Лабораторна робота №2

Тема: Використання методів розширень та узагальнень у C#.

Мета роботи: навчитися використовувати методи розширення та узагальнення у мові програмування C#.

Посилання на гіт: https://github.com/Swargon/DotNetLab2

Хід роботи:

Завдання:

Лістинг програми:

```
using System;
using System.Linq;
using System.Collections.Generic;

namespace Task1
{
    class Program
    {
        static void Main(string[] args)
        {
            // String section
            Console.WriteLine("Do you want to use the default string? (yes/no)");
            string response = Console.ReadLine() ?? "".ToLower();
            string inputString;
            if (response == "yes")
            {
                inputString = "Hello world";
                Console.WriteLine("Using default string: 'Hello world'");
            }
            else if (response == "no")
            {
                Console.WriteLine("Enter a string:");
                inputString = Console.ReadLine() ?? "";
            }
            else
            {
                Console.WriteLine("Invalid input. Using default string: 'Hello world'");
                inputString = "Hello world";
            }

            Console.WriteLine("Reverse the string");
            response = Console.ReadLine() ?? "".ToLower();
            {
                Console.WriteLine($"Reverse of inputString:
{inputString.ReverseString()}");
            }

            Console.WriteLine("Enter the character you want to count:");
            char charToCount = char.Parse(Console.ReadLine() ?? "");
            Console.WriteLine($"Count of '{charToCount}' in inputString:
{inputString.CountOccurrences(charToCount)}");

            // Integer array section
            Console.WriteLine("Do you want to use the default integer array? (yes/no)");
            response = Console.ReadLine() ?? "".ToLower();
            int[] intArray;
            if (response == "yes")
```

```
            {
                intArray = new int[] { 1, 2, 3, 4, 5, 9, 8, 7, 2 };
                Console.WriteLine("Using default integer array.");
            }
            else if (response == "no")
            {
                Console.WriteLine("Enter elements for the integer array (comma-
separated):");
                string input = Console.ReadLine() ?? "";
                intArray = input.Split(',').Select(int.Parse).ToArray();
            }
            else
            {
                Console.WriteLine("Invalid input. Using default integer array.");
                intArray = new int[] { 1, 2, 3, 4, 5, 9, 8, 7, 2 };
            }

            Console.WriteLine($"Array: [{string.Join(", ", intArray)}]");
            Console.WriteLine("Enter the element you want to count:");
            int elementToCount = int.Parse(Console.ReadLine() ?? "");
            Console.WriteLine($"Count of '{elementToCount}' in intArray:
{intArray.CountOccurrences(elementToCount)}");

            // Double array section
            Console.WriteLine("Do you want to use the default double array? (yes/no)");
            response = Console.ReadLine() ?? "".ToLower();
            double[] doubleArray;
            if (response == "yes")
            {
                doubleArray = new double[] { 1, 2, 3, 4, 5, 9, 8, 7, 2 };
                Console.WriteLine("Using default double array.");
            }
            else if (response == "no")
            {
                Console.WriteLine("Enter elements for the double array (comma-
separated):");
                string input = Console.ReadLine() ?? "";
                doubleArray = input.Split(',').Select(double.Parse).ToArray();
            }
            else
            {
                Console.WriteLine("Invalid input. Using default double array.");
                doubleArray = new double[] { 1, 2, 3, 4, 5, 9, 8, 7, 2 };
            }

            Console.WriteLine($"Array: [{string.Join(", ", doubleArray)}]");
            Console.WriteLine("Unique elements in doubleArray:");
            double[] uniqueDoubles = doubleArray.GetUniqueElements();
            foreach (double uniqueDouble in uniqueDoubles)
            {
                Console.Write($"{uniqueDouble} ");
            }
            Console.WriteLine();

            // String array section
            Console.WriteLine("Do you want to use the default string array? (yes/no)");
            response = Console.ReadLine() ?? "".ToLower();
            string[] stringArray;
            if (response == "yes")
            {
                stringArray = new string[] { "Dog", "Zebra","Boar", "Bacteria", "Stas",
"Boar", "Banana", "Stas" };
                Console.WriteLine("Using default string array.");
            }
            else if (response == "no")
```

```
            {
                Console.WriteLine("Enter elements for the string array (comma-
separated):");
                string input = Console.ReadLine() ?? "";
                stringArray = input.Split(',');
            }
            else
            {
                Console.WriteLine("Invalid input. Using default string array.");
                stringArray = new string[] { "Dog", "Zebra", "Boar", "Bacteria", "Stas",
"Boar", "Banana", "Stas" };
            }

            Console.WriteLine($"Array: [{string.Join(", ", stringArray)}]");
            Console.WriteLine("Unique elements in stringArray:");
            string[] uniqueStrings = stringArray.GetUniqueElements();
            foreach (string uniqueString in uniqueStrings)
            {
                Console.Write($"{uniqueString} ");
            }
            Console.WriteLine();
        }
    }

    public static class ArrayExtensions
    {
        public static int CountOccurrences<T>(this T[] arr, T item)
        {
            return arr.Count(x => Equals(x, item));
        }

        public static T[] GetUniqueElements<T>(this T[] arr)
        {
            if (arr == null)
                throw new ArgumentNullException(nameof(arr));

            return arr.Distinct().ToArray();
        }
    }

    public static class StringExtensions
    {
        public static int CountOccurrences(this string str, char c)
        {
            int counter = 0;
            foreach (char chr in str)
            {
                if (chr == c)
                    counter++;
            }
            return counter;
        }

        public static string ReverseString(this string str)
        {
            char[] charArray = str.ToCharArray();
            Array.Reverse(charArray);
            return new string(charArray);
        }
    }
}
```

