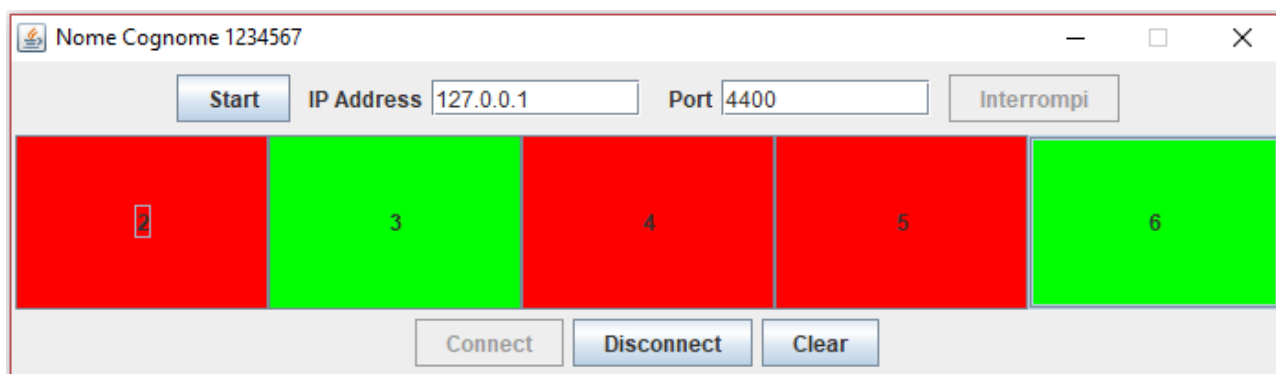


# Progettazione del Software

## Prova Pratica

12/05/2020 – Tempo a disposizione: 120 minuti

Si vuole realizzare un'applicazione client/server che permetta di giocare ad una variante del "Gioco del lotto". Lato client, l'utente inserisce 5 numeri (da 0 a 9) ed avvia l'estrazione da parte del server. Il server genera 5 numeri casuali da 0 a 9, uno per ogni casella. Due caselle differenti possono contenere due numeri uguali. Il server è multithreading ed accetta connessioni da più client. I client, una volta connessi, possono avviare una sessione di gioco, come di seguito specificato. Una volta avviata la sessione, il server invierà stringhe indicanti la posizione della casella e il numero estratto. Le stringhe inviate dal server sono elaborate dal client secondo una logica descritta in seguito. Due client connessi allo stesso server gestiscono partite differenti. Durante la trasmissione delle stringhe, il client può decidere di interrompere la ricezione in qualsiasi momento. La comunicazione è basata unicamente su scambio di stringhe. 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, in grado di comunicare con un server multithreading (fornito).**

L'interfaccia grafica del client (si consiglia 700x200 come risoluzione) dovrà essere composta da un frame che abbia per titolo *nome cognome* e *matricola* dello studente, da due campi testuali (l'indirizzo IP e la porta a cui connettersi che verranno comunicati in aula), da cinque pulsanti (*Connect*, *Disconnect*, *Start*, *Interrompi* e *Clear*) che permettono rispettivamente di gestire la connessione col server, di disconnettersi dal server, di avviare/interrompere una sessione di gioco e di ripristinare il contenuto ed il colore delle 5 caselle. Inoltre, l'interfaccia grafica dovrà essere composta da 1 pannello che a sua volta contiene 5 istanze di `ColoredButton` (caselle) disposte in una griglia 1\*5 come in figura.

All'avvio le caselle devono essere tutte del colore `LIGHT_GRAY`. L'utente deve poter cliccare le caselle per inserire dei numeri da estrarre da 0 a 9. Il click su una casella è già gestito nella classe `ColoredButton`, fornita funzionante. Lo studente, tuttavia, dovrà implementare il metodo `read()` richiamato all'interno della classe `ColoredButton`, per leggere da input una cifra compresa tra 0 e 9. (Suggerimento: si veda la classe `NumberInput` e la relativa interfaccia `Reader`). Se la cifra non è compresa tra 0 e 9, la lettura dovrà essere ripetuta affinché tale condizione sia rispettata.

Si richiede di gestire correttamente la possibilità di premere i pulsanti, in particolare:

- All'avvio solamente i pulsanti *Connect* e *Clear* sono abilitati. Non deve essere possibile avviare 2 connessioni dallo stesso client spingendo ripetutamente il bottone *Connect*.
- *Start* e *Disconnect* possono essere premuti solo se la connessione col server è già avviata.
- Dopo la pressione di *Start* l'unico pulsante che può essere premuto è *Interrompi*.
- Alla pressione di *Interrompi* dovranno tornare attivi solo i pulsanti *Start*, *Disconnect* e *Clear*, per effettuare un nuovo gioco/estrazione, disconnettersi dal server, e ripristinare le 5 istanze di *ColoredButton*.
- *Clear* può essere premuto solo se non si stanno ricevendo stringhe dal server.

