

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

Domanda 1 - (punti: 10 di 30)

Punti assegnati:  
10

Si implementi un programma con grafica java swing per la gestione delle prenotazioni di un laboratorio che contempli le seguenti informazioni: classe, nome docente, data, durata minuti. L'interfaccia grafica dovrà consentire l'inserimento di tutti i suddetti dati e tramite la pressione di un pulsante salvare ogni prenotazione in un elenco di prenotazioni (non è necessaria la visualizzazione su interfaccia dell'elenco). Tramite la voce di menu File > Salva procedere al salvataggio su file CSV dell'elenco inserito. Tramite la voce di menu File > Carica procedere al caricamento dei dati dal file precedentemente creato (non è necessaria la visualizzazione su interfaccia grafica).

Risposta

Si veda il file allegato

Allegato:

[prenotazioniLaboratorio/frontend/gui/mainFrame/GestionePrenotazioniGUI.java](#)  
**package** prenotazioniLaboratorio.frontend.gui.mainFrame;

```
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.io.File;
import java.io.IOException;
import java.util.Date;
import java.util.GregorianCalendar;

import javax.swing.JButton;
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.JPanel;
import javax.swing.JSpinner;
import javax.swing.JTextField;
import javax.swing.SpinnerDateModel;
import javax.swing.filechooser.FileNameExtensionFilter;

import prenotazioniLaboratorio.backend.bean.ElencoPrenotazioni;
import prenotazioniLaboratorio.backend.bean.PrenotazioneLaboratorio;

public class GestionePrenotazioniGUI extends JFrame implements
```



## Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
ActionListener {  
  
    private static final long serialVersionUID = -6609203161800904175L;  
  
    private JTextField txtClasse;  
    private JTextField txtDocente;  
    private JSpinner spinData;  
    private JTextField txtDurata;  
  
    private JButton btnConferma;  
  
    private JMenuItem itemSalva;  
    private JMenuItem itemApri;  
  
    private ElencoPrenotazioni ePren;  
  
    public GestionePrenotazioniGUI() {  
        super("Gestione prenotazioni");  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        initComponents();  
        pack();  
        ePren = new ElencoPrenotazioni();  
    }  
  
    public void initComponents() {  
        setJMenuBar(initJMenuBar());  
  
        setLayout(new GridBagLayout());  
        GridBagConstraints gbc = new GridBagConstraints();  
        gbc.gridx = gbc.gridy = 0;  
        gbc.weightx = gbc.weighty = 5;  
        gbc.fill = GridBagConstraints.BOTH;  
        gbc.insets = new Insets(5, 5, 5, 5);  
  
        add(initDataPanel(), gbc);  
  
        gbc.gridy = 1;  
        gbc.weighty = 1;  
  
        add(initButsPanel(), gbc);  
    }  
  
    private JMenuBar initJMenuBar() {  
        JMenuBar bar = new JMenuBar();  
        bar.add(initFileJMenu());  
        return bar;  
    }
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
}

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

    itemApri = new JMenuItem("Apri");
    itemApri.addActionListener(this);

    itemSalva = new JMenuItem("Salva");
    itemSalva.addActionListener(this);

    mnuFile.add(itemApri);
    mnuFile.add(itemSalva);

    return mnuFile;
}

private JPanel initDataPanel() {
    JPanel pan = new JPanel(new GridLayout(4, 2, 5, 5));

    pan.add(new JLabel("Classe"));
    pan.add(txtClasse = new JTextField(20));
    pan.add(new JLabel("Docente"));
    pan.add(txtDocente = new JTextField(20));
    pan.add(new JLabel("Data"));

    SpinnerDateModel model = new SpinnerDateModel(new Date(), null, null,
GregorianCalendar.DAY_OF_YEAR);
    spinData = new JSpinner(model);

