

DOCUMENTAȚIE PROIECT FINAL

Andron Flavia Călina

Blaga Radu Cătălin

Cărturoiu Bogdan Marian

1.Descriere proiect

Proiect final OOP își propune să illustreze interfața dedicată unui angajat al unei companii ce produce cablaje electrice. Ne dorim ca în urma realizării unui cont de utilizator, angajatul sa aibă o aplicație prin care poate să adauge informații legate de modul de desfășurare a activității prestate de către el, în departament de producție. De asemenea, ne dorim sa includem ferestre suplimentare pentru departamentul de vânzări și departamentul de achiziții.

2.Prezentare ferestre

2.1. PANOUL CONECTARE

Pentru început, noi am implementat o clasă 'login', prin care utilizatorul se poate conecta la meniul creat de noi, numit 'Meniu principal'. Pentru a se conecta, utilizatorul trebuie să introducă de la tastatură un nume și o parolă, parola poate fi văzută integral apăsând pe butonul 'Vizualizare parolă', în caz că acesta a greșit și vrea să se corecteze fără a șterge tot ce a scris în câmpul parolei. Butonul resetare șterge tot ce este scris în câmpurile pentru nume și parolă. După apăsarea butonului 'Conectare', utilizatorul este redirecționat către meniul principal și acolo își alege ce face mai departe.

Captură de ecran cu fereastra de login:



The screenshot shows a Windows-style application window with a pink title bar containing the text 'Hirschmann Automotive' and standard window control icons. The main content area has a blue background. At the top, the word 'CONECTARE' is displayed in large white capital letters. Below this, the label 'Nume complet:' is followed by a white text input field. Further down, the label 'Parolă:' is followed by another white text input field. Below the password field is a checkbox with the label 'Vizualizare parolă'. At the bottom of the form are two white buttons with blue borders and text: 'Conectare' and 'Resetare'.

Cod sursă login:

```
package Project;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class login extends JFrame{
    private JPanel rootPanel;
    private JButton btnConectare;
    private JTextField txtNume;
    private JButton btnReset;
    private JPasswordField txtParola;
    private JCheckBox btnVizualizare;

    login() {
        btnVizualizare.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(btnVizualizare.isSelected()){
                    txtParola.setEchoChar((char)0);
                }
                else{
                    txtParola.setEchoChar('*');
                }
            }
        });
        btnReset.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(e.getSource()==btnReset);
                txtNume.setText("");
                txtParola.setText("");
            }
        });
        btnConectare.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JFrame frame=new JFrame("Meniu principal");
                frame.setContentPane(new
panouprincipal().getRootPane());

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.pack();
                frame.setSize(550,450);
                frame.setLocationRelativeTo(null);
                frame.setResizable(false);
                frame.setVisible(true);
            }
        });
    }
}
```

```

        ImageIcon imagine=new ImageIcon("logo.png");
        frame.setIconImage(imagine.getImage());
    }
    });
}

public static void main(String[] args){
    JFrame frame=new JFrame("Hirschmann Automotive");
    frame.setContentPane(new login().rootPanel);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.setSize(550,450);
    frame.setLocationRelativeTo(null);
    frame.setResizable(false);
    frame.setVisible(true);
    Color culoare =new Color(15,77,196);
    frame.getContentPane().setBackground(culoare);
    ImageIcon imagine=new ImageIcon("logo.png");
    frame.setIconImage(imagine.getImage());
}
}

```

2.2. MENU PRINCIPAL

După apăsarea butonului 'Conectare' va apărea meniul principal:



În fereastra meniului principal, utilizatorul poate alege din cele 4 butoane implementate, fiecare dintre acestea redirectionându-l către ferestre diferite.

