

Lucrare de laborator nr. 4

Șabloanele arhitecturale MVC și MVVM

4.1 Scopul lucrării

În cadrul acestei lucrări de laborator se urmărește consolidarea abilităților în utilizarea eficientă a șabloanelor arhitecturale MVC (Model-View-Controller) și MVVM (Model-View-ViewModel).

4.2 Exerciții rezolvate

Exercițiu 4.1. Dezvoltați, utilizând șablonul arhitectural MVC, o aplicație soft care să permită determinarea produsului cartezian a două mulțimi de numere întregi.

Soluție:

Soluția prezentată utilizează varianta șablonului arhitectural MVC în care pachetele *View* și *Model* sunt decuplate (figura 4.1).

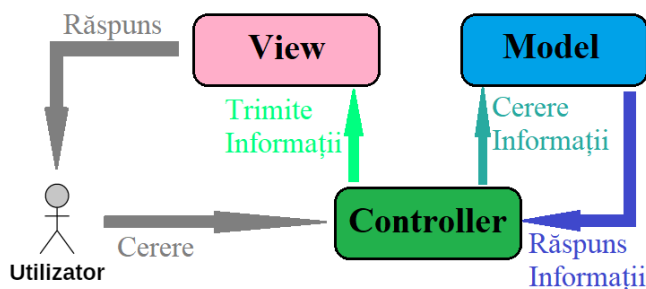


Figura 4.1. Arhitectură MVC

Pachetul *Model* cuprinde 2 clase: *Multime* și *ProdusCartezian* (figura 4.2). Între cele 2 clase există relația de compunere, un obiect de tip *ProdusCartezian* fiind compus din 2 obiecte de tip *Multime*.

Pachetul *View* conține o clasă (*VProdusCartezian*) care are doar responsabilitatea de a gestiona interacțiunile cu utilizatorul.

Pachetul *Controller* conține o clasă *CProdusCartezian* care are 2 atribute:

- ❖ Un obiect ce reprezintă modelul utilizat în rezolvarea problemei;
- ❖ Un obiect ce reprezintă interfața grafică a aplicației.

Clasa *CProdusCartezian* are 2 responsabilități:

- ❖ Gestionarea evenimentelor;
- ❖ Implementarea opțiunilor aplicației.

