var dotaz = from z in zoo

where (z.Jmeno.StartsWith("L") && z.Vaha > 150)

group z by z.Vaha into grouped

select new { Weight = grouped.Key, Animals = grouped.Select(a => a.Jmeno) };

foreach (var group in dotaz)

{

Console.WriteLine($"Weight: {group.Weight}");

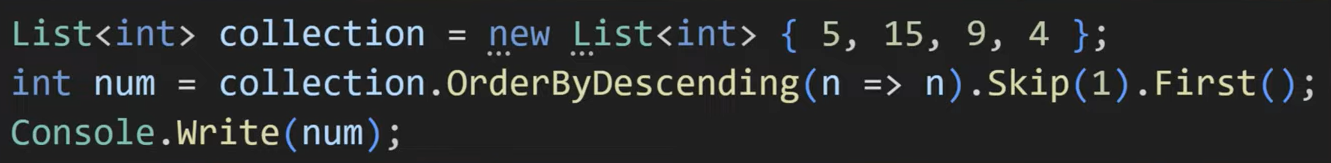
foreach (var name in group.Animals)

{

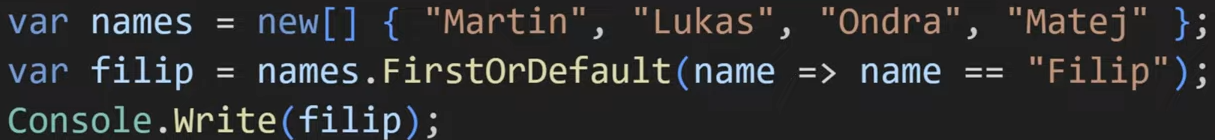
Console.WriteLine($" - Name: {name}");

}

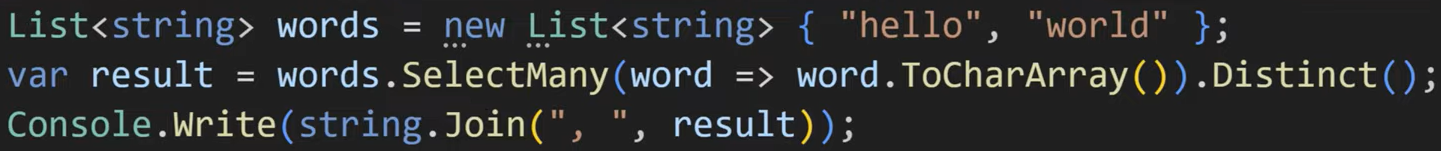
}



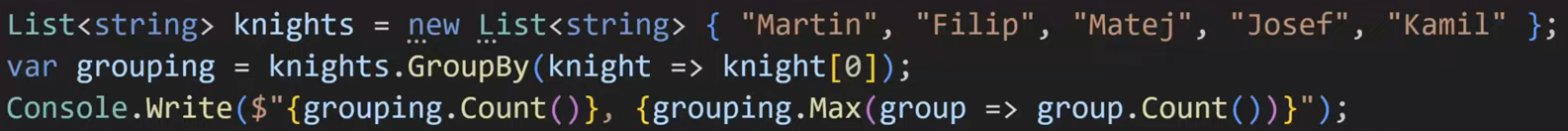
Seřadí sestupně, první vynechá (15) a vypíše zbývající první => 9



Žádné jméno nesplňuje podmínku, díky FirstOrDefault nedojde k exception ale výsledek je null a nevypíše se nic



ToCharArray() rozdělí slovo na seznam (list) jednotlivých znaků a Distinct odstraní duplicity a Join vloží čárku mezi každé dva prvky seznamu (za poslední už ne) => h, e, l, o, w, r, d



Seznam jmen (stringů) chceme seskupit dle prvního písmena, grouping.Count() spočítá počet skupin (4 tj. 4 různá počáteční písmena) a grouping.Max(group => group.Count()) najde maximum výskytů (počtů) v rámci dané skupiny (tj. 2 jména od M) obdobně jako grouping.Select(group => group.Count()).Max() => 4, 2

var booksByAuthor = nazevKnihovny.GroupBy(book => book.Author)

                                .Select(group => new { Author = group.Key, Count = group.Count() });

foreach (var group in booksByAuthor)

        {

            Console.WriteLine($"Autor {group.Author}: {group.Count} knih");

        }

To stejné pomocí LINQ query syntax, který se podobá SQL zápisu:

var booksByAuthor =

from book in nazevKnihovny

group book by book.Author into authorGroup

select new

{

Author = authorGroup.Key,

Count = authorGroup.Count()

};

// Počet unikátních slov v názvech knih

        var uniqueWords = nazevKnihovny

            .SelectMany(book => book.Title.Split(new char[] { ' ', ',', '.', '!', '?', ';', ':' }, StringSplitOptions.RemoveEmptyEntries))

            .Select(word => word.ToLower())

            .Distinct();

        Console.WriteLine($"Počet unikátních slov v názvech: {uniqueWords.Count()}");