Si implementi il seguente protocollo:

- Alla pressione del pulsante *Connect*, il client invierà una richiesta di connessione al server utilizzando Indirizzo IP e porta indicati negli appositi campi.
- Alla pressione del tasto *Start*, dopo la connessione, il client deve controllare che l'utente abbia inserito 5 numeri da estrarre. In caso positivo, deve mandare il comando "*start*" al server, abilitare inoltre il pulsante *Interrompi*, e disabilitare *Start*, *Disconnect* e *Clear*. Il server inizierà ad inviare ad intervalli regolari stringhe nel seguente formato: "*posizione\_griglia;numero\_estratto*".  
Esempio: 3;4. Il dominio del campo *posizione\_griglia* è {'0', '1', '2', '3', '4'}, mentre quello del campo *numero\_estratto* è {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'}.

Durante l'estrazione, il client alla ricezione della stringa dovrà cambiare colore alla casella nel pannello alla posizione indicata da *posizione\_griglia*. Il colore diventerà *Color.GREEN* se la casella è vincente, *Color.RED* altrimenti. Una casella alla posizione *posizione\_griglia* è vincente se *numero\_estratto* per quella casella è uguale a quello che l'utente ha inserito nella fase precedente all'estrazione. Durante l'estrazione (quando si preme *Start*) il click sulle caselle dovrà essere disabilitato.

L'estrazione da parte del server termina quando quest'ultimo, dopo aver estratto il quinto numero invia una stringa che avrà *posizione\_griglia* = '\*' e *numero\_estratto* = '\*'. Alla ricezione di questi due caratteri, il client dovrà smettere di ricevere stringhe, e comunicare, tramite un messaggio in popup il vincitore (Server o Utente). Il criterio di vittoria a fine partita, è il seguente:

*if (n° delle caselle colorate Color.GREEN > 0) then User Wins;*  
*else Server Wins*

Al termine dello scambio, le caselle dovranno rimanere colorate e il click su di esse dovrà essere nuovamente abilitato. Il client dovrà inoltre disabilitare il pulsante *Interrompi* e riabilitare *Start*, *Disconnect* e *Clear*.

Nel caso in cui si avvii una nuova comunicazione dopo l'interruzione (si preme di nuovo *Start*), preliminarmente il colore di ogni casella deve essere riportato a *LIGHT\_GRAY* ed il contenuto delle caselle deve rimanere inalterato.

(**suggerimento:** fare riferimento al metodo `changeColor(Color c)` della classe `ColoredButton` per i colori).

(**suggerimento 2:** fare riferimento al metodo `getDigit()` della classe `ColoredButton` per ottenere la cifra all'interno di una casella, inserita nella fase precedente all'estrazione)

(**suggerimento 3:** fare riferimento al metodo `isGreen()` della classe `ColoredButton` per verificare se la casella su cui viene richiamato tale metodo è di colore verde, così da determinare il vincitore della partita).

- Alla pressione del tasto *Interrompi*, il client deve inviare il comando “*interrompi*”. Il server risponderà inviando una stringa che avrà *posizione\_griglia* = ‘-1’ e *numero\_estratto* = ‘-1’. Alla ricezione di questi due caratteri, il client dovrà smettere di ricevere stringhe, e comunicare, tramite un messaggio in popup, la sconfitta del client. L’interruzione dell’estrazione determina la sconfitta automatica del client. Al termine dello scambio, le caselle dovranno rimanere colorate e il click su di esse dovrà essere nuovamente abilitato. Il client dovrà inoltre disabilitare il pulsante *Interrompi* e riabilitare *Start*, *Disconnect*, e *Clear*.
- Alla pressione del pulsante *Disconnect*, deve essere inviata la stringa “*disconnect*”, e devono essere chiusi i canali di comunicazione generati in fase di connessione. Deve inoltre essere abilitato nuovamente il pulsante *Connect* in quanto deve essere possibile instaurare una nuova connessione senza che sia necessario il riavvio del client.
- Alla pressione del pulsante *Clear*, il colore di ogni casella deve essere riportato a *Color.LIGHT\_GRAY* ed il contenuto ripristinato.

(**suggerimento:** fare riferimento al metodo `setTextDigit(String digit)` della classe `ColoredButton` per ripristinare il contenuto di una casella).