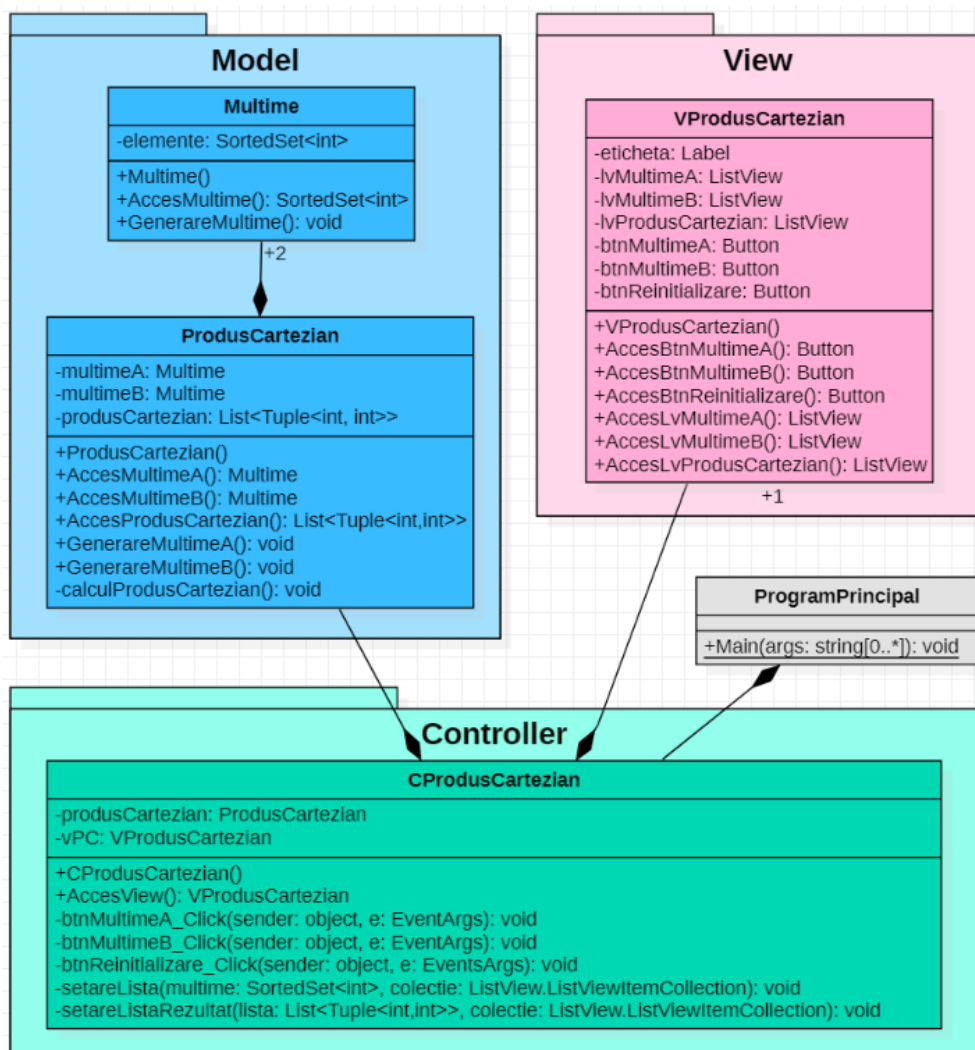


Figura 4.2. Diagrama de clase

Conform arhitecturii MVC crearea obiectelor este realizată în ordinea prezentată în diagrama de secvență din figura 4.3. Diagrama cuprinde și ordinea de apelare a metodelor pentru afișarea produsului cartezian când utilizatorul modifică prima mulțime.

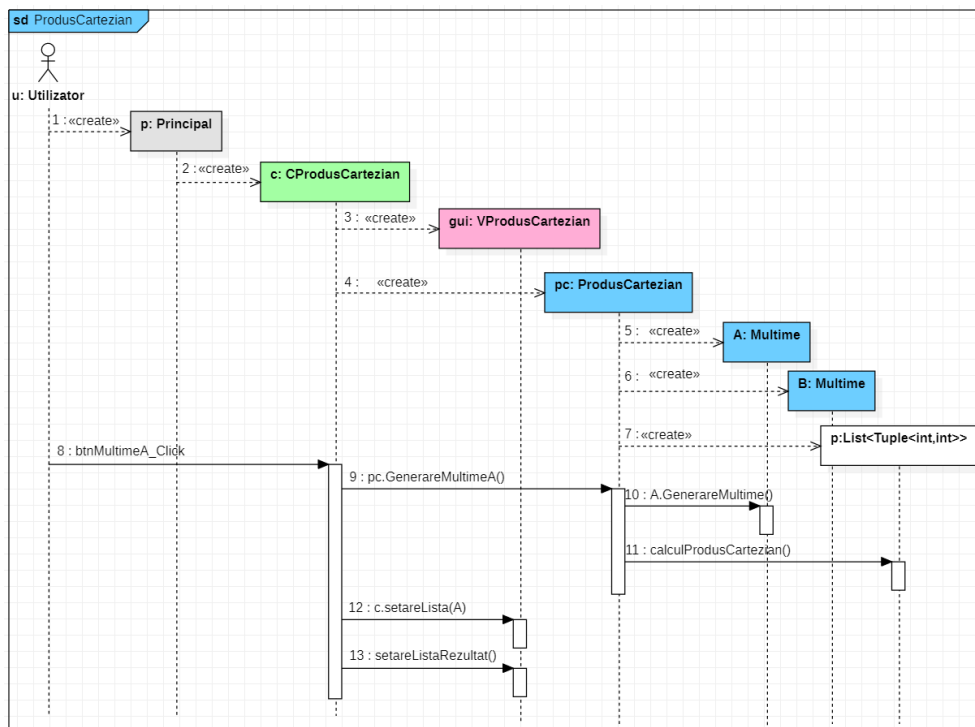


Figura 4.3. Diagramă de secvență

Clasa **Multime** din pachetul **MultimePC.Model** are următoarea implementare:

```

using System;
using System.Collections.Generic;
namespace MultimePC.Model
{
    public class Multime
    {
        private SortedSet<int> elemente;
        public Multime() // multime vida
        {
            this.elemente = new SortedSet<int>();
        }
        public SortedSet<int> Elemente
        {
            get { return this.elemente; }
        }
        public void GenerareMultime()
        {
            this.elemente = new SortedSet<int>();
            Random r = new Random();
            int cardinal = r.Next(1, 21);
            for (int i = 1; i <= cardinal; i++)
            {

