**基于字符界面的C/S结构的简易聊天室程序**

1. **程序要求**

**①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。

**②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；

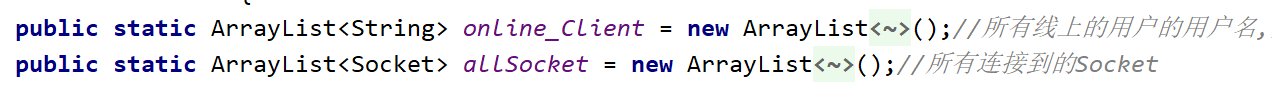
4、聊天室内要能显示所有用户的聊天信息；

5、在聊天室内用户可以输入两类字符串：①普通字符串，为广播类聊天信息，所有人都可以看到；②@+命令，为系统命令；

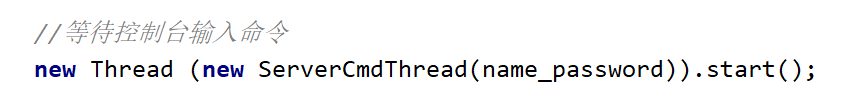
6、系统命令有：list：列出当前在线用户；quit：退出系统。

1. **整体设计思路**
2. 需要两个可运行的类（Server类和Client类）
3. Server类首先需要读入一个装有所有用户名和密码的txt文档，可使用输入输出流完成该操作，并将用户名--密码当作一对key和value存进一个Treemap，方便之后检查客户端输入的密码是否正确
4. Server类需要一个专门处理Server的控制台输入的命令的线程（ServerCmdThread）
5. Server还需要一个类来处理每个客户端的登录（CheckClient），即检查客户端输入的用户名和密码是否匹配，匹配的话，启动该客户端的聊天线程（ClientChatThread）
6. Client类只需要两个线程，一个负责读入服务端发送过来的消息（ReadThread），另一个负责读入控制台的输入并将其发送给服务器
7. **每个类的具体实现**
8. **Server类（服务器类）**

