

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

Domanda 1 - (punti: 10 di 30)

Punti assegnati:  
10

Si crei un programma java in grado di gestire le prenotazioni di uno studio dentistico. Per ogni prenotazione occorre memorizzare cognome, nome, tipo di intervento, data e ora. Il programma deve mostrare nel frame principale l'elenco delle prenotazioni future e consentire l'inserimento di una nuova prenotazione utilizzando una finestra di dialogo modale accessibile tramite la voce di menu Prenotazione > Aggiungi.

Risposta

Si veda il file allegato

Allegato:

[studioDentistico/backend/dati/TipolIntervento.java](#)**package** studioDentistico.backend.dati;**public** enum TipolIntervento {DETARTRASI,  
OTTURAZIONE,  
ASPORTAZIONE;

}

[studioDentistico/frontend/dialogs/InterventoDialog.java](#)**package** studioDentistico.frontend.dialogs;**import** java.awt.FlowLayout;**import** java.awt.GridBagConstraints;**import** java.awt.GridBagLayout;**import** java.awt.GridLayout;**import** java.awt.Insets;**import** java.awt.event.ActionEvent;**import** java.awt.event.ActionListener;**import** java.text.ParseException;**import** java.text.SimpleDateFormat;**import** java.util.GregorianCalendar;**import** java.util.NoSuchElementException;**import** java.util.StringTokenizer;**import** javax.swing.JButton;**import** javax.swing.JComboBox;**import** javax.swing.JDialog;**import** javax.swing.JFrame;**import** javax.swing.JLabel;**import** javax.swing.JPanel;**import** javax.swing.JTextField;**import** studioDentistico.backend.dati.Intervento;**import** studioDentistico.backend.dati.TipolIntervento;



## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: 4 E	Data: 06/12/2018	Penalita':	Punti: 30.0 / 30	Voto: 10.0
-------------------	----------------	---------------------	------------	---------------------	---------------

```
public class InterventoDialog extends JDialog implements ActionListener {

    private static final long serialVersionUID = -9093856105182188086L;

    private Intervento intR;

    private JTextField txtCognome;
    private JTextField txtNome;
    private JComboBox<TipoIntervento> cmbTipo;
    private JTextField txtData;
    private JTextField txtOra;

    private JButton btnConferma;

    public InterventoDialog(JFrame owner, boolean modale) {
        super(owner, "Intervento", modale);
        initComponents();
        pack();
    }

    private void initComponents() {
        JPanel inPan = createInputPanel();
        JPanel butPan = createButtonPanel();

        this.setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(5, 5, 5, 5);
        gbc.gridx = gbc.gridy = 0;
        gbc.weightx = gbc.weighty = 10;
        this.add(inPan, gbc);

        gbc.gridy = 1;
        gbc.weighty = 2;
        gbc.fill = GridBagConstraints.BOTH;
        this.add(butPan, gbc);
    }

    private JPanel createInputPanel() {
        txtCognome = new JTextField(20);
        txtNome = new JTextField(20);
        cmbTipo = new JComboBox<TipoIntervento>(TipoIntervento.values());
        txtData = new JTextField(20);
        txtOra = new JTextField(20);
    }
}
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
JPanel inPan = new JPanel(new GridLayout(5, 2, 5, 5));
inPan.add(new JLabel("Cognome"));
inPan.add(txtCognome);
inPan.add(new JLabel("Nome"));
inPan.add(txtNome);
inPan.add(new JLabel("Tipo"));
inPan.add(cmbTipo);
inPan.add(new JLabel("Data"));
inPan.add(txtData);
inPan.add(new JLabel("Ora"));
inPan.add(txtOra);

return inPan;
}

private JPanel createButtonPanel() {
    btnConferma = new JButton("Conferma");
    btnConferma.addActionListener(this);

    JPanel pan = new JPanel(new FlowLayout(FlowLayout.RIGHT));
    pan.add(btnConferma);
    return pan;
}

public Intervento creaIntervento() {
    Intervento i = null;
    if(!txtCognome.getText().equals("") &&!txtNome.getText().equals("")
    &&!txtData.getText().equals("") &&!txtOra.getText().equals("")) {
        GregorianCalendar c = new GregorianCalendar();
        SimpleDateFormat fmt = new SimpleDateFormat("dd/MM/yyyy");
        try {
            c.setTime(fmt.parse(txtData.getText()));
        } catch (ParseException e) {}

        try {
            StringTokenizer sT = new StringTokenizer(txtOra.getText(), ":");
            int h = Integer.parseInt(sT.nextToken());
            if(h > 23) {
                h = 23;
            }
        }
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        int m = Integer.parseInt(sT.nextToken());
        if(m > 59) {
            m = 59;
        }
        c.set(GregorianCalendar.HOUR_OF_DAY, h);
        c.set(GregorianCalendar.MINUTE, m);

    }
    catch(NoSuchElementException | NumberFormatException e) {}

    i = new Intervento(txtCognome.getText(), txtNome.getText(),
(TipoIntervento) cmbTipo.getSelectedItem(), c);
    }

    return i;
}

@Override
public void actionPerformed(ActionEvent aE) {
    if(aE.getSource().equals(btnConferma)) {
        intR = creaIntervento();
        dispose();
    }
}

public Intervento getIntR() {
    return intR;
}

}
```

