Curs10

FullCode

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LINQ_Curs9 {
  public static class MyExtensionsOperators{
    public static List<T> MyWhere<T>(this List<T> source, Func<T, bool>
predicate) {
      var result = new List<T>();
      foreach(var elem in source)
        if(predicate(elem))
          result.Add(elem);
      return result;
    }

    public static List<TResult> MySelect<T, TResult>(this List<T> source, Func<T,
TResult> selector){
      var result = new List<TResult>();
      foreach(var elem in source)
        result.Add(selector(elem));
      return result;
    }
  }

  class Program {
    class Car {
      public string Model { get; set; }
      public int Year { get; set; }
      public string Color { get; set; }
    }
    class ColorModel {
      public string Model { get; set; }
      public string Color { get; set; }
    }
    class Predicates {
      public static bool P1(Car c) {
        return c.Color == "red" && c.Year> 2000;
      }
    }

    static void ProcesarePagini() {
      List<int> values = new List<int>();
      for(int i = 0; i<10000; i++)
    values.Add(i);
   // Procesare in pagini de cate 100
   var pageIndex = 0;
   var pageSize = 100;
   do {
    var currentPage = values.Skip(pageIndex *
pageSize).Take(pageSize).ToList();
    if(currentPage.Count == 0)
     break;
    // Procesez pagina curenta
    Console.WriteLine(string.Join(",", currentPage));
    pageIndex++;
   } while(true);
  }

  static void Main(string[] args) {
   ProcesarePagini();
   Console.ReadKey();
   return;

   // Operatorul WHERE
   // List<int> values = new List<int>() { 3, 2, 5, 6, 7, 8, 35, 7 };
   // var values1 = values.Where(x => x % 2 == 0).ToList();

   List<Car> cars = new List<Car>() {
    new Car() { Model = "bmw", Year = 2000, Color = "red" },
     new Car() { Model = "toyota", Year = 2018, Color = "black" },
     new Car() { Model = "mercedes", Year = 2016, Color = "red" },
     new Car() { Model = "dacia", Year = 2013, Color = "black" },
     new Car() { Model = "bmw", Year = 2012, Color = "blue" },
     new Car() { Model = "bmw", Year = 2010, Color = "green" }
   };

   // GroupBy
   var groups = cars.GroupBy(x => x.Color).ToList();

   // FirstOrDefault
   var first2012 = cars.First(x => x.Year == 2011);

   // Any
   var condition = cars.Any(x => x.Year> 2015);

   // Procesarea paginata a unei colectii cu multe elemente('Skip' si 'Take')
   // Func<Car, bool> f = Predicates.P1;
   // var c1 = cars.MyWhere(x => x.Model == "bmw");
   // Selecteaza anii de fabricatie pentru masinile de culoare rosie
   var cars1 = cars
    .Where(x => x.Color == "red")
    .Select(x => new ColorModel() { Color = x.Color, Model = x.Model })
    .ToList();

   // var years = cars.Select(x => x.Year).OrderByDescending(x => x).ToList();
   var cars2 = cars.OrderByDescending(x => x.Year).ToList();
   var distinctColors = cars.Select(x => x.Color).Distinct().ToList();
   var maxYear = cars.Max(x => x.Year);

   Console.ReadKey();
  }
  }
}
```