

23. ANEXA. APLICAȚIA MAGAZIN

În această anexă prezentăm diagrama de clase de proiectare (Figura 1) a proiectului Magazin a cărui descriere a problemei este dată în capitolul 4 și programul complet.

```
public class SpecificatieProdus implements Comparable<SpecificatieProdus>{
    private String denumire;
    private double pretUnitar;
    public SpecificatieProdus(String denumire, double pret){
        this.denumire=denumire;
        this.pretUnitar=pret;
    }
    public String getDenumire(){
        return denumire;
    }
    public double getPret(){
        return pretUnitar;
    }
    public int compareTo(SpecificatieProdus o){
        if (o==null) return -1000;
        return denumire.compareTo(o.getDenumire());
    }
    public String toString(){
        return denumire+" "+pretUnitar;
    }
}

public class Articol{
    private SpecificatieProdus produs;
    private int cantitate;
    public Articol(String d, double p, int c){
        produs=new SpecificatieProdus(d,p);
        cantitate=c;
    }
    public Articol(SpecificatieProdus p, int c){
        produs=p;
        cantitate=c;
    }
    public SpecificatieProdus getProdus(){
        return produs;
    }
    public int getCantitate(){
        return cantitate;
    }
    public double calculeazaCost(){
        return produs.getPret()*cantitate;
    }
}

public interface Client{
    void alegeArticol(String den, double pret, int c);
}

import java.util.*;
public class Vanzare extends ArrayList<Articol>{
    private Client c;
    public static int tva=20;
    private double total;
    public Vanzare(Client c){
        this.c=c;
    }
}
```

```

public Client getClient(){
    return c;
}
public void addArticol(String d, double p, int cant){
    super.add(new Articol(d,p,cant));
}
public double calculeazaTotal(){
    ListIterator<Articol> li=listIterator();
    Articol a;
    while(li.hasNext()) {
        a=li.next();
        total+=a.calculeazaCost();
    }
    Calendar cl=Calendar.getInstance();
    if(cl.get(Calendar.DAY_OF_WEEK)>=2&&cl.get(Calendar.DAY_OF_WEEK)<=6)
        total-=5*total/100.0;
    return total+calculeazaTVA();
}
public ArrayList<Articol> getArticole(){
    return this;
}
public double calculeazaTVA(){
    return total*tva/100.0;
}
}

```

```

import java.io.*;
public class Persoana implements Client,Serializable{
    private String nume, prenume, adresa;
    private Vanzare v;
    public Persoana(String n, String p){
        this(n,p,"");
    }
    public Persoana(String n, String p, String a){
        nume=n;
        prenume=p;
        adresa=a;
    }
    public String getNume(){
        return nume;
    }
    public String getPrenume(){
        return prenume;
    }
    public String getAdresa(){
        return adresa;
    }
    public void alegeArticol(String den, double p, int c){
        if (v==null) v=new Vanzare(this);
        v.addArticol(den, p, c);
    }
}

```

```

public class Magazin{
    private Persoana proprietar;
    private int oraD, oraI;
    public Magazin(Persoana p){
        proprietar=p;
        oraD=10;
        oraI=18;
    }
}