```

```

        int nr = r.Next(0, 1000);
        this.elemente.Add(nr);
    }
}
}
}

```

Clasa *ProdusCartezian* din pachetul *MultimePC.Model* are următoarea implementare:

```

using System;
using System.Collections.Generic;
namespace MultimePC.Model
{
    public class ProdusCartezian
    {
        private Multime multimeA;
        private Multime multimeB;
        private List<Tuple<int, int>> produsCartezian;
        public ProdusCartezian()
        {
            this.multimeA = new Multime();
            this.multimeB = new Multime();
            this.produsCartezian = new List<Tuple<int, int>>();
        }
        public Multime MultimeA
        {
            get { return this.multimeA; }
        }
        public Multime MultimeB
        {
            get { return this.multimeB; }
        }
        public List<Tuple<int, int>> Produscartezian
        {
            get { return this.produsCartezian; }
        }
        public void GenerareMultimeA()
        {
            this.multimeA.GenerareMultime();
            this.calculProdusCartezian();
        }
        public void GenerareMultimeB()
        {
            this.multimeB.GenerareMultime();
            this.calculProdusCartezian();
        }
        private void calculProdusCartezian()
        {
            this.produsCartezian = new List<Tuple<int, int>>();
        }
    }
}

```

```

        foreach (int a in this.multimeA.Elemente)
        {
            foreach (int b in this.multimeB.Elemente)
            {
                Tuple<int, int> element = new Tuple<int, int>(a, b);
                this.produsCartezian.Add(element);
            }
        }
    }
}

```

Clasa *VProdusCartezian* din pachetul *MultimePC.View*, ce reprezintă interfața grafică a aplicației (figura 4.4), are următoarea implementare:

```

using System.Windows.Forms;
namespace MultimePC.View
{
    public partial class VProdusCartezian : Form
    {
        public VProdusCartezian()
        {
            InitializeComponent();
        }
        public Button AccesBtnMultimeA()
        {
            return this.btnMultimeA;
        }
        public Button AccesBtnMultimeB()
        {
            return this.btnMultimeB;
        }
        public Button AccesBtnReinitializare()
        {
            return this.btnReinitializare;
        }
        public ListView AccesLvMultimeA()
        {
            return this.lvMultimeA;
        }
        public ListView AccesLvMultimeB()
        {
            return this.lvMultimeB;
        }
        public ListView AccesLvProdusCartezian()
        {
            return this.lvProdusCartezian;
        }
    }
}

