ChatApplication

import java.util.\*;

class ChatApplication {

public static void main(String args[]) throws Exception {

Scanner s = new Scanner(System.in); // Replace with file to take inputs from a file.

String []userAlias = s.nextLine().split(" "); // Take user aliases logged into the chatroom

ChatRoom cr = new ChatRoom(userAlias);

while(s.hasNext()) {

String what = s.next();

if (what.equals("Q")) { // Print user's sent/received messages

String user = s.next();

User queryUser = cr.getUserReference(user);

System.out.println(user+"'s sent messages:");

queryUser.printSentMessages();

System.out.println(user+"'s received messages:");

queryUser.printReceivedMessages();

System.out.println();

}

else if (what.equals("C")) { // A user is sending a message to another user

String from = s.next();

String to = s.next().split(":")[0];

String message = s.nextLine();

User fromUser = cr.getUserReference(from);

User toUser = cr.getUserReference(to);

fromUser.writeMessage(message, toUser);

}

}

}

}

ChatRoom

/\*\*

\* The chatroom maintains the users logged into it.

\*/

class ChatRoom {

User users[];

public ChatRoom(String[] userAlias) {

users = new User[userAlias.length];

for (int i = 0; i < userAlias.length; i++)

this.users[i] = new User(userAlias[i], this);

}

public void transferMessage(String message, User to) {

to.receiveMessage(message);

}

public User getUserReference(String alias) {

for (int i = 0; i < users.length; i++) {

if (users[i].getAlias().equals(alias)) {

return users[i];

}

}

return null;

}

}

User

/\*\*

\* The user has an attribute for alias, and keeps track of her sent/received messages.

\*/

class User {

public static final int MAX\_CHAT = 10;

private String sentMessages[] = new String[MAX\_CHAT];

private String receivedMessages[] = new String[MAX\_CHAT];

private String alias;

private int totalSent = 0;

private int totalReceived = 0;

private ChatRoom assignedChatroom;

public User(String alias, ChatRoom cr) {

this.alias = alias;

this.assignedChatroom = cr;

}

public void writeMessage(String message, User u2) {

sentMessages[totalSent++] = message;

assignedChatroom.transferMessage(message, u2);

}

public void receiveMessage(String message) {

receivedMessages[totalReceived++] = message;

}

public void printSentMessages() {

for (int i = 0; i < totalSent; i++)

System.out.println(sentMessages[i]);

}

public void printReceivedMessages() {

for (int i = 0; i < totalReceived; i++)

System.out.println(receivedMessages[i]);

}

public String getAlias() {

return alias;

}

}