

Nome e cognome: Giuseppe Carlino;

Progetto: ChatLab.

Relazione per “ChatLabClient”

Per la strutturazione del client si è deciso di usufruire dell’interfaccia grafica offerta dal package swing, facendo puro ausilio del motore grafico di NetBeans (v. 20).



Dalla classe ClientFrame viene fatto partire un thread client della classe ChatClient, il quale risulta essere fondamentale per tutta la durata dell’esecuzione dell’applicativo, dato che trattasi dell’elemento cardine per lo sviluppo della comunicazione con il relativo software server-side.

```
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    Look and feel setting code (optional)  
    //</editor-fold>  
  
    // Dichiarazione ed implementazione variabili  
    String host = "192.168.153.1";  
    int porta = 12345;  
  
    // Dichiarazione ed implementazione oggetti  
    ClientFrame interfacciaGrafica = new ClientFrame();  
    ChatClient client = new ChatClient(interfacciaGrafica, host, porta);  
  
    // Avvio del client istanziato  
    client.start();  
}  
  
public class ChatClient extends Thread {  
  
    // Dichiarazione attributi  
    private final ClientFrame interfacciaGrafica;  
    private final String host;  
    private final int porta;  
  
    // Metodi costruttori  
    public ChatClient(ClientFrame interfacciaGrafica, String host, int porta) {  
        this.interfacciaGrafica = interfacciaGrafica;  
        this.host = host;  
        this.porta = porta;  
    }  
}
```

Nome e cognome: Giuseppe Carlino;

Progetto: ChatLab.

Mediante il client precedentemente istanziato, l'applicazione effettua una richiesta di autenticazione all'utente. I dati raccolti (in particolar modo, il suo nickname) vengono trasmessi successivamente al server, l'utilizzo dei quali non è scopo principale di questa documentazione (cfr. ChatLabServer – Relazione.pdf).

```
// Messaggio di conferma di connessione al server in ascolto su di una determinata porta
interfacciaGrafica.setAreaOutputClient("Connesso al server " + host + " sulla porta " + porta + ".\n");

// Messaggio di segnalazione riguardante la modalità da utilizzare per effettuare l'autenticazione
interfacciaGrafica.getCampoInput().setToolTipText("NICK <Nickname>");

// Autenticazione alla rete di comunicazione
do {

    // Se è stato premuto il pulsante di invio
    if (interfacciaGrafica.isInviato()) {

        // Chiamare la procedura setInviato associata al frame interfacciaGrafica
        interfacciaGrafica.setInviato();

        // Memorizzare il messaggio scritto in campoInput
        messaggioClient = interfacciaGrafica.getCampoInput().getText();

        // Invio del messaggio
        messaggioClientOut.println(messaggioClient);

    }

    // Attesa di listener
    join(17);
} while (!messaggioClient.contains("NICK ") || listener.getMessaggioServer().equals("App: Attenzione! Utente già esistente."));
```

Effettuata con successo l'autenticazione, l'utente è invitato a comunicare con gli altri utenti della rete riferendosi a quanto stabilito dalle modalità stampate a video (nell'area di visualizzazione dell'output) dall'applicativo client (quindi, quello che l'utente sta utilizzando), senza dimenticarsi di dare un'occhiata anche alla guida allegata (cfr. ChatLabClient – Guida.pdf).

```
// Disattivare la possibilità di uscire dall'applicativo senza aver prima comunicato "QUIT"
interfacciaGrafica.setDefaultCloseOperation(javax.swing.WindowConstants.DO_NOTHING_ON_CLOSE);

// Modalità di comunicazione
interfacciaGrafica.setAreaOutputClient("""
\n\nModalità:
- CHAT: comunicazione con un particolare utente (inserirne il relativo nickname);
- BROADCAST: comunicazione con tutti gli utenti;
- NICK: modifica del proprio nickname;
- LIST: visualizzazione di tutti gli utenti connessi;
- QUIT: disconnessione.\n
""");

// Messaggio di segnalazione per l'inserimento di un messaggio
interfacciaGrafica.getCampoInput().setToolTipText("<Modalità> [<Nickname>][<Corpo>]");
```

```
// Scambio di messaggi
do {

    // Se è stato premuto il pulsante di invio
    if (interfacciaGrafica.isInviato()) {

        // Chiamare la procedura setInviato associata al frame interfacciaGrafica
        interfacciaGrafica.setInviato();

        // Memorizzare il messaggio scritto in campoInput
        messaggioClient = interfacciaGrafica.getCampoInput().getText();

        // Invio del messaggio
        messaggioClientOut.println(messaggioClient);

        // Attesa di listener
        join(17);

        // Comunicazione listener
        if (messaggioClient.contains("QUIT ") || listener.getMessaggioServer().equals("App: Attenzione! Utente inesistente.") || listener.getMessaggioServer().equals("App: Attenzione! Non è consentito comunicare con se stessi!")) {

            // Messaggio di segnalazione riguardante la modalità da utilizzare per effettuare l'autenticazione
            interfacciaGrafica.getCampoInput().setToolTipText("NICK <Nickname>");

            do {

                // Se è stato premuto il pulsante di invio
                if (interfacciaGrafica.isInviato()) {

                    // Chiamare la procedura setInviato associata al frame interfacciaGrafica
                    interfacciaGrafica.setInviato();

                    // Memorizzare il messaggio scritto in campoInput
                    messaggioClient = interfacciaGrafica.getCampoInput().getText();

                    // Invio del messaggio
                    messaggioClientOut.println(messaggioClient);

                }

                // Attesa di listener
                join(17);

            } while (!messaggioClient.equals("QUIT"));

        }

        // Attesa di listener
        join(17);
    } while (!messaggioClient.equals("QUIT"));

}
```

Nome e cognome: Giuseppe Carlino;

Progetto: ChatLab.

Per terminare la comunicazione, quindi digitare "QUIT".

```
} while (!messaggioClient.equals("QUIT"));

// Disconnessione
// Chiusura del socket di comunicazione
socket.close();
// Chiusura dello stream verso il server
messaggioClientOut.close();
// Chiudere il frame interfacciaGrafica
interfacciaGrafica.dispose();
```

Durante la comunicazione, il server trasmette di tanto in tanto dei messaggi (magari provenienti da altri utenti della rete), che l'utente è in grado di ascoltare grazie al thread listener della classe ServerListener. Ogni qualvolta l'utente invia un messaggio, l'applicativo client-side si mette in pausa per 17 ms, rispettando la concorrenza con il thread listener (soffermare l'attenzione sui vari "join(17)" riportati nel codice in alto, quello specializzato nello scambio dei messaggi), nell'attesa che arrivi qualcosa.

```
listener = new ServerListener(interfacciaGrafica, socket);
listener.start();

public class ServerListener extends Thread {

    // Dichiarazione attributi
    private final ClientFrame interfacciaGrafica;
    private final Socket socket;
    private String messaggioServer;

    // Metodi costruttori
    public ServerListener(ClientFrame interfacciaGrafica, Socket socket) {
        this.interfacciaGrafica = interfacciaGrafica;
        this.socket = socket;
        messaggioServer = "";
    }

    // Avvio all'ascolto
    do {
        messaggioServer = messaggioServerIn.readLine();
        interfacciaGrafica.setAreaOutputClient('\n' + messaggioServer);
    } while (interfacciaGrafica.isActive());
```