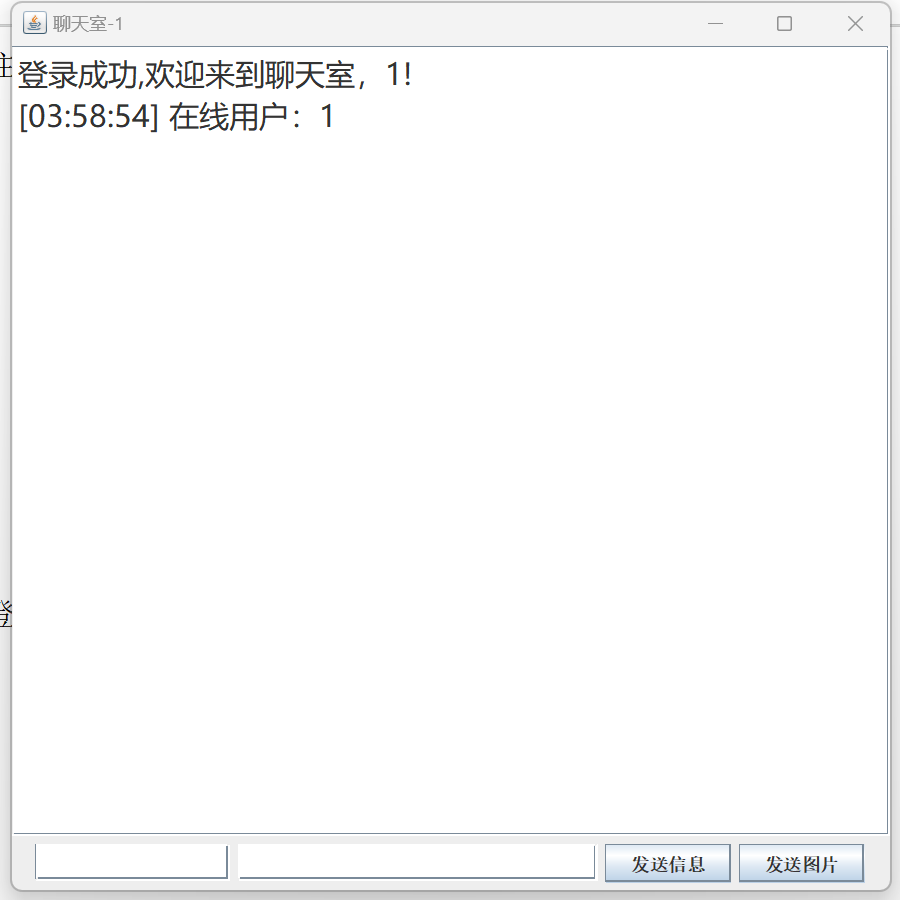
## 启动客户端

启动Client.java文件，在登录界面输入users.txt文件中存在的账号密码登录账号



若选择注册新账号，则点击注册按钮，输入注册的账号和密码

在成功登录后，客户端弹出聊天界面



输入信息类型

不输入：面向所有用发送信息

输入@+其他用户名：向该用户发私信