```

```

public void setOraD(int ora){
    oraD=ora;
}
public void setOraI(int ora){
    oraI=ora;
}
public int getOraD(){
    return oraD;
}
public int getOraI(){
    return oraI;
}
public String getProprietar(){
    return proprietar.getNume()+" "+ proprietar.getPrenume();
}
}

import java.util.*;
import java.text.*;
public class Factura{
    private static int id=1;
    private int numar;
    private Calendar data;
    private Vanzare v;
    private Client c;
    private ArrayList<Articol> articole;
    public Factura(Vanzare v){
        numar=id++;
        data=Calendar.getInstance();
        this.v=v;
        c=v.getClient();
        articole=v.getArticole();
    }
    public String toString(){
        String sir="Factura: "+ numar+"\r\n";
        sir+="Data emiterii: "+
DateFormat.getDateInstance(DateFormat.LONG).format(data.getTime())+"\r\n";
        sir+="-----
        \r\n\r\n";
        sir+="Cumparator: "+((Persoana)c).getNume()+" "+ ((Persoana)c).getPrenume() +
        "\r\n";
        sir+="Adresa: "+((Persoana)c).getAdresa()+"\r\n";
        sir+="-----
        \r\n\r\n";
        sir+=String.format("%-30s","Denumire")+ String.format("%-15s", "Pret unitar")+
        String.format("%-20s","Cantitate")+String.format("%-
20s","Pret/articol")+
        "\r\n";
        Iterator<Articol> it=articole.iterator();
        Articol el;
        double t=v.calculeazaTotal();
        while(it.hasNext()){
            el=it.next();
            sir+=String.format("%-30s",el.getProdus().getDenumire())+
                String.format("%-15.2f",el.getProdus().getPret())+
                String.format("%-20d",el.getCantitate())+
                String.format("%-20.2f",el.calculeazaCost())+"\r\n";
        }
    }
}

```

```

sir+="-----
    \r\n\r\n";
sir+="TOTAL\t\t"+String.format("%-20.2f",t)+" lei \r\n";
sir+="Din care TVA 20%\t"+String.format("%-20.2f",v.calculeazaTVA())+
    " lei \r\n";
sir+="\r\n\r\nTOTAL GENERAL\t"+String.format("%-20.2f",t)+" lei \r\n";
return sir;
}
public int getNumar(){
    return numar;
}
}

import java.util.*;
import java.io.*;
public class ListaFacturi{
    private Properties tabel;
    public ListaFacturi(){
        tabel=new Properties();
    }
    public void adaugaFactura(Factura f){
        tabel.put(String.valueOf(f.getNumar()),f.toString()+"|");
    }
    public void descarca(){
        try{
            FileOutputStream out=new FileOutputStream("Arhiva.dat", true);
            tabel.store(out,"|");
            out.close();
        }catch (IOException io){}
    }
}

public interface OperatiiObiecte{
    public void scrieObiect(Object o);
    public Object citesteObiect(String cheie);
}

import java.io.*;
import java.util.*;
public class OperatiiProduce implements OperatiiObiecte{
    private PrintWriter pw;
    private BufferedReader br;
    public void scrieObiect(Object o){
        if (!(o instanceof SpecificatieProduce)) {
            System.out.println("instanta invalida");
            return;
        }
        else {
            try{
                if (pw==null) pw=new PrintWriter(new FileWriter("produce.txt"));
                SpecificatieProduce p=(SpecificatieProduce)o;
                pw.println(p.getDenumire()+"_"+p.getPret());
                pw.flush();
            }catch(IOException io){io.printStackTrace();}
        }
    }
}

```

```

public Object citesteObiect(String cheie){
    SpecificatieProdus sp=null;
    try{
        if (br==null) br=new BufferedReader(new FileReader("produse.txt"));
        StringTokenizer st;
        String linie;
        while((linie=br.readLine())!=null){
            st=new StringTokenizer(linie, "_");
            if(st.nextToken().equals(cheie)) {
                sp=new SpecificatieProdus(cheie, Double.parseDouble(st.nextToken()));

                break;
            }
        }
    }catch(IOException io){io.printStackTrace();}
    return sp;
}

public void inchide(){
    pw.close();
}
}

import java.io.*;
import java.util.*;
public class OperatiiClient implements OperatiiObiecte{
    private ObjectOutputStream oos;
    private ObjectInputStream ois;
    private String[] nume;
    public OperatiiClient(){
        File f=new File(".");
        nume=f.list(new Filtru());
    }
    public void scrieObiect(Object o){
        if (!(o instanceof Persoana)) {
            System.out.println("instanta invalida");
            return;
        }
        else {
            try{
                oos=new ObjectOutputStream(
                    new FileOutputStream("persoana"+((Persoana)o).hashCode()+".txt", true));
                oos.writeObject(o);
            }catch(IOException io){io.printStackTrace();}
            catch(Exception e){}
        }
    }
    public Object citesteObiect(String cheie){
        if (nume.length==0) return null;
        Persoana c;
        try{
            for (int i=0; i<nume.length;i++){
                ois=new ObjectInputStream(new FileInputStream(nume[i]));
                c=(Persoana)ois.readObject();
                if (cheie.equals(c.getNume()+" "+c.getPrenume()+" "+c.getAdresa()))
                    return c;
            }
        }
    }
}

