SEW-Übungen

LInq - Abfragen

Inhaltsverzeichnis

[Aufgabenstellung 3](#_Toc102761478)

[Aufgaben 3](#_Toc102761479)

[Testfälle 4](#_Toc102761480)

[Average 4](#_Toc102761481)

[Count 4](#_Toc102761482)

[Max 4](#_Toc102761483)

[Min 4](#_Toc102761484)

[Sum 4](#_Toc102761485)

[First 5](#_Toc102761486)

[Last 5](#_Toc102761487)

[Single 5](#_Toc102761488)

[Lösungen 5](#_Toc102761489)

[Average 5](#_Toc102761490)

[}Count 5](#_Toc102761491)

[Max 6](#_Toc102761492)

[Min 6](#_Toc102761493)

[}Sum 6](#_Toc102761494)

[First 7](#_Toc102761495)

[Last 7](#_Toc102761496)

[SIngle 7](#_Toc102761497)

# Aufgabenstellung

Erstelle zwei Abfragen jeweils für Average, Count, Max, Min, Sum, First, Last und Single.

# Aufgaben

1. Geben Sie den Gesamtdurchschnitt von dem Preis jedes Kleidungsstückes aus.
2. Geben Sie den Preisdurchschnitt von den Produkten aus, welche im Gefrierfach gelagert werden.
3. Geben Sie den Preisdurchschnitt der Dekoration aus.
4. Geben Sie die Anzahl der Kleidungsstücke aus, welche lieferbar sind.
5. Geben Sie aus wie viele Metalldinge es im Möbelgeschäft gibt aus.
6. Geben Sie den Maximum Preis der Bioprodukte aus.
7. Geben Sie die Maximalhöhe der Holztische aus.
8. Geben Sie den Minimalpreis der Bademode aus.
9. Geben Sie die Minimalbreite eines lieferbaren Möbelstücks aus.
10. Berechne die Summe des Preises der roten Kleidungsstücke.
11. Gebe die Summe des Preises der regionalen Produkte aus.
12. Geben Sie das erste Möbelstück aus, welches aus Plastik besteht.
13. Geben Sie das erste lieferbare Kleidungsstück aus.
14. Geben Sie das letzte Lebensmittel aus, welches in Körben gelagert wird.
15. Geben Sie das letzte Accessoire aus.
16. Geben Sie ein Kleidungsstück aus, welches schwarz ist.
17. Geben Sie ein Möbelstück aus, welches aus Glas besteht.

# Testfälle

### Average







### Count





### Max





### Min





### Sum





### First





### Last





### Single





# Lösungen

### Average

void AverageExample(List<ClothingStore> clothes)

{

var avg = clothes.Average(c => c.Price);

Console.WriteLine(avg);

}

void AverageExample2(List<GroceryStore> grocery)

{

var avg = (grocery.Where(g => g.Storage == EStorage.FREZZER)).Average(g => g.Price);

Console.WriteLine(avg);

}

void AverageExample3(List<FurnitureStore> furniture)

{

var avg = (from f in furniture

where f.Type == ETypes.DECO

select f).Average(f => f.Price);

Console.WriteLine(avg);

### }Count

void CountExample1(List<ClothingStore> clothes)

{

var count = (from c in clothes

where c.IsDelieverable == true

select c).Count();

Console.WriteLine(count);

}void CountExample2(List<FurnitureStore> furniture)

{

var result = furniture.Count(f => f.Material == EMaterial.METAL);

Console.WriteLine(result);

}

### Max

void MaxExample1(List<GroceryStore> grocery)

{

var max = (from g in grocery

where g.IsBIO == true

select g).Max(g => g.Price);

Console.WriteLine(max);

}

void MaxExample2(List<FurnitureStore> furniture)

{

var max = (from f in furniture

where f.Type == ETypes.TABLE && f.Material == EMaterial.WOOD

select f).Max(f => f.Height);

Console.WriteLine(max);

}

### Min

void MinExample1(List<ClothingStore> clothes)

{

var min = (clothes.Where(c => c.Categories == ECategories.BEACHWEAR)).Min(clothes => clothes.Price);

Console.WriteLine(min);

}

void MinExample2(List<FurnitureStore> furniture)

{

var min = (from f in furniture

where f.IsDelieverable == true

select f).Min(f => f.Width);

Console.WriteLine(min);

### }Sum

void SumExample1(List<ClothingStore> clothes)

{

var sum = (from c in clothes

where c.Color == EColor.RED

select c).Sum(c => c.Price);

Console.WriteLine(sum);

}

void SumExample2(List<GroceryStore> grocery)

{

var sum = (grocery.Where(r => r.Region == "Austria")).Sum(r => r.Price);

Console.WriteLine(sum);

}

### First

void FirstExample1(List<FurnitureStore> furniture)

{

var first = (from f in furniture

select f).First(f => f.Material == EMaterial.PLASTIC);

Console.WriteLine(first.Name);

}

void FirstExample2(List<ClothingStore> clothes)

{

var first = clothes.First(c => c.IsDelieverable == true);

Console.WriteLine(first.Name);

}

### Last

void LastExample1(List<GroceryStore> grocery)

{

var last = (from g in grocery select g).Last(g => g.Storage == EStorage.BASKET);

Console.WriteLine(last.Name);

}

void LastExample2(List<ClothingStore> clothes)

{

var last = clothes.Last(c => c.Categories == ECategories.ACCESSOIRES);

Console.WriteLine(last.Name);

}

### SIngle

void SingleExample1(List<ClothingStore> clothes)

{

var single = (from c in clothes select c).Single(c => c.Color == EColor.BLACK);

Console.WriteLine(single.Name);

}

void SingleExample2(List<GroceryStore> grocery)

{

var single = grocery.Single(g => g.Storage == EStorage.SHELF);

Console.WriteLine(single.Name);

}