



# Esercizi

1. Si crei un server multithread, che dopo la connessione invii al client la stringa 'La tua connessione è gestita dal <nome del thread>'
  - a) Introdurre anche un file di log delle connessioni avvenute e in corso. Il file deve memorizzare IP e porta dell'host remoto. Il file va condiviso tra i thread oppure no?
  - b) Introdurre un file di log SOLO delle connessioni aperte
  - c) Introdurre una struttura dati che memorizzi le connessioni aperte
  - d) Il server invia anche una serie di stringhe contenenti IP e porta delle connessioni aperte. Per esempio  
145.255.23.1:674  
143.67.90.2:6321

2. Si implementi una classe User con i seguenti campi:

boolean isLoggedIn

String email (campo univoco), non possono esserci due users con la stessa e-mail.

String password

Definendo un costruttore vuoto, un costruttore con 2 argomenti Stringa e uno con 3 argomenti. Si definiscano i metodi di set e get.

3. Si implementi la classe Users, modellante un insieme di oggetti User, che fornisce i seguenti metodi:

User getUser(String email): restituisce l'oggetto User associato alla email

Boolean inserisciUser(User u): inserisce nell'insieme un oggetto User, restituisce true se l'inserimento ha avuto successo, false altrimenti (utente già esistente).

4. Si implementi la seguente funzione di 'login' modificando il prodotto nell'esercizio 1. Utilizzando un qualsiasi client (telnet, putty per esempio), si invia una stringa del tipo

login?user=<email>&psw=<password>

1. Il server deve visualizzare sul suo standard output la stringa ricevuta. Quale thread visualizza l'output (main server o thread di servizio)?
2. Il server deve visualizzare solo le informazioni relative al campo user e psw
3. Il server deve cercare l'oggetto user relativo alla mail specificata e inviare al client la stringa corrispondente



# Esercizi

All'utente associato alla mail, se l'utente esiste

4. Il server setta l'utente come loggato utilizzando la seguente regola, se lo stato dell'utente è NonLoggato cambia lo stato e invia al client il messaggio: Utente loggato con successo, se l'utente risulta già loggato invia il messaggio: Utente già loggato.
  5. Si modifichi il secondo messaggio, specificando da quale indirizzo IP + porta l'utente è loggato
  6. Il server, dopo il messaggio deve inviare al client loggato tutte le connessioni aperte
- PS: All'atto della creazione del server, gli utenti vengono caricati da un file utenti.txt che segue il formato: <email>\t<password>\r\n
5. Si stampino tutti gli indirizzi IP associati al nome www.cnn.com
  6. Si crei un programma che genera un indirizzo IP casuale e si cerchi di risolverlo.
  7. Implementare un server multithread che, quando contattato, mandi al client una serie di stringhe contenenti la sequenza dei numeri naturali. Stringhe inviate ad intervalli regolari di un secondo. L'output non termina mai