```

```

        }catch(IOException io){io.printStackTrace();}
        catch(ClassNotFoundException io){io.printStackTrace();}
        return null;
    }
    public void inchide(){
        try{
            if (oos!=null) oos.close();
        }catch(IOException io){io.printStackTrace();}
    }
}

import java.io.*;
public class Filtru implements FilenameFilter{
    public boolean accept(File f, String nume){
        if(nume.startsWith("persoana") && nume.endsWith(".txt")) return true;
        return false;
    }
}

import java.util.*;
public class ListaClienti{
    private HashMap hm;
    private OperatiiObiecte oo;
    public ListaClienti(){
        hm=new HashMap();
        oo=new OperatiiClient();
    }
    public void adaugaClient(Persoana c){
        if (!hm.containsValue(c))
            hm.put(c.getNume()+" "+c.getPrenume()+" "+c.getAdresa(), c);
    }
    public Persoana getClient(String cheie){
        Persoana c=null;
        if(hm.containsKey(cheie)) c=(Persoana)hm.get(cheie);
        else c=(Persoana)oo.citesteObiect(cheie);
        return c;
    }
    public void descarca(){
        Collection s=hm.values();
        Iterator i=s.iterator();
        while(i.hasNext()) oo.scrieObiect(i.next());
        ((OperatiiClient)oo).inchide();
        hm.clear();
    }
}

import java.util.*;
import java.io.*;
public class Catalog extends TreeSet<SpecificatieProdus>{
    private static Catalog c;
    private OperatiiProduse osp;
    private BufferedReader br;
    private int dim=0;
    private Catalog(){}
    public static Catalog getInstanta(){
        if(c==null) c=new Catalog();
        return c;
    }
}

```

```

public boolean add(SpecificatieProdus sp){
    return super.add(sp);
}
public SpecificatieProdus getProdus(String nume){
    Iterator<SpecificatieProdus> it=iterator();
    SpecificatieProdus sp=null;
    while(it.hasNext()){
        sp=it.next();
        if(sp.getDenumire().equals(nume)) return sp;
    }
    osp=new OperatiiProduse();
    sp=(SpecificatieProdus)osp.citesteObiect(nume);
    if(sp!=null) add(sp);
    return sp;
}
public String[] incarca(){
    String[] rez=null;
    try{
        br=new BufferedReader(new FileReader("produse.txt"));
        dim=0;
        while(br.readLine()!=null) dim++;
        br.close();
        rez=new String[dim];
        osp=new OperatiiProduse();
        SpecificatieProdus sp=null;
        int i=0;
        br=new BufferedReader(new FileReader("produse.txt"));
        String line;
        while((line=br.readLine())!=null) {
            rez[i]=line.substring(0, line.indexOf("_"));
            sp=(SpecificatieProdus)osp.citesteObiect(rez[i]);
            add(sp);
            i++;
        }
        br.close();
    }catch(IOException io){io.printStackTrace();}
    return rez;
}
public void descarca(){
    osp=new OperatiiProduse();
    Iterator<SpecificatieProdus> it=iterator();
    while(it.hasNext()) osp.scrieObiect(it.next());
    osp.inchide();
}
public int getDimensiune(){
    return Math.max(dim, size());
}
}

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class ClientPanel extends JPanel implements ActionListener{
    private JLabel numeL,prenumeL,adresaL;
    private JTextField numeT,prenumeT,adresaT;
    private JButton submit;
    private GridBagLayout gb;
    private GridBagConstraints gbc;