输入@@：输入系统信息

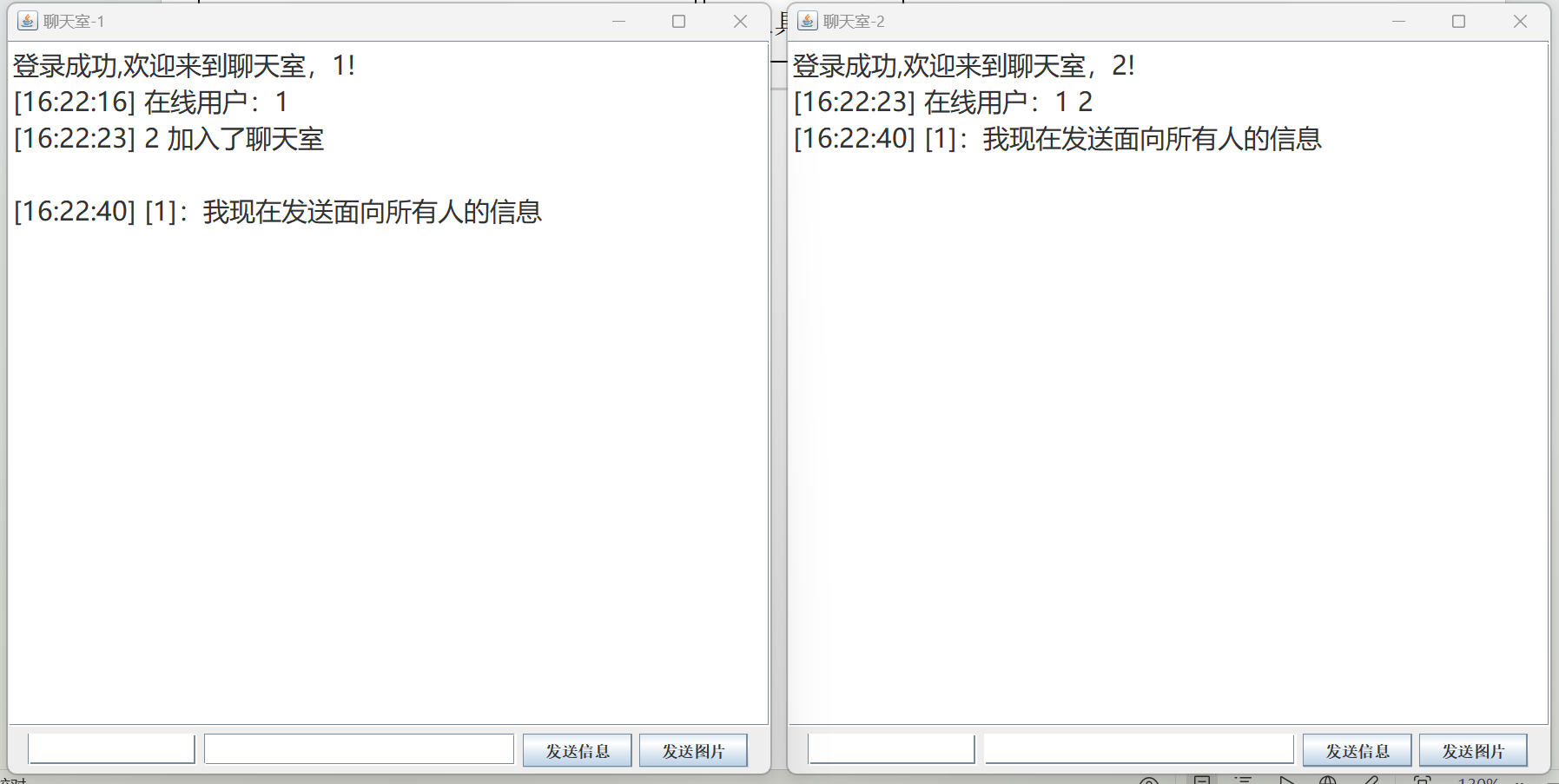
输入具体信息

点击发送文字信息或系统信息

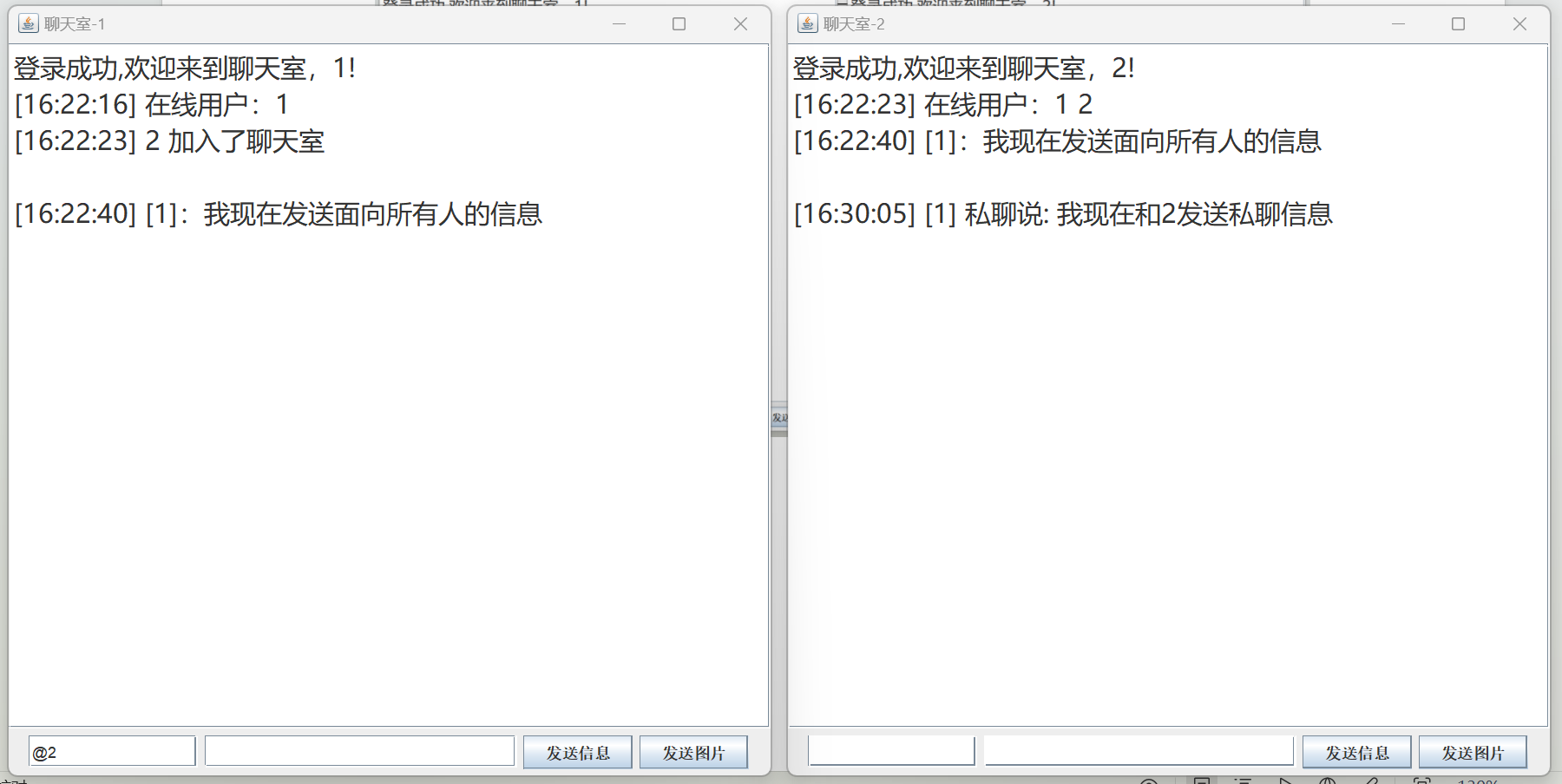
点击发送图片

### 发送信息

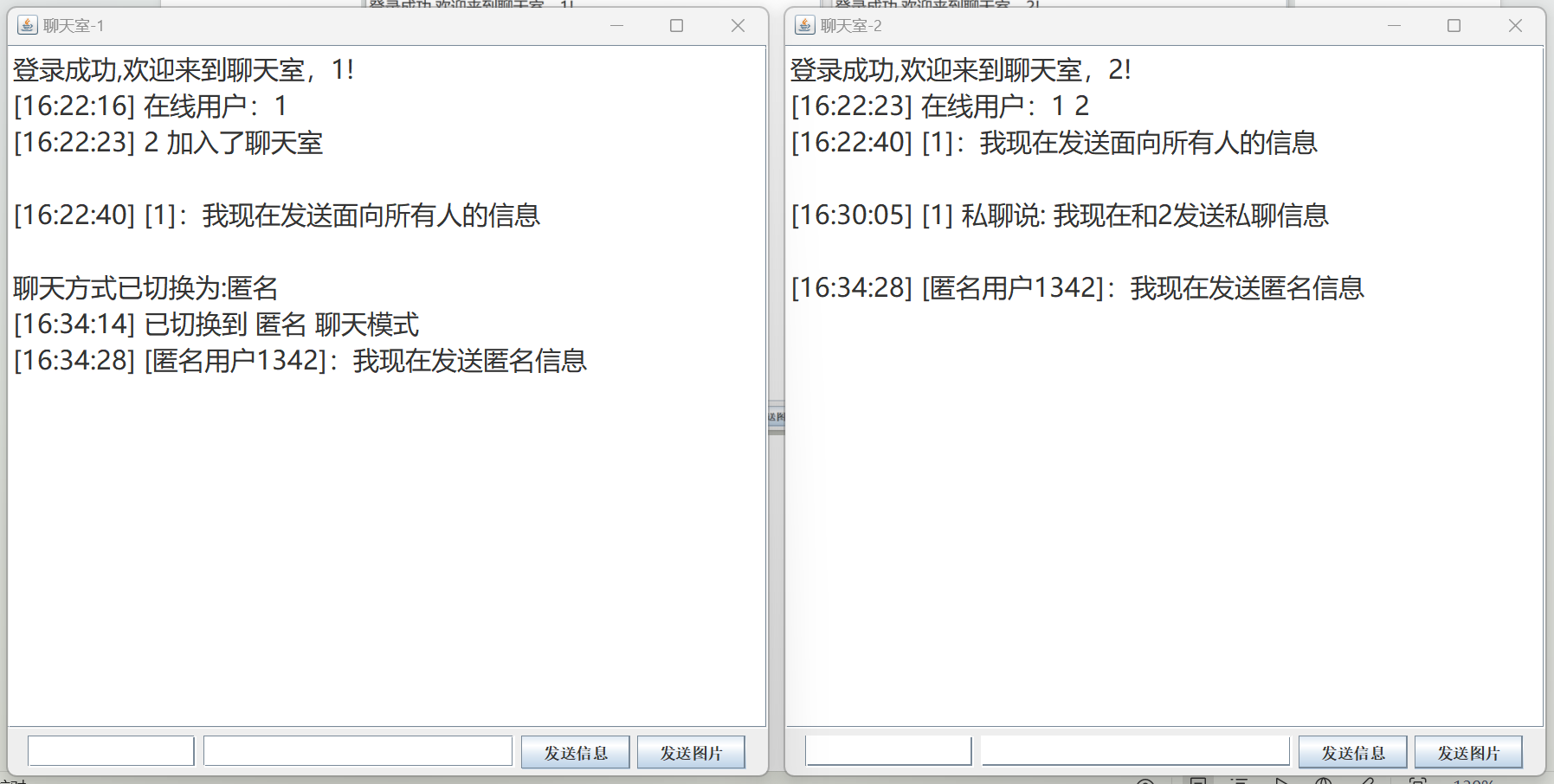
发送面向所有人的信息



