

Progetto Chat tra Server-Client

indice

- 1) *Descrizione panoramica*
- 2) *Descrizione soluzione*
- 3) *Descrizione del protocollo*
- 4) *Descrizione del progetto chetServer*
 - *Descrizione della classe Server*
 - *Descrizione della classe GestioneChat*
- 5) *Descrizione del progetto chatClient*
- 6) *Conclusione*

1) Descrizione panoramica

Il progetto consiste gestione di una chat, una semplice applicazione di chat multiutente con Java e swing. La chat possiede un protocollo per far comunicare tra client- server.

2) Descrizione soluzione

Per la risoluzione del problema ho utilizzato i socket perché è un *oggetto software* che permette l'invio e la ricezione di dati, tra host remoti (tramite una rete) o tra processi locali(nello stesso pc). Quindi per la comunicazione tra server e client. Per fare ciò Socket devono sapere l'indirizzo e porta di client e server; tra di loro c'è una connessione logica. Utilizziamo i socket di tipo Stream socket: orientati alla connessione (connection-oriented), basati su protocolli affidabili come TCP. Visto che non gestiremo un client perché è una chat tra cui client comunica con altri client, il server a il compito della gestione, quindi inoltrare i vari messaggi e la loro connessione/disconnessione utilizziamo i thread visto che c'e bisogno della gestione di client in contemporaneo.

3) Descrizione del protocollo

Codice	Cosa fa/serve?
IAM " "nickname "	<i>Va inserito dopo la richiesta del server CHI rispondi con il messaggio. Serve per identificare l'utente.</i>
CHI	<i>digitando questo comando, serve per vedere tutti gli utenti collegati</i>
MSG "testo".	<i>questa sintassi serve per inviare messaggi in chat pubblica</i>
PRV [user1 user2] "testo".	<i>questa sintassi serve per inviare messaggi in chat privata</i>
BYE	<i>questa sintassi serve per uscire dalla chat</i>

Nota:

- il nome utente non deve contenere alcuno spazio
- In tutti gli altri casi il server risponderà al messaggio: "MESSAGGIO Sconosciuto".
- Significa che il messaggio non è stato inviato e l'utente deve utilizzare uno dei comandi delle sezioni precedenti.

4) Descrizione del progetto chatServer

In questo progetto che solo la creazione 2 due classi che sono:

- ★ Server
- ★ GestioneChat

Descrizione della classe Server

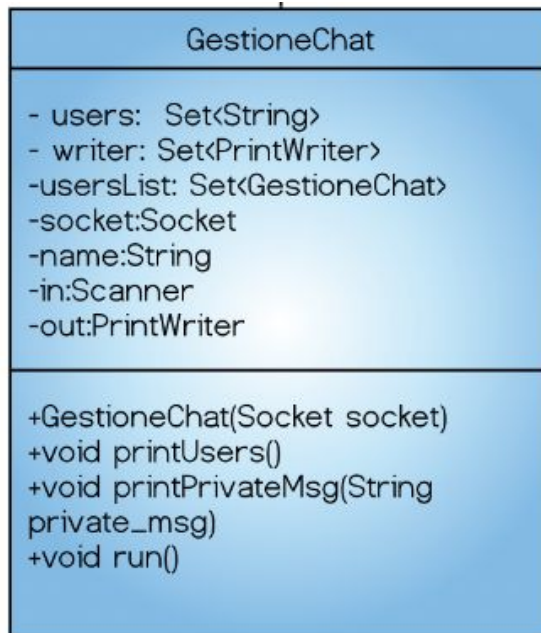
In questa classe avviene l'istanza della classe Gestione Chat, dopo avviene tutta la gestione. Utilizzo un pool perché mi consente di isolare le applicazioni l'una dall'altra, anche se sono in esecuzione sullo stesso server. In questo modo, se si verifica un errore in un'app, non eliminerà altre applicazioni. Inoltre, i pool di applicazioni consentono di separare app diverse che richiedono diversi livelli di sicurezza. E io lo utilizzo come controllo se il server è in funzione.

Descrizione della classe GestioneChat

In questa gestisce le varie funzionalità che sono:

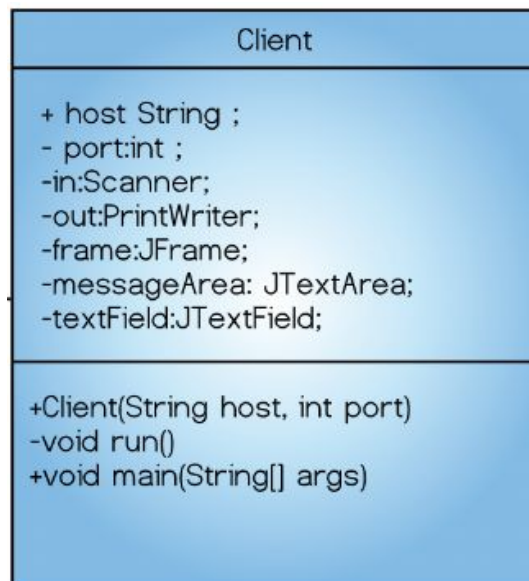
- ★ deve inoltrare i messaggi all'altro client
- ★ deve gestire la situazione in cui ci sia solo un solo client collegato (impossibile inoltrare il messaggio)
- ★ il client deve gestire attraverso un messaggio speciale la chiusura della connessione verso il server (utilizzando il protocollo di comunicazione)
- ★ gestione della disconnessione del client
- ★ i client devono conoscere i nomi dei client connessi
- ★ avere la possibilità di avviare chat one to one oppure one to all.

Questa all'interno della classe ciò che si vedrà nella finestra, visto che il server inoltra tutto al cliente, saranno presenti delle scritte che illustrano il funzionamento, grazie al protocollo di comunicazione avverranno i scambi di messaggi. Nella finestra del server vedremo i collegamenti dei vari clienti quindi la connessione e disconnessione. Utilizzare gli array per potere gestire le varie informazioni.



5) Descrizione del progetto chatClient

In questa classe avviene solo la creazione dei Client, che poi saranno identificati da un nome, così che il server li riconosca. In questa classe avviene anche la gestione grafica della chat. La gestione messaggistica avviene tutto nel server. Quindi dopo essersi connesso con il server, il server invia la richiesta del nome dell'utente, successivamente, il client dopo aver inviato il nome, può inviare messaggi.



6) Conclusione

Alla conclusione, la chat funziona ma con qualche errore sulla comunicazione in chat privata, la grafica è basilare, ma complessivamente funzionante.

