

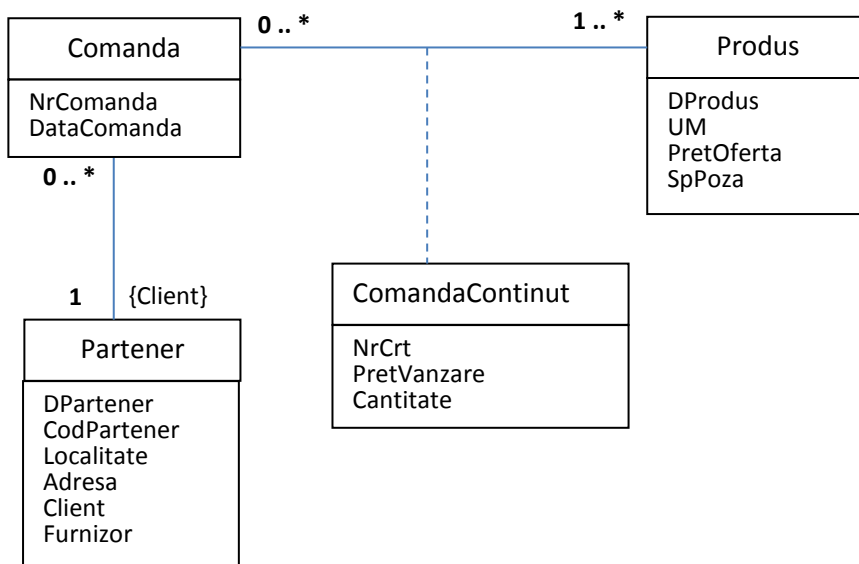
TEHNICI DE PROGRAMARE AVANSATA – Laborator 11

Un model de incarcare date intr-o tabela *asociere*

Tabelele asociere sunt necesare pentru implementarea relatiilor de asociere *many to many*.

Vom realiza un form pentru introducerea de comenzi ale clientilor pentru anumite produse.

Diagrama de clase



Pas 1. Proiectarea bazei de date

Comenzi(IdComanda, NrComanda, DataComanda, IdClient)

	Field Name	Data Type
🔑	IdComanda	AutoNumber
	NrComanda	Number
	DataComanda	Date/Time
	IdClient	Number

IdComanda	NrComanda	DataComanda	IdClient
1	101	01.10.2014	1
2	102	01.10.2014	3

Parteneri(IdPartener, DPartener, CodPartener, Localitate, Adresa, Client, Furnizor))

🔑	IdPartener	AutoNumber
	DPartener	Text
	CodPartener	Number
	Localitate	Text
	Adresa	Text
	Client	Yes/No
	Furnizor	Yes/No

IdPartener	DPartener	CodPartener	Localitate	Adresa	Client	Furnizor
1	Partener 01	1234	Constanta	Bd.Ferdinand I	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Partener 02	2345	Bucuresti	Str. Liliacului N	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Partener 03	3456	Medgidia	Str. Republicii	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Partener 04	4567	Constanta	Bd.Lapusneanu	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8	Partener 05	2222	Bucuresti		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Produse(IdProdus, DProdus, UM, PretOferta, SpPoza)

IdProdus	AutoNumber
DProdus	Text
Um	Text
PretOferta	Currency
SpPoza	Text

IdProdus	DProdus	Um	PretOferta	SpPoza
1	Produs 01	Buc	10,00 lei	d:\ISS2014\Poze\P01.jpeg
2	Produs 02	Kg	12,00 lei	d:\ISS2014\Poze\P02.jpg
3	Produs 03	Kg	5,00 lei	d:\ISS2014\Poze\P03.jpeg
4	Produs 04	Kg	4,00 lei	d:\ISS2014\Poze\P04.jpeg
5	Produs 05	Kg	7,00 lei	d:\ISS2014\Poze\P05.jpeg
6	Produs 06	Kg	100,00 lei	D:\ISS2014\Poze\P03.jpeg
7	Produs 07	Buc	100,00 lei	D:\ISS2014\Poze\P04.jpeg

ComenziContinut(IdComanda, NrCrt, IdProdus, PretVanzare, Cantitate)

IdComanda	Number
NrCrt	Number
IdProdus	Number
PretVanzare	Currency
Cantitate	Currency

IdComanda	NrCrt	IdProdus	PretVanzare	Cantitate
1	1	1	10,00 lei	2,50
1	2	2	12,00 lei	3,20
1	3	3	5,00 lei	4,50
2	1	4	4,00 lei	3,80
2	2	5	7,00 lei	2,00

Pas 2. DataSet-ul trebuie sa contina urmatoarele obiecte DataTable:

Produse IdProdus DProdus Um PretOferta SpPoza ProduseTableAdapter Fill, GetData ()	Clienti IdPartener DPartener CodPartener Localitate Adresa ClientiTableAdapter Fill, GetData ()	ComandaContinutManevra NrCrt DProdus UM PretVanzare Cantitate Valoare IdProdus SpPoza
---	--	--

