```
package com.sxt.chat04;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.concurrent.CopyOnWriteArrayList;

/**
 * 在线聊天室 : 服务器
 * 目标：加入容器实现群聊
 * @author 江
 *
 */

public class Chat {
        private static CopyOnWriteArrayList<Channel> all=new
CopyOnWriteArrayList<>();


    public static void main(String[] args) throws IOException {
            System.err.println("--server--");
            //创建服务器端
            ServerSocket server=new ServerSocket(8888);
            while(true) {
            //阻塞式连接
            Socket client=server.accept();
            System.err.println("一个客户端建立连接");
            Channel c=new Channel(client);
            all.add(c);    //管理所有的成员
            new Thread(c).start();

            }
        }
    //一个客户代表一个Channel
    static class Channel implements Runnable{
        private DataInputStream dis;
        private DataOutputStream dos;
        private Socket client;
        private boolean isRunning;
        private String name;

        public Channel(Socket client) {
                this.client=client;
                try {
                                dis=new
DataInputStream(client.getInputStream());
                                dos=new
DataOutputStream(client.getOutputStream());
                                isRunning=true;
                                //获取名称
                                this.name=receive();
                                //欢迎您的到来
                                this.send("欢迎您的到来");
                                sendOthers(this.name+"来到了上海尚学堂聊天
室",true);
                        } catch (IOException e) {
                                e.printStackTrace();
                                release();
                        }


        }
        //接收消息
        private String receive() {
                String msg="";
                try {
                                msg=dis.readUTF();
                        } catch (IOException e) {
```

```
                        e.printStackTrace();
                        release();
                }
        return msg;
}

//发送消息
private void send(String msg) {
        try {
                        dos.writeUTF(msg);
                        dos.flush();
                } catch (IOException e) {
                        e.printStackTrace();
                        release();
                }
}

/**
 * 群聊：获取自己的信息，发送给其他人
 * 私聊：约定数据格式：@xxx：msg
 * @param msg
 */
private void sendOthers(String msg,boolean isSys) {
    boolean isPrivate=msg.startsWith("@");
    if(isPrivate) {//私聊
        int idx=msg.indexOf(":");
        //获取目标和数据
        String targetName=msg.substring(1,idx);
        msg=msg.substring(idx+1);
        for(Channel others:all) {
                if(others.name.equals(targetName)) {//目标
                        others.send(this.name+"对您说："+msg);
                        break;
                }
        }
```

```
            }else {
                        for(Channel others:all) {
                                if(others==this) {
                                        continue;
                                }
                                if(!isSys) {
                                others.send(this.name+"对所有人说:"+msg);
                                }else {
                                        others.send(msg);
                                }


                        }
            }
        }
        //释放资源
        private void release() {
                this.isRunning=false;
                        Utils.close(dis,dos,client);
            all.remove(this);
            sendOthers(this.name+"离开了群聊",true);
        }
                @Override
                public void run() {
                          while(isRunning) {
                                  String msg=receive();
                                  if(!msg.equals("")) {
                                          //send(msg);
                                          sendOthers(msg,false);
                                  }
                          }
                }
    }


}
```