```

```

private ListaClienti lista;
private Persoana c;
private Vanzare vanzare;
private ComandaFrame f;
public ClientPanel(ComandaFrame f, ListaClienti lista){
    super();
    this.f=f;
    this.lista=lista;
    gb=new GridBagLayout();
    gbc=new GridBagConstraints();
    gbc.fill=GridBagConstraints.HORIZONTAL;
    setLayout(gb);
    setPreferredSize(new Dimension(400,100));
    gbc.anchor=GridBagConstraints.EAST;
    numeL=new JLabel("Nume");
    addComponent(numeL,0,0,1,1);
    numeT=new JTextField(10);
    numeT.addActionListener(this);
    addComponent(numeT,0,1,1,1);
    prenumeL=new JLabel("Prenume");
    numeT.addActionListener(this);
    addComponent(prenumeL,1,0,1,1);
    prenumeT=new JTextField(10);
    prenumeT.addActionListener(this);
    addComponent(prenumeT,1,1,1,1);
    adresaL=new JLabel("Adresa");
    addComponent(adresaL,2,0,1,1);
    adresaT=new JTextField(20);
    addComponent(adresaT,2,1,1,2);
    submit=new JButton("Submit");
    submit.addActionListener(this);
    addComponent(submit,1,2,1,1);
}
public void addComponent(Component c, int linie,int coloana, int lat, int
                           inaltime){
    gbc.gridx=coloana;
    gbc.gridy=linie;
    gbc.gridwidth=lat;
    gbc.gridheight=inaltime;
    gb.setConstraints(c,gbc);
    add(c);
}
public Vanzare getVanzare(){
    return vanzare;
}
public void actionPerformed(ActionEvent e){
    if(e.getSource()==numeT){
        prenumeT.requestFocus();
    }
    if(e.getSource()==prenumeT){
        adresaT.requestFocus();
    }
    if (e.getSource()==submit){
        c=lista.getClient(numeT.getText().trim()+" "+prenumeT.getText().trim()+
            " "+adresaT.getText().trim());
    }
}

```



```

        if (c==null){
            c=new Persoana(numet.getText().trim(),prenumeT.getText().trim(),
                adresaT.getText().trim());
            lista.adaugaClient(c);
            lista.descarca();
            JOptionPane.showMessageDialog(null, "A fost inregistrat un client nou!",
                "Information",
JOptionPane.INFORMATION_MESSAGE);
        }
        vanzare=new Vanzare(c);
        f.add.setEnabled(true);f.add.repaint();
    }
}
public void curata(){
    numet.setText("");
    prenumeT.setText("");
    adresaT.setText("");
    vanzare=null;
}
}

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class CatalogPanel extends JPanel implements ItemListener{
    private JComboBox listaP;
    private int indexProdus;
    public CatalogPanel(Catalog lista){
        super();
        String[] elemente=lista.incarca();
        listaP=new JComboBox(elemente);
        listaP.addItemListener(this);
    }
    public void itemStateChanged(ItemEvent e){
        indexProdus=listaP.getSelectedIndex();
    }
    public int getIndexProdus(){
        return indexProdus;
    }
    public JComboBox getComboBox(){
        return listaP;
    }
}

import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.*;
public class TabelPanel extends JPanel{
    private GridBagLayout gb;
    private GridBagConstraints gbc;
    private JTable tabel;
    private Catalog c;
    public TabelPanel(Catalog c){
        this.c=c;
        gb=new GridBagLayout();
        gbc=new GridBagConstraints();
        gbc.fill=GridBagConstraints.HORIZONTAL;

