Міністерство освіти і науки України

Державний університет „Житомирська політехніка”

Кафедра ФІКТ

Група: ВТ-21-1

Технології розробки додатків .NET Core

Лабораторна робота №2

«Використання методів розширень та узагальнень у C#.»

Виконав: Дзюбчук Д.Р.

Прийняв: Чижмотря О. В.

					ІРТР.420001.121-ЗЛ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Дзюбчук Д.Р.			Звіт з лабораторної роботи	Літ.	Арк. 1 / Аркушів 10
Перевір.		Чижмотря О.В					
Керівник						ФІКТ, гр. ВТ-21-1[1]	
Н. контр.							
Затверд.							

Мета: навчитися використовувати оголошувати та використовувати делегати та події у мові програмування C#.

Виконання роботи:

Завдання 1: Реалізувати методи розширення:

Для класу String:

- інвертування рядка;

- підрахунок кількості входжень заданого у параметрі символа у рядок.

Лістинг програми:

```
public static class Extension_String
{
    public static int Count_Symbols(this string str, char c)
    {
        int counter = 0;
        for (int i = 0; i < str.Length; i++)
        {
            if (str[i] == c) counter++;
        }
        return counter;
    }
    public static string Reverse(this string str)
    {
        StringBuilder s = new StringBuilder();
        for (int i = str.Length - 1; i >= 0; i--)
        {
            s.Append(str[i]);
        }
        return s.ToString();
    }

}
```

Для одновимірних масивів:

- метод, що визначає скільки разів зустрічається задане значення у масиві (метод має працювати для одновимірних масивів усіх типів, для реалізації даного методу розширення використайте узагальнення та їх обмеження за допомогою “where”);

- метод, що повертає новий масив такого ж типу і формує його з унікальних елементів (видаляє повтори);

Лістинг програми:

```
public class RequireClass<TypeArr> where TypeArr : class { }
```

```
        public static int Count<TypeArr>(this TypeArr[] arr, TypeArr elem) where TypeArr
:
             struct
        {
            int count = 0;
            for (int i = 0; i < arr.Length; i++)
            {
                if (object.Equals(arr[i], elem))
                {
                    count++;
                }
            }
            return count;
        }
        public static int Count<TypeArr>(this TypeArr[] arr, TypeArr elem,
RequireClass<TypeArr> ignore = null) where TypeArr :
             class
        {
            int count = 0;
            for (int i = 0; i < arr.Length; i++)
            {
                if (object.Equals(arr[i], elem))
                {
                    count++;
                }
            }
            return count;
        }
        public static TypeArr[] Special<TypeArr>(this TypeArr[] arr) where TypeArr :
            struct
        {
            List<TypeArr> list = new List<TypeArr>();
            for (int i = 0; i < arr.Length; i++)
            {
                if (list.IndexOf(arr[i]) == -1)
                {
                    list.Add(arr[i]);
                }
            }
            TypeArr[] ArrUniq = new TypeArr[list.Count];
            for (int i = 0; i < ArrUniq.Length; i++)
            {
                ArrUniq[i] = list[i];
            }
            return ArrUniq;
        }
        public static TypeArr[] Special<TypeArr>(this TypeArr[] arr,
RequireClass<TypeArr> ignore = null) where TypeArr :
            class
        {
            List<TypeArr> list = new List<TypeArr>();
            for (int i = 0; i < arr.Length; i++)
            {
                if (list.IndexOf(arr[i]) == -1)
                {
                    list.Add(arr[i]);
                }
            }
            TypeArr[] ArrUniq = new TypeArr[list.Count];
            for (int i = 0; i < ArrUniq.Length; i++)
            {
                ArrUniq[i] = list[i];
            }
            return ArrUniq;
```

```
        }
```

Написати код для демонстрації роботи реалізованих методів розширення.

