12.4-Lambda expressions

Program

```
using System;

using System.Collections.Generic;


class Product

{

    public int Id { get; set; }

    public string Name { get; set; }

    public decimal Price { get; set; }

}


class Program

{

    static void Main()

    {

        List<Product> products = new List<Product>

        {

            new Product { Id = 1, Name = "Laptop", Price = 999.99M },

            new Product { Id = 2, Name = "Smartphone", Price = 699.99M },

            new Product { Id = 3, Name = "Tablet", Price = 399.99M },

            new Product { Id = 4, Name = "Headphones", Price = 149.99M },

        };


        Console.WriteLine("Original list of products:");

        DisplayProducts(products);
        {
            // Sorting products by name using selection sort
            SelectionSortByName(products);
            Console.WriteLine("\nSorted list of products by name:");
            DisplayProducts(products);

            // Sorting products by price using selection sort
            SelectionSortByPrice(products);
            Console.WriteLine("\nSorted list of products by price (descending):");
            DisplayProducts(products);
        }

        static void DisplayProducts(List<Product> products)
        {
            foreach (var product in products)
            {
                Console.WriteLine($"ID: {product.Id}, Name: {product.Name}, Price: {product.Price:C}");
            }
            Console.WriteLine();
        }

        static void SelectionSortByName(List<Product> products)
        {
            for (int i = 0; i < products.Count - 1; i++)
            {
                int minIndex = i;
                for (int j = i + 1; j < products.Count; j++)
                {
                    if (string.Compare(products[j].Name, products[minIndex].Name,
    StringComparison.OrdinalIgnoreCase) < 0)
                    {
                        minIndex = j;
                    }
                }
                if (minIndex != i)
                {
                    Product temp = products[i];
                    products[i] = products[minIndex];
                    products[minIndex] = temp;
                }
            }
        }

        static void SelectionSortByPrice(List<Product> products)
        {
            for (int i = 0; i < products.Count - 1; i++)
            {
                int maxIndex = i;
                for (int j = i + 1; j < products.Count; j++)
                {
                    if (products[j].Price > products[maxIndex].Price)
                    {
                        maxIndex = j;
                    }
                }
                if (maxIndex != i)
                {
                    Product temp = products[i];
                    products[i] = products[maxIndex];
                    products[maxIndex] = temp;
                }
            }
        }
    }
```

Exercise

Write a C# program that manages a list of products with their respective IDs, names, and prices.

It demonstrates sorting products by name and price in descending order using selection sort algorithms.

Contains properties for ID, name, and price.

Initializes a list of products with sample data.

Displays the original list of products.

Sorts the products by name using the SelectionSortByName method.

Displays the sorted list of products by name.

Sorts the products by price in descending order using the SelectionSortByPrice method.

Displays the sorted list of products by price.

Helper method to display the details of each product in the list.

Sorts the list of products by name using the selection sort algorithm.

Compares product names using case-insensitive comparison.

Swaps products based on name comparisons to arrange them in ascending order.

Sorts the list of products by price in descending order using the selection sort algorithm.

Compares product prices and swaps products accordingly to arrange them in descending order of price.

Utilizes the selection sort algorithm for simplicity.

Selection sort repeatedly selects the minimum (or maximum) element from the unsorted portion of the list and moves it to the sorted portion.

Hint

Analyze the SelectionSortByName method, which implements the selection sort algorithm to sort the list of products by name in ascending order.

Pay attention to the comparison logic used to compare product names and the swapping mechanism to rearrange the products.

Examine the SelectionSortByPrice method, which sorts the list of products by price in descending order using the selection sort algorithm.

Understand how product prices are compared and products are swapped to arrange them in descending order of price.

Explanation

Program sorting a list of products by name and price using the selection sort algorithm. Let's break down the program and explain it in a paragraph:

The program begins by defining a Product class with properties Id, Name, and Price. It also contains the Program class with the Main method serving as the entry point. Inside Main, a list of Product objects is created and initialized with sample data representing different products. The original list of products is displayed to the console using the DisplayProducts method.

The program then sorts the products by name using the SelectionSortByName method, which implements the selection sort algorithm. This algorithm iterates through the list, finding the minimum element by comparing product names, and swaps elements accordingly. After sorting by name, the sorted list is displayed.

Similarly, the program sorts the products by price using the SelectionSortByPrice method, which applies the selection sort algorithm but compares product prices instead. The sorted list of products by price in descending order is displayed to the console.