Cererile SQL pentru Produse si Clienti

```
SELECT    IdProdus, DProdus, Um, PretOferta, SpPoza
FROM      Produse
ORDER BY  DProdus
```

```
SELECT    IdPartener, DPartener, CodPartener, Localitate, Adresa
FROM      Parteneri
WHERE     (Client = True)
ORDER BY  DPartener
```

Observatii:

1. DataTable-ul ComandaContinutManevra nu are TableAdapter, deci nu are nici o legatura cu baza de date. Este, asa cum ii spune si numele o structura tabelara de manevra, in care se editeaza comanda curenta, care la confirmare va fi salvata in baza de date.
2. ComandaContinutManevra este insa sursa de date pentru grid-ul din form. Exista chiar si un BindingSource pentru el.
3. ComandaContinutManevra se creaza prin ClickDr pe spatiul liber al Dataset-ului => Add => DataTable
4. Campurile DataTable-ului ComandaContinutManevra se creaza prin ClickDr pe bara de titlu al DataTable-ului => Add => Column.
5. Tipurile de date pentru campurile lui ComandaContinutManevra sunt:
 NrCrt => System.Int16
 DProdus => System.String (implicit)
 UM => System.String (implicit)
 PretVanzare => System.Decimal
 Cantitate => System.Decimal
 Valoare => System.Decimal
 IdProdus => System.Int32
 SpPoza => System.String (implicit)
6. Pentru campul Valoare trebuie setata proprietatea Expression => PretVanzare * Cantitate

Pas 3. Se creaza un form nou FComenziAct avand structura

Labels pointing to form elements:

- txtNrComanda
- dateTimePicker1
- cmbClient
- btnConfirmare
- lblOp
- txtTotal

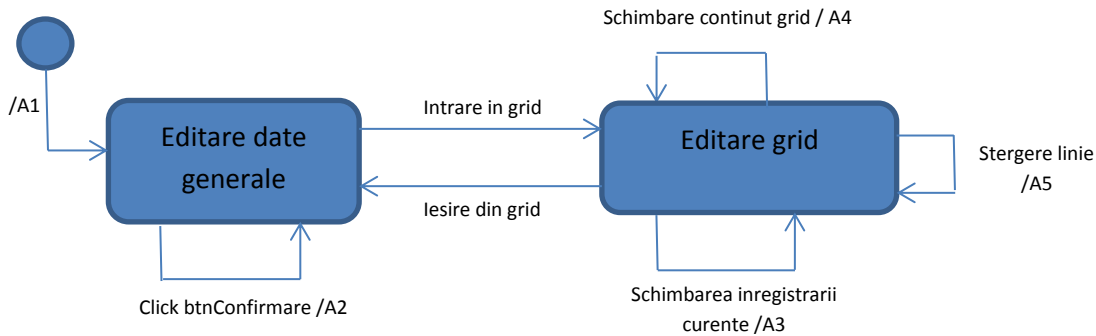
Form structure:

- Nr Comanda
- Data: 1 decembrie 2014
- Client
- Total
- COMANDA NOUA
- Table with columns: NrCrt, DProdus, UM, PretVanzare, Cantitate, Valoare, IdProdus, SpPoza

Data sources at the bottom:

- dataSet1
- clientiBindingSource
- clientiTableAdapter
- comandaContinutManevraBindingSource
- produseBindingSource
- produseTableAdapter

Pas 4 Proiectarea logicii form-ului



Pas 5. Configurare comboBox Clienti

DataSource => clientiBindingSource

DisplayMember => DPartener

ValueMember => IdPartener

Pas 6. Configurare Grid

DataSource => comandaContinutManevraBindingSource

Edit Columns => coloana DProdus => ColumnType => DataGridViewComboBoxColumn

DisplayStyleForCurrentCellOnly => True

DataSource => produseBindingSource

DisplayMember => DProdus

Pas 7. Codul sursa

Variabile locale clasei

```
private OleDbConnection con = new OleDbConnection();
private OleDbCommand cmd = new OleDbCommand();
private OleDbDataReader rdr;
private int idCda;
```