studioDentistico/frontend/mainFrame/GestoreInterventiGUI.java

**package** studioDentistico.frontend.mainFrame;

```
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

```
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JScrollPane;
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

import studioDentistico.backend.dati.Intervento;
import studioDentistico.backend.dati.ListaInterventi;
import studioDentistico.frontend.dialogs.InterventoDialog;

public class GestoreInterventiGUI extends JFrame implements ActionListener {

    private static final long serialVersionUID = 4818558612874915938L;

    private static final String TITLE = "Gestore Interventi";
    private static final String[] TABLE_HEADER = {"Cognome", "Nome", "Tipo",
"Data e ora"};

    private ListaInterventi listaInt;

    private JMenuItem itemAggiungi;

    private DefaultTableModel tblModel;
    private JTable tblOut;

    public GestoreInterventiGUI() {
        super(TITLE);
        listaInt = new ListaInterventi();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setExtendedState(MAXIMIZED_BOTH);
        initComponents();
    }

    private void initComponents() {
        this.setJMenuBar(createMenuBar());

        tblModel = new DefaultTableModel(TABLE_HEADER, 0);
        tblOut = new JTable(tblModel);
        JScrollPane scrITbl = new JScrollPane(tblOut);

        this.setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.gridx = gbc.gridy = 0;
        gbc.weightx = gbc.weighty = 10;
        gbc.fill = GridBagConstraints.BOTH;
        this.add(scrITbl, gbc);

    }

    private JMenuItem createPrenotazioneMenu() {
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        itemAggiungi = new JMenuItem("Agggiungi");
        itemAggiungi.addActionListener(this);
        JMenu mnuPren = new JMenu("Prenotazione");
        mnuPren.add(itemAggiungi);
        return mnuPren;
    }

    private JMenuBar createMenuBar() {
        JMenuBar bar = new JMenuBar();
        bar.add(createPrenotazioneMenu());
        return bar;
    }

    public void aggiornaTable() {
        tblModel.setRowCount(0);
        for(Intervento i: listaInt) {
            tblModel.addRow(i.toStringArray());
        }
    }

    public void performAggiungi() {
        InterventoDialog iD = new InterventoDialog(this, true);
        iD.setVisible(true);
        try {
            listaInt.add(iD.getIntR());
        }
        catch(IllegalArgumentException e) {}
    }

    @Override
    public void actionPerformed(ActionEvent aE) {
        if(aE.getSource().equals(itemAggiungi)) {
            performAggiungi();
            aggiornaTable();
        }
    }

    public static void main(String[] args) {
        GestoreInterventiGUI gl = new GestoreInterventiGUI();
        gl.setVisible(true);
    }
}
```

## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
}  
}
```

studioDentistico/backend/dati/ListaInterventi.java

**package** studioDentistico.backend.dati;

**import** java.util.ArrayList;

**public class** ListaInterventi **extends** ArrayList<Intervento> {

**private static** final **long** serialVersionUID = 5033554638095116452L;

**public** boolean add(Intervento i) throws IllegalArgumentException {

**if**(i != null) {  
            return super.add(i);

        }  
        **else** {  
            throw **new** IllegalArgumentException();  
        }