发送私聊信息

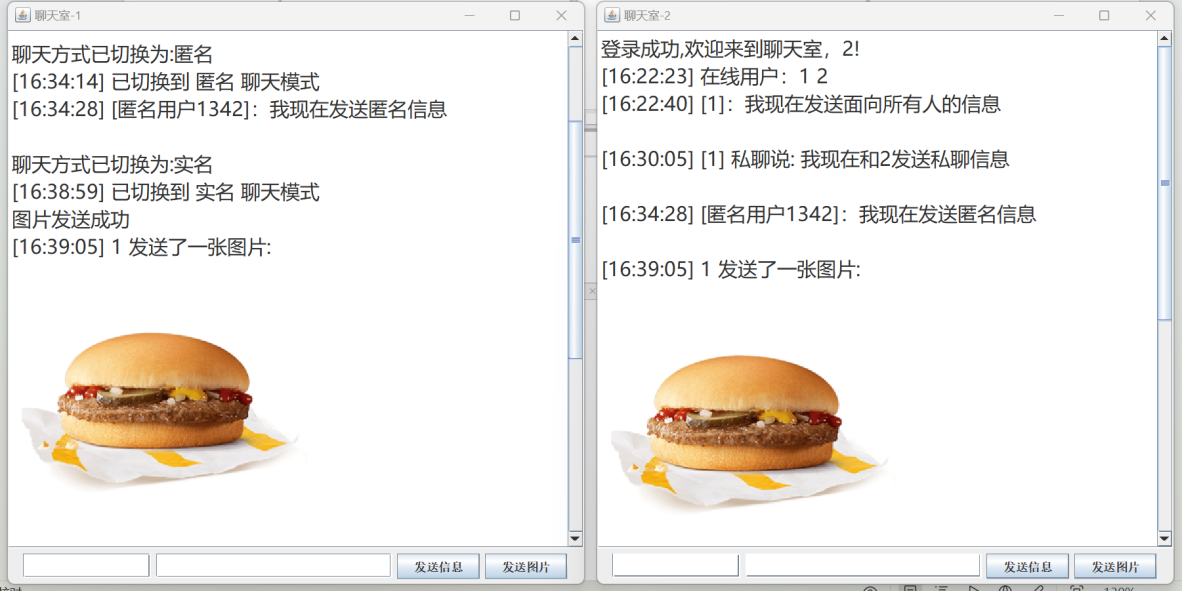


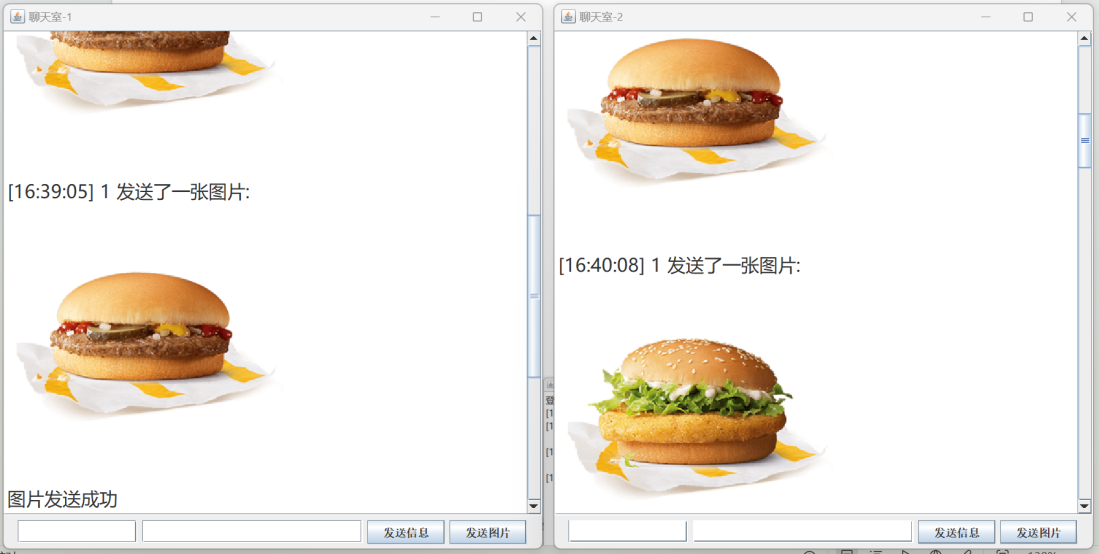
匿名发送信息



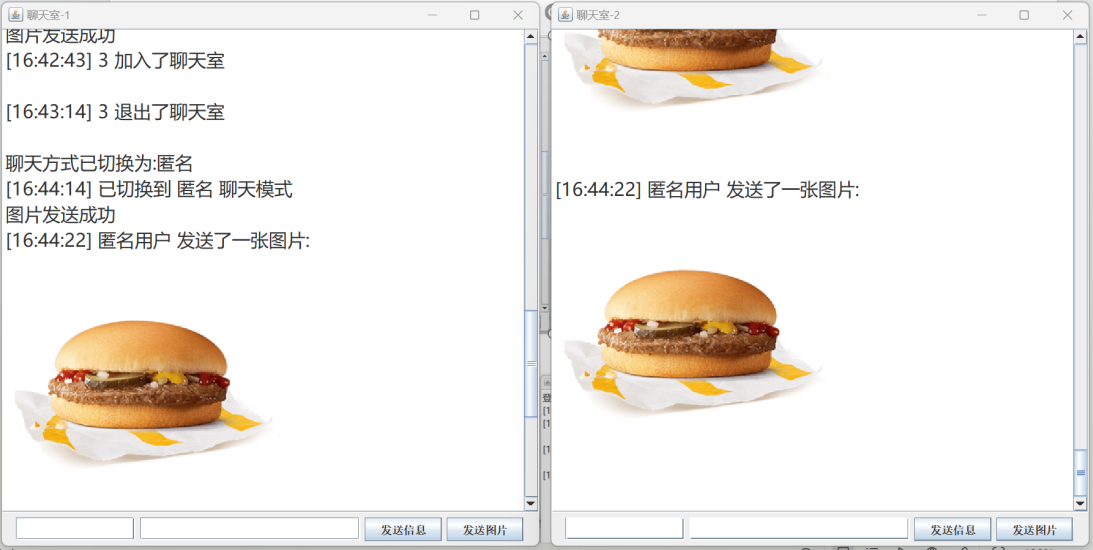
### 发送图片

发送面向所有人的图片



发送私聊图片

匿名发送信息