Cod sursă Meniu principal:

```
package Project;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ComponentAdapter;
import java.awt.event.ComponentEvent;

public class panouprincipal extends JFrame {
    private JButton sesizareDefectiuniButton;
    private JButton achizitiiButton;
    private JButton dateEconomiceButton;
    private JButton vanzariButton;
    private JPanel rootPanel;

    panouprincipal() {
        super("Meniu principal");
        this.setContentPane(this.rootPanel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.pack();
        sesizareDefectiuniButton.addActionListener(new
        ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JFrame frame=new JFrame("Meniu principal
                defectiuni");

                frame.setContentPane(new
                Defectiuni().getRootPane());
                frame.pack();
                frame.setSize(700,600);
                frame.setLocationRelativeTo(null);
                frame.setResizable(true);
                frame.setVisible(true);
                ImageIcon imagine=new ImageIcon("logo.png");
                frame.setIconImage(imagine.getImage());
            }
        });
        dateEconomiceButton.addActionListener(new
        ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JFrame frame=new JFrame("Calcul economic");
                frame.setContentPane(new
                Economic().getRootPane());
            }
        });
    }
}
```

```

        frame.pack();
        frame.setSize(680,500);
        frame.setLocationRelativeTo(null);
        frame.setResizable(true);
        frame.setVisible(true);
        ImageIcon imagine=new ImageIcon("logo.png");
        frame.setIconImage(imagine.getImage());
    }

});
achizitiiButton.addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent e) {
        JFrame frame2=new JFrame("Achizitii");
        frame2.setContentPane(new
Achizitii().getRootPane());
        frame2.pack();
        frame2.setSize(680,500);
        frame2.setLocationRelativeTo(null);
        frame2.setResizable(false);
        frame2.setVisible(true);
        ImageIcon imagine=new ImageIcon("logo.png");
        frame2.setIconImage(imagine.getImage());
    }
});
vanzariButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JFrame frame2=new JFrame("Vanzari");
        frame2.setContentPane(new
Vanzari_Form().getRootPane());
        frame2.pack();
        frame2.setResizable(false);
        frame2.setVisible(true);
        frame2.setSize(600, 500);
        frame2.setLocationRelativeTo(null);
        ImageIcon imagine=new ImageIcon("logo.png");
        frame2.setIconImage(imagine.getImage());
    }
});
}

public static void main(String[] args) {
    JFrame frame=new JFrame("Hirschmann Automotive");
    frame.setContentPane(new panouprincipal().rootPanel);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.setSize(450,300);
}

```

```

frame.setLocationRelativeTo(null);
frame.setResizable(false);
frame.setVisible(true);
ImageIcon imagine=new ImageIcon("logo.png");
frame.setIconImage(imagine.getImage());
}
}

```

2.3. SESIZARE DEFECTIUNI

După apăsarea butonului ‘Sesizare defecțiuni’ va apărea fereastra următoare. Prin această fereastră ne propunem ca utilizatorul să poată introduce defecțiunile și problemele întâmpinate în producție. Aceste date se păstrează într-un ArrayList care se afișează în partea stângă a panoului. De asemenea, aceste informații se pot actualiza și modifica prin faptul că am realizat în cod o funcție ce actualizează informații.

Meniu principal defectiuni

DEFECȚIUNI TEHNICE PRODUCȚIE

Nume

Vârsta

Număr marcă

Secția

Completează defecțiunea pe care ai întâmpinat-o!