    }

}

studioDentistico/backend/dati/Intervento.java

**package** studioDentistico.backend.dati;

**import** java.text.SimpleDateFormat;

**import** java.util.GregorianCalendar;

**public class** Intervento {

**private** String cognome;

**private** String nome;

**private** TipoIntervento tipo;

**private** GregorianCalendar dataOra;

**public** Intervento(String cognome, String nome, TipoIntervento tipo,  
GregorianCalendar dataOra) {

        super();  
        this.cognome = cognome;  
        this.nome = nome;  
        this.tipo = tipo;  
        this.dataOra = dataOra;

    }



## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
public String[] toStringArray() {  
    String[] _ris = new String[4];  
    SimpleDateFormat fmt = new SimpleDateFormat("dd/MM/yyyy HH:mm");  
    _ris[0] = cognome;  
    _ris[1] = nome;  
    _ris[2] = tipo.toString();  
    _ris[3] = fmt.format(dataOra.getTime());  
    return _ris;  
}  
  
public String getCognome() {  
    return cognome;  
}  
  
public String getNome() {  
    return nome;  
}  
  
public TipoIntervento getTipo() {  
    return tipo;  
}  
  
public GregorianCalendar getDataOra() {  
    return dataOra;  
}  
  
}
```

Note  
correzione

OK

Soluzione  
proposta

```
package liceocuneo.studiodentistico;  
  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
  