Результат виконання програми:

```
Do you want to use the default string? (yes/no)
yes
Using default string: 'Hello world'
Reverse the string

Reverse of inputString: dlrow olleH
Enter the character you want to count:
o
Count of 'o' in inputString: 2
Do you want to use the default integer array? (yes/no)
yes
Using default integer array.
Array: [1, 2, 3, 4, 5, 9, 8, 7, 2]
Enter the element you want to count:
2
Count of '2' in intArray: 2
Do you want to use the default double array? (yes/no)
no
Enter elements for the double array (comma-separated):
1,2,3,4,5,1,2,3,3,3,4,5,6,7,8
Array: [1, 2, 3, 4, 5, 1, 2, 3, 3, 3, 4, 5, 6, 7, 8]
Unique elements in doubleArray:
1 2 3 4 5 6 7 8
Do you want to use the default string array? (yes/no)
yes
Using default string array.
Array: [Dog, Zebra, Boar, Bacteria, Stas, Boar, Banana, Stas]
Unique elements in stringArray:
Dog Zebra Boar Bacteria Stas Banana
```

Завдання 2:

Лістинг програми:

(ExtendedDictionary)

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace ClassLibrary1
{
    public class ExtendedDictionary<Key, Value1, Value2>
    {
        private readonly List<ExtendedDictionaryElement<Key, Value1, Value2>> dict = new
List<ExtendedDictionaryElement<Key, Value1, Value2>>();

        public void AddOrUpdate(Key key, Value1 val1, Value2 val2)
        {
            var existingElement = dict.FirstOrDefault(x => Equals(x.Key, key));
            if (existingElement != null)
            {
                existingElement.Value1 = val1;
                existingElement.Value2 = val2;
            }
            else
            {
                var newElement = new ExtendedDictionaryElement<Key, Value1, Value2>
                {
                    Key = key,
                    Value1 = val1,
                    Value2 = val2
                };
                dict.Add(newElement);
```

```
            }
        }

        public void Print()
        {
            foreach (var item in dict)
            {
                Console.WriteLine($"{item.Key} - {item.Value1} - {item.Value2}");
            }
        }

        public void Remove(Key key)
        {
            var itemToRemove = dict.FindIndex(x => Equals(x.Key, key));
            if (itemToRemove != -1)
            {
                dict.RemoveAt(itemToRemove);
            }
        }

        public bool ContainsKey(Key key)
        {
            return dict.Exists(x => Equals(x.Key, key));
        }

        public bool ContainsValues(Value1 val1, Value2 val2)
        {
            return dict.Exists(x => Equals(x.Value1, val1) && Equals(x.Value2, val2));
        }

        public int Count()
        {
            return dict.Count;
        }

        public ExtendedDictionaryElement<Key, Value1, Value2> this[Key key]
        {
            get
            {
                return dict.Find(x => Equals(x.Key, key));
            }
        }

        public IEnumerator<ExtendedDictionaryElement<Key, Value1, Value2>> GetEnumerator()
        {
            return dict.GetEnumerator();
        }

        public bool ExistKey(Key key)
        {
            return ContainsKey(key);
        }

        public bool ExistValues(Value1 val1, Value2 val2)
        {
            return ContainsValues(val1, val2);
        }