Salvare Modificare

Cod sursă Sesizare defectiuni :

```
package Project;
import javax.swing.*;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

//am realizat clasa Defectiuni

public class Defectiuni extends JFrame {
    private JPanel PanelSus;
    private JPanel PanelDreapta;
    private JLabel Nume;
    private JLabel Varsta;
    private JLabel Greutate;
    private JLabel Inaltime;
    private JLabel Cancere;
    private JTextField textNume;
    private JTextField textVarsta;
    private JTextField textMarca;
    private JTextField textSectia;
    private JButton CONECTAREButton;
    private JPanel PanelPrincipal;
    private JList list1;
    private JPanel PanelStanga;
    private JButton PacientNouButon;
    private JTextField textDefectiune;
    private JPanel compPanel;
    private JButton ButonSugestii;
    private ArrayList <Project> utilizator;
    private DefaultListModel listaUtilizatoriModel;

    Defectiuni() {

        super("Defectiuni");
        this.setContentPane(this.PanelPrincipal );
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.pack();
    }

    //am utilizat un arraylist in care am pastrat toate defectiunile
    utilizator =new ArrayList<Project>();
    listaUtilizatoriModel=new DefaultListModel();
    list1.setModel(listaUtilizatoriModel);
}
```



```

// butonul de conectare are drept rol instantierea listei cu
elemente scrise in campurile libere
    CONECTAREButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            Project p=new Project(
                textNume.getText(),
                textVarsta.getText(),
                textMarca.getText(),
                textSectia.getText(),
                textDefectiune.getText()
            );
            utilizator.add(p);
            AcutualizareLista();
        }
    });

//acest buton are rolul de a actualiza informatiile cu privire la
reclamatiiile deja existente in platforma
    PacientNouButon.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            int numardepacienti= list1.getSelectedIndex();
            if(numardepacienti>=0) {
                Project p=utilizator.get(numardepacienti);
                p.setNume(textNume.getText());
                p.setVarsta(textVarsta.getText());
                p.setMarca(textMarca.getText());
                p.setSectia(textSectia.getText());
                p.setDefectiune(textDefectiune.getText());
                AcutualizareLista();
            }
        }
    });

list1.addListSelectionListener(new ListSelectionListener()
{
    @Override
    public void valueChanged(ListSelectionEvent e) {
        int numardepacienti= list1.getSelectedIndex();
        if (numardepacienti>=0){
            Project p=utilizator.get(numardepacienti);
            textNume.setText(p.getNume());
            textVarsta.setText(p.getVarsta());
            textMarca.setText(p.getMarca());
            textSectia.setText(p.getSectia());
            textDefectiune.setText(p.getDefectiune());
            PacientNouButon.setEnabled(true);
        }
    }
}