public class Prenotazione {  
  
    private String nome;  
    private String cognome;  
    private String tipoIntervento;  
    private Calendar dataOra;
```



**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
public Prenotazione(String nome, String cognome) {
    this.nome = nome;
    this.cognome = cognome;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getCognome() {
    return cognome;
}

public void setCognome(String cognome) {
    this.cognome = cognome;
}

public String getTipoIntervento() {
    return tipoIntervento;
}

public void setTipoIntervento(String tipoIntervento) {
    this.tipoIntervento = tipoIntervento;
}

public Calendar getDataOra() {
    return dataOra;
}

public void setDataOra(Calendar dataOra) {
    this.dataOra = dataOra;
}

@Override
public String toString() {
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm");
    String dataOraStr = sdf.format(dataOra.getTime());
    return nome + " " + cognome + " - "
        + tipoIntervento + " - (" + dataOraStr + ")";
}
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
}

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Calendar;
import java.util.GregorianCalendar;

import javax.swing.DefaultListModel;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JMenuItem;
import javax.swing.JScrollPane;

public class PrenotazioniFrame extends JFrame implements ActionListener {

    private static final String TITLE = "Studio Dentistico IL-DENTE";

    private JMenuItem itemAggP;
    private JList listaS;
    private DefaultListModel listaModel;

    public PrenotazioniFrame(){
        super(TITLE);
        setExtendedState(MAXIMIZED_BOTH);
        initComponents();
    }

    private void initComponents() {
        JMenuBar bar= new JMenuBar();
        JMenu mS = new JMenu("Prenotazioni");
        itemAggP = new JMenuItem("Aggiungi");
        itemAggP.addActionListener(this);

        mS.add(itemAggP);
        bar.add(mS);

        setJMenuBar(bar);

        listaModel = new DefaultListModel();
        listaS = new JList(listaModel);
        JScrollPane sp= new JScrollPane(listaS);
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        add(sp);
    }

    public void aggiungiPrenotazione(Prenotazione p){
        if( p != null) {
            listaModel.addElement(p);
        }
        //filtro le prenotazioni in modo da visualizzare solo le future
        int index = 0;
        Calendar now = new GregorianCalendar();
        while(index < listaModel.size()) {
            Prenotazione pT = (Prenotazione)listaModel.getElementAt(index);
            if((pT.getDataOra() != null) && (pT.getDataOra().compareTo(now) < 0)) {
                listaModel.remove(index);
            }
            else {
                index++;
            }
        }
    }

    public void actionPerformed(ActionEvent e) {
        if(e.getSource().equals(itemAggP)) {
            PrenotazioniAddDialog pd = new PrenotazioniAddDialog(this);
            pd.setVisible(true);
        }
    }

    public static void main(String[] args) {
        PrenotazioniFrame pf = new PrenotazioniFrame();
        pf.setVisible(true);
    }
}

import java.awt.FlowLayout;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Calendar;
import java.util.Date;
import java.util.Properties;

import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSpinner;
import javax.swing.JTextField;
import javax.swing.SpinnerDateModel;

import org.jdatepicker.impl.DateComponentFormatter;
import org.jdatepicker.impl.JDatePanelImpl;
import org.jdatepicker.impl.JDatePickerImpl;
import org.jdatepicker.impl.UtilDateModel;

public class PrenotazioniAddDialog extends JDialog implements ActionListener{

    private JTextField nome;
    private JTextField cognome;
    private JTextField tipoIntervento;
    private UtilDateModel data;
    private JSpinner oraSpinner;
    private JButton btnSalva;
    private JButton btnAnnulla;
    private PrenotazioniFrame parent;

    public PrenotazioniAddDialog(PrenotazioniFrame parent) {
        super(parent , true);
        this.parent = parent;
        setSize((int)(parent.getWidth()*0.5),(int)(parent.getHeight()*0.5));
        initComponents();
    }

    private void initComponents() {

        JPanel pc = new JPanel();
        pc.setLayout(new GridLayout(5,2,0,20));
        pc.add(new JLabel("Nome paziente:"));
        nome = new JTextField(30);
        pc.add(nome);
        pc.add(new JLabel("Cognome paziente:"));
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
cognome = new JTextField(30);
pc.add(cognome);
pc.add(new JLabel("Tipo Intervento:"));
tipoIntervento = new JTextField(30);
pc.add(tipoIntervento);
pc.add(new JLabel("Data intervento:"));

data = new UtilDateModel();
//libreria https://github.com/JDatePicker/JDatePicker
Properties p = new Properties();
p.put("text.today", "Oggi");
p.put("text.month", "Mese");
p.put("text.year", "Anno");
JDatePanelImpl datePanel = new JDatePanelImpl(data, p);
JDatePickerImpl datePicker = new JDatePickerImpl(datePanel, new
DateComponentFormatter());
pc.add(datePicker);

pc.add(new JLabel("Ora intervento:"));
//utilizzo spinner

SpinnerDateModel sm = new SpinnerDateModel(new Date(), null, null,
Calendar.MINUTE);

oraSpinner = new JSpinner(sm);
JSpinner.DateEditor de = new JSpinner.DateEditor(oraSpinner, "hh:mm a");
de.getTextField().setEditable( false );
oraSpinner.setEditor(de);

pc.add(oraSpinner);

JPanel pb = new JPanel();
pb.setLayout(new FlowLayout());
setLayout(new GridBagLayout());
btnSalva = new JButton("Salva");
btnSalva.addActionListener(this);
pb.add(btnSalva);
btnAnnulla = new JButton("Annulla");
btnAnnulla.addActionListener(this);
pb.add(btnAnnulla);

GridBagConstraints gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.weightx=100;
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
gbc.weighty = 100;
gbc.insets = new Insets(10,10,10,10);
add(pc, gbc);

gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 1;
gbc.weightx=1;
gbc.weighty = 1;
add(pb, gbc);
}

public void actionPerformed(ActionEvent e) {
    if(e.getSource().equals(btnSalva)) {
        Prenotazione p = new Prenotazione(nome.getText(),cognome.getText());
        p.setTipoIntervento(tipoIntervento.getText());
        if(data.getValue() != null) {
            Calendar c = Calendar.getInstance();
            c.setTime(data.getValue());
            Date hm = (Date)oraSpinner.getModel().getValue();
            c.set(Calendar.HOUR, hm.getHours());
            c.set(Calendar.MINUTE, hm.getMinutes());
            p.setDataOra(c);
        }
        parent.aggiungiPrenotazione(p);
        this.dispose();
    }
    else if(e.getSource().equals(btnAnnulla)) {
        this.dispose();
    }
}
}
```

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

Domanda 2 - (punti: 8 di 30)

Punti assegnati: 8

Si spieghi con un esempio come può essere letto in java un file CSV.

Risposta

Si veda il file in allegato

Allegato:

letturaCSV/LettoreCSV.java**package** letturaCSV;**import** java.io.BufferedReader;**import** java.io.File;**import** java.io.FileNotFoundException;**import** java.io.FileReader;**import** java.io.IOException;**import** java.util.ArrayList;**public class** LettoreCSV {**public** ArrayList<Persona> leggiCSV(File f) throws IOException {  
    ArrayList<Persona> arr = **new** ArrayList<Persona>();

//Stream di lettura file di testo

FileReader fR = null;

BufferedReader buffR = null;

**try** {        fR = **new** FileReader(f);

//lettore per leggere singole righe dal file

        buffR = **new** BufferedReader(fR);    **try** {

//leggo una linea dal file

String line = buffR.readLine();