Лістинг програми:

```
using System.Text;

namespace Console_Sorted_Array
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.OutputEncoding = UTF8Encoding.UTF8;
            string str = "Без діла слабіє сила";
            Console.WriteLine($"Текст: {str}");
            Console.WriteLine($"Кількість літер 'c' у слові: {str.Count_Symbols('c')}");
            Console.WriteLine($"Текст задом наперед: {str.Reverse()}");
            // Опрацювання рядка

            int[] numbers = new int[] { 2, 0, 0, 4, 1, 2, 0, 4 };
            Console.WriteLine("\nСписок чисел: [{0}]", string.Join(", ", numbers));
            int sub_number = 0;
            Console.WriteLine($"Кількість {sub_number} в
масиві:{numbers.Count<int>(sub_number)}");
            // Опрацювання масиву чисел

            string[] words = new string[] { "Zhytomyr", "Socks", "Ice-cream", "Victory"
};
            Console.WriteLine("\nСписок слів: [{0}]", string.Join(", ", words));
            string sub_word = "Zhytomyr";
            Console.WriteLine($"Кількість слів
{sub_word}:{words.Count<string>(sub_word)}");
            // Опрацювання масиву строк

            char[] symbols = new char[] { 'l', 'o', 'r', 'e', 'a', 'r', 't', 'x', 'd' };
            Console.WriteLine("\nСписок символів: [{0}]", string.Join(", ", symbols));
            char sub_symbol = 'r';
            Console.WriteLine($"Кількість символів
{sub_symbol}:{symbols.Count<char>(sub_symbol)}");
            // Опрацювання масиву символів

            double[] array_not_special = new double[] { 5, 9, 4, 1, 8, 5, 2, 4 };
            Console.WriteLine("\nСписок чисел: [{0}]", string.Join(", ",
array_not_special));
            Console.Write($"Перетворюємо масив на унікальний без повторень : ");
            double[] arr_special = array_not_special.Special<double>();
            foreach (double i in arr_special)
            {
                Console.Write($" {i} ");
            }
            // Перетворення на унікальний масив без повторень
        }
    }
}
```

Перевірка:

```
Текст: Без діла слабіє сила
Кількість літер 'с' у слові: 2
Текст задом наперед: алис єібалс алід зеБ

Список чисел: [2, 0, 0, 4, 1, 2, 0, 4]
Кількість 0 в масиві:3

Список слів: [Zhytomyr, Socks, Ice-cream, Victory]
Кількість слів Zhytomyr:1

Список символів: [l, o, r, e, a, r, t, x, d]
Кількість символів r:2

Список чисел: [5, 9, 4, 1, 8, 5, 2, 4]
Перетворюємо масив на унікальний без повторень :  5  9  4  1  8  2
```

Завдання 2 : Реалізувати узагальнені класи для:

- Реалізувати узагальнений клас для зберігання “розширеного словника” (для ключа передбачається два значення).

ExtendedDictionary<T, U, V>, де T - тип даних ключа, U - тип даних першого значення, V - тип даних другого значення. Передбачити операції:

- додавання елемента у словник;
- видалення елемента з словника за заданим ключем;
- перевірка наявності елемента із заданим ключем;
- перевірка наявності елемента із заданим значенням (значення1 та значення2);
- повернення елемента за заданим ключем (реалізувати операцію індексування);
- властивість, що повертає кількість елементів;

Представлення елемента словника реалізувати у вигляді окремого класу ExtendedDictionaryElement<T, U, V>, передбачивши властивості для доступу до ключа, першого та другого значення.

Словник повинен мати можливість використання у циклах foreach:

foreach(var elem in array) { … }

Лістинг програми:

```
public class ExtendedDictionaryElement<Key, Value1, Value2>
        {
            public Key key { set; get; }
            public Value1 value1 { set; get; }
            public Value2 value2 { set; get; }
        }
        public class ExtendedDictionary<Key, Value1, Value2>
        {

            protected List<ExtendedDictionaryElement<Key, Value1, Value2>> dict = new
List<ExtendedDictionaryElement<Key, Value1, Value2>>();
            public void Add(Key key, Value1 val1, Value2 val2)
            {
                foreach (ExtendedDictionaryElement<Key, Value1, Value2> i in dict)
                {
                    if (object.Equals(i.key, key))
                    {
                        Console.WriteLine("Ключ існує");
                        return;
                    }
                }
                ExtendedDictionaryElement<Key, Value1, Value2> line = new
ExtendedDictionaryElement<Key, Value1, Value2>();
                line.key = key;
                line.value1 = val1;
                line.value2 = val2;
                dict.Add(line);
            }

            public void Print()
            {
                foreach (ExtendedDictionaryElement<Key, Value1, Value2> i in dict)
                {
                    Console.WriteLine($"{i.key}-{i.value1}-{i.value2}");
                }
            }

            public void Remove(Key elem)
            {
                for (int i = 0; i < dict.Count; i++)
                {
                    if (object.Equals(dict[i].key, elem)) { dict.RemoveAt(i); return; }
                }
            }
            public void ExistKey(Key elem)
            {
                int iss = 0; int ind = 0;
                for (int i = 0; i < dict.Count; i++)
                {
                    if (object.Equals(dict[i].key, elem))
                    {
                        iss++;
                        ind = i;
                    }
                }
                if (iss == 0) Console.WriteLine("Немає такого ключа ;(");
                else Console.WriteLine($"Ключ знайдено:{ind}");
            }
            public int Count() { return dict.Count; }


            public void ExistValues(Value1 val1, Value2 val2)
            {
                bool ter = false;
```

```
                for (int i = 0; i < dict.Count; i++)
                {
                    if ((object.Equals(dict[i].value1, val1)) &&
(object.Equals(dict[i].value2, val2)))
                    {
                        ter = true;
                    }
                }
                if (ter)
                {
                    Console.WriteLine($"Елемент із значеннями {val1} і {val2} існує");
                }
                else Console.WriteLine($"Елемент із значеннями {val1} і {val2} не
існує");
            }

            private int ind = 0;
            public ExtendedDictionaryElement<Key, Value1, Value2> this[Key key]
            {
                get
                {
                    for (int i = 0; i < dict.Count; i++)
                    {
                        if (object.Equals(dict[i].key, key))
                        {
                            ind = i;
                        }
                    }
                    return dict[ind];
                }
            }

            public IEnumerator<ExtendedDictionaryElement<Key, Value1, Value2>>
GetEnumerator()
            {
                for (int i = 0; i < dict.Count; i++)
                {
                    yield return dict[i];
                }
            }
              }
```