    JSpinner.DateEditor dEdit = new JSpinner.DateEditor(spinData,
"dd/MM/yyyy");
    dEdit.getTextField().setEditable(false);
    spinData.setEditor(dEdit);

    pan.add(spinData);

    pan.add(new JLabel("Durata [min]"));
    pan.add(txtDurata = new JTextField(20));

    return pan;
}

private JPanel initButsPanel() {
    JPanel pan = new JPanel(new FlowLayout(FlowLayout.RIGHT));
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        btnConferma = new JButton("Conferma");
        btnConferma.addActionListener(this);

        pan.add(btnConferma);

        return pan;
    }

    public void performConferma() {
        if(!txtClasse.getText().equals("") &&!txtDocente.getText().equals("")
        &&!txtDurata.getText().equals("")) {
            GregorianCalendar data = new GregorianCalendar();
            data.setTime((Date) spinData.getValue());
            int durata = 0;
            try {
                durata = Integer.parseInt(txtDurata.getText());
            }
            catch(NumberFormatException e) {

            }

            PrenotazioneLaboratorio pren = new
PrenotazioneLaboratorio(txtClasse.getText(), txtDocente.getText(), data, durata);
            ePren.add(pren);

            txtClasse.setText("");
            txtDocente.setText("");
            spinData.setValue(new Date());
            txtDurata.setText("");
        }
    }

    public JFileChooser initFileChooser() {
        JFileChooser fC = new JFileChooser();
        fC.setFileFilter(new FileNameExtensionFilter("Testo CSV", "csv"));
        return fC;
    }

    public void performApri() throws IOException {
        JFileChooser fc = initFileChooser();
        if(fc.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
            try {
                ePren.importaCSV(fc.getSelectedFile());
            }
        }
    }
}
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        } catch (IOException e) {
            throw e;
        }
    }
}

public void performSalva() throws IOException {
    JFileChooser fc = initFileChooser();
    if(fc.showSaveDialog(this) == JFileChooser.APPROVE_OPTION) {
        File f = fc.getSelectedFile();
        if(!f.getAbsolutePath().endsWith(".csv")) {
            f = new File(f.getAbsolutePath() + ".csv");
        }
        try {
            ePren.esportaCSV(f);
        } catch (IOException e) {
            throw e;
        }
    }
}

public void showExceptionMessage(Exception e) {
    JOptionPane.showMessageDialog(this, e.getMessage(), "Errore",
    JOptionPane.ERROR_MESSAGE);
}

@Override
public void actionPerformed(ActionEvent aE) {
    if(aE.getSource().equals(btnConferma)) {
        performConferma();
    }
    else if(aE.getSource().equals(itemApri)) {
        try {
            performApri();
        } catch (IOException e) {
            showExceptionMessage(e);
        }
    }
    else if(aE.getSource().equals(itemSalva)) {
        try {
            performSalva();
        } catch (IOException e) {
            showExceptionMessage(e);
        }
    }
}
}
```



## Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

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

}



```

prenotazioniLaboratorio/backend/bean/ElencoPrenotazioni.java
package prenotazioniLaboratorio.backend.bean;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;

import prenotazioniLaboratorio.backend.service.FileServiceFactory;
import prenotazioniLaboratorio.backend.service.IFileService;

public class ElencoPrenotazioni extends ArrayList<PrenotazioneLaboratorio> {

    private static final long serialVersionUID = 2839220487963455623L;

    public boolean add(PrenotazioneLaboratorio p) {
        boolean ris = false;
        if(p != null) {
            ris = super.add(p);
        }
        else {
            throw new IllegalArgumentException();
        }
        return ris;
    }

    public void esportaCSV(File f) throws IOException {
        IFileService service = FileServiceFactory.getFileService();
        try {
            service.salva(this, f);
        } catch (IOException e) {
            throw e;
        }
    }

    public void importaCSV(File f) throws IOException {
        IFileService service = FileServiceFactory.getFileService();
        ElencoPrenotazioni _e;
        try {

```