//fino a che non arrivo alla fine del file

**while**(line != null) {

//converto la stringa in oggetto

//invocando il costruttore da String

            arr.add(**new** Persona(line));

//leggo la linea successiva del file

line = buffR.readLine();

}

    } **catch** (IOException e) {

throw e;

}

## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        } catch (FileNotFoundException e) {
            //rilancio l'eccezione al chiamante
            throw e;
        }
        finally {
            //anche in caso di eccezione chiudo gli stream per rendere il file
            //disponibile ad altri processi
            if(buffR != null) {
                buffR.close();
            }
            if(fR != null) {
                try {
                    fR.close();
                } catch (IOException e) {}
            }
        }

        return arr;
    }

    public static void main(String[] args) {
        LettoreCSV l = new LettoreCSV();
        try {
            ArrayList<Persona> arr = l.leggiCSV(new
            File("C:\\Users\\Francesco\\Desktop\\Test.csv"));
            for(Persona p: arr) {
                System.out.println(p.toString());
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

letturaCSV/Persona.java

**package** letturaCSV;

**import** java.util.StringTokenizer;





## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
public class Persona {  
  
    private String nome;  
    private String cognome;  
  
    public Persona(String s) {  
        //Per separare la stringa in tokens individuati da ;  
        StringTokenizer sT = new StringTokenizer(s, ";");  
        nome = sT.nextToken();  
        cognome = sT.nextToken();  
    }  
  