```

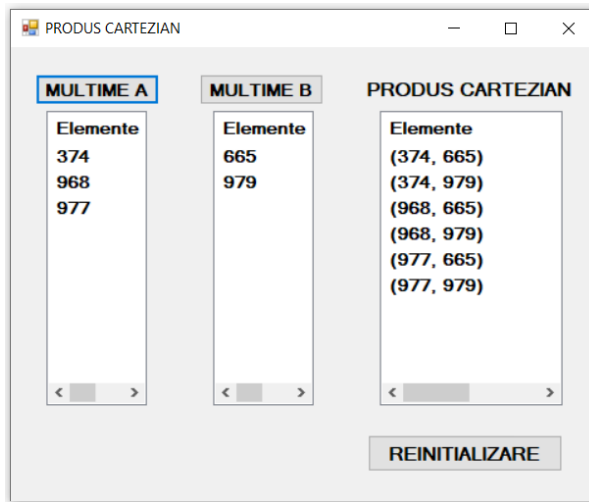


Figura 4.4. Interfața grafică utilizator

Clasa *CProdusCartezian* din pachetul *MultimePC.View* are următoarea implementare:

```
using MultimePC.Model;
using MultimePC.View;
using System;
using System.Collections.Generic;
using System.Windows.Forms;
namespace MultimePC.Controller
{
    public class CProdusCartezian
    {
        private ProdusCartezian produsCartezian = null;
        private VProdusCartezian vPC = null;
        public CProdusCartezian()
        {
            this.vPC = new VProdusCartezian();
            this.produsCartezian = new ProdusCartezian();
            this.vPC.AccesBtnMultimeA().Click += new
EventHandler(btnMultimeA_Click);
            this.vPC.AccesBtnMultimeB().Click += new
EventHandler(btnMultimeB_Click);
            this.vPC.AccesBtnReinitializare().Click += new
EventHandler(btnReinitializare_Click);
        }
        public VProdusCartezian AccesView()
        {
            return this.vPC;
        }
        private void btnMultimeA_Click(object sender, EventArgs e)
        {
            this.produsCartezian.GenerareMultimeA();
        }
    }
}
```

```

        this.setareLista(this.produsCartezian.MultimeA.Elemente,
this.vPC.AccesLvMultimeA().Items);
this.setareListaRezultat(this.produsCartezian.Produscartezian,
this.vPC.AccesLvProdusCartezian().Items);
    }
    private void btnMultimeB_Click(object sender, EventArgs e)
    {
        this.produsCartezian.GenerareMultimeB();
        this.setareLista(this.produsCartezian.MultimeB.Elemente,
this.vPC.AccesLvMultimeB().Items);
this.setareListaRezultat(this.produsCartezian.Produscartezian,
this.vPC.AccesLvProdusCartezian().Items);
    }
    private void btnReinitializare_Click(object sender,
EventArgs e)
    {
        this.produsCartezian = new ProdusCartezian();
        this.setareLista(this.produsCartezian.MultimeA.Elemente,
this.vPC.AccesLvMultimeA().Items);
        this.setareLista(this.produsCartezian.MultimeB.Elemente,
this.vPC.AccesLvMultimeB().Items);
this.setareListaRezultat(this.produsCartezian.Produscartezian,
this.vPC.AccesLvProdusCartezian().Items);
    }
    private void setareLista(SortedSet<int> multime,
ListView.ListViewItemCollection colectie)
    {
        colectie.Clear();
        foreach (int element in multime)
        {
            ListViewItem item = new ListViewItem(element.ToString());
            colectie.Add(item);
        }
    }
    private void setareListaRezultat(List<Tuple<int, int>>
lista, ListView.ListViewItemCollection colectie)
    {
        colectie.Clear();
        foreach (Tuple<int, int> element in lista)
        {
            ListViewItem item = new ListViewItem(element.ToString());
            colectie.Add(item);
        }
    }
}

```

Exercițiu 4.2. Dezvoltați, utilizând șablonul arhitectural MVVM, o aplicație soft care să permită administrarea unei liste de persoane.

Soluție:

Soluția prezentată utilizează șablonului arhitectural MVVM prezentat în figura 4.5.

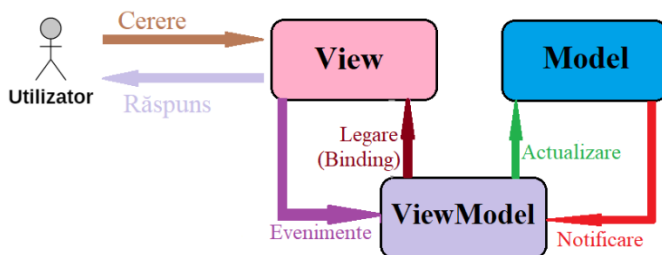


Figura 4.5. Arhitectură MVVM

Pachetul *Model* cuprinde 2 clase: *Persoana* și *PersoanaServiciu* (figura 4.6). Între cele 2 clase există relația de compunere, un obiect de tip *PersoanaServiciu* fiind compus din mai multe obiecte de tip *Persoana*.