①需要两个public static变量，方便在其他线程调用；记录了所有在线的用户用户名和socket

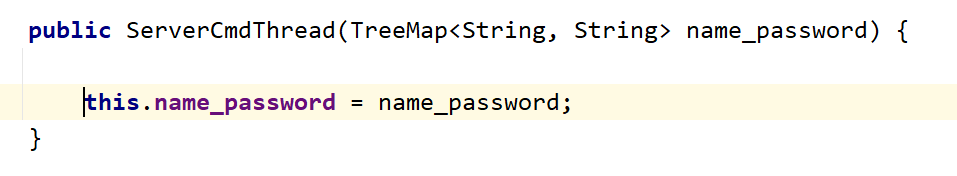
②用输入输出流读入所有用户名和密码，存入一个Treemap

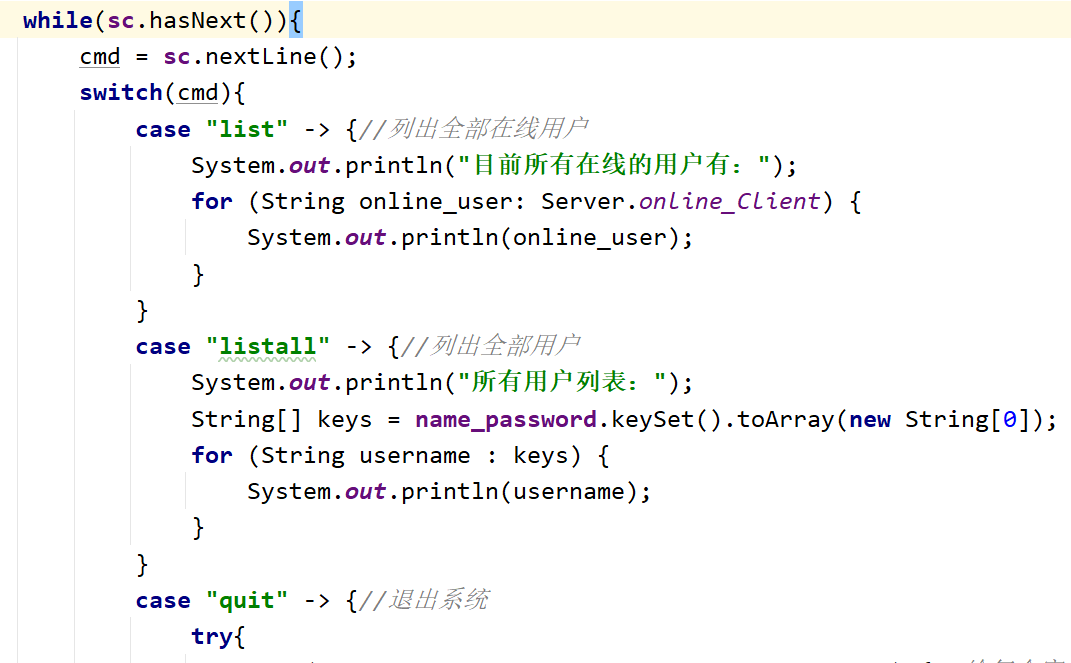
③启动处理控制台命令的线程

④开始侦听8300端口，每侦听到一个用户尝试接入，就启动检查密码的线程

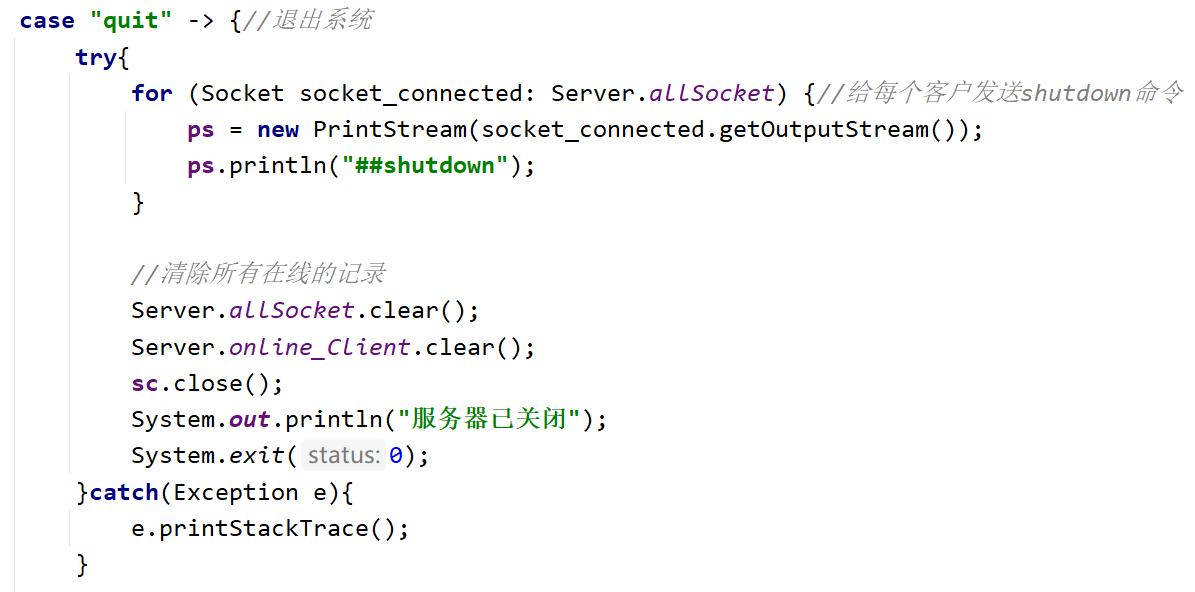
1. **ServerCmdThread类（处理控制台命令的线程）**

①new这个线程时需要传入所有用户的用户名和密码

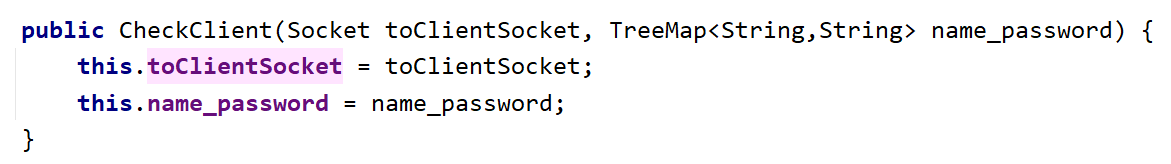
②持续不断地读入控制台的输入，根据不同的输入选择执行不同的语句（switch-case）

③list和listall命令均只需调用相应的ArrayList并使用for-each输出

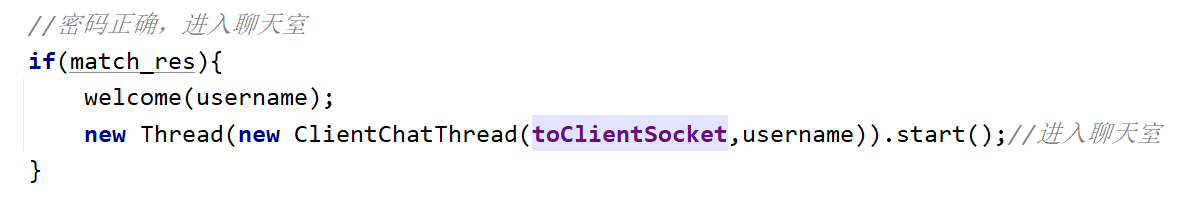
④quit命令需要给所有在线的用户发送关闭连接的命令“##shutdown”，并退出程序（System.exit(0)）