```

```

        gbc.anchor=GridBagConstraints.NORTH;
        setLayout(gb);
        setSize(new Dimension(425,100));
        tabel=new JTable(c.getDimensiune()+1,4);
        tabel.setValueAt(new String("Numar"),0,0);
        tabel.setValueAt(new String("Denumire"),0,1);
        tabel.setValueAt(new String("Cantitate"),0,2);
        tabel.setValueAt(new String("Pret unitar"),0,3);
        addComponent(tabel,0,0,4,c.getDimensiune()+1);
    }
    public void addComponent(Component c, int linie, int coloana, int lat,
        int inaltime){

        gbc.gridx=coloana;
        gbc.gridy=linie;
        gbc.gridwidth=lat;
        gbc.gridheight=inaltime;
        gb.setConstraints(c,gbc);
        add(c);
    }
    public JTable getTabel(){
        return tabel;
    }
    public void curata(){
        for (int i=1; i<=c.getDimensiune(); i++)
            for (int j=0; j<=3; j++)
                tabel.setValueAt("",i,j);
    }
}

import java.awt.*;
import javax.swing.*;
public class DateMagazin extends JFrame{
    public DateMagazin(){
        super("Date despre magazinul Tomis!");
        JPanel p=new JPanel();
        Magazin m=new Magazin(new Persoana("Ionescu", "Pop"));
        JLabel ta=new JLabel("Proprietar: "+m.getProprietar());
        JLabel tal=new JLabel("Orar: "+m.getOraD()+" :00 - "+m.getOraI()+" :00");
        p.setLayout(new GridLayout(2,1, 10,0));
        p.add(ta);
        p.add(tal);
        add(p);
    }
}

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class StocFrame extends JFrame{
    private JButton adauga, cancel;
    private JTextField tf1, tf2;
    private ControllerStocFrame c;
    private Catalog cat;
    public StocFrame(){
        super("Administrazione stoc");
        JPanel p=new JPanel();
        p.setLayout(new GridLayout(3, 2, 10, 10));
        p.add(new Label("Nume produs"));

```

```

        tf1=new JTextField(10);
        p.add(tf1);
        p.add(new Label("Pret unitar"));
        tf2=new JTextField(10);
        p.add(tf2);
        c=new ControllerStocFrame();
        adauga=new JButton("Adauga la stoc");
        adauga.addActionListener(c);
        p.add(adauga);
        cancel=new JButton("Cancel");
        cancel.addActionListener(c);
        p.add(cancel);
        add(p);
        cat=Catalog.getInstanta();
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e){
                cat.descarca();
                dispose();
            }
        });
    }
}

class ControllerStocFrame implements ActionListener{
    public void actionPerformed(ActionEvent e){
        String et=e.getActionCommand();
        if (et.equals("Adauga la stoc")){
            cat.add(new SpecificatieProdus(tf1.getText(),
                                           Double.parseDouble(tf2.getText())));

            tf1.setText(null);
            tf2.setText(null);
        }
        else {
            tf1.setText(null);
            tf2.setText(null);
        }
    }
}
}

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.awt.print.*;
public class FacturaFrame extends JFrame{
    private JButton print, cancel, printF, arhivare;
    private JTextArea arie;
    private JPanel p;
    private ControllerFacturaFrame c;
    private Factura factura;
    private ListaFacturi facturi;
    private GridBagLayout gb;
    private GridBagConstraints gbc;
    public FacturaFrame(Factura f){
        super("Factura");
        factura=f;
        gb=new GridBagLayout();
        gbc=new GridBagConstraints();
        gbc.fill=GridBagConstraints.HORIZONTAL;

