**ЛАБОРАТОРНА РОБОТА № 2**

**Використання методів розширень та узагальнень у**

**C#.**

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

**Хід роботи:**

**Завдання 2**

Program.cs

using ConsoleApp.classes;

using System.ComponentModel;

using System.Text;

Console.OutputEncoding = Encoding.UTF8;

// Розширення string

string s = "subscribe";

Console.WriteLine("Початковий рядок: " + s);

Console.WriteLine("Інвертований рядок: " + s.MyReverse());

Console.WriteLine("Кількість символів s: " + s.MyCount('s'));

Console.WriteLine();

// Розширення узагальненого List

List<short> list = new List<short>() { 2, 4, 6, 2, 2, 4, 7 };

Console.Write("Початковий масив: ");

foreach (short item in list)

Console.Write(item + " ");

Console.WriteLine();

Console.WriteLine("Кількість елементів 2: " + list.MyCount((short)2));

Console.Write("Масив без повторень: ");

foreach (short item in list.MyUnic())

Console.Write(item + " ");

Console.WriteLine();

Console.WriteLine();

// Узагальнений "Розширений словник"

ExtendedDictionary<string, int, char> extendedDictionary = new ExtendedDictionary<string, int, char>();

extendedDictionary.Add("Стандартизація та документування програмних систем", 82, 'B');

Console.WriteLine("Додавання предмету: Стандартизація та документування програмних систем - 82 - В");

extendedDictionary.Add("Інформаційна безпека та захист ПЗ", 60, 'E');

Console.WriteLine("Додавання предмету: Інформаційна безпека та захист ПЗ - 60 - E");

extendedDictionary.Add("Математичні методи дослідження операцій", 67, 'D');

Console.WriteLine("Додавання предмету: Математичні методи дослідження операцій - 67 - D");

if (extendedDictionary.ContainsKey("Інформаційна безпека та захист ПЗ"))

Console.WriteLine("Предмет \"Інформаційна безпека та захист ПЗ\" знайдено");

else

Console.WriteLine("Предмет \"Інформаційна безпека та захист ПЗ\" не знайдено");

extendedDictionary.Remove("Інформаційна безпека та захист ПЗ");

Console.WriteLine("Видалення предмету: Інформаційна безпека та захист ПЗ");

if (extendedDictionary.ContainsKey("Інформаційна безпека та захист ПЗ"))

Console.WriteLine("Предмет \"Інформаційна безпека та захист ПЗ\" знайдено");

else

Console.WriteLine("Предмет \"Інформаційна безпека та захист ПЗ\" не знайдено");

if (extendedDictionary.ContainsValues(84, 'B'))

Console.WriteLine("Предмет з оцінкою 84,В знайдено");

else

Console.WriteLine("Предмет з оцінкою 84,В не знайдено");

extendedDictionary["Стандартизація та документування програмних систем"] = (84, 'B');

Console.WriteLine("Виставлення нової оцінки: Стандартизація та документування програмних систем - 84 В");

if (extendedDictionary.ContainsValues(84, 'B'))

Console.WriteLine("Предмет з оцінкою 84,В знайдено");

else

Console.WriteLine("Предмет з оцінкою 84,В не знайдено");

Console.WriteLine("Загальна кількість предметів: " + extendedDictionary.Count());

Console.WriteLine();

StringExtensions.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp.classes

{

static class StringExtensions

{

public static string MyReverse(this string s)

{

string result = "";

foreach (char c in s)

result = c + result;

return result;

}

public static int MyCount(this string s, char C)

{

int result = 0;

foreach (char c in s)

if (c == C)

result++;

return result;

}

}

}

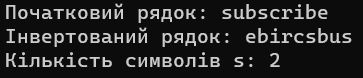


Рис.1. Результат виконання

**Завдання 3**

ListExtensions.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp.classes

{

public static class ListExtensions

{

public static int MyCount<T>(this List<T> list, T t) where T : IComparable<T>

{

int result = 0;

foreach (T item in list)

if (item.Equals(t))

result++;

return result;

}

public static List<T> MyUnic<T>(this List<T> list) where T : IComparable<T>

{

List<T> result = new List<T>();

Dictionary<T, bool> dic = new Dictionary<T, bool>();

foreach (T item in list)

{

dic[item] = true;

if (dic.Count > result.Count)

result.Add(item);

}

return result;

}

}

}

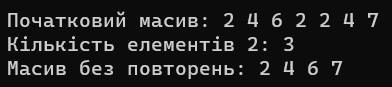


Рис.2. Результат виконання

ExtendedDictionary.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp.classes

{

public class ExtendedDictionary<T, U, V>

{

private Dictionary<T, U> dicU { get; set; }

private Dictionary<T, V> dicV { get; set; }

public ExtendedDictionary()

{

dicU = new Dictionary<T, U>();

dicV = new Dictionary<T, V>();

}

public void Add (T key, U value1, V value2)

{

dicU[key] = value1;

dicV[key] = value2;

}

public void Remove (T key)

{

dicU.Remove(key);

dicV.Remove(key);

}

public bool ContainsKey(T key)

{

return dicU.ContainsKey(key);

}

public bool ContainsValues (U value1, V value2)

{

List<T> keys = new List<T>();

foreach (var (key, item) in dicU)

if (item.Equals(value1))

keys.Add(key);

foreach (var (key, item) in dicV)

if (item.Equals(value2))

foreach (T k in keys)

if (k.Equals(key))

return true;

return false;

}

public (U, V) this[T ind]

{

get => (dicU[ind], dicV[ind]);

set => Add(ind, value.Item1, value.Item2);

}

public int Count()

{

return dicU.Count;

}

}

}

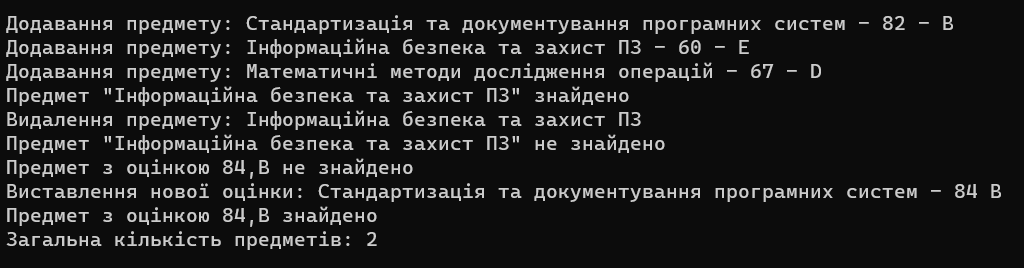


Рис.3. Результат виконання

**Завдання 4**

Репозиторій: <https://github.com/Kochubei-Kostiantyn/-DotNetLab2>

***Висновки:*** в ході виконання лабораторної роботи ми навчиkися використовувати методи розширення та узагальнення у мові програмування C#.