

# Curs08

## Curs08

Metode extinse (Extension methods)

Introducere în LINQ to Objects

## Metode extinse (Extension methods)

O metoda care **extinde** metodele publice deja existente pentru o anumită clasă căreia nu i se poate modifica codul sursă. (Este într-o librărie la care nu avem acces).

Se declară într-o clasă statică, de tip static. Sintaxă generală:

```
public static Extensions {  
    public static [tip_returnat] MethodName(this NumeClasaExtinsa numeRef,  
        [lista_param]) {  
        //...  
    }  
}
```

Sintaxă de apelare: `numeRef.MethodName([lista_param]);`

### Exemplu

1. Să se extindă clasa `string` (deja existentă) cu o metodă care verifică câte cifre apar în stringul respectiv.

```
public static class Extensions {  
    public static int GetDigitsCount(this string textValue) {  
        if (string.IsNullOrEmpty(textValue))  
            return 0;  
        int count = 0;  
        foreach(var c in textValue)  
            if (char.IsDigit(c))  
                count++;  
        return count;  
    }  
}  
  
class Program {  
    static void Main(string[] args) {  
        string s = "adasfasr3443f afa 45 ";  
        var nr = s.GetDigitsCount();  
  
        Console.WriteLine("Numarul de cifre: ", nr);  
        Console.ReadKey();  
    }  
}
```

2. Se consideră clasa deja existentă `List<T>`. Să se adauge o metodă extinsă care verifică dacă un obiect de tip `List<T>` este `null` sau este lista goală (`IsEmpty`).

```
public static bool IsNullOrEmpty<T> (this List<T> list) {
    if (list == null)
        return true;
    if (list.Count == 0)
        return true;
    return false;
}
```

3. Să se adauge clasei `List<T>` o metodă care verifică de câte ori apare o valoare dată în listă `GetOccurrences`.

```
public static int GetOccurrences<T> (this List<T> list, T value) {
    int result = 0;
    foreach(var v in list)
        if (v.Equals(value))
            result++;
    return result;
}

List<string> l = new List<string>() { "aasda", "as", "asda", "as", "asdasdas",
    "as" };
Console.WriteLine(l.GetOccurrences<string>("as"));
```

## Introducere în LINQ to Objects

`LINQ` (Language Integrated Query) permite interogarea colecțiilor de obiecte cu operatori asemănători cu cei din `SQL` (`SELECT`, `WHERE`, `ORDERBY`, ...).

Sunt implementați ca metode extinse în biblioteca `LINQ`:

```
List<string> list = new List<string>() { "Aasda", "as", "asda", "as", "Asdasdas",
    "as" };
var values = list.Where(x => char.IsUpper(x[0])).ToList();
```