Esame 10/06/2019

```
package esami.giugnoDieci;

import javax.swing.*;

public class HexMain {

    public static void main (String[] args){

        Runnable init = new Runnable() {
            @Override
            public void run() {
                new HexFrame();
            }
        };

        SwingUtilities.invokeLater(init);
    }
}

package esami.giugnoDieci;

import javax.swing.*;
import java.awt.*;

public class HexFrame extends JFrame {

    protected JLabel ipLabel = new JLabel("IP Address");
    protected JLabel portLabel = new JLabel("Port");
    protected JTextField ipBox = new JTextField(10);
    protected JTextField portBox = new JTextField(10);
    protected JButton connect = new JButton("Connetti");
    protected JButton disconnect = new JButton("Disconnetti");
    protected JButton start = new JButton("Start");
    protected JButton stop = new JButton("Stop");
    protected JButton converti = new JButton("Converti");
    protected JButton numeri = new JButton("0 - 9");
    protected JButton lettere = new JButton("A - F");
    protected JTextField hexBox = new JTextField(40);
    protected JTextField decBox = new JTextField(40);
    protected JTextField binBox = new JTextField(40);
    HexListener listener;

    public HexFrame(){
        super("Dario Pietrosanto");
        HexFrame frame = this;
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setResizable(false);
        frame.setLocationRelativeTo(null);
        frame.setLayout(new BorderLayout(10,10));
        listener = new HexListener(frame);

        frame.add(top(), BorderLayout.NORTH);
        frame.add(middle(), BorderLayout.CENTER);
        frame.add(bottom(), BorderLayout.SOUTH);

        frame.pack();
        frame.setVisible(true);
    }

    private JPanel top(){
        JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER,10,10));
        ipBox.setText("127.0.0.1");
        portBox.setText("4400");
        connect.addActionListener(listener);
        connect.setActionCommand(HexListener.CONNECT);
        disconnect.addActionListener(listener);
        disconnect.setActionCommand(HexListener.DISCONNECT);
        disconnect.setEnabled(false);

        panel.add(ipLabel);
        panel.add(ipBox);
        panel.add(portLabel);
        panel.add(portBox);
        panel.add(connect);
        panel.add(disconnect);

        return panel;
    }

    private JPanel middle(){
        JPanel panel = new JPanel(new BorderLayout(10,10));
        JPanel subpanel1 = new JPanel(new BorderLayout(5,5));
        JPanel subpanel2 = new JPanel(new GridLayout(3,1));
        JPanel subpanel3 = new JPanel(new GridLayout(3,1));

        lettere.addActionListener(listener);
        lettere.setActionCommand(HexListener.LETTERE);
        lettere.setEnabled(false);
        numeri.addActionListener(listener);
        numeri.setActionCommand(HexListener.NUMERI);
        numeri.setEnabled(false);
        hexBox.setEditable(false);
        decBox.setEditable(false);
        binBox.setEditable(false);

        subpanel1.add(lettere,BorderLayout.WEST);
        subpanel1.add(numeri,BorderLayout.EAST);
        subpanel2.add(new JLabel("Hexadecimal"));
        subpanel2.add(new JLabel("Decimal"));
        subpanel2.add(new JLabel("Binary"));
        subpanel3.add(hexBox);
        subpanel3.add(decBox);
        subpanel3.add(binBox);
        panel.add(subpanel1, BorderLayout.WEST);
        panel.add(subpanel2, BorderLayout.CENTER);
        panel.add(subpanel3, BorderLayout.EAST);

        return panel;
    }

    private JPanel bottom(){
        JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER,10,10));
        start.addActionListener(listener);
        start.setActionCommand(HexListener.START);
        start.setEnabled(false);
        stop.addActionListener(listener);
        stop.setActionCommand(HexListener.STOP);
        stop.setEnabled(false);
        converti.addActionListener(listener);
        converti.setActionCommand(HexListener.CONVERTI);
        converti.setEnabled(false);

        panel.add(start);
        panel.add(stop);
        panel.add(converti);

        return panel;
    }

}
package esami.giugnoDieci;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

public class HexListener implements ActionListener {
    public static final String START = "start";
    public static final String CONNECT = "connect";
    public static final String DISCONNECT = "disconnect";
    public static final String STOP = "stop";
    public static final String CONVERTI = "converti"; //todo
    public static final String LETTERE = "lettere";
    public static final String NUMERI = "cifre";

    private HexFrame frame;
    private Socket socket;
    private PrintWriter printer;
    private Scanner scanner;
    private HexWorker worker;

    public HexListener(HexFrame frame) {
        this.frame=frame;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();

        if (cmd.equals(CONNECT)){
            String ip;
            Integer port;
            ip=frame.ipBox.getText();
            try {
