Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: **4 E**	Data: **30/05/2019**	Penalita':	Punti: **20.0 / 20**	Voto: **10.0**

Domanda 1 - (punti: 16 di 20) Punti assegnati: 16

Si implementi un software java client-server in grado di consentire l'invio del file csv a questo link ad un server ed in particolare:
- Il client consente la selezione del file tramite apposita voce di menu File > Carica.
- Il client invia tramite socket il contenuto del file al server
- Il server riceve il file csv e lo visualizza in un JTable.

La valutazione dell'esercizio prevede:
- creazione di entrambe le interfacce grafiche client e server: 6 punti
- lettura del file csv: 3 punti
- scrittura/lettura del file csv tramite socket: 7

Risposta

si veda il file allegato

Allegato:
clientServer/server/backend/beam/CSVServer.java

```
package clientServer.server.backend.beam;

import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Observer;

@SuppressWarnings("deprecation")
public class CSVServer implements Runnable {

    private ServerSocket servSocket;
    private Observer forwarded;



    public CSVServer(int port) throws IOException {
        try {
            servSocket = new ServerSocket(port);
        } catch (IOException e) {
            throw e;
        }
    }

    public CSVServer(int port, Observer forwarded) throws IOException {
        this(port);
        this.forwarded = forwarded;
    }

    @Override
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
    public void run() {
        if(servSocket != null) {
            int id = 0;
            while(true) {
                try {
                    Socket _s = servSocket.accept();
                    ClientHandler c = new ClientHandler(id, _s);
                    if(forwarded != null) {
                        c.addObserver(forwarded);
                    }
                    new Thread(c).start();
                } catch (IOException e) {

                }

                id++;
            }

        }

    }

}
```

---

clientServer/server/backend/beam/EventoAssicurazione.java

```
package clientServer.server.backend.beam;

public class EventoAssicurazione {

    private static final String CSV_SEPARATOR = ";";

    private String policyID;
    private String stateCode;
    private String county;
    private String eq_site_limit;

    public EventoAssicurazione(String csvString) {
        String[] data = csvString.split(CSV_SEPARATOR);
        this.policyID = data[0];
        this.stateCode = data[1];
        this.county = data[2];
        this.eq_site_limit = data[3];
    }

    public String getPolicyID() {
        return policyID;
    }
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
    public void setPolicyID(String policyID) {
        this.policyID = policyID;
    }

    public String getStateCode() {
        return stateCode;
    }

    public void setStateCode(String stateCode) {
        this.stateCode = stateCode;
    }

    public String getCounty() {
        return county;
    }

    public void setCounty(String county) {
        this.county = county;
    }

    public String getEq_site_limit() {
        return eq_site_limit;
    }

    public void setEq_site_limit(String eq_site_limit) {
        this.eq_site_limit = eq_site_limit;
    }



}
```

---

clientServer/client/backend/beam/CSVCommunicator.java

```
package clientServer.client.backend.beam;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;

null
public class CSVCommunicator extends Thread {
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
    private Socket conn;

    private File csvFile;

    public CSVCommunicator(String hostName, int port, File csvFile) throws
IOException {
        try {
            InetAddress addr = InetAddress.getByName(hostName);
            conn = new Socket(addr, port);

        } catch (IOException e) {
            throw e;
        }
        this.csvFile = csvFile;
    }


    public void run() {
        if(conn != null) {
            sendFileLineByLine();
        }
    }

    public void sendFileLineByLine() {
        BufferedReader reader = null;
        PrintWriter writer = null;
        try {
            reader = new BufferedReader(new FileReader(csvFile));
            writer = new PrintWriter(new
OutputStreamWriter(conn.getOutputStream()));

            //legge e invia le righe del file una per volta, all'invio di stringa vuoto
considero la file del file
            String line = reader.readLine();
            while(line != null) {
                writer.println(line);
                line = reader.readLine();
            }


            writer.flush();




        } catch (IOException e) {
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        finally {
            if(reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }


            try {
                conn.close();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }
    }
}
```

