

TCP-ChatRoulette

Questo compito di programmazione offre l'opportunità di implementare una semplice applicazione di chat utilizzando il protocollo TCP. Il collegamento chat dovrebbe avvenire simile a ChatRoulette, dove due utenti vengono collegati a caso. Conta il 10% per il voto finale.

Istruzioni: Potete lavorare da soli o in gruppi di due/tre studenti.

Vi viene chiesto di sviluppare un sistema di chat su Internet, basato su un'architettura client/server, utilizzando il linguaggio Java e sfruttando il protocollo TCP:

- Il vostro sistema deve permettere di connettere *molteplici* client remoti a un singolo server centrale.
- Quando un utente inserisce un messaggio di testo sul proprio client, il messaggio viene consegnato attraverso il server e visualizzato da qualsiasi altro client che è attualmente connesso al server, compreso il client originario.
- Gli utenti possono entrare e uscire dalla chat in qualsiasi momento, purché il server sia attivo.
- Quando si uniscono alla chat, gli utenti scelgono dei nickname, che appariranno insieme ai loro messaggi. L'indirizzo del server deve essere specificato alla riga di comando quando si esegue il client, oppure tramite un menu interattivo dedicato.

Caratteristiche

Per semplicità, il client può essere eseguito in una singola finestra in modalità testo. Sono necessarie le seguenti caratteristiche d'implementazione:

- Gli utenti possono disconnettersi in modo controllato digitando “/quit”.
- I client possono scegliere i nomi utente che vengono visualizzati accanto ai loro messaggi di chat e per tutte le loro azioni (registrazione, messaggistica, disconnessione) nel log del server.

- una volta collegato, un client viene "connesso" a caso, per esempio dopo un ritardo di 30 secondi, a un altro client libero (se presente) e chatta solo con esso. (testate con almeno 4 client). Questo step si ripete se uno dei client si scollega
- Il sistema nel suo complesso deve essere robusto per quanto riguarda i crash dei client e/o le connessioni bloccate, vale a dire che qualsiasi problema con un singolo client non deve avere ripercussioni sugli altri utenti.

Opzionale: Messaggistica privata a terzi e/o cambio partner digitando "/shuffle"

Compiti

Vi viene chiesto di:

1. Scrivere i requisiti di ingegneria del software per il sistema, includendo ogni diagramma necessario, oppure in alternativa un codice sorgente ben scritto (struttura chiara, nomi di identificatori significativi, ecc.) e correttamente commentato (struttura generale, scopo delle classi e dei metodi, eventuale concorrenza delle classi e dei metodi, o thread).
2. Descrivere la progettazione dell'algoritmo, mostrando quali interazioni avvengono tra i client e il server.
3. Sviluppare un'implementazione Java basata su TCP sia del client che del server.
4. Testate la vostra implementazione e segnalate eventuali bug e/o comportamenti inaspettati che dovrete trovare.
5. Discutere brevemente una possibile implementazione alternativa che sfrutta il protocollo UDP, facendo un confronto critico tra le due.

Deliverables

Presentare un singolo file zip ChatSystem.zip che contenga

1. Una relazione tecnica di 2-3 pagine in formato PDF che discute i punti 1*, 2, 4 e 5 di cui sopra
2. (Almeno) Due classi Client.java e Server.java che implementano le funzionalità del client e del server nel loro metodo principale.

Termine di consegna: Mercoledì 6 Novembre, ore 23:55 via Ole.