### 设计思路

本聊天系统采用了基于客户端-服务器（C/S）架构的设计，服务器端负责管理客户端的连接、处理用户登录、消息转发和维护在线用户列表。客户端负责与服务器通信，发送和接收消息。整个系统的实现使用了Java的网络编程和图形用户界面（GUI）组件

**服务端设计思路:**

1. **接受客户端连接**：服务器通过一个端口监听客户端的连接请求。
2. **处理客户端请求**：服务器接收到客户端的请求后，进行相应的处理，如消息转发、命令处理等。
3. **维护客户端列表**：服务器需要维护一个客户端列表，以便在需要时向所有或特定客户端发送消息。

**客户端设计思路:**

1. **连接服务器**：客户端通过指定的服务器IP和端口与服务器建立连接。
2. **发送消息**：客户端将用户输入的消息发送到服务器。
3. **接收消息**：客户端接收从服务器转发过来的消息，并在GUI中显示。

**类的功能**

**服务器端 ServerChar 类**

1. **属性**
   * PORT: 服务器监听的端口号
   * JTextArea jta: 用于显示服务器日志的文本区域
   * JPanel south: 位于窗口南部的面板，包含显示在线人数的标签和停止服务器的按钮
   * JLabel jb: 显示在线人数的标签
   * JButton stopBtn: 停止服务器的按钮
   * boolean isStart: 服务器运行状态的标识
   * ServerSocket serverSocket: 服务器套接字
   * ArrayList<ClientConn> clientConn: 存储客户端连接的列表
   * int peopleOnline: 在线人数计数器
   * Map<String, String> users: 存储用户名和密码的映射
2. **构造方法**
   * 初始化窗口组件，设置界面布局
   * 加载用户数据
   * 启动服务器
3. **方法**
   * listOnlineUsers(): 列出在线用户
   * listAllUsers(): 列出所有用户
   * appendToLog(String message): 将消息追加到日志
   * loadUsers(): 从文件加载用户数据
   * startServer(): 启动服务器，等待客户端连接
   * broadcast(String str): 广播消息给所有客户端
   * logToFile(String message): 将日志写入文件
   * getCurrentTime(): 获取当前时间
4. **内部类 ClientConn**
   * 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

**客户端 ClientChar 类**

1. **属性**
   * boolean isConn: 标记是否连接到服务器
   * String name:客户端用户名
   * boolean isAnonymous:是否选择匿名聊天模式
   * JTextArea jta:用于显示聊天信息的文本区域
   * JScrollPane jsp:滚动面板，用于显示文本区域
   * JPanel jp:面板，用于放置输入文本框和发送按钮
   * JTextField jtf: 输入消息的文本框
   * JButton jb1: 发送消息的按钮
   * Socket socket: 套接字对象
   * String CONNSTR: 连接服务器的IP地址
   * int CONNPORT: 连接服务器的端口号
   * DataOutputStream dos = null; // 数据输出流
2. **构造方法**
   * 初始化窗口组件，设置界面布局
3. **方法**
   * init (): 设置GUI界面、处理用户登录、连接服务器
   * sendMessage(String str): 发送消息到服务器
   * handleSystemCommand(String command): 处理系统命令
   * keyPressed(): 处理键盘按键按下事件
4. **内部类 Recive**
   * 接收服务器发送的消息

**类的关系**

### 服务端代码类关系

1. **ServerChar**
   * **关系**: 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，接受客户端连接，并为每个客户端启动一个新的处理线程。
   * **包含的类**:
     + ClientHandler: 内部类，用于处理客户端通信。
2. **ClientHandler**
   * **关系**: ServerChar 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。

**客户端代码类关系**

1. **ClientChar**
   * **关系**: 主类，继承 JFrame 并实现 KeyListener 接口。
   * **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**:
     + Receive: 内部类，实现 Runnable 接口，用于接收服务器发送的消息。
2. **Receive**
   * **关系**: ClientChar 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。

**客户端和服务端的相互关系**

* **Socket 连接**:
  + ClientChar 类在 init 方法中通过 Socket 连接到服务器 ServerChar。
  + ServerChar 类接受客户端连接，并为每个连接创建一个新的 ClientHandler 线程处理。
* **数据流**:
  + ClientChar 和 ClientHandler 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**:
  + ClientChar 负责发送消息和处理系统命令，并通过 Receive 类接收服务器的消息。
  + ServerChar 的 ClientHandler 负责处理客户端发送的消息，并转发给其他连接的客户端。

