I .I.S "A. AVOGADRO" di TORINO

RFC per Calcolatrice

TPSI

Autore: Ramli Adam

ANNO SCOLASTICO 2025/2026

Riepilogo

Il protocollo realizzato dal sottoscritto serve a far comunicare due calcolatori, corrispettivamente un client che richiede che una serie di calcoli vengano fatti da un dispositivo server e quest'ultimo restituisca il risultato quando richiesto.

Panoramica del protocollo

Il funzionamento del protocollo è molto semplice. Un computer ha bisogno di eseguire dei calcoli ma quest'ultimi invece di essere eseguiti localmente (per localmente si intende senza un'architettura Client-Server quindi facendo uso di un codice che viene eseguito da un singolo calcolatore senza fare affidamento a un computer su di una rete) vengono affidati ad un secondo calcolatore che ha il ruolo di Server che in base ai comandi inviategli eseguirà delle operazioni su di un risultato. La connessione da parte del client viene al momento dell'invio dell'username. Infatti il server funziona tramite una mappa Hash che tiene conto degli username connessi poiché la calcolatrice supporta la funzione di connessione di più client in contemporanea.

Il protocollo a livello di trasporto usato è il TCP. Anche se non è una scelta personale poiché la libreria `java.net` fa uso della pila protocollare TCP/IP, trovo sia corretto far uso di questo protocollo poiché garantisce che i dati richiesti vengano inviati con successo grazie al controllo del flusso a differenza di UDP.

Il numero di porta su cui è in ascolto il server non è di rilevanza particolare ma è stata scelta la 10000 (1-1024 porte conosciute.).

Il controllo degli errori implementato consiste nell'uso delle librerie per la gestione di eccezioni. In particolare si fa uso della `NumberFormatException` per gli errori di formato numerico nel caso ad esempio l'utente immettesse valori invalidi come stringhe. Inoltre si fa uso della `ArithmeticException` per gli errori aritmetici casuali di ogni tipo. Infine l'ultimo controllo a livello di input numerico sono le divisioni per 0 che vengono ignorate e mandate in output nello `standard error` insieme agli altri due casi citati precedentemente.

L’altra serie di controlli che viene effettuata si presenta quando il comando immesso dal client è invalido e questo errore viene mostrato come ultima istruzione di un blocco condizionale.

TUTTI GLI ERRORI VENGONO MANDATI AL CLIENT COME FORMA DI ERRORE;NUM. IL CLIENT MOSTRERà UNA FINESTRA ISTANTANEA DI ERRORE MENTRE IL SERVER PRESENTA L’ERRORE IN PARTICOLARE **(L’unico errore che fa eccezione è sull’I/O che viene presentato solo al Server.).**

Per quanto riguarda la connessione non ci sono controlli sugli errori particolari poiché è gestito dalle eccezioni presenti nella libreria `java.net`.

Il protocollo non presenta alcuna conferma del servizio.

Comandi del server

Il server non manda comandi particolari dopo le operazioni richieste dal client ma solo una conferma di tipo “SUCCESS”. Un ulteriore conferma che viene mandata è quando avviene un errore o più in particolare quando il client fa uso del comando `EXIT`. Infatti con quest’ultimo richiede la chiusura definitiva della connessione anche da parte del server e quest’ultimo apparato manda una conferma ulteriore di `EXIT` e solo dopo questa il client potrà chiudersi definitivamente insieme al server.

Comandi del client

I comandi del client sono i seguenti:

- **+F** : richiede la somma di un numero di tipo float al risultato (inizialmente 0);
- **-F** : richiede la differenza di un numero di tipo float al risultato;
- ***F** : richiede la moltiplicazione di un numero di tipo float al risultato;
- **/F** : richiede la divisione di un numero di tipo float al risultato;
- **=** : richiede che il risultato venga restituito;
- **EXIT** : richiede la chiusura della connessione e di conseguenza del programma da parte del client e del server;
- **C** : richiede il reset del risultato a 0;
- **username**: inserimento dell’username.

Errori del protocollo

Gli errori che prevede il protocollo richiedono tutti che il client abbia una finestra di errore quando si verificano (tranne per gli errori di I/O del server). Inoltre il server è l'unico ad avere nello standard error gli errori che si verificano nello specifico mentre il client riceve solo ERRORE;NUM. Eccoli elencati:

- ERRORE;00: errori di formato numerico o input/output.
- ERRORE;01: divisione per 0.
- ERRORE;02: errore generale quindi comando invalido.

Sequenza temporale