

## TEHNICI DE PROGRAMARE AVANSATA – Laborator 7

- **Implementarea modelului simplificat de vizualizare si actualizare tabela utilizand fisiere XML**

### **Implementarea modelului simplificat de vizualizare si actualizare tabela utilizand fisiere XML**

Nu vom mai folosi baze de date ci fisiere XML din care vom prelua date sau in care vom salva date. Fisierele XML le vom crea in directorul aplicatiei.

Pas 1. Se creaza o noua aplicatia.

Se adauga aplicatiei o component DataSet.

Pas 2. Se creaza in DataSet doua obiecte DataTable (click dr => Add => DataTable).

Se configureaza obiectele DataTable din proprietati (Name -> Persoane, respectiv Name -> Auto).

Pas 3. Se definesc campurile tabelii Persoane (click dr. pe tabela => ADD => Column) si se configureaza din proprietati:

Name -> IdPersoana, DataType -> System.Int32, AutoIncrement -> True,

AutoIncrementSeed -> 1, AutoIncrementStep -> 1, ReadOnly -> True,

Unique -> True, AllowDBNull -> False

Obs. Cand s-a setat Unique -> True s-a marcat campul respective prin Cheie.

Name-> Nume

Name-> CNP

Name-> Salariu, DataType->System.Decimal

Pas 4. Se definesc campurile tabelii Auto (click dr. pe tabela => ADD => Column) si se configureaza din proprietati:

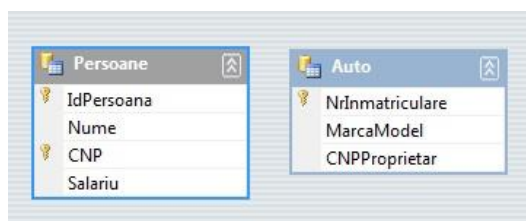
Name-> NrInmatriculare

Name-> MarcaModel

Name-> CNPProprietar

Toate de tip string (implicit)

Se obtine o imagine Designer a dataset-ului de forma:



Pas 5. Se configureaza form-ul Persoane :

	IdPersoana	Nume	CNP	Salariu
*				

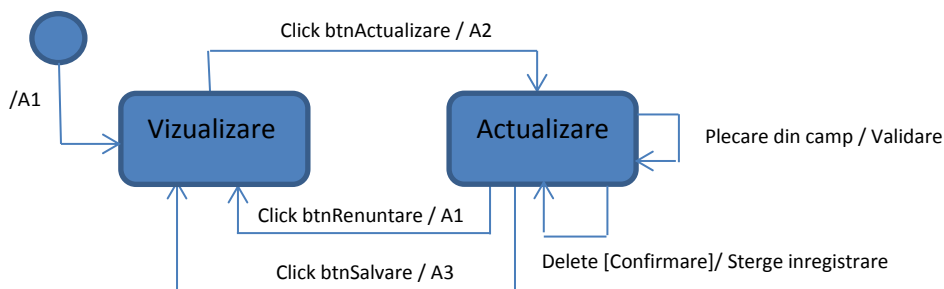
	NrInmatriculare	MarcaModel	CNPProprietar
*			

Pas 6. Se leaga Grid-ul din stanga la DataTable-ul Persoane: ChooseDataSource => OtherDataSources => ProjectDataSources => DataSet1 => Persoane, iar cel din dreapta la Auto.

Obs. Se observa includerea a 3 componente: DataSet1, persoaneBindingSource si autoBindingSource.

Nu se mai include persoaneTableAdapter si autoTableAdapter pt. ca nu mai lucram cu o baza de date.

Pas 7. Logica de functionare este data de diagrama de stari:



Descrierea actiunilor din diagrama de stari:

A1 – config(true); refresh()

A2 – config(false)

A3 - salvare *dataTable-uri* Persoane si Auto in fisiere XML; config(true)

Validari

1. validare de tip numeric pentru salariu;
2. protejare la actualizare campuri. Campul IdPersoana nu poate fi modificat de catre utilizator (este generat automat);

## Modulul de cod

```
using System;
using System.Windows.Forms;
using System.Data;

namespace C03
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            dataGridView1.Columns[0].ReadOnly = true; //IdPersoana read only
        }

        private void config(bool v)
        {
            //pt.vizualizare v = true pt.actualizare v=false

            dataGridView1.AllowUserToAddRows = !v;
            dataGridView1.AllowUserToDeleteRows = !v;
            dataGridView1.Columns[1].ReadOnly = v;
            dataGridView1.Columns[2].ReadOnly = v;
            dataGridView1.Columns[3].ReadOnly = v;

            btnActualizare.Enabled = v;

            btnSalvare.Visible = !v;
            btnRenuntare.Visible = !v;
        }

        private void refresh(){
            dataSet1.Persoane.Clear();
            dataSet1.Auto.Clear();
            try
            {
                dataSet1.Persoane.ReadXml("Persoane.xml");
                dataSet1.Auto.ReadXml("Auto.xml");
            }
            catch { }
        }

        private void btnActualizare_Click(object sender, EventArgs e)
        {
            //A2
            config(false);
        }

        private void btnSalvare_Click(object sender, EventArgs e)
        {
            //A3
            dataSet1.Persoane.WriteXml("Persoane.xml");
            dataSet1.Auto.WriteXml("Auto.xml");
            config(true);
        }

        private void btnRenuntare_Click(object sender, EventArgs e) {
            config(true);
            refresh();
        }

        private void dataGridView1_DataError(object sender, DataGridViewDataErrorEventArgs e){
            MessageBox.Show("Format eronat");
        }
    }
}
```

```

private void Form1_Load(object sender, EventArgs e)
{
    //A1
    config(true);
    refresh();
}

private void dataGridView1_UserDeletingRow(object sender,
                                           DataGridViewRowCancelEventArgs e)
{
    const string mesaj = "Confirmati stergerea";
    const string titlu = "Stergere inregistrare";

    var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
                                   MessageBoxIcon.Warning);
    if (rezultat == DialogResult.No) { e.Cancel = true; return; }
}

private void dataGridView1_CellValidating(object sender,
                                           DataGridViewCellValidatingEventArgs e)
{
    MessageBox.Show("Validare camp " + e.ColumnIndex);
}
}