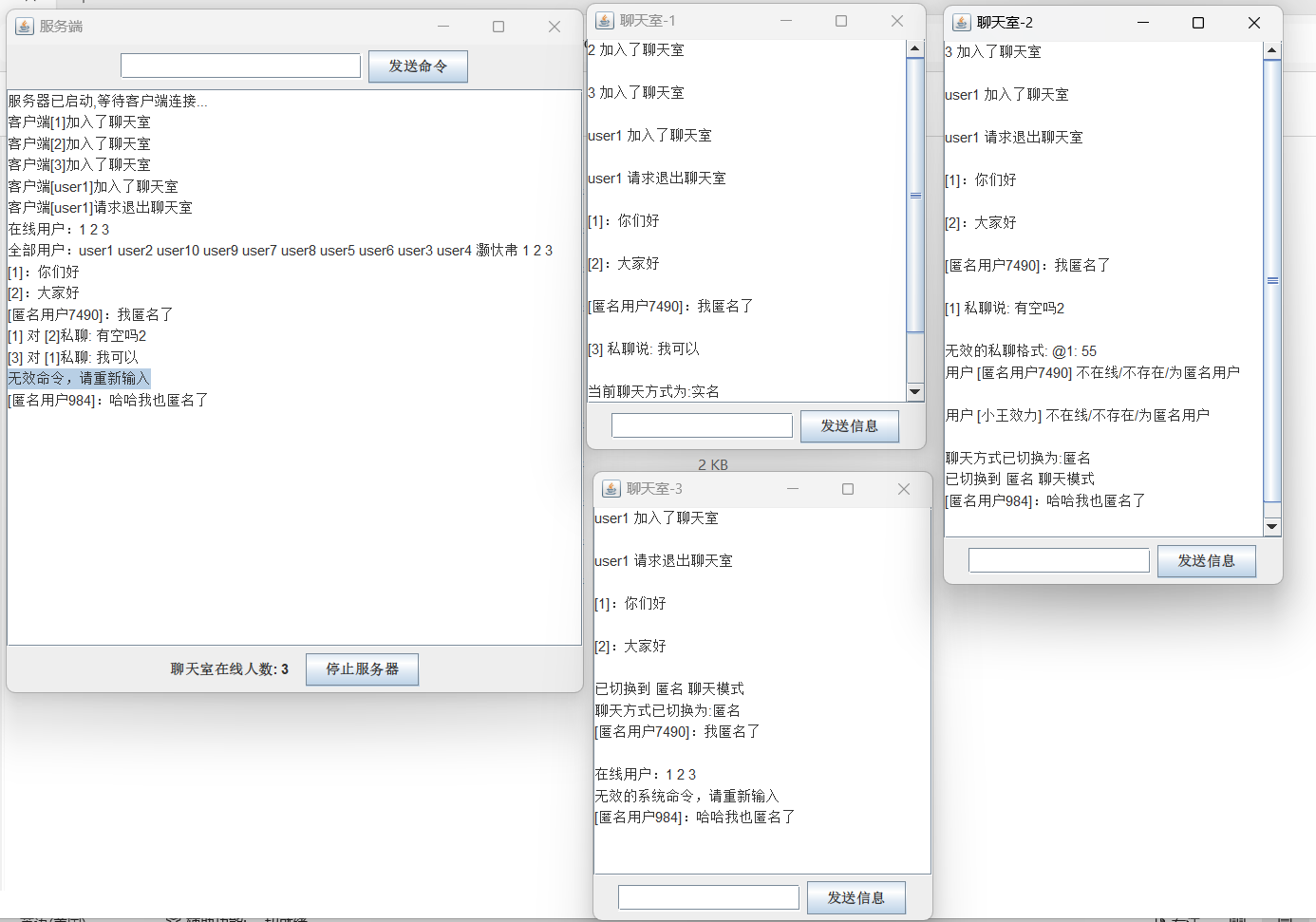
**使用说明**

 **服务器端**

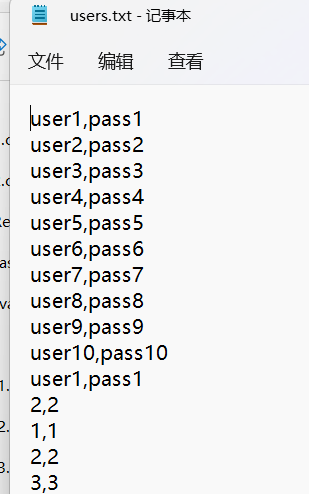
* 在同级目录下创建users.txt储存用户的用户名与密码，格式为：user1,pass1。并创建log.txt作为日志记录每个用户的行为
* 启动 ServerChar 类的主方法，运行服务器。
* 界面上显示发送命令按钮、当前客户端连接状态、在线人数、停止服务器按钮、转发的客户端聊天信息
* 可通过输入命令（list, listall, quit）管理服务器（list：列出全部在线用户；listall：列出全部用户；quit：退出系统），输入错误会显示“无效命令，请重新输入”。
* 可在log.txt中查看用户每次登录成功信息（包含IP地址，时间）、每次登录失败信息（包含IP地址、时间等）、每次退出聊天室时间等

 **客户端**

* 启动 ClientChar 类的主方法，运行客户端。
* 登录窗口输入已在users.txt中记录的用户名和密码，点击登录按钮，输入错误可以重新输入。
* 成功登录后进入聊天界面，可以发送和接收消息。
* 每个用户的聊天室界面会显示广播聊天信息、用户连接/退出聊天室信息、对自己的私聊信息，与自己发送的命令回复信息
* 可使用“@目标用户名：”格式私聊在线的用户（注意为中文冒号，否则为无效的私聊格式），若当前私聊的用户不在线或不存在则有错误提示
* 输入@@命令，为系统命令。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。输入错误的命令会显示“无效的系统命令，请重新输入”。
* 当处于匿名聊天方式时，用户在聊天室中的广播发言用户名会更改为“匿名用户+随机数”。请注意：对于匿名状态下的私聊消息对方消息界面仍显示匿名用户的真是用户名，避免匿名私聊骚扰情况，同时也不可以匿名用户的名字被私聊。

**样例演示**



**** ****