    }
    public class ExtendedDictionaryElement<KeyElement, Value1Element, Value2Element>
    {
        public KeyElement Key { get; set; }
        public Value1Element Value1 { get; set; }
        public Value2Element Value2 { get; set; }
    }
```

```
}
```
(Program.cs)
```
using System;
using System.Linq;
using System.Collections.Generic;

namespace Task1
{
    class Program
    {
        static void Main(string[] args)
        {
            // String section
            Console.WriteLine("Do you want to use the default string? (yes/no)");
            string response = Console.ReadLine() ?? "".ToLower();
            string inputString;
            if (response == "yes")
            {
                inputString = "Hello world";
                Console.WriteLine("Using default string: 'Hello world'");
            }
            else if (response == "no")
            {
                Console.WriteLine("Enter a string:");
                inputString = Console.ReadLine() ?? "";
            }
            else
            {
                Console.WriteLine("Invalid input. Using default string: 'Hello world'");
                inputString = "Hello world";
            }

            Console.WriteLine("Reverse the string");
            response = Console.ReadLine() ?? "".ToLower();
            {
                Console.WriteLine($"Reverse of inputString:
{inputString.ReverseString()}");
            }

            Console.WriteLine("Enter the character you want to count:");
            char charToCount = char.Parse(Console.ReadLine() ?? "");
            Console.WriteLine($"Count of '{charToCount}' in inputString:
{inputString.CountOccurrences(charToCount)}");

            // Integer array section
            Console.WriteLine("Do you want to use the default integer array? (yes/no)");
            response = Console.ReadLine() ?? "".ToLower();
            int[] intArray;
            if (response == "yes")
            {
                intArray = new int[] { 1, 2, 3, 4, 5, 9, 8, 7, 2 };
                Console.WriteLine("Using default integer array.");
            }
            else if (response == "no")
            {
                Console.WriteLine("Enter elements for the integer array (comma-
separated):");
                string input = Console.ReadLine() ?? "";
                intArray = input.Split(',').Select(int.Parse).ToArray();
            }
            else
            {
                Console.WriteLine("Invalid input. Using default integer array.");
```

```
            intArray = new int[] { 1, 2, 3, 4, 5, 9, 8, 7, 2 };
        }

        Console.WriteLine($"Array: [{string.Join(", ", intArray)}]");
        Console.WriteLine("Enter the element you want to count:");
        int elementToCount = int.Parse(Console.ReadLine() ?? "");
        Console.WriteLine($"Count of '{elementToCount}' in intArray:
{intArray.CountOccurrences(elementToCount)}");

        // Double array section
        Console.WriteLine("Do you want to use the default double array? (yes/no)");
        response = Console.ReadLine() ?? "".ToLower();
        double[] doubleArray;
        if (response == "yes")
        {
            doubleArray = new double[] { 1, 2, 3, 4, 5, 9, 8, 7, 2 };
            Console.WriteLine("Using default double array.");
        }
        else if (response == "no")
        {
            Console.WriteLine("Enter elements for the double array (comma-
separated):");
            string input = Console.ReadLine() ?? "";
            doubleArray = input.Split(',').Select(double.Parse).ToArray();
        }
        else
        {
            Console.WriteLine("Invalid input. Using default double array.");
            doubleArray = new double[] { 1, 2, 3, 4, 5, 9, 8, 7, 2 };
        }

        Console.WriteLine($"Array: [{string.Join(", ", doubleArray)}]");
        Console.WriteLine("Unique elements in doubleArray:");
        double[] uniqueDoubles = doubleArray.GetUniqueElements();
        foreach (double uniqueDouble in uniqueDoubles)
        {
            Console.Write($"{uniqueDouble} ");
        }
        Console.WriteLine();