```

```

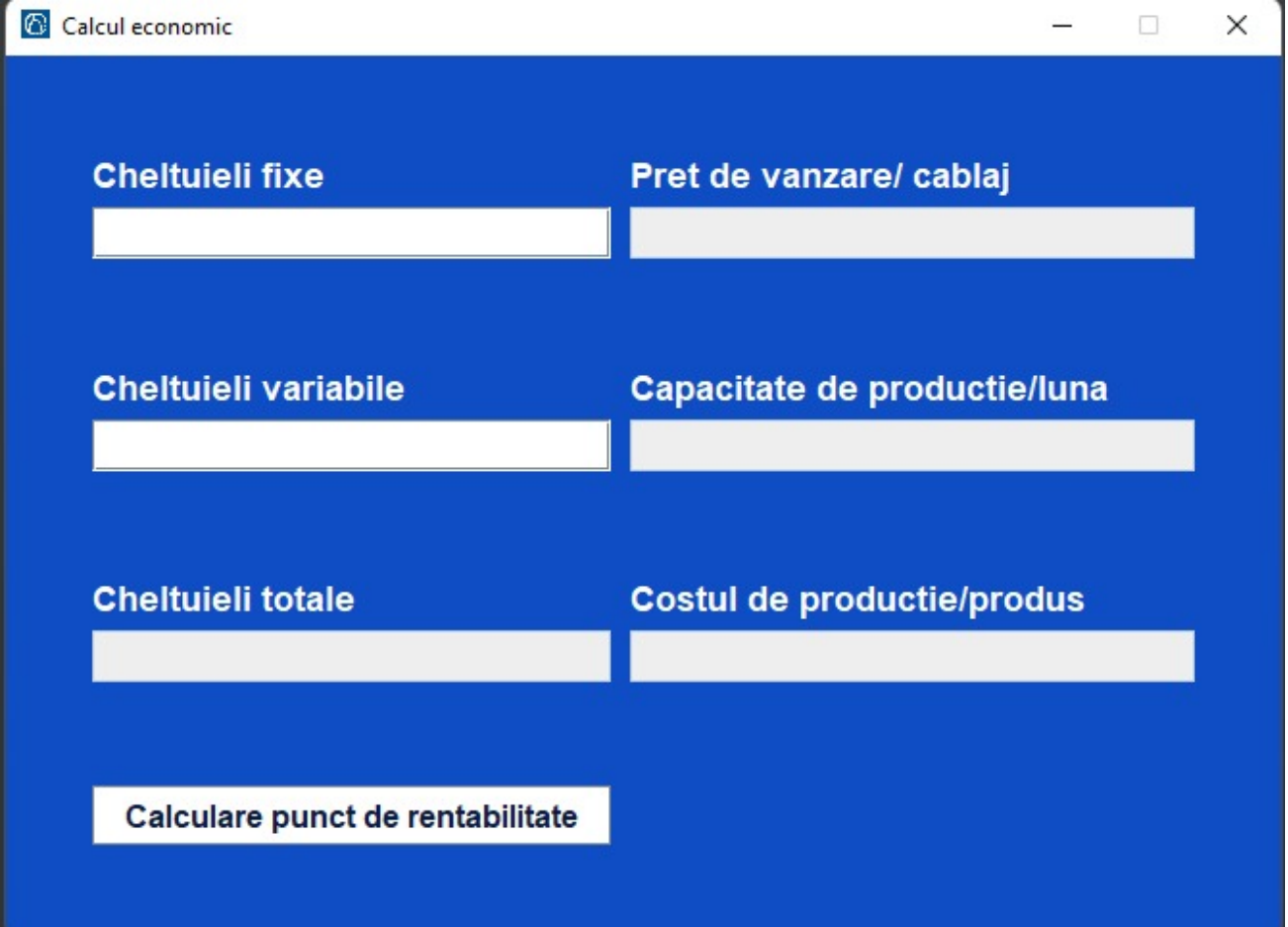
        else PacientNouButon.setEnabled(false);
    }
});
}
//functia dedicata actualizarii listei
public void AcutualizareLista(){
    listaUtilizatoriModel.removeAllElements();
    System.out.println("Sterge toti utilizatorii");
    for (Project p:utilizator) {
        System.out.println("Adauga persoana la lista");
        listaUtilizatoriModel.addElement("Nume:
"+p.getNum() +"; "+"Marca: "+p.getMarca()+"; "+"Defectiune:
"+p.getDefectiune());
    }
}
//functia dedicata adaugarii unei noi defectiuni
public void addProject (Project u){
    utilizator.add(u);
    AcutualizareLista();
}
public static void main(String[] args) {
    Defectiuni d=new Defectiuni();
    d.setVisible(true);
    Project u1= new Project("Ana Popa", "20", "70B23", "TM1",
    "Inserare gresita in pin");
    Project u2= new Project("Maria Popa", "23", "74C34", "TM2",
    "Capat terminal defect");
    Project u3= new Project("Mariana Vasile", "53", "84A23",
    "Magazie", "Lipsa cutie 234C");
    Project u4= new Project("Flavia Andron", "45", "89F35",
    "Foaming", "Gromete defect");
    Project u5= new Project("Carmen Moldovan", "23", "66D#4",
    "TM2", "Inserare incompleta terminal");
    Project u6= new Project("Vasilica Mihaila", "53", "84F33",
    "Foaming", "Gromete lipsa");
    d.addProject(u1);
    d.addProject(u2);
    d.addProject(u3);
    d.addProject(u4);
    d.addProject(u5);
    d.addProject(u6);
// Utilizare de ToString si StringBuffer
    String u=u1.toString();
    StringBuffer s = new StringBuffer(u);
    int l = s.length();
    System.out.println("Lungimea Numelui primului utilizator
="+ l);
}

```

}

2.4. DATE ECONOMICE

După apăsarea butonului 'Date economice' va apărea fereastra următoare. Utilizatorul introduce de la tastatură cheltuielile fixe și cheltuielile variabile și prin apăsarea butonului se vor inițializa prin intermediul unei interfețe prețul de vânzare, capacitatea de producție și costul de producție. În plus, tot prin apăsarea butonului, se vor calcula simultan costurile totale și punctul de rentabilitate. Astfel, într-un mod foarte simplu, se poate afla, în funcție de cheltuielile asimilate, la ce număr de produse vândute compania înregistrează un punct de rentabilitate, adică cheltuielile sunt egale cu veniturile.