```



## Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        _e = service.carica(f);
        this.clear();
        this.addAll(_e);
    } catch (IOException e) {
        throw e;
    }
}
```

prenotazioniLaboratorio/backend/service/IFileService.java

**package** prenotazioniLaboratorio.backend.service;

**import** java.io.File;

**import** java.io.IOException;

**import** prenotazioniLaboratorio.backend.bean.ElencoPrenotazioni;

**public interface** IFileService {

**public void** salva(ElencoPrenotazioni el, File f) throws IOException;

**public** ElencoPrenotazioni carica(File f) throws IOException;

}

prenotazioniLaboratorio/backend/service/FileServiceFactory.java

**package** prenotazioniLaboratorio.backend.service;

**public class** FileServiceFactory {

**public static** IFileService getFileService() {  
 return (IFileService)**new** FileServiceImpl();  
 }

}

prenotazioniLaboratorio/backend/bean/PrenotazioneLaboratorio.java

**package** prenotazioniLaboratorio.backend.bean;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.GregorianCalendar;

**public class** PrenotazioneLaboratorio {

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
private static final String CSV_SEPARATOR = ";";

private String classe;
private String nomeDocente;
private GregorianCalendar data;
private int durataMin;

public PrenotazioneLaboratorio(String classe, String nomeDocente,
GregorianCalendar data, int durataMin) {
    super();
    this.classe = classe;
    this.nomeDocente = nomeDocente;
    this.data = data;
    this.durataMin = durataMin;
}

public PrenotazioneLaboratorio(String daCSV) {
    String[] campi = daCSV.split(CSV_SEPARATOR);
    this.classe = campi[0];
    this.nomeDocente = campi[1];
    SimpleDateFormat fmt = new SimpleDateFormat("dd/MM/yyyy");
    GregorianCalendar d = new GregorianCalendar();
    try {
        d.setTime(fmt.parse(campi[2]));
    } catch (ParseException e) {
    }
    this.data = d;
    this.durataMin = Integer.parseInt(campi[3]);
}

public String toCSVString() {
    StringBuffer sB = new StringBuffer();
    sB.append(classe);
    sB.append(CSV_SEPARATOR);
    sB.append(nomeDocente);
    sB.append(CSV_SEPARATOR);
    SimpleDateFormat fmt = new SimpleDateFormat("dd/MM/yyyy");
    sB.append(fmt.format(data.getTime()));
    sB.append(CSV_SEPARATOR);
    sB.append(String.valueOf(durataMin));
    sB.append(CSV_SEPARATOR);
}
```



**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        return sB.toString();
    }

    public String getClasse() {
        return classe;
    }
    public String getNomeDocente() {
        return nomeDocente;
    }
    public GregorianCalendar getData() {
        return data;
    }
    public int getDurataMin() {
        return durataMin;
    }
}
```

*prenotazioniLaboratorio/backend/service/FileServiceImpl.java*

**package** prenotazioniLaboratorio.backend.service;

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** prenotazioniLaboratorio.backend.bean.ElencoPrenotazioni;

**import** prenotazioniLaboratorio.backend.bean.PrenotazioneLaboratorio;

**public class** FileServiceImpl **implements** IFileService {

FileServiceImpl() {

}

@Override

**public void** salva(ElencoPrenotazioni el, File f) throws IOException {  
 FileWriter fW = null;

**try** {

fW = **new** FileWriter(f);

**for**(PrenotazioneLaboratorio p: el) {

fW.write(p.toCSVString());

fW.write("\r\n");

}

## Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

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

@Override
public ElencoPrenotazioni carica(File f) throws IOException {
    BufferedReader buffR= null;
    ElencoPrenotazioni ris = new ElencoPrenotazioni();
    try {
        buffR = new BufferedReader(new FileReader(f));

        String line = buffR.readLine();
        while(line != null) {
            ris.add(new PrenotazioneLaboratorio(line));
            line = buffR.readLine();
        }

    } catch (IOException e) {
        throw e;
    }
    finally {
        if(buffR != null) {

            try {
                buffR.close();
            } catch (IOException e) {
```



## Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

return ris;

}
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

Note correzione	ok
Soluzione proposta	<pre>import java.text.SimpleDateFormat; import java.util.Calendar;  public class Prenotazione {      private static final String SEPARATORE_CSV = ",";      private String classe;     private String nomeDocente;     private Calendar data;     private int durataMin;     public Prenotazione(String classe, String nomeDocente, Calendar data, int durataMin) {         super();         this.classe = classe;         this.nomeDocente = nomeDocente;         this.data = data;         this.durataMin = durataMin;     }     public String getClasse() {         return classe;     }     public void setClasse(String classe) {         this.classe = classe;     }     public String getNomeDocente() {         return nomeDocente;     }     public void setNomeDocente(String nomeDocente) {         this.nomeDocente = nomeDocente;     }     public Calendar getData() {         return data;     }     public void setData(Calendar data) {         this.data = data;     }     public int getDurataMin() {         return durataMin;     }     public void setDurataMin(int durataMin) {         this.durataMin = durataMin;     }      public String toString() {</pre>

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
StringBuffer buf = new StringBuffer();
buf.append(getClasse());
buf.append(SEPARATORE_CSV);
buf.append(getNomeDocente());
buf.append(SEPARATORE_CSV);
buf.append(sdf.format(getData().getTime()));
buf.append(SEPARATORE_CSV);
buf.append(getDurataMin());
return buf.toString();
}
}
```

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
```

```
public class ElencoPrenotazioni extends ArrayList<Prenotazione>{
```

```
    public void salvaSuFileCsv(String percorso) {
        FileWriter fw = null;
        try {
            File f = new File(percorso);
            fw = new FileWriter(f);
            for(int i=0; i<this.size(); i++) {
                Prenotazione p = this.get(i);
                fw.write(p.toString());
                fw.write("\r\n");
            }
            fw.flush();
        }catch(Exception exc) {
            exc.printStackTrace();
        }
        finally {
            if(fw != null) {
                try {
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

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

public void caricaDaFileCsv(String percorso) {
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
    FileReader fr = null;
    BufferedReader br = null;
    try {
        File f = new File(percorso);
        fr = new FileReader(f);
        br = new BufferedReader(fr);
        String line = null;
        do {
            line = br.readLine();
            if(line != null) {
                String[] parti = line.split(";");

                Date d = sdf.parse(parti[2]);
                Calendar cal = Calendar.getInstance();
                cal.setTime(d);
                int dMin = Integer.parseInt(parti[3]);
                Prenotazione p = new Prenotazione(parti[0], parti[1], cal, dMin);
                this.add(p);
            }
        }while(line != null);

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

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        } catch (IOException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
    }  
}  
}  
}  
}  
  
import java.awt.FlowLayout;  
import java.awt.GridBagLayout;  
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
  
import javax.swing.*.*;  
  
public class PrenotazioneGUI extends JFrame implements ActionListener{  
  
    private JLabel lblClasse;  
    private JTextField txtClasse;  
    private JLabel lblDocente;  
    private JTextField txtDocente;  
    private JLabel lblDurata;  
    private JTextField txtDurata;  
    private JLabel lblData;  
    private JTextField txtData;  
    private JButton salva;  
  
    private JMenuBar bar;  
    private JMenu file;  
    private JMenuItem salva1;  
    private JMenuItem carica;  
  
    private ElencoPrenotazioni elenco;
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
public PrenotazioneGUI(){
    elenco=new ElencoPrenotazioni();
    init();
}

public void init(){
    lblClasse=new JLabel ("inserisci la classe");
    txtClasse=new JTextField(30);
    lblDocente=new JLabel ("inserisci il docente");
    txtDocente=new JTextField(30);
    lblDurata=new JLabel ("inserisci la durata in minuti");
    txtDurata=new JTextField(30);
    lblData=new JLabel ("inserisci la data dd/MM/yyyy");
    txtData=new JTextField(30);

    bar=new JMenuBar();
    file=new JMenu("file");
    salva1=new JMenuItem("Salva");
    file.add(salva1);
    carica=new JMenuItem("carica");
    file.add(carica);
    carica.addActionListener(this);
    salva1.addActionListener(this);
    bar.add(file);
    setJMenuBar(bar);

    JPanel p=new JPanel();
    p.setLayout(new GridLayout(4,2));
