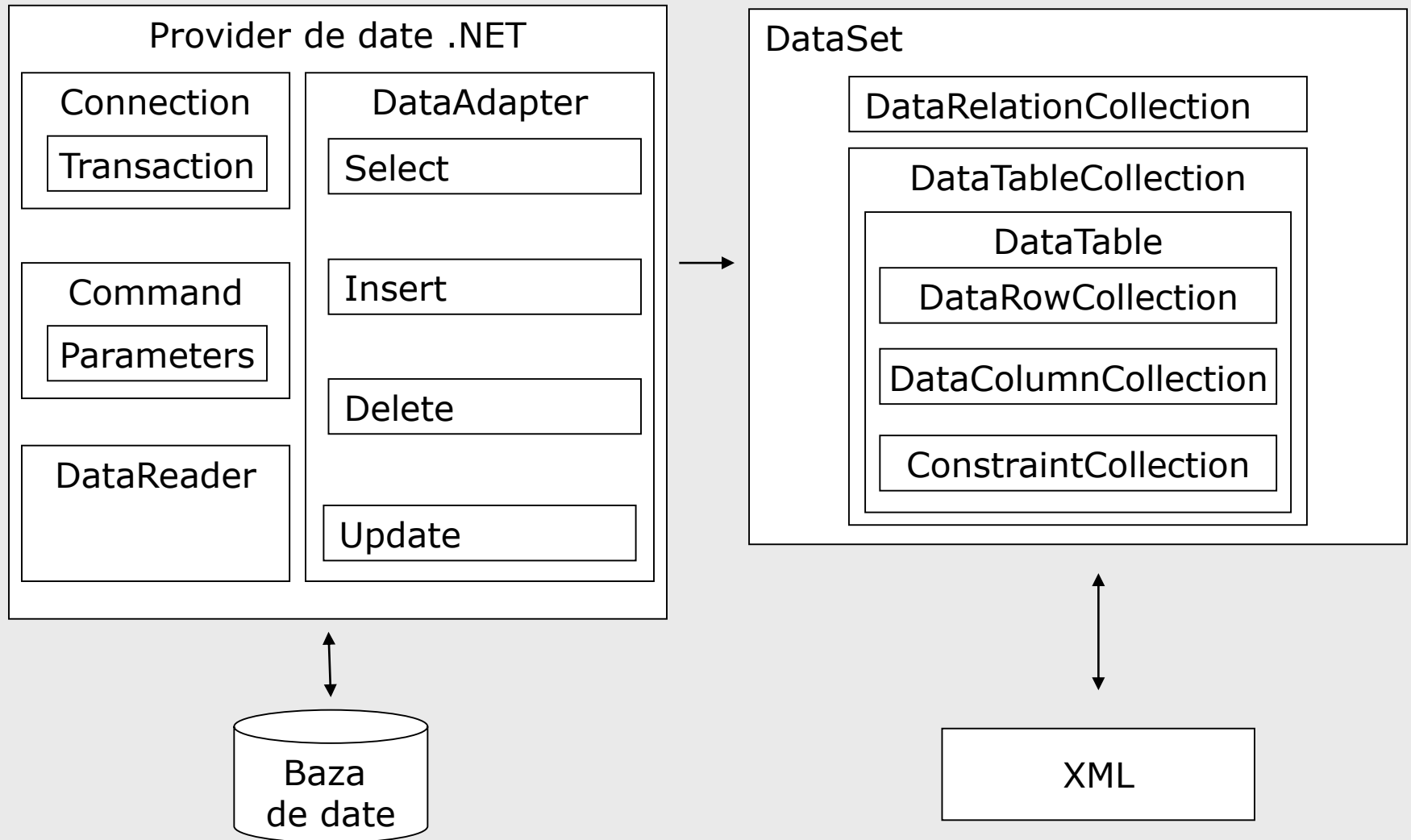


Seminar 2

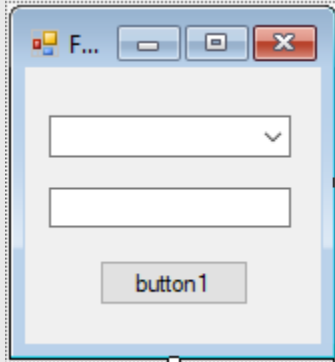
ADO.NET
Data binding

Arhitectura ADO.NET



Data binding - Windows Forms

Windows Form



Binding
Context

Currency
Manager

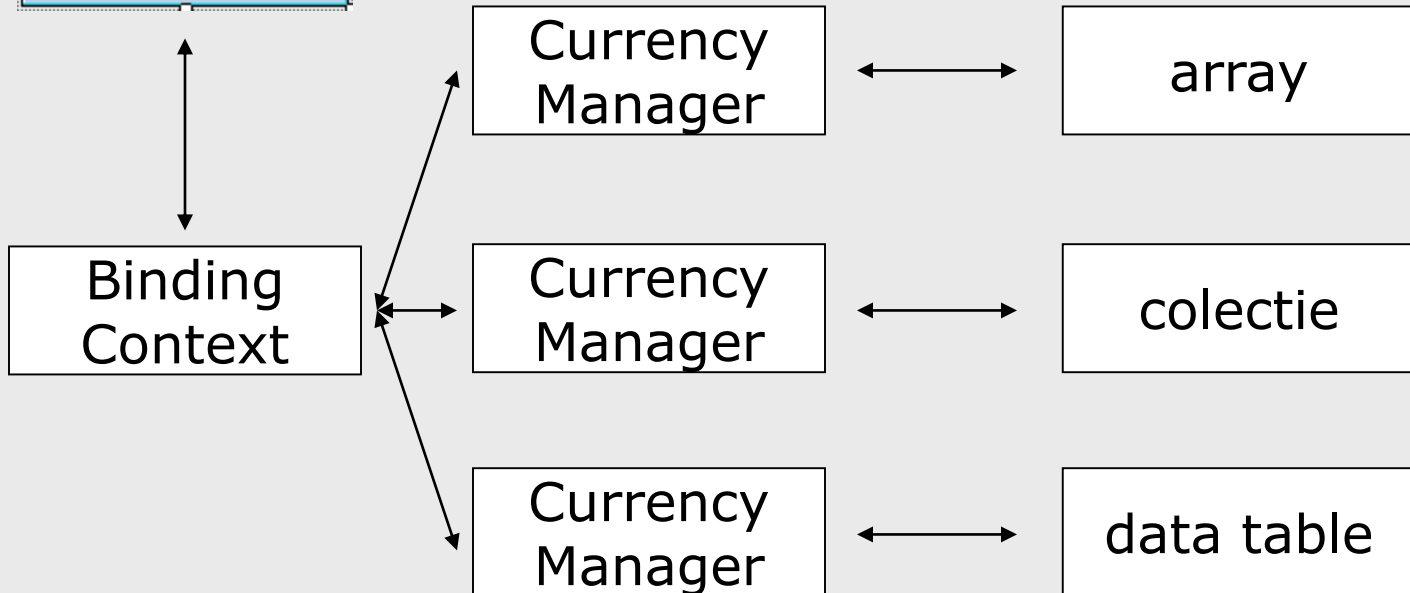
array

Currency
Manager

colectie

Currency
Manager

data table



Provideri de date în .NET

- Pentru a putea îndeplini funcția de sursă de date, o listă trebuie să implementeze interfața *IList*
- ADO.NET oferă structuri de date adecvate pentru *binding*:
 - DataColumn
 - DataTable – format din coloane, rânduri, constrângeri
 - DataView – view customizat al unui tabel
 - DataSet – format din tabele, relații
 - DataViewManager – view customizat al unui DataSet

Consumatori de date în .NET

■ *Currency Manager*

- Menține controalele *data-bound* sincronizate între ele
- Pentru fiecare sursă de date asociată cu un Windows Form, există un obiect *CurrencyManager*
- *Currency* = poziția curentă într-o structură de date
- *Position* - determină poziția curentă a tuturor controalelor conectate la același *CurrencyManager*

Consumatori de date în .NET

- *BindingContext*

- Gestionează colecția de obiecte
CurrencyManager pentru fiecare control
container

Data binding

■ *DataBindings*

- Proprietate a unui control
- Stabilirea unei legături între orice proprietate a unui control și o proprietate a unui obiect

■ Clasa *Binding*

- asociere simplă: valoare proprietate obiect - valoare proprietate control
- *Binding* (*String numeProprietate*, *Object sursaDate*, *String membruDate*)

Populare DataSet cu date

- Un dataset nu conține date implicit
- Tabelele sunt populate prin execuția interogărilor / comenzilor *TableAdapter* / data adapter (*SqlDataAdapter*)

```
unTableAdapter.Fill(unDataSet.NumeTabel);
```