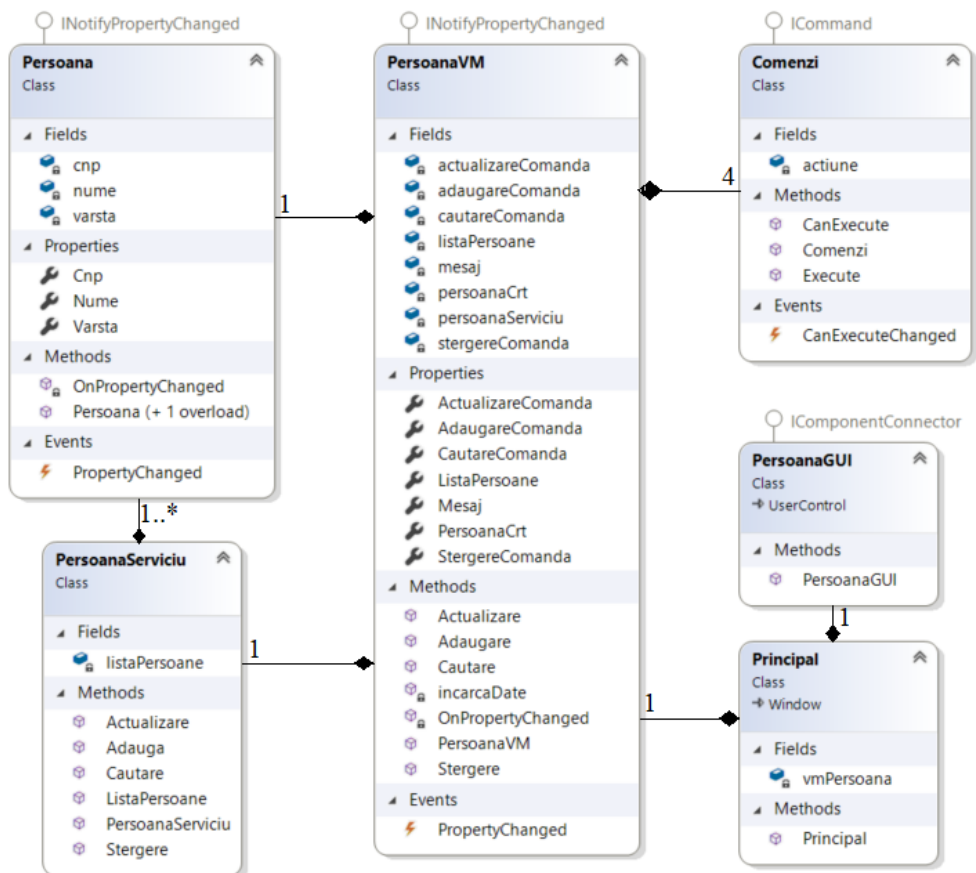


Figura 4.6. Diagrama de clase

Clasa **Persoana** din pachetul **ExempluMVVM.Model** are următoarea implementare:

```
using System.ComponentModel;
namespace ExempluMVVM.Model
{
    public class Persoana : INotifyPropertyChanged
    {
        private string cnp;
        private string nume;
        private byte varsta;
        public event PropertyChangedEventHandler PropertyChanged;
        private void OnPropertyChanged(string numeProprietate)
        {
            if (PropertyChanged != null)
                PropertyChanged(this, new PropertyChangedEventArgs
(numeProprietate));
        }
        public Persoana()
        {
            this.nume = "";
            this.cnp = "";
            this.varsta = 0;
        }
        public Persoana(string nume, string cnp, byte varsta)
        {
            this.nume = nume;
            this.cnp = cnp;
            this.varsta = varsta;
        }
        public string Cnp
        {
            get { return this.cnp; }
            set { this.cnp = value; OnPropertyChanged("Cnp"); }
        }
        public string Nume
        {
            get { return this.nume; }
            set { this.nume = value; OnPropertyChanged("Nume"); }
        }
        public byte Varsta
        {
            get { return this.varsta; }
            set { this.varsta = value; OnPropertyChanged("Varsta"); }
        }
    }
}
```

