

# Tehnici Avansate de Programare

LINQ, Entity Framework & Repository Pattern

---

Petru Rebeja, Marius Apetrii

2 Aprilie 2020

Facultatea de Matematică

Universitatea Alexandru Ioan Cuza, Iași

# Introdurre

---

# Recapitulare

- Delegates
- Expresii lambda
- Attribute
- Extension methods

# Despre ce vom discuta azi

- LINQ
- Entity Framework
- Repository Pattern

LINQ

---

# Ce este LINQ

- LINQ = Language Integrated Query.
- O mulțime de funcții ale platformei .net și particularități ale limbajului de programare care permit interogarea uniformă atât a colecțiilor de date din memorie cât și a surselor de date externe<sup>1</sup>.

---

<sup>1</sup>Joseph Albahari and Ben Albahari. 2012. C# 5.0 in a Nutshell: The Definitive Reference (5th. ed.). O'Reilly Media, Inc.

Interogările LINQ pot fi scrise folosind două tipuri de sintaxă:

- Sintaxa de interogare (`Query syntax`),
- Sintaxa pe bază de înlănțuire de funcții (`Fluent syntax`).

- O sintaxă simplificată.
- Nu este o transpunere a sintaxei SQL în C#.
- A fost inspirată în mare parte de limbajele funcționale (LISP, Haskell); SQL a avut o influență mai mult cosmetică<sup>1</sup>.



## Query syntax — exemplu

```
var numbers = new int[]{1, 2, 3, 4};
```

```
var odd = from n in numbers  
          where n % 2 == 1  
          orderby n descending  
          select n;
```

- Sintaxa fundamentală și cea mai flexibilă.
- Reprezintă o înlănțuire a operatorilor (metode de extensiune care operează pe colecții de elemente).

# Fluent syntax — exemplu

```
var numbers = new int[]{1, 2, 3, 4};
```

```
var odd = numbers  
    .Where(n => n % 2 == 1)  
    .OrderByDescending(n => n);
```

# Operatori standard

Platforma .net pune la dispoziție o mulțime de **operatori LINQ standard** în clasa `Enumerable`.

Exemple:

- `Where()`
- `Select()`
- `Average()`
- `GroupBy()`
- etc.

- O interogare LINQ este executată abia atunci când începem să parcurgem mulțimea-rezultat.
- Excepții:
  - Operatorii care au ca rezultat un scalar: `First()`, `Count()` etc.,
  - Operatorii care convertesc într-o colecție: `ToList()` etc.

# Execuție întârziată — exemplu<sup>1</sup>

```
var numbers = new List<int>(){1};  
var multiples = numbers.Select(n => n * 10);  
numbers.Add(2);  
  
foreach(var n in multiples)  
{  
    Console.Write($"{n}|");  
}
```

**Rezultat**  
10|20|

# Entity Framework

---

# Ce este un ORM

## Object-Relational Mapping

Este o tehnică prin care interogarea și manipularea datelor dintr-o bază de date se face folosind paradigma orientat-obiect<sup>2</sup>.

---

<sup>2</sup><https://stackoverflow.com/a/1279678/844006>



## **Entity Framework**

Este o bibliotecă pentru platforma .net care oferă funcționalitatea ORM.

Entity Framework oferă două moduri de a modela baza de date:

1. Din cod — programatorul definește asocierile dintre clase și tabele precum și relațiile dintre diferite tabele.
2. Folosind interfața grafică — utilizatorul definește schema bazei de date cu ajutorul unei interfețe grafice iar codul este generat de platformă.

## Alegerea fluxului de lucru<sup>3</sup>

	Prefer să scriu cod	Prefer interfață grafică
Bază de date nouă	Code First	Model First
Bază de date existentă	Code First	Database First

---

<sup>3</sup><https://docs.microsoft.com/en-us/ef/ef6/modeling/#ef-workflows>

## **Modelare**

Este procedeul prin care definim tipurile de obiecte și relațiile dintre ele a.î. să reprezinte procesele din viața reală pe care le simulează aplicația curentă.

## **Modelul bazei de date**

Totalitatea claselor și altor tipuri de date care reprezintă structura bazei de date a aplicației.

## Data context / DbContext

- Reprezintă o sesiune pentru interogarea și manipularea datelor din baza de date.
- Interfață pentru acces la date și proprietăți conexe (conexiune, tranzacții etc.).

## Entity

- Denumire generică pentru fiecare tip de date din modelul bazei de date.

## Relație / Navigational Property

- O proprietate a unei entități prin care se modelează relația cu altă entitate,
- Ex: `Student.Classes` este o listă ce modelează relația 1:N dintre entitățile `Student` și `Class`.



## CRUD

- Un acronim care denotă cele 4 funcții ale unui model de date:  
Create, Read, Update, Delete.

## Eager vs Lazy loading

- Implicit Entity Framework va încărca entitățile relaționate atunci când acestea sunt parcurse (Lazy loading).
- Acest comportament poate fi modificat a.î. entitățile relaționate să fie încărcate în același timp cu entitatea principală (Eager loading).

# Eager vs Lazy loading

- Lazy loading scade presiunea asupra bazei de date dar necesită o conexiune activă la parcurgere,
- Eager loading — invers.

# Repository Pattern

---

- Șablon de proiectare care decuplează logica aplicației de accesul la date.
- Interfață pentru a implementa operațiunile CRUD pentru o anumită entitate.

- Decuplează logica aplicației de accesul la date,
- Decuplează aplicația de modificările aplicate bazei de date,
- Promovează reutilizarea codului,
- Logica aplicației devine mai ușor de testat.

```
public interface IStudentRepository
{
    Student GetById(int studentId);
    void Update(Student student);
    void Insert(Student student);
    void Delete(int studentId);
    IQueryable<Student> Query();
    void Save();
}
```

# Înceiere

---



# Recapitulare

LINQ

## LINQ

O mulțime de funcții ce permit interogarea uniformă a colecțiilor de date din diverse surse.

## LINQ

O mulțime de funcții ce permit interogarea uniformă a colecțiilor de date din diverse surse.

## Entity Framework

## **LINQ**

O mulțime de funcții ce permit interogarea uniformă a colecțiilor de date din diverse surse.

## **Entity Framework**

O platformă care permite interogarea și manipularea datelor folosind paradigma orientat-obiect.

## **LINQ**

O mulțime de funcții ce permit interogarea uniformă a colecțiilor de date din diverse surse.

## **Entity Framework**

O platformă care permite interogarea și manipularea datelor folosind paradigma orientat-obiect.

## **Repository Pattern**

## **LINQ**

O mulțime de funcții ce permit interogarea uniformă a colecțiilor de date din diverse surse.

## **Entity Framework**

O platformă care permite interogarea și manipularea datelor folosind paradigma orientat-obiect.

## **Repository Pattern**

Șablon de proiectare ce decuplează logica aplicației de accesul la date.

# Vă mulțumesc!

Mulțumesc pentru atenție!