//DISPLAY.JAVA

package citta;

import java.awt.BorderLayout;

import java.awt.event.ActionListener;

import javax.swing.\*;

public class Display extends JFrame {

//CAMPI

private JLabel serveraddresslabel = new JLabel("Server Address");

JTextField serveraddressfield = new JTextField(30);

private JLabel portlabel = new JLabel("Port");

JTextField portfield = new JTextField(10);

private JButton connect = new JButton("Connect");

private JButton disconnect = new JButton("Disconnect");

JTextArea usa = new JTextArea(5, 20);

JTextArea italia = new JTextArea(5, 20);

JTextArea log = new JTextArea(5, 20);

private JScrollPane scrollusa;

private JScrollPane scrollitalia;

private JScrollPane scrolllog;

private JButton start = new JButton("Start");

private JButton stop = new JButton("Stop");

public boolean connected = false;

public boolean started = false;

//COSTRUTTORE

public Display() {

super("Sofia Lapi 1837563");

JPanel north = new JPanel();

north.add(serveraddresslabel);

north.add(serveraddressfield);

north.add(portlabel);

north.add(portfield);

north.add(connect);

north.add(disconnect);

usa.setEditable(false);

italia.setEditable(false);

log.setEditable(false);

scrollusa = new JScrollPane(usa);

scrollitalia = new JScrollPane(italia);

scrolllog = new JScrollPane(log);

scrollusa.setBorder(BorderFactory.createTitledBorder("USA"));

scrollitalia.setBorder(BorderFactory.createTitledBorder("Italia"));

scrolllog.setBorder(BorderFactory.createTitledBorder("Log"));

JPanel center = new JPanel();

center.add(scrollusa);

center.add(scrollitalia);

center.add(scrolllog);

JPanel south = new JPanel();

south.add(start);

south.add(stop);

this.getContentPane().add(north, BorderLayout.NORTH);

this.getContentPane().add(center, BorderLayout.CENTER);

this.getContentPane().add(south, BorderLayout.SOUTH);

//ASCOLTATORI

ActionListener ascoltatore = new ClientListener(this);

connect.addActionListener(ascoltatore);

disconnect.addActionListener(ascoltatore);

start.addActionListener(ascoltatore);

stop.addActionListener(ascoltatore);

connect.setActionCommand(ClientListener.CONNECT);

disconnect.setActionCommand(ClientListener.DISCONNECT);

start.setActionCommand(ClientListener.START);

stop.setActionCommand(ClientListener.STOP);

//PARTE DI VISIBILITà

this.setButtons();

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

this.pack();

this.setVisible(true);

}

//METODO PER GESTIONE DI STATI

public void setButtons() {

if (!connected) {

connect.setEnabled(true);

disconnect.setEnabled(false);

start.setEnabled(false);

stop.setEnabled(false);

}else {

connect.setEnabled(false);

disconnect.setEnabled(true);

start.setEnabled(true);

stop.setEnabled(false);

if (started) {

start.setEnabled(false);

stop.setEnabled(true);

}else {

stop.setEnabled(false);

}

}

}

}

//CLIENTLISTENER.JAVA

package citta;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.net.Socket;

import java.net.UnknownHostException;

import java.util.Scanner;

import javax.swing.JOptionPane;

public class ClientListener implements ActionListener{

//CAMPI PER L'EVOLUZIONE DEL DISPLAY

public final static String CONNECT = "Connect";

public final static String DISCONNECT = "Disonnect";

public final static String START = "Start";

public final static String STOP = "Stop";

private Display displayAscoltato = null;

//CAMPI PER LA CONNESSIONE

private Socket socket = null;

private PrintWriter pw = null;

private Scanner sc = null;

//COSTRUTTORE

public ClientListener(Display display) {

this.displayAscoltato = display;

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getActionCommand().equals(CONNECT)) {

//PARTE DI CONNESSIONE

try {

this.setupConnection();

} catch (IOException ex) {

JOptionPane.showMessageDialog(null, "impossibile connettersi");

return;

}

//PARTE DI EVOLUZIONE PULSANTI

this.displayAscoltato.connected = true;

this.displayAscoltato.setButtons();

}else if(e.getActionCommand().equals(DISCONNECT)) {

try {

pw.println("DISCONNECT");

pw.flush();

this.closeConnection();

} catch (IOException ex) {

JOptionPane.showMessageDialog(null, "impossibile disconnettersi");

return;

}

this.displayAscoltato.connected = false;

this.displayAscoltato.setButtons();

}else if(e.getActionCommand().equals(START)) {

this.displayAscoltato.started = true;

this.displayAscoltato.setButtons();

//INVIO COMANDO AL SERVER

pw.println("START");

pw.flush();

//THREAD DI RICEZIONE DATI

ReceiverThread rt = new ReceiverThread(this.sc, this.displayAscoltato);

Thread thread = new Thread(rt);

thread.start();

}else if(e.getActionCommand().equals(STOP)) {

this.displayAscoltato.started = false;

this.displayAscoltato.setButtons();

//INVIO COMANDO AL SERVER

pw.println("STOP");

pw.flush();

}

}

private void setupConnection() throws IOException {

String stampaindirizzo = this.displayAscoltato.serveraddressfield.getText();

String stampaport = this.displayAscoltato.portfield.getText();

System.out.println(stampaindirizzo);

System.out.println(stampaport);

socket = new Socket(this.displayAscoltato.serveraddressfield.getText(), Integer.parseInt(this.displayAscoltato.portfield.getText()));

pw = new PrintWriter(new OutputStreamWriter(socket.getOutputStream()));

sc = new Scanner (socket.getInputStream());

}

private void closeConnection() throws IOException {

sc.close();

pw.close();

socket.close();

}

}

//RECEIVERTHREAD.JAVA

package citta;

import java.util.Scanner;

public class ReceiverThread implements Runnable{

//CAMPI

private Scanner scanner;

private Display display;

//COSTRUTTORE

public ReceiverThread(Scanner scanner, Display display) {

this.scanner = scanner;

this.display = display;

}

@Override

public void run() {

boolean running = true;

while (running) {

String cmd = scanner.nextLine();

String commands[] = cmd.split(":");

if (commands[0].equals("START")) {

continue;

}else if (commands[0].equals("END")||commands[1].equals("END")) {

//FINE TRASMISSIONE

running = false;

continue;

}else if(commands[0].equals("INTERRUPTED")||commands[1].equals("INTERRUPTED")) {

//TRASMISSIONE INTERROTTA

running = false;

continue;

}else {

//CASO RICEZIONE CITTA

String attualeusa = display.usa.getText();

String attualeitalia = display.italia.getText();

String attualelog = display.log.getText();

//mettere nel log

String cittaRicevuta = commands[1];

display.log.setText(attualelog + "\n" + cittaRicevuta);

if(commands[0].equals("USA")) {

//mettere in usa

display.usa.setText(attualeusa + "\n" +cittaRicevuta);

}else if(commands[0].equals("ITALIA")) {

//mettere in ita

display.italia.setText(attualeitalia + "\n" + cittaRicevuta);

}

}

}

}

}

//PRINCIPALE.JAVA

package citta;

public class Principale {

public static void main(String[] args) {

Display display = new Display();

}

}