```

```

gbc.anchor=GridBagConstraints.NORTH;
p=new Panel();
p.setLayout(gb);
arie=new JTextArea(factura.toString(),20, 50);
arie.setEditable(false);
addComponent(arie,0,0,2,5);
c=new ControllerFacturaFrame();
print=new Button("Print");
addComponent(print,0,6,1,1);
print.addActionListener(c);
printF=new Button("PrintToFile");
addComponent(printF,1,6,1,1);
printF.addActionListener(c);
arhivare=new Button("Arhivare");
addComponent(arhivare,2,6,1,1);
arhivare.addActionListener(c);
cancel=new Button("Cancel");
addComponent(cancel,3,6,1,1);
cancel.addActionListener(c);
add(p);
facturi=new ListaFacturi();
addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e){
    facturi.descarca();
    dispose();
}
});
}
void addComponent(Component c, int linie,int col, int lat, int inal){
    gbc.gridx=col;
    gbc.gridy=linie;
    gbc.gridwidth=lat;
    gbc.gridheight=inal;
    gb.setConstraints(c,gbc);
    p.add(c);
}
class ControllerFacturaFrame implements ActionListener{
    private PrintWriter out;
    public void actionPerformed(ActionEvent e){
        if (e.getSource()==print){
            PrinterJob imprimanta=PrinterJob.getPrinterJob();
            Book bk=new Book();
            bk.append(new PanouContinut(),imprimanta.defaultPage());
            imprimanta.setPageable(bk);
            if(imprimanta.printDialog()){
                try{
                    imprimanta.print();
                }catch (PrinterException pe){
                    arie.append("Imprimanta nu exista");
                    arie.repaint();
                }
                catch(ArrayIndexOutOfBoundsException ae){
                    System.out.println("Ce se printeaza???");
                }
            }
        }
    }
}

```

```

else if (e.getSource()==printF){
    try{
        out=new PrintWriter(new FileOutputStream("factura.dat"));
        out.write(factura.toString());
        out.flush();
        out.close();
    }catch(FileNotFoundException fe){}
    catch(IOException ioe){}
}
else if (e.getSource()==arhivare){
    facturi.adaugaFactura(factura);
}
else if (e.getSource()==cancel) setVisible(false);
}
}

class PanouContinut extends JPanel implements Printable{
    private BufferedReader br;
    private String sir="";
    public int print(Graphics g, PageFormat pf,int pageIndex)
                                                throws PrinterException{
        g.setColor(Color.black);
        try{
            StringReader continut=new StringReader(arie.getText());
            br=new BufferedReader(continut);
            int i=0;
            while((sir=br.readLine())!=null) {
                if (sir.length()==0) sir=" ";
                g.drawString(sir,100,100+i);i+= 20;
            }
        }catch(IOException io){}
        catch (IllegalArgumentException ie){}
        return Printable.PAGE_EXISTS;
    }
}

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class ComandaFrame extends JFrame implements ActionListener{
    private JPanel panel;
    private JLabel produsL,cantitateL;
    private JComboBox listaP;
    private JTextField cantitateT;
    JButton add, execute;
    private Articol articol;
    private Factura factura;
    private Client c;
    private GridBagLayout gb;
    private GridBagConstraints gbc;
    private TabelPanel panelTabel;
    private JTable tabel;
    private ClientPanel cp;
    private int numar=0;
    private Catalog lista;
    private Vanzare vanzare;
    public ComandaFrame(Catalog cat, ListaClienti lc){

