

# Programmazione Orientata agli Oggetti


## Esercitazione del 5.12.2017

(se svolta come simulazione di una prova di progetto di esame - tempo a disposizione 2h:15m)

Si vuole realizzare un'applicazione client/server che permetta lo scambio di caratteri binari. Un server multithreading accetta connessioni da più client. I client, una volta connessi, possono iniziare a scaricare dal server sequenze di valori 0-1 (stringhe). I valori ricevuti dal server sono scritti dal client in un file. 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 tre campi testuali (Matricola, IP Address, Porta) e da quattro bottoni (Connetti, Disconnetti, Start, Stop) disposti secondo l'immagine riportata:

The image shows a graphical user interface (GUI) for a client application. It consists of a window with a title bar and a close button (X) in the top right corner. Inside the window, there are three text input fields arranged vertically. The first field is labeled 'Matricola', the second 'IP Address', and the third 'Porta'. Below these fields, there are four buttons arranged horizontally: 'Connect', 'Disconnect', 'Start', and 'Stop'. The buttons have a blue gradient and white text.

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 “Porta”.
- Alla pressione del pulsante “Start”, dopo la connessione, il client deve mandare il comando `start` al server, il quale inizierà ad inviare una sequenza di stringhe, ognuna di lunghezza uno, contenente i valori “0” oppure “1”. Il client deve ricevere le stringhe e salvarle su un'unica riga in un file `<matricola>.txt`, in cui il valore di `<matricola>` è il contenuto del campo testuale ‘Matricola’.
- Alla pressione del pulsante “Stop”, il client deve mandare al server il comando `stop`. Il server risponderà inviando in sequenza 3 stringhe: \*, un codice di terminazione e `stop`. Il client dovrà scrivere su una nuova riga dello stesso file il codice di terminazione ed interrompere la ricezione dei messaggi.
- Alla pressione del pulsante “Disconnect” il client invia al server il comando `disconnect` e chiude la connessione

Notare che il file scritto dal client deve contenere due linee: la prima linea formata solo dalla sequenza di caratteri 0 o 1 ricevuti, la seconda linea contenente solo il codice di terminazione (senza il carattere asterisco o altre stringhe di controllo provenienti dal server).

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