

Seminar 1

ADO.NET

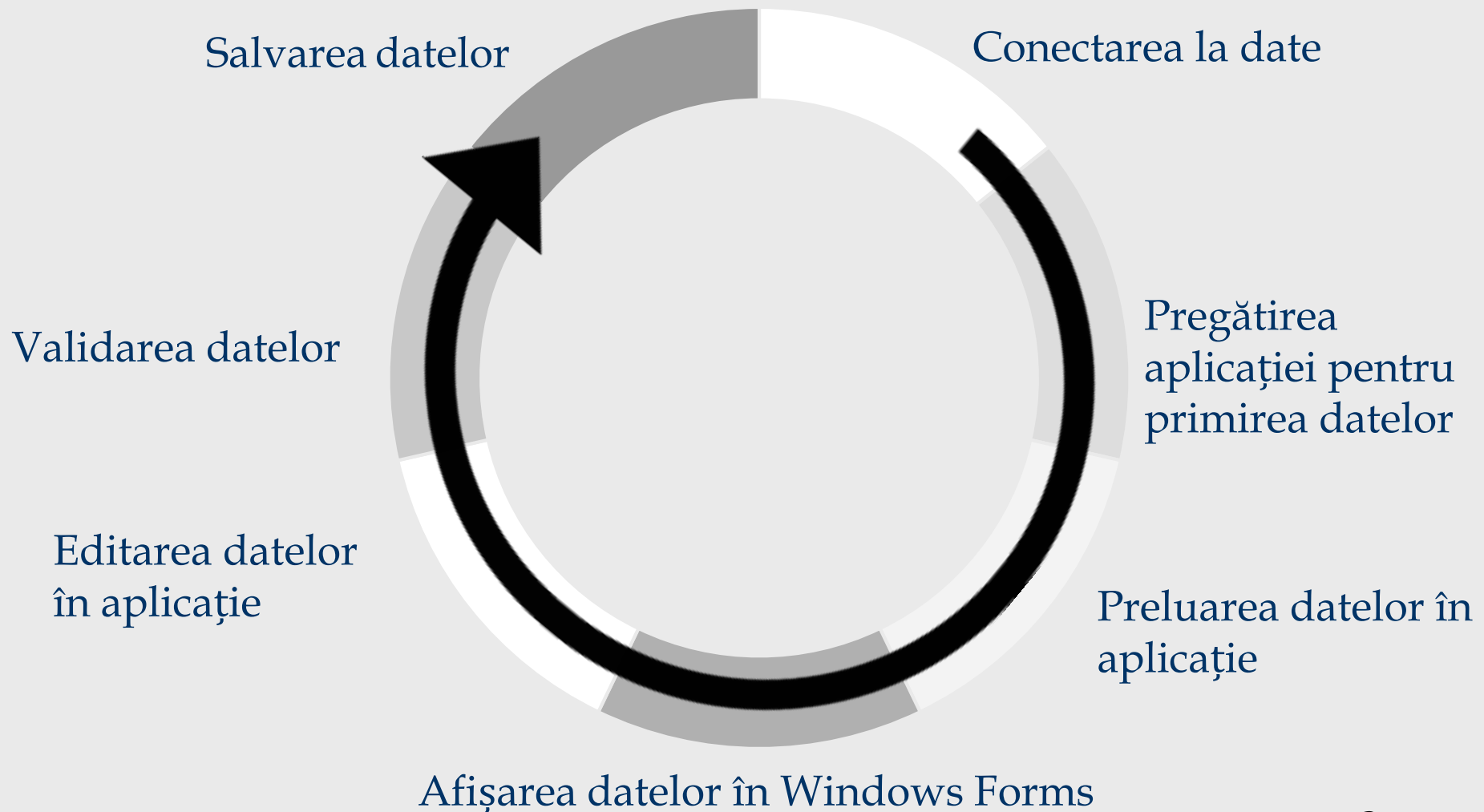
ADO.NET

ADO.NET este un set de clase care expun servicii de acces a datelor pentru programatorii .NET

ADO.NET:

- Oferă un set bogat de componente pentru crearea aplicațiilor distribuite, care partajează date
- Este o parte integrantă a .NET Framework, care oferă acces la date provenite din surse de date cum ar fi XML sau SQL Server
- Include .NET Framework data providers pentru conectare la baza de date, execuție comenzi și returnare a rezultatelor

Ciclul datelor



Ciclul datelor

- Conectarea la date: stabilește o comunicare bidirecțională între aplicație și serverul de baze de date (*TableAdapter,ObjectContext*)
- Pregătirea aplicației pentru primirea datelor: când se utilizează un model deconectat, anumite obiecte stochează datele temporar (*dataset-uri, entități, obiecte LINQ to SQL*)
- Preluarea datelor: execuția interogărilor și a procedurilor stocate (*TableAdapters, LINQ to Entities, conexiune directă între entități și procedurile stocate*)