Clasa **PersoanaServiciu** din pachetul **ExempluMVVM.Model** are următoarea implementare:

```

using System.Collections.Generic;
namespace ExempluMVVM.Model
{
    public class PersoanaServiciu
    {
        private List<Persoana> listaPersoane;
        public PersoanaServiciu()
        {
            this.listaPersoane = new List<Persoana>();
            this.listaPersoane.Add(new Persoana("Pop Ioan",
"1971212345678", 25));
            this.listaPersoane.Add(new Persoana("Ionescu Lucia",
"2991025345678", 23));
        }
        public List<Persoana> ListaPersoane()
        {
            return this.listaPersoane;
        }
        public bool Adauga(Persoana persoana)
        {
            if (this.Cautare(persoana.Cnp) == null)
            {
                this.listaPersoane.Add(persoana);
                return true;
            }
            return false;
        }
        public bool Actualizare(Persoana persoana)
        {
            bool actualizare = false;
            for(int i=0; i< this.listaPersoane.Count; i++)
                if (this.listaPersoane[i].Cnp == persoana.Cnp)
                {
                    this.listaPersoane[i].Nume = persoana.Nume;
                    this.listaPersoane[i].Varsta = persoana.Varsta;
                    actualizare = true;
                    break;
                }
            return actualizare;
        }
        public bool Stergere(string cnp)
        {
            bool stergere = false;
            for (int i = 0; i < this.listaPersoane.Count; i++)
                if (this.listaPersoane[i].Cnp == cnp)
                {
                    this.listaPersoane.RemoveAt(i);
                    stergere = true;
                    break;
                }
            return stergere;
        }
        public Persoana Cautare(string cnp)
        {
            for (int i = 0; i < this.listaPersoane.Count; i++)

```

```

        if (this.listaPersoane[i].Cnp == cnp)
            return this.listaPersoane[i];
        return null;
    }
}

```

Pachetul **ViewModel** conține o clasă **PersoanaVM** caracterizată de următoarele atribute:

- ❖ Un obiect ce reprezintă obiectul curent de tip **Persoana**;
- ❖ Un obiect de tip **PersoanaServiciu** ce reprezintă lista de persoane
- ❖ Patru obiecte de tip **Comenzi** utilizate pentru a gestiona evenimentele generate de utilizator.

Clasa **PersoanaVM** din pachetul **ExempluMVVM.ViewModel** are următoarea implementare:

```

using System;
using System.ComponentModel;
using ExempluMVVM.Model;
using ExempluMVVM.Commands;
using System.Collections.Generic;
namespace ExempluMVVM.ViewModel
{
    public class PersoanaVM : INotifyPropertyChanged
    {
        PersoanaServiciu persoanaServiciu;
        private Persoana persoanaCrt;
        private List<Persoana> listaPersoane;
        private string mesaj;
        private Comenzi adaugareComanda;
        private Comenzi actualizareComanda;
        private Comenzi stergereComanda;
        private Comenzi cautareComanda;
        public PersoanaVM()
        {
            this.persoanaCrt = new Persoana();
            this.persoanaServiciu = new PersoanaServiciu();
            this.incarcaDate();
            this.adaugareComanda = new Comenzi(this.Adaugare);
            this.actualizareComanda = new Comenzi(this.Actualizare);
            this.stergereComanda = new Comenzi(this.Stergere);
            this.cautareComanda = new Comenzi(this.Cautare);
        }
        public event PropertyChangedEventHandler PropertyChanged;
        private void OnPropertyChanged(string numeProprietate)
        {
            if (PropertyChanged != null)
                PropertyChanged(this, new PropertyChangedEventArgs(
numeProprietate));
        }
        public List<Persoana> ListaPersoane

```

```

    {
        get { return this.listaPersoane; }
        set { this.listaPersoane = value; OnPropertyChanged
("ListaPersoane"); }
    }
    public Persoana PersoanaCrt
    {
        get { return this.persoanaCrt; }
        set { this.persoanaCrt = value; }
    }
    public string Mesaj
    {
        get { return this.mesaj; }
        set { this.mesaj = value; OnPropertyChanged("Mesaj"); }
    }
    public Comenzi AdaugareComanda
    {
        get { return this.adaugareComanda; }
    }
    public Comenzi ActualizareComanda
    {
        get { return this.actualizareComanda; }
    }
    public Comenzi StergereComanda
    {
        get { return this.stergereComanda; }
    }
    public Comenzi CautareComanda
    {
        get { return this.cautareComanda; }
    }
    private void incarcaDate()
    {
        this.ListaPersoane = new
List<Persoana>(this.persoanaServiciu.ListaPersoane());
    }
    public void Adaugare()
    {
        try
        {
            bool succes = persoanaServiciu.Adauga
(this.PersoanaCrt);
            if (succes)
            {
                this.incarcaDate();
                this.Mesaj = "Persoana adaugata cu succes";
            }
            else
                this.Mesaj = "Esec adaugare persoana";
        }
        catch (Exception ex)
        {
            this.Mesaj = ex.Message;
        }
    }
}