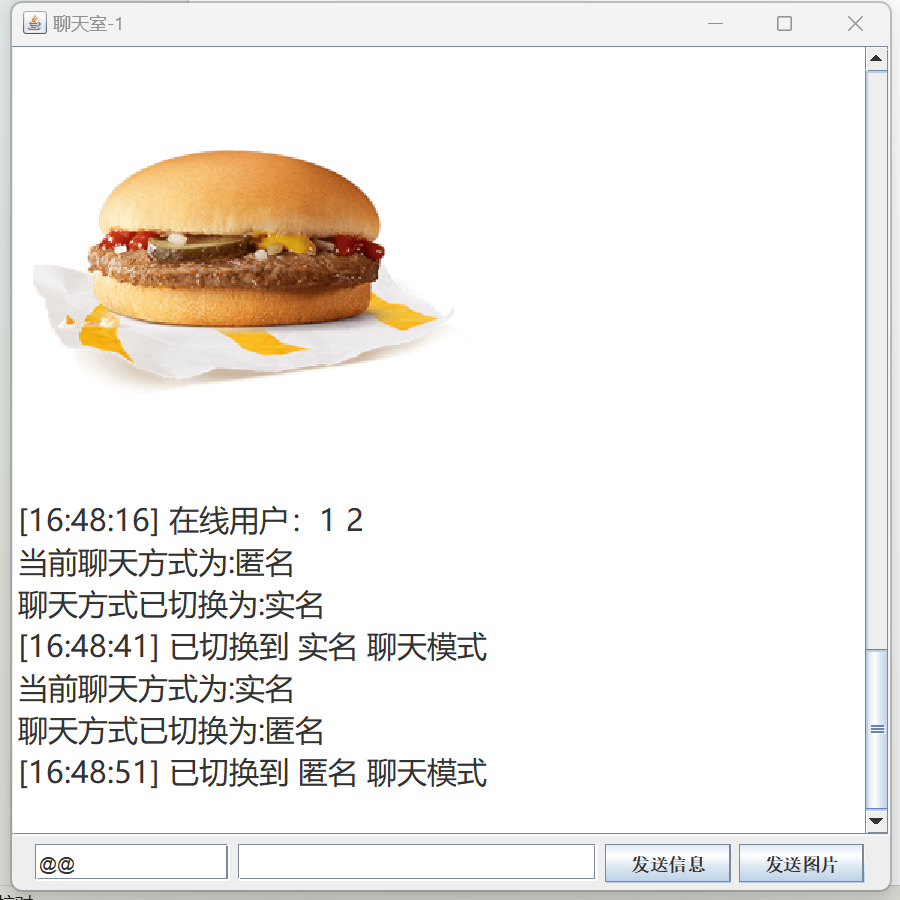
### 系统命令

在消息类型框输入“@@”，在消息输入框中输入“list”指令列出全部在线用户

在消息类型框输入“@@”，在消息输入框中输入“quit”指令退出

在消息类型框输入“@@”，在消息输入框中输入“showanonymous”指令显示当前聊天方式是否为匿名

在消息类型框输入“@@”，在消息输入框中输入“anonymous”指令切换聊天方式，即使用匿名聊天，还是实名聊天



# 课程设计总结

这次课程设计为使用java语言设计，我之前有用C语言，C++，python等语言设计项目的经验，但使用java进行设计还是第一次。但java项目的设计在C++和python中都能找到类似的设计思路，如和C++一样面向对象编程，用不同的类而不是不同的函数实现功能，和python一样引入其他项目文件和系统库。但由于时间紧迫，导致在设计过程中用户图形化界面不太美观，仅实现了功能，而且只用了两个文件，使用两个类完成功能，导致略显臃肿，这都是我需要再优化更新的点。

总而言之，通过学习这门课程，让我对java这门编程语言有了更加深入地了解，对面向对象编程的思想有了更加清楚地认识。感谢李劼老师的谆谆教诲，以及助教老师们的悉心指导。