```

```

super("Comandati produse din acest magazin");
lista=cat;
cp=new ClientPanel(this, lc);
panel=new JPanel();
gb=new GridBagLayout();
panel.setLayout(gb);
gbc=new GridBagConstraints();
gbc.fill=GridBagConstraints.BOTH;
gbc.anchor=GridBagConstraints.NORTH;
produsL=new JLabel("Produs");
addComponent(produsL,0,0,1,1);
cantitateL=new JLabel("Cantitate");
addComponent(cantitateL,0,1,1,1);
listaP=new CatalogPanel(cat).getComboBox();
addComponent(listaP,1,0,1,1);
cantitateT=new JTextField(10);
addComponent(cantitateT,1,1,1,1);
add=new JButton("Adauga");
add.addActionListener(this);
if (c==null) add.setEnabled(false);
addComponent(add,1,2,1,1);
execute=new JButton("Executa");
execute.addActionListener(this);
execute.setEnabled(false);
addComponent(execute,1,3,1,1);
add(cp, BorderLayout.NORTH);
add(panel);
panelTabel=new TabelPanel(cat);
tabel=panelTabel.getTabel();
add(panelTabel, BorderLayout.SOUTH);
setSize(450,430);
this.setResizable(false);
setLocation(250,150);
addWindowListener(new WindowAdapter(){
    public void windowClosing(WindowEvent e){
        dispose();
    }
});
}
private void addComponent(Component c, int linie,int col, int lat, int inal){
    gbc.gridx=col;
    gbc.gridy=linie;
    gbc.gridwidth=lat;
    gbc.gridheight=inal;
    gb.setConstraints(c,gbc);
    panel.add(c);
}
public void actionPerformed(ActionEvent e){
    if (e.getSource()==add){
        vanzare=cp.getVanzare();
        if(vanzare!=null){
            try{
                Integer.parseInt(cantitateT.getText());
                execute.setEnabled(true);execute.repaint();
                numar++;
                SpecificatieProdus p=lista.getProdus((String)listaP.getSelectedItem());
                vanzare.addArticol(p.getDenumire(), p.getPret(),
                    Integer.parseInt(cantitateT.getText()));
            }
        }
    }
}

```

```

        tabel.setValueAt(""+numar, numar, 0);
        tabel.setValueAt(p.getDenumire(), numar, 1);
        tabel.setValueAt(cantitateT.getText(), numar, 2);
        tabel.setValueAt(String.valueOf(p.getPret()), numar, 3);
        panelTabel.repaint();
        cantitateT.setText("");
    }
    catch(NumberFormatException nf) {JOptionPane.showMessageDialog(this,
        "Eroare!\n Trebuie sa introduceti un numar pentru cantitate",
        "Reminder", JOptionPane.WARNING_MESSAGE);
        cantitateT.setText(""); numar--;}
    }
    else {JOptionPane.showMessageDialog(null, "Mai intai clientul trebuie sa fie
        inregistrat!", "Reminder", JOptionPane.WARNING_MESSAGE);
        }
    }
    else if (e.getSource()==execute){
        factura=new Factura(vanzare);
        JFrame frame=new FacturaFrame(factura);
        frame.setSize(450,430);
        frame.setLocation(250,100);
        frame.setVisible(true);
        add.setEnabled(false);
        add.repaint();
        cp.curata();
        numar=0;
        panelTabel.curata();
        cantitateT.setText("");
        pack();
    }
}

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class FereastraPrincipala extends JFrame{
    private JButton b1, b2, b3;
    private ControllerButoane cb;
    public FereastraPrincipala(){
        super("Bine ati venit la magazinul Tomis!");
        JPanel p=new JPanel();
        cb=new ControllerButoane();
        b1=new JButton("Date despre magazin");
        p.add(b1);
        b1.addActionListener(cb);
        b2=new JButton("Comandati produse");
        p.add(b2);
        b2.addActionListener(cb);
        b3=new JButton("Administrare stoc");
        p.add(b3);
        b3.addActionListener(cb);
        add(p);
    }
    private class ControllerButoane implements ActionListener{
        private JFrame f, g, h;
        public void actionPerformed(ActionEvent e){
            if (e.getSource()==b1){

```

```

        if (f==null) f=new DateMagazin();
        f.setSize(150,100);
        f.setVisible(true);
    }
    if (e.getSource()==b2){
        g=new ComandaFrame(Catalog.getInstanta(), new ListaClienti());
        g.pack();
        g.setVisible(true);
    }
    if (e.getSource()==b3){
        h=new StocFrame();
        h.pack();
        h.setVisible(true);
    }
}
}
public static void main(String[] args){
    JFrame f=new FereastraPrincipala();
    f.setSize(500,75);
    f.setVisible(true);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}