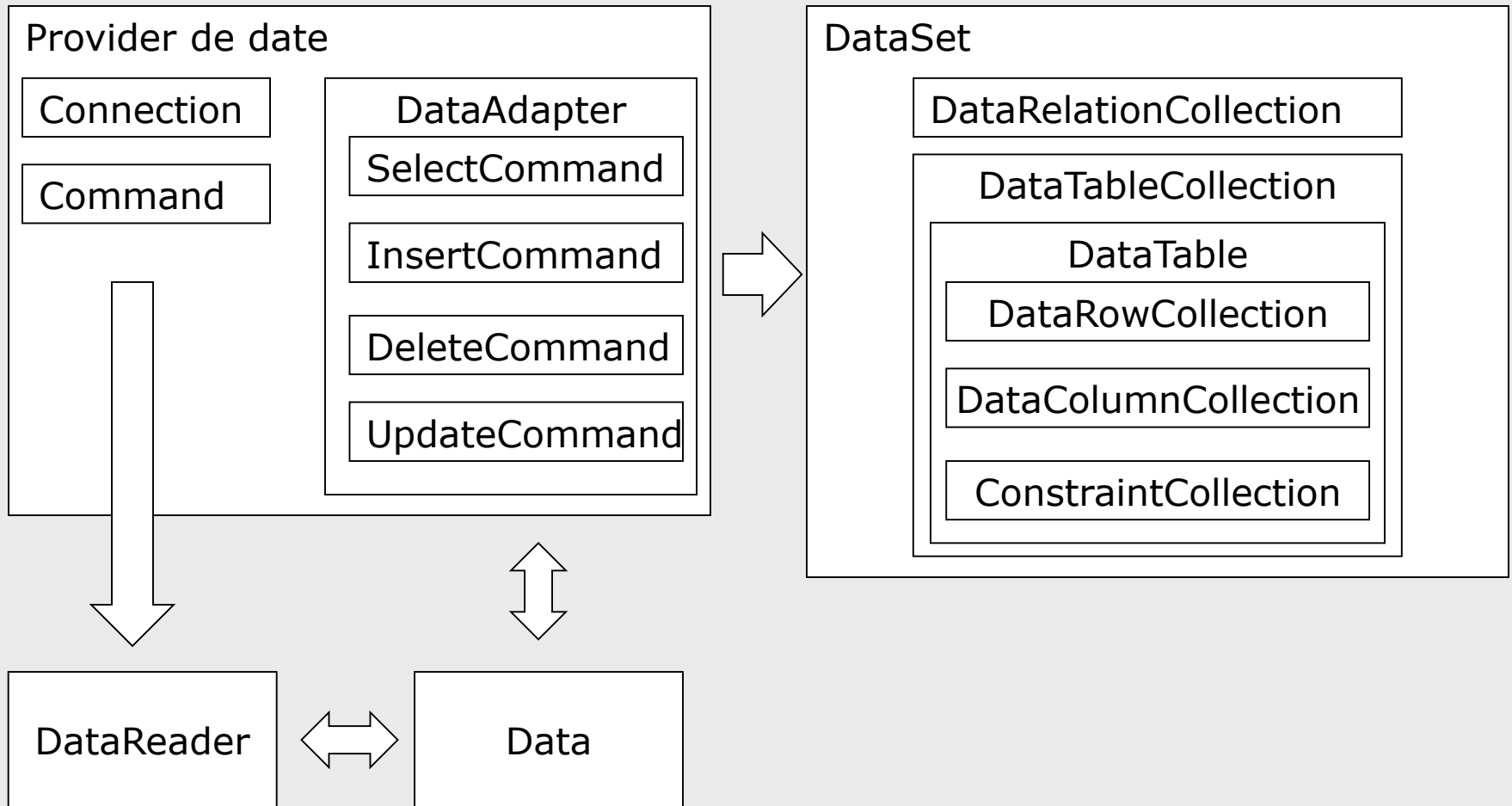
Ciclul datelor

- Afișarea datelor: controale conectate la date (*data-bound*)
- Editarea și validarea datelor: adăugarea / modificarea / ștergerea înregistrărilor și verificarea noilor valori; acestea din urmă trebuie să îndeplinească cerințele aplicației
- Salvarea datelor: *TableAdapterManager*, *SaveChanges*
 - **TableAdapterManager** – o componentă care oferă posibilitatea de a salva modificările făcute în *data tables* conectate (determină ordinea corectă a operațiilor de adăugare, actualizare și ștergere astfel încât să se evite încălcarea cheilor străine definite)

Modele de date

- Dataset-uri tipizate/netipizate (*typed / untyped*)
- Model conceptual bazat pe *Entity Data Model*. Poate fi utilizat de *Entity Framework* sau *WCF Data Services*
- Clase *LINQ to SQL*

ADO.NET



DataSet

- Obiect care conține tabele care pot stoca temporar datele utilizate în aplicație
- Tipizat / netipizat
- Cache local, în memorie
- Funcționează și dacă aplicația se deconectează de la baza de date
- Structură similară cu cea a unei baze de date relaționale (are tabele, rânduri, coloane, constrângeri, relații)

DataSet

■ Proprietăți:

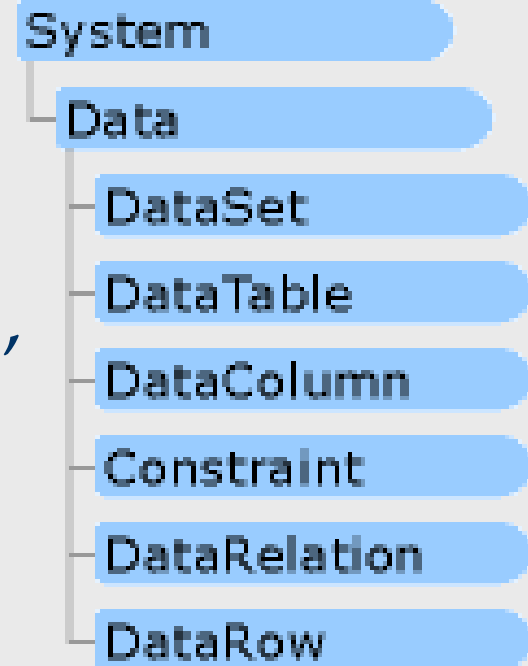
- *Tables* (*DataTableCollection*) – colecție de tabele din DataSet
- *Relations* (*DataRelationCollection*) – colecție de relații (tabele copil / părinte)

■ Metode:

- *Clear()* – elimină toate rândurile din toate tabelele
- *HasChanges()* – indică dacă există rânduri noi, șterse sau modificate

DataSet

- Clasa *DataTable* - proprietăți
 - *Rows* (*DataRowCollection*),
Columns (*DataColumnCollection*),
ChildRelations și *ParentRelations*
(*DataRelationCollection*), etc
- Clasa *DataRow* - proprietatea
RowState (valori: *Added*, *Deleted*,
Modified, *Unchanged*)



SqlConnection

- Reprezintă o conexiune deschisă la baza de date
- Nu poate fi moștenită
- Dacă iese din domeniul de vizibilitate, nu este închisă - conexiunea trebuie închisă explicit (*Close*)
- Proprietăți:
 - *ConnectionString* - șir utilizat pentru deschiderea unei baze de date SQL Server (<http://www.connectionstrings.com/>)
 - *ConnectionTimeout* - timpul de așteptare pentru stabilirea unei conexiuni; la expirare, încercarea de stabilire a conexiunii se termină și se generează o eroare

SqlConnection

- Metode:

- *Open()*

- *Close()*

- Dacă se generează *SQLException*, *SqlConnection* rămâne deschisă când nivelul de gravitate ≤ 19

SqlCommand

- Reprezintă o instrucțiune sau procedură stocată Transact-SQL care se dorește a fi executată pe o bază de date SQL Server
- Proprietăți
 - *CommandText*
 - *CommandTimeout*
- Metode
 - *ExecuteNonQuery* – returnează numărul de rânduri afectate
 - *ExecuteScalar* – returnează valoarea primei coloane a primului rând

SqlCommand

- *ExecuteReader* – construiește un *SqlDataReader*

SqlDataReader

- Citește un flux de rânduri dintr-o bază de date SQL Server; acesta nu poate decât să înainteze (*forward-only*)

SqlDataAdapter

- Punte între DataSet și SQL Server pentru obținerea și salvarea datelor
- Proprietăți:
 - *UpdateCommand* - instrucțiune sau procedură stocată utilizată la modificarea înregistrărilor în sursa de date
 - *InsertCommand* - instrucțiune sau procedură stocată pentru inserarea înregistrărilor în sursa de date
 - *DeleteCommand* - instrucțiune sau procedură stocată pentru ștergerea înregistrărilor

SqlDataAdapter

■ Metode:

- *Fill(DataSet, String)* - adaugă sau reîncarcă rândurile în obiectul DataSet, astfel încât acestea să corespundă celor din sursa de date
- *Update(DataSet, String)* - modifică valorile din baza de date, executând instrucțiunile INSERT, UPDATE sau DELETE pentru fiecare rând adăugat, modificat sau șters

Console

- Reprezintă fluxurile standard input, output, respectiv de eroare pentru aplicațiile de consolă
- Proprietăți:
 - *WindowLeft, WindowTop, WindowHeight, WindowWidth, BackgroundColor, Title* etc
- Metode:
 - *Write(...), WriteLine(...)*
 - *Read(), ReadLine(), ReadKey()*

Data binding

- DataBindings – proprietate a unui control care dă posibilitatea de a lega orice proprietate a controlului respectiv de o proprietate a unui obiect
- Cu ajutorul data binding-ului, valorile proprietăților a două obiecte diferite pot fi sincronizate
- Clasa Binding – reprezintă legarea simplă dintre valoarea unei proprietăți a unui obiect și valoarea unei proprietăți a unui control

`Binding(String propertyName, Object dataSource, String dataMember)`

- Pentru a simplifica data binding-ul, Windows Forms permite legarea sursei de date de o componentă BindingSource, la care sunt legate apoi controale