Cheltuieli fixe	Pret de vanzare/ cablaj
<input type="text"/>	<input type="text"/>
Cheltuieli variabile	Capacitate de productie/luna
<input type="text"/>	<input type="text"/>
Cheltuieli totale	Costul de productie/produs
<input type="text"/>	<input type="text"/>
<input type="button" value="Calculare punct de rentabilitate"/>	

Cod sursă Date economice:

```
package Project;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.lang.invoke.StringConcatException;
interface E{
    public int getPretVanzare();
    public int getCapacitatedePrductie();
    public int getCostuldeProductie();
}
public class Economic extends JFrame {
    private JTextField Pretunitate;
    private JTextField Cprod;
    private JTextField Costprod;
    private JButton button1;
    private JPanel Panel;
    private JTextField Cf;
    private JTextField Cv;
    private JTextField Ct;
    private JLabel Prentab;

    Economic(){
        Calcul d= new Calcul() {
            @Override
            public int getCapacitatedePrductie() {
                return 0;
            }
        };
        this.setContentPane(this.Panel);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.pack();
        button1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                double cheltuieli_fixe =
Integer.parseInt(Cf.getText());
                double cheltuieli_variabile =
Integer.parseInt(Cv.getText());
                double cheltuieli_totale=
cheltuieli_fixe+cheltuieli_variabile;
                Ct.setText(String.valueOf(cheltuieli_totale)+"
lei");
            }
        });
    }
}
```

```

        Pretunitate.setText(String.valueOf(d.getPretVanzare())+"
lei");
Cprod.setText(String.valueOf(d.getCapacitatedeProductie())+" lei");
Costprod.setText(String.valueOf(d.getCostuldeProductie())+" lei");
        double
punct_rentabilitate=cheltuieli_fixe/(d.getPretVanzare()-d.getCostu
ldeProductie());
        Prentab.setText(String.valueOf(punct_rentabilitate)
+" produse vandute");
    }
    });
}
public static void main(String[] args) {
    JFrame frame=new JFrame("Hirschmann Automotive");
    frame.setContentPane(new Economic().rootPane);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.setSize(1000,500);
    frame.setResizable(false);
    frame.setVisible(true);
    ImageIcon imagine=new ImageIcon("logo.png");
    frame.setIconImage(imagine.getImage());
}
}

```

2.5. VÂNZĂRI

După apăsarea butonului 'Vânzări' va apărea fereastra următoare:

Vânzări

Selectați tipul de produs

Produs

Marca

Model

Versiune

Garanție

Preț

Cumpără

Calculează Rată

Șterge

Când este selectat un produs va apărea așa:



The screenshot shows a software window titled "Vanzari" (Sales) with a blue header and a red title bar. The main area is blue and contains a form for selecting a product. The form has the following fields and values:

Selectați tipul de produs	
Produs	Senzori sistem frânare
Marca	DELPHI
Model	Senzor ABS
Versiune	SS20005
Garanție	4 ani
Preț	66 LEI

Below the form are three buttons: "Cumpără", "Calculează Rată", and "Șterge".

În fereastra Vanzari, putem selecta unul dintre produsele noastre cu ajutorul comboBox-ului adăugat și să ne afișeze detalii tehnice despre produs. După acționarea butonului "Cumapara" ne va afișa un mesaj în care ne informează ce produs am cumpărat. Butonul "CalculeazaRata", ne calculează rata produsului selectat pe o perioadă de 24 luni și dobânda + prețul cu dobânda. Butonul "Șterge" ne afișează o fereastră în care ne întreabă dacă vrem sau nu să ștergem datele.