A1:

```
private void A1()
{
    //Incarcare DataTable Produse
    produseTableAdapter.Fill(dataSet1.Produse);

    //Incarcare DataTable Clienti
    clientiTableAdapter.Fill(dataSet11.Clienti);

    //Setare lblOp
    lblOp.Text = "COMANDA NOUA";

    // Protectie la modificare
    txtNrComanda.ReadOnly = true;
    txtTotal.ReadOnly = true;
    nrCrtDataGridViewTextBoxColumn.ReadOnly = true;
    uMDataGridViewTextBoxColumn.ReadOnly = true;
    pretVanzareDataGridViewTextBoxColumn.ReadOnly = true;
    idProdusDataGridViewTextBoxColumn.ReadOnly = true;

    // Initializare cmbClient
    cmbClient.SelectedIndex = -1;
}
```

```

A2:
private void A2() {
    if (!validareCampuriObligatorii()) return;
    generez_nr_cda();
    // Scriu inregistrare in Comenzi si ComenziContinut
    adaugaInregistrareComenzi();
    cautaInregistrare();
    adaugaInregistrariComenziContinut();
    golireCampuri();
}

A3:
private void A3() {
    // Generare NrCrt
    DataRowView current = (DataRowView)comandaContinutManevraBindingSource.Current;
    current["NrCrt"] = comandaContinutManevraBindingSource.Position + 1;
}

A4:
private void A4() {
    try
    {
        if (dataGridView1.CurrentCell.ColumnIndex == 1)
        {
            DataRowView current = (DataRowView)produseBindingSource.Current;

            dataGridView1.CurrentRow.Cells[2].Value = current["UM"];
            dataGridView1.CurrentRow.Cells[3].Value = current["PretOferta"];
            dataGridView1.CurrentRow.Cells[6].Value = current["IdProdus"];
            calcTotal();
        }
        if (dataGridView1.CurrentCell.ColumnIndex == 4)
        {
            comandaContinutManevraBindingSource.EndEdit();
            calcTotal();
        }
    }
    catch { }
}

A5:
private void A5(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;
}

```

Metode eveniment

```

private void Form1_Load(object sender, EventArgs e) {
    A1();
}

private void comandaContinutManevraBindingSource_CurrentChanged(object sender, EventArgs e){
    A3();
}

private void dataGridView1_CellValueChanged(object sender, DataGridViewCellEventArgs e){
    A4();
}

private void dataGridView1_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e){
    A5(e);
}

```

```

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

private void dataGridView1_UserDeletedRow(object sender, DataGridViewRowEventArgs e){
    calcTotal();
}

private void btnConfirmare_Click(object sender, EventArgs e){
    A2();
}

```

Metode de uz general

```

private void calcTotal() // Calcul total valori
{
    decimal t = 0;
    foreach (DataRow r in dataSet1.ComandaContinutManevra)
    {
        if (r["Valoare"] != DBNull.Value )
            t += (decimal)r["Valoare"];
    }

    txtTotal.Text = Convert.ToString(t);
}

private void generez_nr_cda() {
    con.ConnectionString= produseTableAdapter.Connection.ConnectionString;

    cmd.Connection = con;
    cmd.CommandText = "SELECT Max(Comenzi.NrComanda) AS NrMax FROM Comenzi";

    con.Open();
    rdr = cmd.ExecuteReader();
    rdr.Read();

    txtNrComanda.Text = Convert.ToString (rdr.GetInt32(0) + 1);
    rdr.Close();
    con.Close();
}

private bool validareCampuriObligatorii() {
    //Validare de completare obligatorie campul Client
    if (cmbClient.Text == "")
    {
        MessageBox.Show("Completati Client !");
        cmbClient.Focus();
        return false;
    }

    // Validare completare continuut
    if (comandaContinutManevraBindingSource.Count == 0)
    {
        MessageBox.Show("Completati continuut comanda !");
        dataGridView1.Focus();
        return false;
    }
    return true;
}

private void adaugaInregistrareComenzi() {
    string listaCampuri;
    string listaValori;
    DateTime d = dateTimePicker1.Value;

    listaCampuri = "NrComanda, DataComanda, IdClient";
    listaValori = txtNrComanda.Text +
        ",#" + Convert.ToString(d.Month) + "/"

```

```

        + Convert.ToString(d.Day) + "/"
        + Convert.ToString(d.Year) + "#,"
        + cmbClient.SelectedValue;

cmd.CommandText = "Insert into Comenzi(" + listaCampuri + ") " +
    "Select " + listaValori;

MessageBox.Show(cmd.CommandText);
con.Open();
cmd.ExecuteNonQuery();
con.Close();
}

private void cautaInregistrare() {
    cmd.CommandText = "SELECT idComanda From Comenzi Where NrComanda = " +
        txtNrComanda.Text;

    con.Open();
    rdr = cmd.ExecuteReader();
    rdr.Read();

    idCda = rdr.GetInt32(0);
    rdr.Close();
    con.Close();
}

private void adaugaInregistrariComenziContinut(){
    string listaCampuri = "IdComanda, NrCrt, IdProdus, Cantitate, PretVanzare";
    string listaValori;

    con.Open();
    foreach (DataRow r in dataSet1.ComandaContinutManevra)
    {
        listaValori = idCda + "," + r["NrCrt"] + "," + r["IdProdus"] + "," +
            r["Cantitate"] + "," + r["PretVanzare"];

        cmd.CommandText = "Insert into ComenziContinut(" + listaCampuri + ") " +
            "Select " + listaValori;

        MessageBox.Show(cmd.CommandText);
        cmd.ExecuteNonQuery();
    }
    con.Close();
}

private void golireCampuri() {
    txtNrComanda.Text = "";
    cmbClient.SelectedIndex = -1;
    txtTotal.Text = "";
    dataSet1.ComandaContinutManevra.Clear();
}

```