    p.add(lblClasse);
    p.add(txtClasse);
    p.add(lblDocente);
    p.add(txtDocente);
    p.add(lblDurata);
    p.add(txtDurata);
    p.add(lblData);
    p.add(txtData);

    setLayout(new FlowLayout());
    add(p);

    salva=new JButton("SALVA");
    salva.addActionListener(this);
    add(salva);
}

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



**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
PrenotazioneGUI f=new PrenotazioneGUI();

f.setSize(600,600);
f.setVisible(true);

}

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getSource().equals(salva)){
        SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
        String c=txtClasse.getText();
        String d=txtDocente.getText();
        String min=txtDurata.getText();
        int m=Integer.parseInt(min);
        String data=txtData.getText();
        try {
            Date cal=sdf.parse(data);
        } catch (ParseException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
        Calendar cal1=Calendar.getInstance();

        Prenotazione p=new Prenotazione(c,d,cal1,m);
        elenco.add(p);

        //svuoto i campi per nuovo inserimento
        txtClasse.setText("");
        txtDocente.setText("");
        txtData.setText("");
        txtDurata.setText("");
        JOptionPane.showMessageDialog(this, "Prenotazione salvata");

    }
    else if(e.getSource().equals(salva1)){
        elenco.salvaSuFileCsv("C:\\Java\\prenotazioni.csv");
    }
    else if(e.getSource().equals(carica)){
        elenco.caricaDaFileCsv("C:\\Java\\prenotazioni.csv");
    }
}
}
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

Domanda 2 - (punti: 4 di 30)		Punti assegnati: 4
Si fornisca la definizione di file.		
Risposta	Con il termine file si intende un archivio di dati (e relative informazioni) solitamente reso persistente, in modo da poter recuperare i dati memorizzati in esso anche dopo uno spegnimento e accensione della macchina. I dati memorizzati sono rappresentati tramite una sequenza di byte che viene scritta fisicamente sul disco rigido, o in generale sul supporto di memorizzazione di massa, al momento della creazione. Per facilitare l'accesso ai file, il loro mantenimento e gestione all'interno di un supporto di memorizzazione, il SO dispone di un modulo chiamato file system in cui per ogni file fisicamente scritto sul disco e' presente una referenza al del tipo CHS o LBA (allocazione fisica) al disco stesso. I moderni file system offrono una visualizzazione gerarchica dei file memorizzati (directory e sotto-directory) per facilitare ulteriormente la gestione dei file sul disco, che pero' fisicamente non e' rispettata.	
Note correzione	ok	
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.	

Domanda 3 - (punti: 8 di 30)		Punti assegnati: 8
Si fornisca l'implementazione dell'interfaccia java <code>IFileOperations</code> , che definisce i seguenti metodi: - <code>public void salvaMultipli(int num, String percorso);</code> che, in base al percorso passato come parametro, salva su file di testo i primi 10 multipli di num; - <code>public void serializzaValori(ArrayList&lt;Double&gt; listaValori, String percorso);</code> che serializza su file l'elenco dei valori passato come parametro; - <code>public ArrayList&lt;Double&gt; deserializzaValori(String percorso);</code> che deserializza e restituisce l'elenco dei valori serializzati su file.  Si proceda a richiamare i metodi nel main.		
Risposta	Si veda il file in allegato Allegato: <u><a href="#">interfaccia/FileOperationsImpl.java</a></u> <b>package</b> interfaccia;  <b>import</b> java.io.FileInputStream; <b>import</b> java.io.FileOutputStream; <b>import</b> java.io.FileWriter; <b>import</b> java.io.IOException; <b>import</b> java.io.ObjectInputStream;	

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
import java.io.ObjectOutputStream;
import java.util.ArrayList;

public class FileOperationsImpl implements IFileOperations {

    @Override
    public void salvaMultipli(int num, String percorso) {
        FileWriter fW = null;
        try {
            fW = new FileWriter(percorso);
            for(int i = 1; i <= 10; i++) {
                fW.write(String.valueOf(num * i));
                fW.write("\r\n");
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        finally {
            if(fW != null) {
                try {
                    fW.flush();
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                finally {
                    try {
                        fW.close();
                    } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }
            }
        }
    }

    @Override
    public void serializzaValori(ArrayList<Double> listaValori, String percorso) {
        ObjectOutputStream oos = null;
        try {
            oos = new ObjectOutputStream(new FileOutputStream(percorso));
            oos.writeObject(listaValori);
        } catch (IOException e) {
```

## Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

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

@Override
public ArrayList<Double> deserializzaValori(String percorso) {
    ArrayList<Double> ris = null;
    ObjectInputStream ois = null;
    try {
        ois = new ObjectInputStream(new FileInputStream(percorso));
        ris = (ArrayList<Double>) ois.readObject();
    } catch (IOException | ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    finally {
        if(ois != null) {
            try {
                ois.close();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
```



## Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E

Ghinamo Francesco	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
-------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
        return ris;
    }

    public static void main(String[] args) {
        FileOperationsImpl i = new FileOperationsImpl();
        i.salvaMultipli(5, "C:\\Users\\franc\\OneDrive\\Documents\\Test.txt");
        ArrayList<Double> test = new ArrayList<Double>();
        test.add(2342D);
        test.add(3123D);
        i.serializzaValori(test, "C:\\Users\\franc\\OneDrive\\Documents\\Ser.dat");
        ArrayList<Double> ris =
        i.deserializzaValori("C:\\Users\\franc\\OneDrive\\Documents\\Ser.dat");

    }
}
```

interfaccia/IFileOperations.java

**package** interfaccia;

**import** java.util.ArrayList;

**public interface** IFileOperations {

**public void** salvaMultipli(**int** num, String percorso);

**public void** serializzaValori(ArrayList<Double> listaValori, String percorso);

**public** ArrayList<Double> deserializzaValori(String percorso);

}

Note  
correzione

ok

Soluzione  
proposta

```
import java.util.ArrayList;
```

```
public interface IFileOperations {
```

```
    public void salvaMultipli(int num, String percorso);
```

```
    public void serializzaValori(ArrayList<Double> listaValori, String percorso);
```

```
    public ArrayList<Double> deserializzaValori(String percorso);
```

```
}
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.lang.reflect.Array;
import java.util.ArrayList;

public class FileOperations implements IFileOperations {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        FileOperations a= new FileOperations();
        ArrayList<Double> array= new ArrayList<Double>();
        array.add((double) 5);
        a.salvaMultipli(2, "testo.txt");
        a.serializzaValori(array, "seria.bin");
        array.clear();
        array= a.deserializzaValori("seria.bin");
        System.out.println(String.valueOf(array.get(0)));
    }

    @Override
    public void salvaMultipli(int num, String percorso) {
        // TODO Auto-generated method stub
        FileWriter fw=null;
        try{File f= new File(percorso);
            fw= new FileWriter(f);
            for(int i=1;i<11;i++){
                fw.write(String.valueOf(num*i));
                fw.write("\r\n");}
            fw.flush();}catch(IOException e){}
        finally{
            if(fw!=null){try {
                fw.close();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }}
        }
    }
}
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
    }

}

@Override
public void serializzaValori(ArrayList<Double> listaValori, String percorso) {
    // TODO Auto-generated method stub
    FileOutputStream out= null;
    ObjectOutputStream fout =null;
    try{
        File f= new File(percorso);
        out= new FileOutputStream(f);
        fout= new ObjectOutputStream(out);
        fout.writeObject(listaValori);
        fout.flush();
    }catch(IOException e){}
    finally{
        if(out!=null){try {
            out.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }}
        if(fout!=null){try {