    @Override  
    public String toString() {  
        return "Persona [nome=" + nome + ", cognome=" + cognome + "];"  
    }  
}
```



## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

Note correzione	OK.
Soluzione proposta	<pre><b>public class</b> Studente {      <b>private</b> String nome;     <b>private</b> String cognome;      <b>public</b> Studente(String nome, String cognome) {         super();         this.nome = nome;         this.cognome = cognome;     }      <b>public</b> String getNome() {         return nome;     }      <b>public void</b> setNome(String nome) {         this.nome = nome;     }      <b>public</b> String getCognome() {         return cognome;     }      <b>public void</b> setCognome(String cognome) {         this.cognome = cognome;     }  }</pre> <hr/> <p><u>Es5/src/ElencoD.java</u></p> <pre><b>import</b> java.io.BufferedReader; <b>import</b> java.io.File; <b>import</b> java.io.FileNotFoundException; <b>import</b> java.io.FileReader; <b>import</b> java.io.IOException; <b>import</b> java.util.ArrayList;  <b>public class</b> ElencoD <b>extends</b> ArrayList &lt;Studente&gt;{</pre>

## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
public void leggiDaCsv(){
    File f=new File("C:\\Users\\studente\\Desktop\\Studenti.txt"); //creo l'oggetto
file passando il percorso definito tra le virgolette
    FileReader fr = null;
    BufferedReader br = null;
    try {
        fr=new FileReader(f); //passo al file reader il mio oggetto file
        br=new BufferedReader(fr); //passo al burred reader il mio oggetto file
reader
        String line=br.readLine(); //leggo la linea sul file
        while(line!=null){ //controllo che non sia stata raggiunta la fine del file
            String[] parti=line.split(";"); //dico che il separatore tra un campo e
l'altro e' il ; (per il file csv)
            Studente p=new Studente(parti[0], parti[1]); //creo l'oggetto studente
con i suoi due campi nome e cognome presi dal file e separati dal ;
            this.add(p); //aggiungo all'arrayList lo studente

            line = br.readLine(); //viene letta la riga successiva del file
            System.out.println("fatto");
        }
    } catch (IOException e) {

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



## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

--	--

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

<b>Domanda 3 - (punti: 5 di 30)</b>	<b>Punti assegnati: 5</b>
-------------------------------------	---------------------------

Si fornisca la definizione di filesystem

Risposta	<p>Il file system e' uno dei moduli che costituiscono il Sistema Operativo. Il file system permette la memorizzazione, la gestione, il mantenimento e la lettura di file memorizzati su un supporto di memoria di massa come hard disk, floppy disk, chiavetta USB...</p> <p>Normalmente i file system offrono una visualizzazione di tipo gerarchico dei file presenti sull'unita', per facilitare cosi' la gestione dell'archivio da parte dell'utente che puo' memorizzare i file in cartelle e sottocartelle per poterli rintracciare piu' agevolmente. Nonostante questo, sul disco i file non vengono memorizzati seguendo una regola gerarchica, ma vengono salvati modificando la struttura atomica del disco in corrispondenza di blocchi (possibilmente contigui per file di dimensioni maggiori a quella di un blocco) che vengono create durante il processo di formattazione del disco. Formattando il disco il file system divide, a livello atomico, la superficie del disco stesso in circonferenze concentriche, dette tracce. La superficie viene ancora divisa in spicchi, chiamati settori, e l'intersezione tra una traccia e un settore definisce appunto un blocco. Piu' blocchi contigui formano un cluster. Un hard disk e infine composto da piu' dischi (piatti) e per ogni faccia di ogni disco e' presente una testina per la lettura/scrittura.</p> <p>Al momento dell'accesso al disco, il file system, agendo sul modulo del SO per il controllo delle periferiche, individua su quale cilindro (insieme di cluster concidenti in dischi sovrapposti che formano un angolo giro (C)) e' presente il file cercato, quindi aziona la testina (H) corrispondente al piatto del disco e infine legge in corrispondenza del settore (S) corretto (Metodo CHS).</p> <p>Per facilitare l'accesso ad un determinato blocco, il file system puo' anche assegnare ai blocchi un LBA (Logic Block Address) che poi viene convertito in CHS.</p> <p>In Microsoft i file system piu' usati sono NTFS, FAT e FAT32</p>
Note correzione	ok
Soluzione proposta	<p>Modulo del Sistema Operativo che si occupa della gestione, organizzazione e mantenimento dei file su un supporto di memorizzazione di massa.</p> <p>L'organizzazione dei file all'interno del filesystem è solitamente di tipo gerarchico. Il tipo di filesystem viene impostato durante la formattazione ad alto livello del disco/partizione. Nel mondo Microsoft, i filesystem più diffusi sono NTFS e FAT (nelle varie versioni), per Linux EXT3, EXT4,..., per Mac APFS.</p>

<b>Domanda 4 - (punti: 7 di 30)</b>	<b>Punti assegnati: 7</b>
-------------------------------------	---------------------------

Si fornisca l'implementazione dell'interfaccia java `IFileOperations`, che definisce il seguente metodo:

- `public void salvaMultipli(int num, String percorso);` che, in base al percorso passato come parametro, salva su file di testo i primi 10 multipli di num (andare a capo dopo ogni numero);

## Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

Risposta	<p>Si veda il file allegato</p> <p>Allegato: <u><a href="#">interfaccia/IFileOperations.java</a></u> <b>package</b> interfaccia;</p> <p><b>public interface</b> IFileOperations {</p> <p>    <b>public void</b> salvaMultipli(<b>int</b> num, String percorso);</p> <p>}</p> <hr/> <p><u><a href="#">interfaccia/FileOperator.java</a></u> <b>package</b> interfaccia;</p> <p><b>import</b> java.io.FileWriter;</p> <p><b>import</b> java.io.IOException;</p> <p><b>public class</b> FileOperator <b>implements</b> IFileOperations {</p> <p>    @Override</p> <p>    <b>public void</b> salvaMultipli(<b>int</b> num, String percorso) {</p> <p>        FileWriter fW = null;</p> <p>        <b>try</b> {</p> <p>            fW = <b>new</b> FileWriter(percorso);</p> <p>            <b>for</b>(<b>int</b> i = 1; i &lt;= 10; i++) {</p> <p>                fW.write(String.valueOf(num * i));</p> <p>                fW.write("\r\n");</p> <p>            }</p> <p>        } <b>catch</b> (IOException e) {</p> <p>            // TODO Auto-generated <b>catch</b> block</p> <p>            e.printStackTrace();</p> <p>        }</p> <p>        <b>finally</b> {</p> <p>            <b>if</b>(fW != null) {</p> <p>                <b>try</b> {</p> <p>                    fW.flush();</p> <p>                    fW.close();</p> <p>                } <b>catch</b> (IOException e) {</p> <p>                    // TODO Auto-generated <b>catch</b> block</p> <p>                    e.printStackTrace();</p> <p>                }</p> <p>            }</p> <p>        }</p> <p>    }</p> <p>}</p>
----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Verifica di Informatica Nr. 2 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>06/12/2018</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

	<pre><b>public static void</b> main(String[] args) {     FileOperator fO = <b>new</b> FileOperator();     fO.salvaMultipli(5, "C:\\Users\\Francesco\\Desktop\\Test.txt"); }</pre>
Note correzione	ok
Soluzione proposta	