```

Validari suplimentare

1. validare de obligativitate a completarii pentru Nume, CNP si Salariu;
2. validare de numericitate pt. campuri declarate text (CNP)
3. validare de unicitate pentru CNP (cheie semantica).
4. integritatea referentiala la stergerea unei persoane, verificand daca exista inregistrari in alte tabele care refera persoana respectiva. Se va considera DataTable-ul Automobile care vor avea un camp cu CNP-ul proprietarului. Aceasta se va incarca tot dintr-un fisier XML (Automobile.xml).

Implementarea lor:

1. validare de obligativitate a completarii pentru Nume, CNP si Salariu;

```

private string numeColoana(int c) {
    if (c == 1) return "Nume";
    else if (c == 2) return "CNP";
    else if (c == 3) return "Salariu";
    return "";
}

private void dataGridView1_CellValidating(object sender, DataGridViewCellValidatingEventArgs e)
{
    string s;

    s = e.FormattedValue.ToString();

    // Validare completare obligatorie
    if (s == "" && !btnActualizare.Enabled) {
        MessageBox.Show("Completati campul " + numeColoana(e.ColumnIndex));
        e.Cancel = true; return;
    }
}

```

2. validare de numericitate pt. campuri declarate text (CNP)

```
private void dataGridView1_CellValidating(object sender, DataGridViewCellValidatingEventArgs e)
{
    decimal d;
    string s;

    s = e.FormattedValue.ToString();

    // Validare completare obligatorie
    if (s == "" && !btnActualizare.Enabled)
    {
        MessageBox.Show("Compleati campul " + numeColoana(e.ColumnIndex));
        e.Cancel = true; return;
    }

    //Validare de tip numeric
    if (e.ColumnIndex == 2)
        try { d = Convert.ToInt64(s); }
        catch { MessageBox.Show("Format eronat"); e.Cancel = true; return; }
}
```

3. validare de unicitate pentru CNP (cheie semantica).

```
// Verificare cheie semantica
private Boolean existaCNP(int id, string cnp)
{
    int idCrt;
    string cnpCrt;
    foreach (DataRow r in dataSet1.Persoane) {
        idCrt = (Int32)(r["IdPersoana"]);
        cnpCrt = r["CNP"].ToString();
        if (cnpCrt == cnp && idCrt != id) return true;
    }
    return false;
}

private void dataGridView1_CellValidating(object sender, DataGridViewCellValidatingEventArgs e) {
    decimal d;
    string s;

    s = e.FormattedValue.ToString();
    string ids = dataGridView1.CurrentRow.Cells[0].Value.ToString();
    Int32 id = Convert.ToInt32(ids);

    // Validare completare obligatorie
    if (s == "" && !btnActualizare.Enabled) {
        MessageBox.Show("Compleati campul " + numeColoana(e.ColumnIndex));
        e.Cancel = true; return;
    }

    //Validare de tip numeric
    if (e.ColumnIndex == 2)
        try { d = Convert.ToInt64(s); }
        catch { MessageBox.Show("Format eronat"); e.Cancel = true; return; }

    //Validare cheie semantica
    if (e.ColumnIndex == 2 && existaCNP(id, s)) {
        MessageBox.Show("CNP deja existent");
        e.Cancel = true;
        return;
    }
}
```

4. integritatea referentiala la stergerea unei persoane, verificand daca exista inregistrari in alte tabele care refera persoana respectiva. Se va considera DataTable-ul Automobile care vor avea un camp cu CNP-ul proprietarului. Aceasta se va incarca tot dintr-un fisier XML (Automobile.xml).

```
// Verificare integritate referentiala la stergere persoana
private Boolean existaAuto(string cnp) {
    string cnpCrt;
    foreach (DataRow r in dataSet1.Auto)
    {
        cnpCrt = r["CNPProprietar"].ToString();
        if (cnpCrt == cnp) return true;
    }
    return false;
}

private void dataGridView1_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e){
    const string mesaj = "Confirmati stergerea";
    const string titlu = "Stergere inregistrare";
    string cnp = dataGridView1.CurrentRow.Cells[2].Value.ToString();

    var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
                                    MessageBoxIcon.Warning);
    if (rezultat == DialogResult.No) { e.Cancel = true; return; }
    //Verificare integritate referentiala

    if (existaAuto(cnp)) {
        MessageBox.Show("Este proprietar Auto! Nu se poate sterge");
        e.Cancel = true;
        return;
    }
}
```