Esercitazione di Programmazione Orientata agli Oggetti

14/12/2017

Realizzare un'applicazione client-server che permetta di visualizzare lato client lo stato di un sistema monitorato dal server. Il server è multithreading ed accetta connessioni da più client. Ogni client connesso visualizza lo stato di un differente sistema monitorato (due client connessi allo stesso server di norma visualizzeranno differenti stati). I client una volta connessi possono iniziare a scaricare dal server stringhe indicanti un codice di stato, una dimensione di font, messaggio da visualizzare. I valori ricevuti dal server sono mostrati all'utente dal client direttamente nell'interfaccia grafica. Tutte le stringhe sono inviate da client a server e viceversa utilizzando il carattere di fine linea come separatore.

Si richiede la realizzazione del client, con interfaccia grafica e networking.

Il client dovrà essere composto da un frame che abbia per titolo *nome cognome e matricola* dello studente e che contenga due campi testuali (IP Address, Port), quattro pulsanti (Connect, Start, Stop, Disconnect) ed un pannello che mostri lo stato comunicato dal server, il tutto disposto secondo l'immagine riportata:

Si implementi il seguente protocollo:

- Alla pressione del pulsante "Connect", il client deve inviare una richiesta di connessione al server utilizzando indirizzo IP e porta indicate nei campi testuali "IP Address" e "Port".

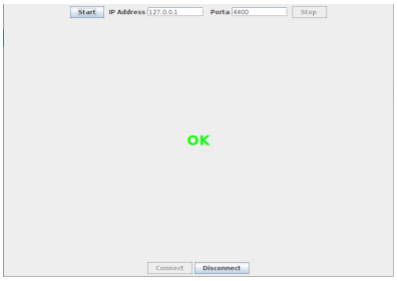
- Alla pressione del pulsante "Start", dopo la connessione, il client deve mandare il comando "start" al server, il quale inizierà ad inviare ad intervalli regolari una stringa composta nel seguente modo:

  codice_colore;dimensione_font;messaggio

  dove:

  -*codice_colore*: stringa rappresentante un intero (0,1,2,3)
  -*dimensione_font*: stringa rappresentante un intero
  -*messaggio*: stringa contenente il messaggio da mostrare

  esempio: "3;28;OK" va visualizzato come nell'immagine →



Il client alla ricezione della stringa dovrà valutare il *codice_colore*, ed in base ad esso mostrare nel pannello il *messaggio* nel colore e nella *dimensione_font* indicata (vedi hint)*:

codice_colore = "3", verde
codice_colore = "2", arancione
codice_colore = "1", rosso.
Alla ricezione del *codice_colore* = "0", il client deve mostrare il *messaggio* nella *dimensione_font* indicata (come nei precedenti casi) in nero e successivamente interrompere la ricezione dei messaggi.
Il messaggio iniziale è lasciato a discrezione dello studente.

- Alla pressione del pulsante "Stop", dopo l'avvio, il client deve mandare al server il comando "stop". Il server risponderà inviando una stringa che avrà *codice_colore* = "0" ed il cui *messaggio* dovrà essere visualizzata sulla GUI come ultimo messaggio."

- Alla pressione del pulsante "Disconnect" il client invia al server il comando "disconnect" e chiude la connessione

Gestire correttamente la possibilità di premere i pulsanti: in particolare

- all'avvio solo il pulsante "Connect" è abilitato
- una volta avviata una connessione, non deve poter essere premuto il pulsante "Connect"
- "Disconnect" può essere premuto solo se una connessione è avviata ma non si stanno ricevendo stringhe dal server
- "Stop" può essere premuto solo se il client è connesso e sta ricevendo stringhe
- "Start" può essere premuto solo se il client è connesso e non sta ricevendo stringhe

*Hints:

Per modificare il colore, utilizzare il metodo *setForeground(Color c)* , per i colori fare riferimento ai campi statici offerti dalla classe *java.awt.Color*. Consultare la documentazione per ulteriori dettagli.

Per modificare il font, utilizzare la classe *java.awt.Font.* In particolare il nuovo font da assegnare al pannello dovrà avere lo stesso nome del vecchio font, stile "BOLD" (fare riferimento ai campi statici della classe Font) e size indicata dal server in *dimensione_font.* Consultare la documentazione per ulteriori dettagli.