            fout.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }}
    }

}

@Override
public ArrayList<Double> deserializzaValori(String percorso) {
    // TODO Auto-generated method stub
    FileInputStream in= null;
    ObjectInputStream fint= null;
    ArrayList<Double> array= new ArrayList<Double>();
    try{
        File f=new File(percorso);
        in=new FileInputStream(f);
        fint= new ObjectInputStream(in);

        try {
            array= (ArrayList<Double>) fint.readObject();
        }
    }
}
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

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

**Domanda 4 - (punti: 8 di 30)****Punti assegnati: 8**

Implementare una classe singleton BlackListWebSite con i seguenti metodi:

- void addWebSite(String url) : che aggiunge l'url passato alla lista dei siti in black list.
- boolean isWebSiteBlocked(String url) : che restituisce true se l'url passato come parametro appartiene alla black list, false altrimenti.

Richiamare i metodi dal main nella classe BlackListMain.

**Risposta**

Si veda il file allegato

Allegato:

[singleton/BlackListMain.java](#)

**package** singleton;

**public class** BlackListMain {

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

    //test in debug

    BlackListWebSite list = BlackListWebSite.getInstance();

    list.addWebSite("http://google.com");



**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
list.addWebSite("http://liceocuneo.it");
boolean ris = list.isWebSiteBlocked("http://google.com");
boolean ris1 = list.isWebSiteBlocked("http://test.it");

System.out.println(ris);
System.out.println(ris1);

BlackListWebSite list1 = BlackListWebSite.getInstance();

}

}
```

---

singleton/BlackListWebSite.java  
**package** singleton;

**import** java.util.Hashtable;

null

**public class** BlackListWebSite **extends** Hashtable<String, String> {

**private static** final **long** serialVersionUID = -7165135693682461062L;

**private static** BlackListWebSite me;

**private** BlackListWebSite() {

    }

**public static** BlackListWebSite getInstance() {

**if**(me == null) {

            me = **new** BlackListWebSite();

        }

        return me;

    }

**public void** addWebSite(String url) {

**if**(url != null) {

            this.put(url, url);

        }

    }



## Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

```
public boolean isWebSiteBlocked(String url) {  
    return this.containsKey(url);  
}  
  
}
```

**Verifica di Informatica Nr. 3 - Anno 2018/2019 - Classe 4E**

<b>Ghinamo Francesco</b>	Classe: <b>4 E</b>	Data: <b>07/02/2019</b>	Penalita':	Punti: <b>30.0 / 30</b>	Voto: <b>10.0</b>
--------------------------	-----------------------	----------------------------	------------	----------------------------	----------------------

Note correzione	ok
Soluzione proposta	<pre>import java.util.ArrayList;  public class BlackListWebSite {      private static BlackListWebSite me;      private ArrayList&lt;String&gt; urlList;      private BlackListWebSite() {         urlList = new ArrayList&lt;String&gt;();     }      public static BlackListWebSite getInstance() {         if(me == null) {             me = new BlackListWebSite();         }         return me;     }      public void addWebSite(String url) {         if((url != null) &amp;&amp; (!urlList.contains(url))) {             urlList.add(url);         }     }      public boolean isWebSiteBlocked(String url) {         return urlList.contains(url);     } }  public class BlackListMain {      public static void main(String[] args) {         BlackListWebSite bl = BlackListWebSite.getInstance();         bl.addWebSite("http://www.sitobrutto.com");         if(bl.isWebSiteBlocked("http://www.sitobrutto.com")) {             System.out.println("Sito bloccato");         }     } }</pre>