1. **CheckClient类（检查用户名是否与密码匹配）**

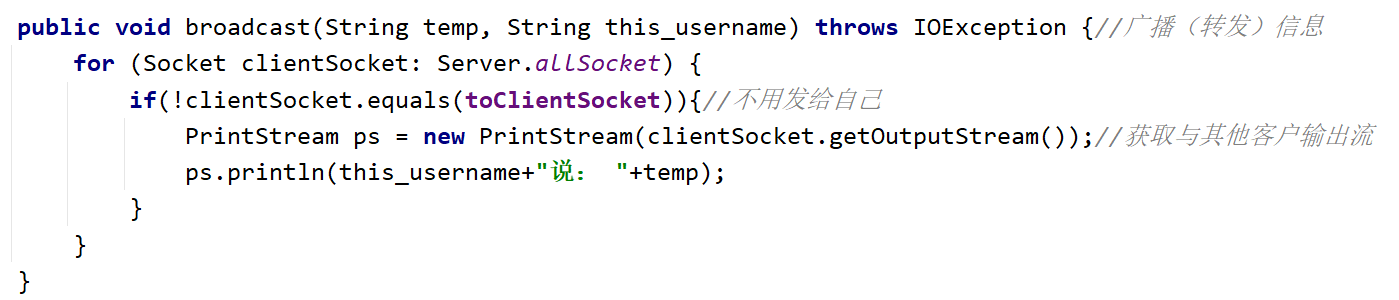
①构造函数需要socket和Treemap两个参数

②读入客户端的输入，并match，直到match成功或者用户放弃登录，否则一直循环读入和match的操作

③若密码正确，则启动该用户的聊天线程（ClientChatThread）

1. **ClientChatThread类（某位用户的聊天线程）**

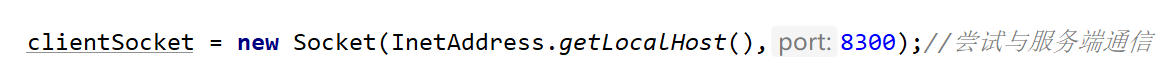
①该线程的主要功能是将该用户输入的消息转发给其他所有用户；注意本程序并不把自己发送的消息显示在自己的控制台上

②若客户端输入的是“@\*”的命令字符，则用switch-case选择相应的语句执行命令



1. **Client类（客户端）**

①与服务器建立连接

②启动读取服务器输出的信息的线程（ReadThread）

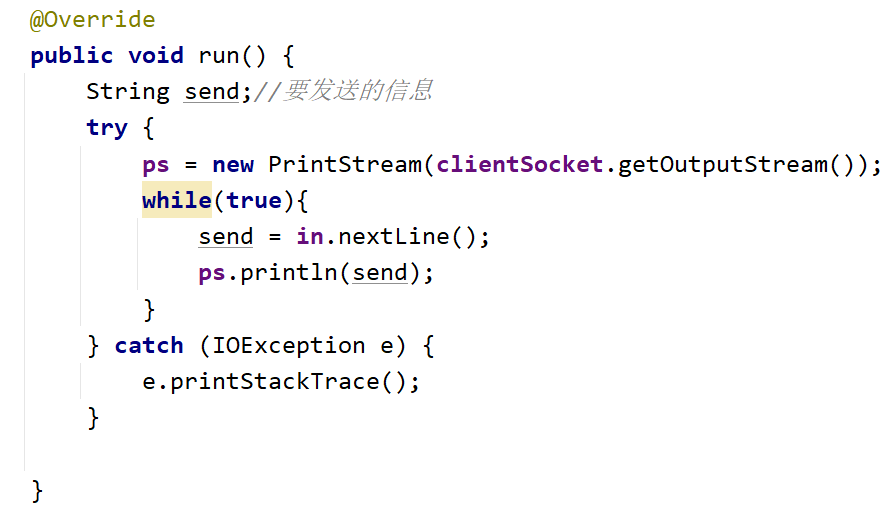
③启动读入控制台输入并发送给服务器的线程（SendThread）

1. **ReadThread类**

①持续循环读入网络输出流的信息，直到服务发来##shutdown的命令

1. **SendThread类**

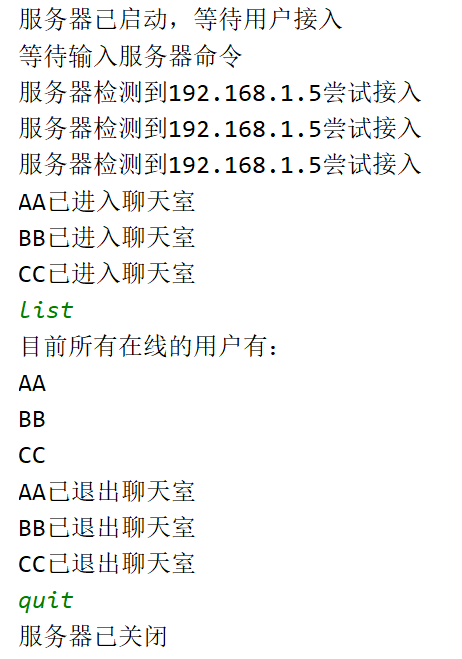
**①持续循环读入控制台的输入，并println到网络输出流**



**四、实验结果**

**A视角： B视角：**



**C视角： 服务器视角：**