```

```

public void Actualizare()
{
    try
    {
        bool succes =
persoanaServiciu.Actualizare(this.PersoanaCrt);
        if (succes)
        {
            this.incarcaDate();
            this.Mesaj = "Persoana actualizata cu succes";
        }
        else
            this.Mesaj = "Esec actualizare persoana";
    }
    catch (Exception ex)
    {
        this.Mesaj = ex.Message;
    }
}

public void Stergere()
{
    try
    {
        bool succes =
persoanaServiciu.Stergere(this.persoanaCrt.Cnp);
        if (succes)
        {
            this.incarcaDate();
            this.Mesaj = "Persoana stearsa cu succes";
        }
        else
            this.Mesaj = "Esec stergere persoana";
    }
    catch (Exception ex)
    {
        this.Mesaj = ex.Message;
    }
}

public void Cautare()
{
    try
    {
        var pers =
persoanaServiciu.Cautare(this.persoanaCrt.Cnp);
        if (pers != null)
        {
            this.PersoanaCrt = pers;
            this.Mesaj = "Persoana gasita!";
        }
        else
            this.Mesaj = "Nu s-a gasit persoana!";
    }
    catch (Exception ex)
    {
        this.Mesaj = ex.Message;
    }
}

```

```

    }
}
}
}

```

Clasa **Comenzi** are următoarea implementare:

```

using System;
using System.Windows.Input;
namespace ExempluMVVM.Commands
{
    public class Comenzi : ICommand
    {
        public event EventHandler CanExecuteChanged;
        private Action actiune;
        public Comenzi(Action actiune)
        {
            this.actiune = actiune;
        }
        public bool CanExecute(object parameter)
        {
            return true;
        }
        public void Execute(object parameter)
        {
            this.actiune();
        }
    }
}

```

Pachetul **View** conține o clasă (**PersoanaGUI**) care este caracterizată de responsabilitățile de a gestiona interacțiunile cu utilizatorul și evenimentele generate de acesta (figura 4.7). Această clasă are asociat fișierul **PersoanaGUI.xml** ce este caracterizat de următoarea structură:

```

<UserControl x:Class="ExempluMVVM.View.PersoanaGUI"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:local="clr-namespace:ExempluMVVM.View"
        mc:Ignorable="d"
        d:DesignHeight="545" Width="623">
    <Grid Margin="15">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="auto" />
            <ColumnDefinition Width="*" />

```

```

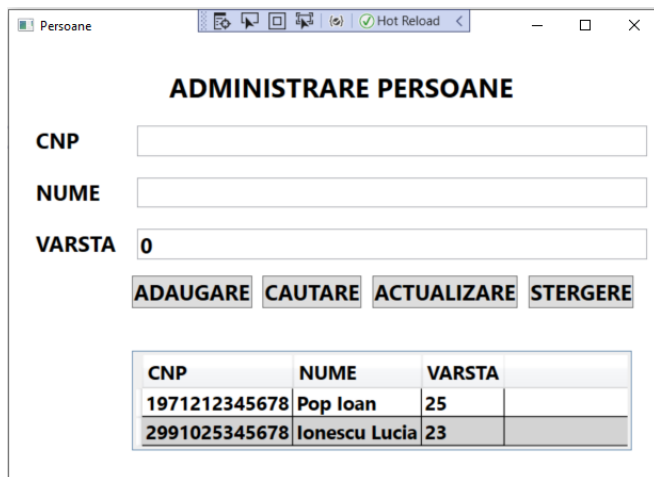
</Grid.ColumnDefinitions>
<Grid.RowDefinitions>
    <RowDefinition Height="auto" />
    <RowDefinition Height="auto" />
    <RowDefinition Height="auto" />
    <RowDefinition Height="auto" />
    <RowDefinition Height="auto" />
    <RowDefinition Height="auto" />
    <RowDefinition Height="auto" />
</Grid.RowDefinitions>
<TextBlock Text="ADMINISTRARE PERSOANE"
    Grid.Row="0" Grid.Column="0"
    Grid.ColumnSpan="2"
    FontSize="25" FontWeight="Bold"
    Margin="10" HorizontalAlignment="Center" >
</TextBlock>
<TextBlock Text="CNP"
    Grid.Row="1" Grid.Column="0"
    FontSize="20" FontWeight="Bold"
    Margin="10" >
</TextBlock>
<TextBlock Text="NUME"
    Grid.Row="2" Grid.Column="0"
    FontSize="20" FontWeight="Bold"
    Margin="10" >
</TextBlock>
<TextBlock Text="VARSTA"
    Grid.Row="3" Grid.Column="0"
    FontSize="20" FontWeight="Bold"
    Margin="10" >
</TextBlock>
<TextBox Name="txtCnp"
    Grid.Row="1" Grid.Column="1"
    FontSize="20" FontWeight="Bold" Margin="10"
    Text="{Binding Path=PersoanaCrt.Cnp,
Mode=TwoWay}" >
</TextBox>
<TextBox Name="txtNume"
    Grid.Row="2" Grid.Column="1"
    FontSize="20" FontWeight="Bold" Margin="10"
    Text="{Binding Path=PersoanaCrt.Nume,
Mode=TwoWay}" >
</TextBox>
<TextBox Name="txtVarsta"
    Grid.Row="3" Grid.Column="1"
    FontSize="20" FontWeight="Bold" Margin="10"
    Text="{Binding Path=PersoanaCrt.Varsta,
Mode=TwoWay}" >
</TextBox>
<StackPanel Orientation="Horizontal"
    Grid.Row="4" Grid.Column="1">
    <Button Name="btnAdaugare"
        FontSize="20" FontWeight="Bold"
        Margin="5" Content="ADAugARE"
        Command="{Binding Path=AdaugareComanda}"/>