```

Programul 1. Codul aplicației proiectului Magazin

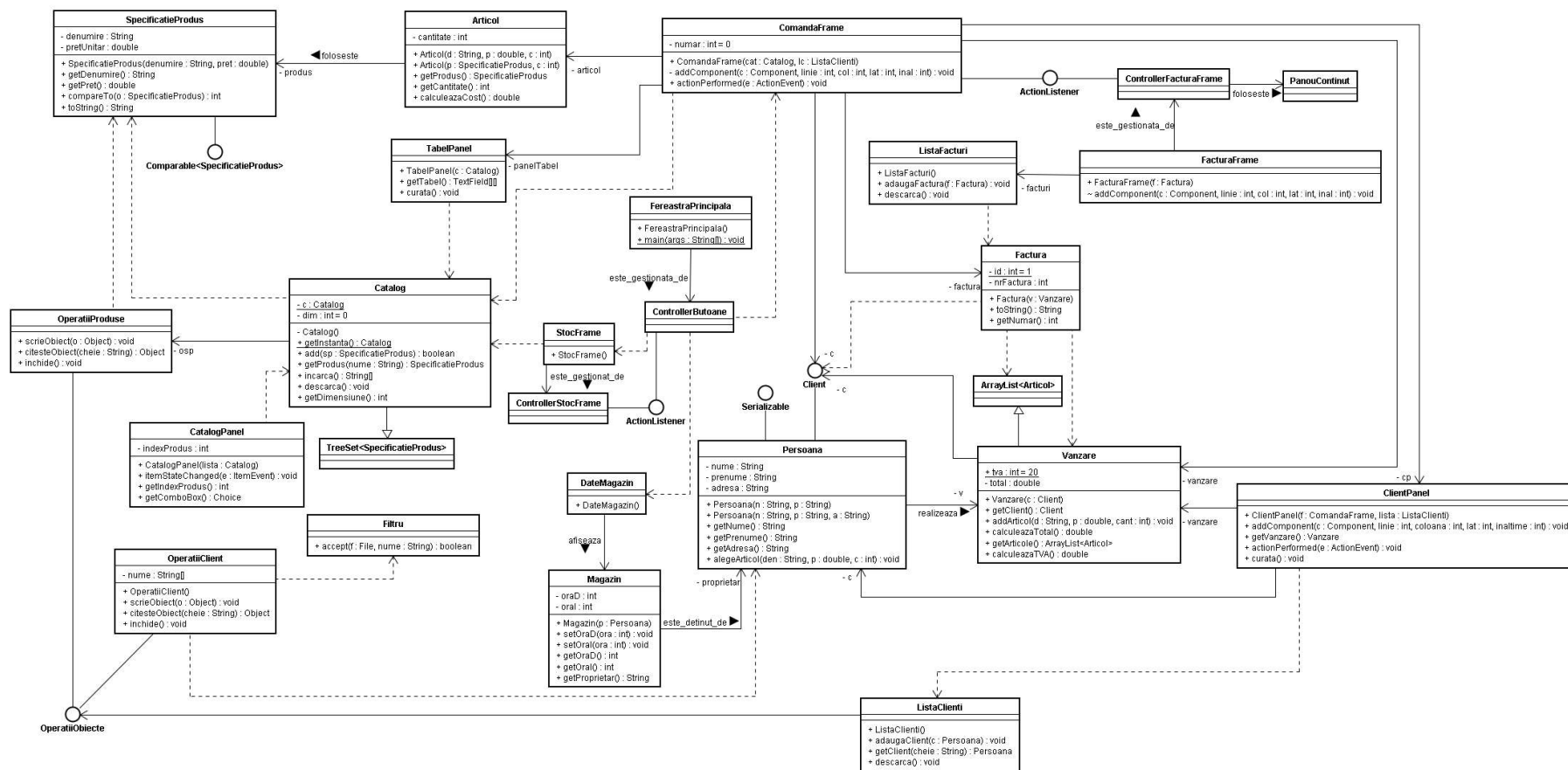


Figura 1. Diagrama de clase de proiectare a programului proiectului Magazin