Cod sursă Vânzări:

```
package Project;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.awt.*;

public class Vanzari_Form extends JFrame {
    private JPanel Main;
    private JButton cautabtn;
    private JButton ratebtn;
    private JButton stergebtn;
    private JButton cumparabtn;
    private JTextField marcatxt;
    private JTextField modeltxt;
    private JTextField versiunetxt;
    private JTextField prettxt;
    private JComboBox produslst;
    private JTextField garantietxt;
    private ArrayList<Produs> produse;

    public Vanzari_Form() {
        this.setContentPane(this.Main );
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.pack();
        produse = new ArrayList<Produs>();
        produse.add(0, new Produs("Senzori sitem frânare",
"DELPHI", "Senzor ABS", "SS20005", "4",66));
        produse.add(1, new Produs("Senzori parcare", "VALEO",
"Senzori Față-Spate cu avertizare sonoră", "SQ632200", "3",327));
        produse.add(2, new Produs("Cablaje pentru oglinzi", "CNH",
"Cablaj conectare oglindă", "RG2997387", "1",152));
        produse.add(3, new Produs("Cutii de viteze automate", "VW
PASSAT", "DSG", "3C2 220864", "2",4500));
        produse.add(4, new Produs("Cutii de viteze manuale", "Audi
A4 B8", "OGS", "0b1301103", "5",1500));

        produslst.setSelectedItem(null);

        stergebtn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
```

```

        int result =
OptionPane.showConfirmDialog(null,"Ești sigur că vrei să ștergi
toate datele introduse?" , "Question",
        JOptionPane.YES_NO_OPTION ,
OptionPane.QUESTION_MESSAGE);

        if (result == JOptionPane.YES_OPTION) {
            marcatxt.setText("");
            modeltxt.setText("");
            versiunetxt.setText("");
            garantietxt.setText("");
            prettxt.setText("");

            JLabel label = new JLabel("Datele au fost
șterse!");

            JOptionPane.showMessageDialog(null, label,
"Atenție", JOptionPane.ERROR_MESSAGE);
        }
        else{
            JLabel label = new JLabel("Datele NU au fost
șterse!");

            JOptionPane.showMessageDialog(null, label,
"Atenție", JOptionPane.INFORMATION_MESSAGE);
        }

        System.out.println(result);
    }
});

ratebtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        int pret =
produse.get(produslst.getSelectedIndex()).getPret();
        int dobanda = 18;
        int nr_rate = 24;

        JLabel label = new JLabel("<html>Rată lunară (24
rate): " + String.format("%.2f", (1 + (double)dobanda / 100) *
pret / nr_rate) + " lei <br> Preț total: " + String.format("%.2f",
(1 + (double)dobanda / 100) * pret) + " lei <br> Dobânda: " +
dobanda + "%");

        JOptionPane.showMessageDialog(null, label, "Rată
lunară", JOptionPane.INFORMATION_MESSAGE);
    }
});

cumparabtn.addActionListener(new ActionListener() {

```



```

        @Override
        public void actionPerformed(ActionEvent e) {
            JLabel label = new JLabel("<html>Felicitări ! <br>
Ai achiziționat "
                                +
produse.get(produslst.getSelectedIndex()).getMarca() + " "
                                +
produse.get(produslst.getSelectedIndex()).getModel() + " "+"!");
            JOptionPane.showMessageDialog(null, label,
"Achiziționat", JOptionPane.INFORMATION_MESSAGE);
        }
    });

    produslst.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

marcatxt.setText(produse.get(produslst.getSelectedIndex()).getMarc
a());

modeltxt.setText(produse.get(produslst.getSelectedIndex()).getMode
l());

versiunetxt.setText(produse.get(produslst.getSelectedIndex()).getV
ersiune());

garantietxt.setText(produse.get(produslst.getSelectedIndex()).getG
arantie()+" ani");

prettxt.setText(Integer.toString(produse.get(produslst.getSelectedI
ndex()).getPret()+" LEI");
        }
    });
}

public static void main(String[] args){
    JFrame frame = new JFrame("Secțiunea Vânzări");
    frame.setContentPane(new Vanzari_Form().Main);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.setVisible(true);
    frame.setSize(800, 500);
    frame.setLocationRelativeTo(null);
    frame.setResizable(false);
    ImageIcon imagine=new ImageIcon("logo.png");
    frame.setIconImage(imagine.getImage());

}

```

}

2.6. ACHIZIȚII

După apăsarea butonului 'Achiziții' va apărea fereastra următoare:

Achiziții

Selecția noastră de produse:

☒ **8PIN MALE CONNECTOR** Selectează:

☐ **RIDE BY WIRE** Selectează:

☐ **HPS40-2 MALE PLUG** Selectează:

☐ **BWM E-GAS** Selectează:

Calculează

Subtotal

Total (+ TVA)

Adaugă în coș **Anulează**

În fereastra 'Achiziții' se poate calcula costul total al produselor cu sau fără tva. Prin intermediul celor 4 radio-buttons putem selecta produsele pe care utilizatorul dorește să le achiziționeze, iar prin intermediul celor 4 combobox-uri putem selecta tipul produselor. Butonul 'Calculează' rezolvă calculul produselor selectate, rezultatul fără TVA fiind afișat în câmpul 'Subtotal', iar cel cu TVA în câmpul 'Total + TVA'. Prin acționarea butonului 'Adaugă în coș' va apărea un mesaj de informare cum că produsele au fost adăugate cu succes în coș, iar prin acționarea butonului 'Anulează' se șterg toate datele calculate.

Când este selectată una sau mai multe variabile dintre cele 4 radio-buttons va apărea astfel:

Achizitii

Selecția noastră de produse:

☒ 8PIN MALE CONNECTOR Selectează:

☐ RIDE BY WIRE HEATING GRIP

☐ HPS40-2 MALE PLUG Selectează:

☐ BWM E-GAS SPEED CONTROL + HEATING

Calculează

Subtotal **225.0**

Total (+ TVA) **267.75**

Adaugă în coș **Anulează**

Cod sursă Achiziții:

```
package Project;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.ImageIcon;

public class Achizitii extends JFrame{
    private JPanel rootPanel;
    private JRadioButton btn1;
    private JRadioButton btn2;
    private JRadioButton btn3;
    private JRadioButton btn4;
    private JComboBox comboBox1;
    private JComboBox comboBox2;
```

```

private JComboBox comboBox3;
private JComboBox comboBox4;
private JTextField txtSubT;
private JTextField txtTotal;
private JButton btnCalcul;
private JButton btnAnulare;
private JButton btnAdaugare;

Achizitii() {
    this.setContentPane(this.rootPanel );
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.pack();
    btnCalcul.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            double total;
            double subtotal=0;

            if (btn1.isSelected()) {
                subtotal =subtotal + 15;
            }if (btn2.isSelected()){
                subtotal=subtotal + 125;
            }if(btn3.isSelected()){
                subtotal=subtotal + 50;
            }if (btn4.isSelected()) {
                subtotal = subtotal + 100;
            }

            txtSubT.setText(Double.toString (subtotal));
            subtotal = Double.parseDouble(txtSubT.getText());
            total=subtotal+.19*subtotal;
            txtTotal.setText(Double.toString(total));
        }
    });
    btnAnulare.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if(e.getSource()==btnAnulare);
            txtSubT.setText("");
            txtTotal.setText("");
        }
    });
    btnAdaugare.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            int rezultat =
JOptionPane.showConfirmDialog(null,"Produsele au fost adăugate în
coş cu succes!" , "Reuşit!",

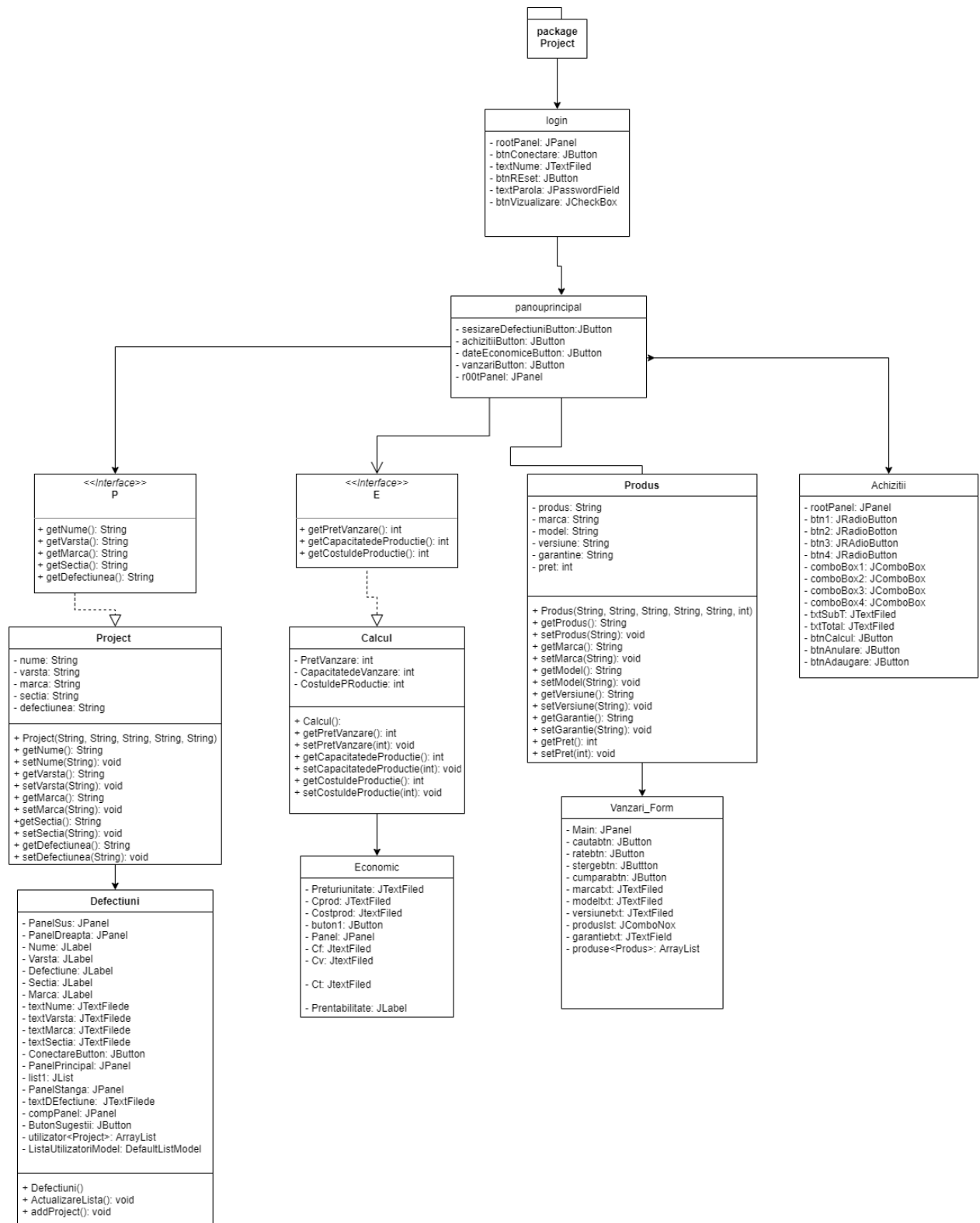
```

```
                JOptionPane.DEFAULT_OPTION ,
JOptionPane.INFORMATION_MESSAGE);
            }
        });
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Hirschmann Automotive");
        frame.setContentPane(new Achizitii().rootPanel);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setSize(700, 500);
        frame.setResizable(true);
        frame.setVisible(true);
        Color culoare =new Color(15,77,196);
        frame.getContentPane().setBackground(culoare);
        ImageIcon imagine = new ImageIcon("logo.png");
        frame.setIconImage(imagine.getImage());

    }
}
```

3. Diagrama UML:



4. Concluzie

În concluzie, prin acest proiect, am reușit să dezvoltăm, prin îmbinarea noțiunilor de bază a programării orientate pe obiect cu noțiunile din implementarea unei interfețe grafice în java, o interfață care poate facilita diverse operații economice și logistice dintr-un mediu de producție precum o companie de la noi din județ. Credem că acest proiect a fost un mod eficient prin care fiecare membru a reușit să își aprofundeze informațiile cu privire la programarea orientată obiect. Acest proces de aprofundare a fost eficientizat mai ales de munca de echipă și orele petrecute alături de membrii echipei.