```

```

        <Button Name="btnCautare"
            FontSize="20" FontWeight="Bold"
            Margin="5" Content="CAUTARE"
            Command="{Binding Path=CautareComanda}"/>
        <Button Name="btnActualizare"
            FontSize="20" FontWeight="Bold"
            Margin="5" Content="ACTUALIZARE"
            Command="{Binding Path=ActualizareComanda}"/>
        <Button Name="btnStergere"
            FontSize="20" FontWeight="Bold"
            Margin="5" Content="STERGERE"
            Command="{Binding Path=StergereComanda}"/>
    </StackPanel>
    <TextBlock Name="txtMesaj"
        Grid.Row="5" Grid.Column="1"
        FontSize="20" FontWeight="Bold"
        Text="{Binding Path=Mesaj}">
    </TextBlock>
    <DataGrid Name="dgPersoane"
        AutoGenerateColumns="False"
        AlternatingRowBackground="LightGray"
        CanUserAddRows="False"
        Grid.Row="6" Grid.Column="1"
        Margin="5,8,24,8" Padding="3,3,3,3"
        FontSize="18" FontWeight="Bold"
        ItemsSource="{Binding Path=ListaPersoane,
Mode=TwoWay}">
        <DataGrid.Columns>
            <DataGridTextColumn Header="CNP"
                Width="auto"
                Binding="{Binding Path=Cnp}">
            </DataGridTextColumn>
            <DataGridTextColumn Header="NUME"
                Width="auto"
                Binding="{Binding Path=Nume}">
            </DataGridTextColumn>
            <DataGridTextColumn Header="VARSTA"
                Width="auto"
                Binding="{Binding Path=Varsta}">
            </DataGridTextColumn>
        </DataGrid.Columns>
    </DataGrid>
</Grid>
</UserControl>

```

ADMINISTRARE PERSOANE

CNP

NUME

VARSTA

CNP	NUME	VARSTA	
1971212345678	Pop Ioan	25	
2991025345678	Ionescu Lucia	23	

Figura 4.7. Interfața grafică utilizator

4.3 Exerciții propuse spre rezolvare

Exercițiu 4.3. Modificați exercițiul 3.2 (lucrare de laborator 3) astfel încât să respecte șablonul arhitectural MVC.

Exercițiu 4.4. Pornind de la diagrama de clase din figura 1.6 (lucrare de laborator 1) dezvoltăți o aplicație software pe baza șablonului arhitectural MVVM care să determine distanța euclidiană, distanța Manhattan și distanța Cebâsev pentru 2 puncte în plan și în spațiu.