- Salvarea datelor

```
unTableAdapter.Update(unDataSet.NumeTabel);
```

- Metoda *Update* examinează valoarea proprietății *RowState* pentru a determina care înregistrări trebuie salvate și care comandă trebuie invocată (*InsertCommand*, *UpdateCommand*, *DeleteCommand*)

Accesarea înregistrărilor

- Fiecare tabel expune o colecție de rânduri
- Ca în orice colecție, rândurile se accesează prin intermediul indexului colecției sau utilizând instrucțiuni specifice colecției în limbajul de programare utilizat
- dataset tipizat

```
TextBox1.Text = ds1.NumeTabel[2].unCamp;
```

- dataset netipizat

```
string val = (string)  
ds1.Tables["NumeTabel"].Rows[0]["unCamp"];
```

Tabele asociate, obiecte *DataRelation*

- Informațiile din tabelele aflate într-un *DataSet* pot fi inter-relaționate
- Crearea obiectelor *DataRelation* permite descrierea relațiilor dintre tabelele din dataset
- Se poate utiliza un obiect *DataRelation* pentru localizarea înregistrărilor asociate apelând metoda *GetChildRows* pe un *DataRow* în tabelul părinte; această metodă returnează un array de înregistrări copil asociate
- Se poate apela metoda *GetParentRow* pe un *DataRow* din tabelul copil; această metodă returnează un singur *DataRow* din tabelul părinte

Tabele asociate, obiecte *DataRelation*

- Returnare înregistrări copil ale unei înregistrări părinte

```
string IDClient = "VINET";  
NorthwindDataSet.OrdersRow[] comenzi;  
comenzi = (NorthwindDataSet.OrdersRow[])  
    northwindDataSet.Customers.  
    FindByCustomerID(IDClient).GetChildRows  
        ("FK_Orders_Customers");  
MessageBox.Show(comenzi.Length.ToString());
```

Tabele asociate, obiecte *DataRelation*

■ Returnare înregistrare părinte a unei înregistrări copil

```
int IDComanda = 10835;
NorthwindDataSet.CustomersRow client;
client = (NorthwindDataSet.CustomersRow)
    northwindDataSet.Orders.
        FindByOrderID(IDComanda).
            GetParentRow("FK_Orders_Customers");
MessageBox.Show(client.CompanyName);
```

Crearea unei aplicații în Visual Studio

Creare proiect nou *Windows Forms*

- Din meniul *File*, se creează un proiect nou
- Se selectează *Windows Forms Application* și se precizează un nume
- Click *OK*. Proiectul este creat și adăugat în *Solution Explorer*

Creare *Data Source*

- Se pornește *Data Source Configuration Wizard* (fereastra *Data Source*)
- Se alege conexiunea
- Se selectează tabelele dorite

Controale *data-bound*

- *DataSet* - dataset tipizat care conține tabele
- *BindingSource* - conectarea controalelor de pe form la tabelul din dataset
- *BindingNavigator* - parcurgerea înregistrărilor din tabel
- *TableAdapter* - comunicarea între baza de date și dataset
- *TableAdapterManager* - controlul ordinii operațiilor de insert, update, delete pentru toate componentele *TableAdapter*

Constrângeri

- 2 tipuri de constrângeri
- Constrângere unique - verifică dacă noile valori dintr-o coloană sunt unice în tabel
- Constrângere foreign-key - definește reguli pentru modificarea înregistrărilor copil asociate la modificarea / ștergerea unei înregistrări în tabelul părinte
- La crearea unui obiect *DataRelation* într-un dataset, se adaugă automat o constrângere foreign-key

Constrângeri

- Constrângerile sunt implementate ca obiecte de tip *UniqueConstraint* sau *ForeignKeyConstraint* și adăugate colecției *Constraints* a unui *DataTable*
- Dataset-ul are o proprietate booleană *EnforceConstraints* care precizează dacă restricțiile sunt impuse sau nu (implicit, este adevărat)

Entity Framework

- Set de tehnologii în ADO.NET
- Susține dezvoltarea aplicațiilor orientate către date
- Permite dezvoltatorilor să lucreze cu date sub forma obiectelor și proprietăților specifice domeniului de aplicație, fără să se mai ocupe de tabelele (coloanele) care stochează aceste date
- Dă viață modelelor, permițând dezvoltatorilor să interogheze entități și relații din modelul domeniului, bazându-se pe Entity Framework pentru a traduce aceste operații în comenzi specifice sursei de date

Entity Framework