                port=Integer.parseInt(frame.portBox.getText());
                if (port<0)
                    throw new NumberFormatException();
            } catch (NumberFormatException e1){
                JOptionPane.showMessageDialog(frame,"Inserisci un numero di porta valido.","Errore",
                        JOptionPane.WARNING_MESSAGE);
                return;
            }
            try {
                socket= new Socket(ip,port);
            } catch (IOException e1){
                JOptionPane.showMessageDialog(frame,"Impossibile connettersi al server
"+ip+":"+port+
                        "\nRiprova","Errore",JOptionPane.WARNING_MESSAGE);
                return;
            }
            try {
                printer = new PrintWriter(socket.getOutputStream());
                scanner = new Scanner(socket.getInputStream());
            } catch (IOException e1){
                JOptionPane.showMessageDialog(frame,"Errore nel connettersi al server
"+ip+":"+port+"\nRiprova",
                        "Errore",JOptionPane.WARNING_MESSAGE);
                return;
            }
            frame.connect.setEnabled(false);
            frame.disconnect.setEnabled(true);
            frame.start.setEnabled(true);
            JOptionPane.showMessageDialog(frame,"Connessione riuscita.","Connessione riuscita",
                    JOptionPane.INFORMATION_MESSAGE);
            return;
        } else if (cmd.equals(DISCONNECT)){
            printer.println(DISCONNECT);
            printer.flush();
            try {
                printer.close();
                scanner.close();
                socket.close();
            } catch (IOException e1) {
                JOptionPane.showMessageDialog(frame,"Errore in chiusura della connessione.",
                        "Errore",JOptionPane.WARNING_MESSAGE);
            }
            frame.hexBox.setText("");
            frame.decBox.setText("");
            frame.binBox.setText("");
            frame.connect.setEnabled(true);
            frame.disconnect.setEnabled(false);
            frame.start.setEnabled(false);
            frame.converti.setEnabled(false);
            JOptionPane.showMessageDialog(frame,"Connessione chiusa.","Connessione chiusa",
                    JOptionPane.INFORMATION_MESSAGE);
            return;
        } else if (cmd.equals(START)){
            frame.hexBox.setText("");
            frame.decBox.setText("");
            frame.binBox.setText("");
            frame.start.setEnabled(false);
            frame.stop.setEnabled(true);
            frame.lettere.setEnabled(true);
            frame.numeri.setEnabled(true);
            frame.converti.setEnabled(false);

            printer.println(START);
            printer.flush();

            worker = new HexWorker(frame, printer, scanner);
            worker.execute();
            return;

        } else if (cmd.equals(STOP)){
            if (!worker.equals(null))
                worker.cancel(true);

//              frame.start.setEnabled(true);
//              frame.stop.setEnabled(false);
//              frame.lettere.setEnabled(false);
//              frame.numeri.setEnabled(false);
//              frame.disconnect.setEnabled(true);

        } else if (cmd.equals(LETTERE) || cmd.equals(NUMERI)){
            printer.println(cmd);
            printer.flush();
            return;
        } else if (cmd.equals(CONVERTI)){
            Long dec = Long.parseUnsignedLong(frame.hexBox.getText(), 16);
            frame.decBox.setText(String.valueOf(dec));
            String bin = Long.toBinaryString(dec);
            frame.binBox.setText(bin);
        }
```

```java
    }
}

package esami.giugnoDieci;

import com.sun.org.apache.xpath.internal.operations.Bool;

import javax.swing.*;
import java.io.PrintWriter;
import java.util.Scanner;

public class HexWorker extends SwingWorker<Boolean, Object> {

    private HexFrame frame;
    private PrintWriter printer;
    private Scanner scanner;

    public HexWorker(HexFrame frame, PrintWriter printer, Scanner scanner) {
        this.frame = frame;
        this.printer = printer;
        this.scanner = scanner;
    }

    @Override
    protected Boolean doInBackground() throws Exception {
        // preferred A-F
        if (!isCancelled()) {
            String res = "";

            while (true) {
                String line=scanner.nextLine();
                if (line.equals("+"))
                    break;
                res+=line;
                frame.hexBox.setText(res);
            }
            frame.hexBox.setText(res);
        }
        return true;
    }

    @Override
    protected void done(){
        printer.println(HexListener.STOP);
        printer.flush();

        frame.start.setEnabled(true);
        frame.stop.setEnabled(false);
        frame.lettere.setEnabled(false);
        frame.numeri.setEnabled(false);
        frame.disconnect.setEnabled(true);
        frame.converti.setEnabled(true);
        JOptionPane.showMessageDialog(frame,"Trasmissione conclusa.", "Trasmissione conclusa",
                JOptionPane.INFORMATION_MESSAGE);
    }
}
```