To stejné pomocí LINQ query syntax, který se podobá SQL zápisu:

var uniqueWords = (

from book in nazevKnihovny

from word in book.Title.Split(new char[] { ' ', ',', '.', '!', '?', ';', ':' }, StringSplitOptions.RemoveEmptyEntries)

select word.ToLower()

).Distinct();

// 4. Seřaďte jména vzestupně

        List<string> jmena = new List<string>()

        {

            "Hana", "Jaroslav", "Xenie", "Michaela", "Borivoj", "Nela

        };

// 4. Řešení

        List<string> vzestupne = null;

        //vzestupne = jmena.Sort().ToList; //nefunguje

vzestupne = jmena.OrderBy(j => j).ToList(); //funguje

        foreach (string text in vzestupne)

        {

            Console.WriteLine(text);

        }

// Definice Banky

public class Banka

{

    public string Symbol { get; set; }

    public string Jmeno { get; set; }

}

// Definice Zákazníka

public class Zakaznik

{

    public string Jmeno { get; set; }

    public double Zustatek { get; set; }

    public string Banka { get; set; }

}

// Definice Skupiny milionářů

public class SkupinaMilionaru

{

    public string Banka { get; set; }

    public IEnumerable<string> Milionari { get; set; }

}

public class Program

{

    public static void Main()

    {

// 7. Zobrazte vsechny milionare v kazde bance např.:

        // CS: Jan Novak a Josef Novotny

        // KB: Jana Nova

        List<Zakaznik> zakaznici = new List<Zakaznik>() {

            new Zakaznik(){ Jmeno="Jan Maly", Zustatek=10345.50, Banka="CS"},

            new Zakaznik(){ Jmeno="Jiri Hladny", Zustatek=452.10, Banka="KB"},

            new Zakaznik(){ Jmeno="Lenka Sporiva", Zustatek=523665.13, Banka="CS"},

            new Zakaznik(){ Jmeno="Marie Bohata", Zustatek=7482184.38, Banka="FIO"},

            new Zakaznik(){ Jmeno="Michal Marny", Zustatek=745234.93, Banka="KB"},

            new Zakaznik(){ Jmeno="Lada Vychytraly", Zustatek=8832937.34, Banka="CS"},

            new Zakaznik(){ Jmeno="Sandra Nedostatecna", Zustatek=942488.48, Banka="KB"},

            new Zakaznik(){ Jmeno="Silvie Zavodou", Zustatek=56198334.72, Banka="FIO"},

            new Zakaznik(){ Jmeno="Tereza Presna", Zustatek=1000000.00, Banka="CITI"},

            new Zakaznik(){ Jmeno="Stefan Pilny", Zustatek=48282.73, Banka="CITI"}

        };

        // 7. Řešení

        List<SkupinaMilionaru> skupinyPodleBanky = null;

        skupinyPodleBanky = zakaznici

                    .Where(z => z.Zustatek >= 1000000)

                    .GroupBy(z => z.Banka)

                    .Select(g => new SkupinaMilionaru

                    {

                        Banka = g.Key,

                        Milionari = g.Select(z => z.Jmeno)

                    })

                    .ToList();

        //nebo

        List<SkupinaMilionaru> skupinyPodleBanky2 = null;

        skupinyPodleBanky2 =

                    (from zak in zakaznici

                     where zak.Zustatek >= 1000000

                     group zak by zak.Banka into g

                     select new SkupinaMilionaru

                     {

                         Banka = g.Key,

                         Milionari = from z in g

                                     select z.Jmeno

                     }).ToList();

        foreach (var skupina in skupinyPodleBanky)

        {

            Console.WriteLine($"{skupina.Banka}: {string.Join(" a ", skupina.Milionari)}");

        }

// 8. Vytisknete jmeno kazdeho milionare a jeho banky např.

        // Jan Novak v Ceska Sporitelna

        // Josef Novotny v Komercni Banka

        List<Banka> banky = new List<Banka>() {

            new Banka(){ Jmeno="Ceska Sporitelna", Symbol="CS"},

            new Banka(){ Jmeno="Komercni Banka", Symbol="KB"},

            new Banka(){ Jmeno="Fio Banka", Symbol="FIO"},

            new Banka(){ Jmeno="Citibank", Symbol="CITI"},

        };

        // 8. Řešení

        List<Zakaznik> reportMilionaru = null;

        reportMilionaru = (

                    from zak in zakaznici

                    join banka in banky on zak.Banka equals banka.Symbol

                    where zak.Zustatek >= 1000000

                    select new Zakaznik

                    {

                        Jmeno = zak.Jmeno,

                        Banka = banka.Jmeno,  // Použijeme plný název banky

                        Zustatek = zak.Zustatek

                    }

                    ).ToList();

        foreach (Zakaznik zakaznik in reportMilionaru)

        {

            Console.WriteLine(zakaznik.Jmeno + " v " + zakaznik.Banka);

        }

    }

}