- Написати код для демонстрації роботи з реалізованими узагальненими класами.

Лістинг програми:

```
        static void Main(string[] args)
        {
            Console.OutputEncoding = UTF8Encoding.UTF8;
            ExtendedDictionary<int, string, string> dict = new ExtendedDictionary<int,
string, string>();
            // Створення списку

            int count = 1;
            int choice;
            do
            {
                Console.WriteLine("\nВведіть перше значення елементу :");
                string var1 = Console.ReadLine();
                Console.WriteLine("Введіть друге значення елементу :");
                string var2 = Console.ReadLine();
                dict.Add(count, var1, var2);
```

```
                // Додавання елементів у список

                count++;

                Console.WriteLine("Якщо бажаєте зупинитись, введіть 0");
                while(!Int32.TryParse(Console.ReadLine(),out choice))
                {
                    Console.WriteLine("Невірно введено значення! Спробуйте ще раз!");
                }

            } while (choice != 0);
            // Надавання елементів для опрацювання

            Console.WriteLine("\n\nЯкий елемент списку ви бажаєте знайти?");
            Console.WriteLine("Введіть перше значення елементу :");
            string fnd_var1 = Console.ReadLine();
            Console.WriteLine("Введіть друге значення елементу :");
            string fnd_var2 = Console.ReadLine();

            PrintTable(dict, fnd_var1,fnd_var2);
        }

        public static void PrintTable(ExtendedDictionary<int, string, string> dict,string
find_var1,string find_var2)
        {
            Console.WriteLine("\n\n\nЕлементи списку:");
            dict.Print();
            Console.WriteLine("-------------------------");
            dict.Remove(1);
            // Видаляємо елемент списку

            Console.WriteLine("Список елементів після видалення першого по індексу
об'єкта:");
            dict.Print();
            Console.WriteLine("-------------------------");
            dict.ExistKey(2);
            dict.ExistKey(4);
            // Перевірки на існування ключа 1 та 4 відповідно

            Console.WriteLine("-------------------------");
            Console.WriteLine($"Кількість елементів:{dict.Count()}");
            // Кількість елеметів списку

            Console.WriteLine("-------------------------");
            dict.ExistValues(find_var1, find_var2);
            // Пошук ключа зі значеннями

            Console.WriteLine("-------------------------");
            Console.WriteLine($"dict[2] ={dict[2].value1}-{dict[2].value2}");
            // Робота з елементом списку по індексу

            Console.WriteLine("-------------------------");
            foreach (var i in dict)
            {
                Console.WriteLine($"Key:{i.key}--{i.value1}--{i.value2}");
            }
            // Проходження по списку циклом foreach
              }
```

Перевірка:

```
Введіть перше значення елементу :
Ulrik
Введіть друге значення елементу :
Hoderd
Якщо бажаєте зупинитись, введіть 0
1

Введіть перше значення елементу :
Koorben
Введіть друге значення елементу :
Nulian
Якщо бажаєте зупинитись, введіть 0
1

Введіть перше значення елементу :
Tuttor
Введіть друге значення елементу :
Fenhrir
Якщо бажаєте зупинитись, введіть 0
0


Який елемент списку ви бажаєте знайти?
Введіть перше значення елементу :
Koorben
Введіть друге значення елементу :
Nulian
```

```
Елементи списку:
1-Ulrik-Hoderd
2-Koorben-Nulian
3-Tuttor-Fenhrir
-------------------------
Список елементів після видалення першого по індексу об'єкта:
2-Koorben-Nulian
3-Tuttor-Fenhrir
-------------------------
Ключ знайдено:0
Немає такого ключа ;(
-------------------------
Кількість елементів:2
-------------------------
Елемент із значеннями Koorben і Nulian існує
-------------------------
dict[2] =Koorben-Nulian
-------------------------
Key:2--Koorben--Nulian
Key:3--Tuttor--Fenhrir
```

Висновок: В ході виконання лабораторної роботи було ознайомлено з середовищем Visual Studio на мові C#. Роботу виконано в повному обсязі.