---

clientServer/server/frontend/gui/CSVServerGUI.java

```
package clientServer.server.frontend.gui;

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Observable;
import java.util.Observer;

import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

import clientServer.server.backend.beam.CSVServer;
import clientServer.server.backend.beam.ClientHandler;

@SuppressWarnings("deprecation")
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
public class CSVServerGUI extends JFrame implements ActionListener,
Observer {


    private static final long serialVersionUID = -1432818586050381729L;

    private static final String CSV_SEPARATOR = ";";

    private static final String[] TABLE_HEADER = {"policyID", "statecode",
"county", "eq_site_limit"};
    private DefaultTableModel tblMod;

    private JMenuItem itemStartServer;

    private CSVServer server;

    public CSVServerGUI() {
        super("CSV - server");
        setExtendedState(MAXIMIZED_BOTH);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        initComponents();
    }

    private void initComponents() {
        setJMenuBar(initJMenuBar());

        setLayout(new BorderLayout());

        tblMod = new DefaultTableModel(TABLE_HEADER, 0);
        JTable tbl = new JTable(tblMod);
        JScrollPane scrl = new JScrollPane(tbl);

        this.add(scrl, BorderLayout.CENTER);


    }

    private JMenuBar initJMenuBar() {
        JMenuBar bar = new JMenuBar();

        bar.add(initServerMenu());

        return bar;
    }

    private JMenu initServerMenu() {
        JMenu mnuServer = new JMenu("Server");
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
        itemStartServer = new JMenuItem("Avvia");
        itemStartServer.addActionListener(this);

        mnuServer.add(itemStartServer);

        return mnuServer;
    }

    @Override
    public synchronized void update(Observable o, Object arg) {
        if(o instanceof ClientHandler) {
            String line = (String) arg;
            String[] cont = line.split(CSV_SEPARATOR);
            if(!cont.equals(TABLE_HEADER)) {
                tblMod.addRow(cont);
            }

        }

    }

    public void performStartServer() {
        try {
            int port = Integer.valueOf(JOptionPane.showInputDialog(this, "Port",
"Connessione", JOptionPane.PLAIN_MESSAGE));
            server = new CSVServer(port, this);
            new Thread(server).start();
        }
        catch(Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
        }

    }


    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource().equals(itemStartServer)) {
            performStartServer();
        }

    }


    public static void main(String[] args) {
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
        new CSVServerGUI().setVisible(true);
    }

}
```

---

clientServer/server/backend/beam/ListaEventiAssicurazione.java

```
package clientServer.server.backend.beam;

import java.util.Observable;
import java.util.Observer;
import java.util.Vector;

@SuppressWarnings("deprecation")
public class ListaEventiAssicurazione extends Vector<EventoAssicurazione>
implements Observer {


    private static final long serialVersionUID = -8128410870827826654L;

    private static ListaEventiAssicurazione me;

    public static ListaEventiAssicurazione getInstance() {
        if(me == null) {
            me = new ListaEventiAssicurazione();
        }
        return me;
    }

    private ListaEventiAssicurazione() {

    }

    public boolean add(EventoAssicurazione e) {
        boolean ris = false;
        if(e != null) {
            ris = super.add(e);
        }
        return ris;
    }

    @Override
    public void update(Observable o, Object arg) {
        if(o instanceof ClientHandler) {
            String line = (String) arg;
            EventoAssicurazione _e = new EventoAssicurazione(line);
            this.add(_e);
        }
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
    }



}
```

---

clientServer/server/backend/beam/ClientHandler.java

```
package clientServer.server.backend.beam;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Socket;
import java.util.Observable;

@SuppressWarnings("deprecation")
public class ClientHandler extends Observable implements Runnable {

    private int id;
    private Socket conn;



    public ClientHandler(int id, Socket conn) {
        super();
        this.id = id;
        this.conn = conn;
        this.addObserver(ListaEventiAssicurazione.getInstance());
    }



    @Override
    public void run() {
        BufferedReader reader = null;
        try {
            reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));

            String line = reader.readLine();
            while(line != null) {
                setChanged();
                notifyObservers(line);
                line = reader.readLine();
            }
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        finally {
            try {
                conn.close();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }

    }



    public int getId() {
        return id;
    }



}
```