        // String array section
        Console.WriteLine("Do you want to use the default string array? (yes/no)");
        response = Console.ReadLine() ?? "".ToLower();
        string[] stringArray;
        if (response == "yes")
        {
            stringArray = new string[] { "Dog", "Zebra","Boar", "Bacteria", "Stas",
"Boar", "Banana", "Stas" };
            Console.WriteLine("Using default string array.");
        }
        else if (response == "no")
        {
            Console.WriteLine("Enter elements for the string array (comma-
separated):");
            string input = Console.ReadLine() ?? "";
            stringArray = input.Split(',');
        }
        else
        {
            Console.WriteLine("Invalid input. Using default string array.");
            stringArray = new string[] { "Dog", "Zebra", "Boar", "Bacteria", "Stas",
"Boar", "Banana", "Stas" };
        }

        Console.WriteLine($"Array: [{string.Join(", ", stringArray)}]");
```

```
            Console.WriteLine("Unique elements in stringArray:");
            string[] uniqueStrings = stringArray.GetUniqueElements();
            foreach (string uniqueString in uniqueStrings)
            {
                Console.Write($"{uniqueString} ");
            }
            Console.WriteLine();
        }
    }

    public static class ArrayExtensions
    {
        public static int CountOccurrences<T>(this T[] arr, T item)
        {
            return arr.Count(x => Equals(x, item));
        }

        public static T[] GetUniqueElements<T>(this T[] arr)
        {
            if (arr == null)
                throw new ArgumentNullException(nameof(arr));

            return arr.Distinct().ToArray();
        }
    }

    public static class StringExtensions
    {
        public static int CountOccurrences(this string str, char c)
        {
            int counter = 0;
            foreach (char chr in str)
            {
                if (chr == c)
                    counter++;
            }
            return counter;
        }

        public static string ReverseString(this string str)
        {
            char[] charArray = str.ToCharArray();
            Array.Reverse(charArray);
            return new string(charArray);
        }
    }
}
```

Результат виконання програми:

```
Hello World!
Do you want to enter custom data for dict? (Yes/No)
no
Printing dictionary:
1 - Sophia - Miller
2 - Alice - Jones
3 - Michael - Garcia
4 - Emma - Martinez
5 - Ivan - Smith

Enter the key to remove from dict:
2
Key 2 removed from dict.

Checking if key exists in dict:
Enter key to check: 4
Key 4 exists in dict: True

Dictionary count of elements in dict: 4

Checking if values 'Michael' and 'Garcia' exist in dict:
Values 'Michael' and 'Garcia' exist in dict: True

Enter the key you want to check in dict:
1
Value of dict[1] = Sophia ~ Miller

Iterating through dict:
Key: 1 ~/~ Sophia ~/~ Miller
Key: 3 ~/~ Michael ~/~ Garcia
Key: 4 ~/~ Emma ~/~ Martinez
Key: 5 ~/~ Ivan ~/~ Smith
```

```
Do you want to enter custom data for dict2? (Yes/No)
yes
Enter the number of entries for dict2:
4
Entry 1 for dict2:
Enter key: 1
Enter name: Alice
Enter value: 2
Entry 2 for dict2:
Enter key: 2
Enter name: Oleg
Enter value: 3
Entry 3 for dict2:
Enter key: 3
Enter name: Klenter
Enter value: 4
Entry 4 for dict2:
Enter key: 4
Enter name: Stas
Enter value: 5
Printing dictionary2:
1 - Alice - 2
2 - Oleg - 3
3 - Klenter - 4
4 - Stas - 5

Enter the key to remove from dict2:
4
Key 4 removed from dict2.

Checking if key exists in dict2:
Enter key to check: 2
Key 2 exists in dict2: True

Dictionary2 count of elements in dict2: 3


Enter name to check: Oleg
Enter value to check: 3
Values 'Oleg' and '3' exist in dict2: True

Enter the key you want to check in dict2:
1
Value of dict2[1] = Alice ~ 2

Iterating through dict2:
Key: 1 ~/~ Alice ~/~ 2
Key: 2 ~/~ Oleg ~/~ 3
Key: 3 ~/~ Klenter ~/~ 4
```

Посилання на гіт: https://github.com/Swargon/DotNetLab2