---

clientServer/client/frontend/gui/CSVClientGUI.java

```
package clientServer.client.frontend.gui;

import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;

import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.filechooser.FileNameExtensionFilter;

import clientServer.client.backend.beam.CSVCommunicator;
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
public class CSVClientGUI extends JFrame implements ActionListener {

    private static final long serialVersionUID = -5835991100801909484L;

    private JMenuItem itemCarica;
    private JLabel lblStatus;

    public CSVClientGUI() {
        super("CSV - client");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(new Dimension(600, 400));
        initComponents();

    }

    private void initComponents() {
        setJMenuBar(initJMenuBar());

        setLayout(new GridBagLayout());

        lblStatus = new JLabel("Status...");
        lblStatus.setFont(new Font("Calibri", Font.BOLD, 30));

        this.add(lblStatus, new GridBagConstraints());

    }

    private JMenuBar initJMenuBar() {
        JMenuBar bar = new JMenuBar();

        bar.add(initFileMenu());

        return bar;
    }

    private JMenu initFileMenu() {
        JMenu mnuFile = new JMenu("File");

        itemCarica = new JMenuItem("Carica");
        itemCarica.addActionListener(this);

        mnuFile.add(itemCarica);

        return mnuFile;
    }
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
    private JFileChooser initJFileChooser() {
        JFileChooser fc = new JFileChooser();
        fc.setFileFilter(new FileNameExtensionFilter("Comma separated text",
"csv"));
        return fc;
    }

    public void performLoadFile() {
        try {
            String host = JOptionPane.showInputDialog(this, "Server IP: ",
"Connessione", JOptionPane.PLAIN_MESSAGE);
            int port = Integer.valueOf(JOptionPane.showInputDialog(this, "Port",
"Connessione", JOptionPane.PLAIN_MESSAGE));
            JFileChooser fc = initJFileChooser();
            if(fc.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
                File f = fc.getSelectedFile();

                CSVCommunicator c = new CSVCommunicator(host, port, f);
                lblStatus.setFont(new Font("Calibri", Font.BOLD, 20));
                lblStatus.setText("Invio del file " + f.getAbsolutePath());
                c.start();

            }
        }
        catch(Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
        }


    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource().equals(itemCarica)) {
            performLoadFile();
        }

    }

    public static void main(String[] args) {
        new CSVClientGUI().setVisible(true);

    }

}
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: **4 E**	Data: **30/05/2019**	Penalita':	Punti: **20.0 / 20**	Voto: **10.0**

Note correzione	
Soluzione proposta	

Domanda 2 - (punti: 3 di 20) Punti assegnati: 3

Implementando l'interfaccia Runnable si crei un Thread in grado di:
- accettare come parametro nel costruttore il percorso di una directory
- nel metodo "run()" contare il numero di file e cartelle presenti nel percorso fornito al punto precedente e visualizzarlo a video

Risposta

Si veda il file allegato

Allegato:
DirDigger.java

```
package threadCartelle;

import java.io.File;

public class DirDigger implements Runnable {

    private String sourcePath;



    public DirDigger(String sourcePath) {
        super();
        this.sourcePath = sourcePath;
    }

    @Override
    public void run() {
        File dir = new File(sourcePath);


        int nFiles = dir.list().length;

        System.out.println("Nr cartelle e file della directory " + sourcePath + ": " +
nFiles);

    }

    public static void main(String[] args) {
        DirDigger d = new DirDigger("C:\\Users\\franc\\OneDrive\\Desktop");
        new Thread(d).start();
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 30/05/2019	Penalita':	Punti: 20.0 / 20	Voto: 10.0

```
    }
}
```

Note correzione

Soluzione proposta

```
public class ContaFile implements Runnable {

   private String percorso;



  public ContaFile(String dir) {

       this.percorso = dir;

  }



  public void run() {

       File f = new File(percorso);

       if(f.exists()) {

            String[] listF = f.list();

            if(listF != null) {

                 System.out.println("numero di file e cartelle: "+listF.length);

            }

       }




 }
```

Verifica di Informatica Nr. 5 - Anno 2018-2019 - Classe 4E

Ghinamo Francesco	Classe: **4 E**	Data: **30/05/2019**	Penalita':	Punti: **20.0 / 20**	Voto: **10.0**

Domanda 3 - (punti: 1 di 20)	Punti assegnati: 1
Si fornisca la definizione di file.	
Risposta	Un file e' un archivio di dati (e relative informazioni), che si puo' intendere come sequenza di byte, resa persistente su una memoria di massa del computer (ad esempio hard disk o solid state drive); essendo persistenti i dati salvati in un file rimangono inalterati anche dopo uno spegnimento e accensione della macchina. Un file e' generalmente caratterizzato da un nome e da un estensione che indica al SO il programma con cui interpretare il file. I file sono gestiti, organizzati e mantenuti da un modulo del SO detto File System.
Note correzione	
Soluzione proposta	Unità di informazione memorizzata in modo persistente su un supporto di memorizzazione di massa. E' il filesystem che si occupa della gestione ed organizzazione dei file.