

КПІ ім. Ігоря Сікорського
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт до лабораторної роботи з курсу
“Основи програмування. Частина 2. Методології програмування”

Прийняв
Куценко М.О..
“30” травня 2025 р.

Виконав
Студент групи ІП-43
Пакалюк В.І.

Київ 2025

Лабораторна робота №7

Тема: Побудова та використання структур даних

Варіант: 5

Завдання:

Написати консольний застосунок для проведення з односпрямованим списком елементів типу “double”, нові елементи якого включаються до початку списку, наступних операцій:

1. Пошук першого елемента більшого за середнє значення.
2. Пошук суми елементів, значення яких більше за задане користувачем.
3. Отримання нового списку зі значень елементів, менших за середнє значення
4. Видалення елементів, що розташовані на парних позиціях (нумерація починається з голови списку).

Посилання на репозиторій

https://github.com/Muh0m0r/OOP_Laba7

Текст програми

```
Program.cs
using OOP_Laba7;
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;

class Program
{
    static void Main()
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;
        LinkedList<double> list = null;
        ListOperations operations = new ListOperations();

        while (true)
        {
            Console.WriteLine("\nМеню:");
            Console.WriteLine("1. Створити новий список");
            Console.WriteLine("2. Вийти");
            Console.Write("Ваш вибір: ");

            string choice = Console.ReadLine();

            switch (choice)
            {
                case "1":
                    list = operations.CreateList();
                    PrintList(list);
                    ListInteractionMenu(ref list, operations);
                    break;
                case "2":
```

```

        return;
    default:
        Console.WriteLine("Некоректний вибір. Спробуйте ще раз.");
        break;
    }
}
}

static void ListInteractionMenu(ref LinkedList<double> list, ListOperations
operations)
{
    while (true)
    {
        Console.WriteLine("\nМеню роботи зі списком:");
        Console.WriteLine("1. Знайти перший елемент > середнього");
        Console.WriteLine("2. Знайти суму елементів > заданого");
        Console.WriteLine("3. Отримати новий список зі значень < середнього");
        Console.WriteLine("4. Видалити елементи на парних позиціях");
        Console.WriteLine("5. Отримати елемент за індексом");
        Console.WriteLine("6. Видалити елемент за індексом");
        Console.WriteLine("7. Видалити список та повернутись назад");
        Console.Write("Ваш вибір: ");

        string choice = Console.ReadLine();

        switch (choice)
        {
            case "1":
                try
                {
                    double result = operations.GetFirstOverAverageValue(list);
                    Console.WriteLine($"Перший елемент більший за середнє значення =
{result}");
                }
                catch (Exception ex)
                {
                    Console.WriteLine(ex.Message);
                }
                break;

            case "2":
                double threshold;
                while (true)
                {
                    Console.Write("Введіть значення порогу: ");
                    string input = Console.ReadLine();
                    if (double.TryParse(input, NumberStyles.Any,
CultureInfo.InvariantCulture, out threshold))
                        break;
                    Console.WriteLine("Некоректне значення. Спробуйте ще раз.");
                }
            }
        }
    }
}

```

```

    }
    try
    {
        double sum = operations.GetSumOfElementsBiggerThan(list, threshold);
        Console.WriteLine($"Сума елементів більших за {threshold} =
{sum}");
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    break;

case "3":
    try
    {
        list = operations.CreateLinkedListUnderAverage(list);
        Console.WriteLine("Оновлений список (елементи менші за середнє
значення оригіналу):");
        PrintList(list);
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    break;

case "4":
    try
    {
        list = operations.DeleteEvenNodes(list);
        Console.WriteLine("Список після видалення елементів на парних
позиціях:");
        PrintList(list);
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    break;

case "5":
    try
    {
        int idx;
        while (true)
        {
            Console.Write("Введіть індекс елемента: ");
            string input = Console.ReadLine();
            if (int.TryParse(input, out idx) && idx >= 0)

```

```

        break;
        Console.WriteLine("Некоректне значення. Спробуйте ще раз.");
    }
    double element = operations.GetNodeValueByIndex(list, idx);
    Console.WriteLine($"Елемент за індексом {idx}: {element}");
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
break;

case "6":
    try
    {
        int idxToDelete;
        while (true)
        {
            Console.Write("Введіть індекс елемента для видалення: ");
            string input = Console.ReadLine();
            if (int.TryParse(input, out idxToDelete) && idxToDelete >= 0)
                break;
            Console.WriteLine("Некоректне значення. Спробуйте ще раз.");
        }
        list = operations.DeleteNodeByIndex(list, idxToDelete);
        Console.WriteLine($"Елемент за індексом {idxToDelete} видалено.");
        PrintList(list);
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    break;

case "7":
    list = null;
    return;

default:
    Console.WriteLine("Некоректний вибір. Спробуйте ще раз.");
    break;
}
}
}

static void PrintList(LinkedList<double> list)
{
    if (list == null || list.Count == 0)
    {
        Console.WriteLine("Список порожній.");
    }
}

```

```

        return;
    }

    Console.Write("Список: [");
    Console.Write(string.Join("; ", list.Select(x => x.ToString("F2",
CultureInfo.InvariantCulture))));
    Console.WriteLine("]");
}
}

```

ListOperations.cs

```

using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OOP_Laba7
{
    internal class ListOperations
    {
        public double GetFirstOverAverageValue(LinkedList<double> list)
        {
            if (list == null || list.Count == 0)
                throw new InvalidOperationException("Список порожній.");

            double average = list.Average();

            foreach (double currentNodeValue in list)
            {
                if (currentNodeValue > average)
                {
                    return currentNodeValue;
                }
            }
            throw new InvalidOperationException("У списку немає елемента, більшого за середнє значення.");
        }

        public double GetSumOfElementsBiggerThan(LinkedList<double> list, double sampleElement)
        {
            if (list == null || list.Count == 0)
                throw new InvalidOperationException("Список порожній.");

            double sum = 0;

```

```

        foreach (double currentNodeValue in list)
        {
            if (currentNodeValue > sampleElement)
            {
                sum += currentNodeValue;
            }
        }

        return sum;
    }

    public LinkedList<double> CreateLinkedListUnderAverage(LinkedList<double>
originalList)
    {
        if (originalList == null || originalList.Count == 0)
            throw new InvalidOperationException("Список порожній.");

        LinkedList<double> resultList = new LinkedList<double>();
        double originalAverage = originalList.Average();

        foreach (double currentNodeValue in originalList)
        {
            if (currentNodeValue < originalAverage)
            {
                resultList.AddLast(currentNodeValue);
            }
        }

        return resultList;
    }

    public LinkedList<double> DeleteEvenNodes(LinkedList<double> originalList)
    {
        if (originalList == null || originalList.Count == 0)
            throw new InvalidOperationException("Список порожній.");

        LinkedListNode<double> current = originalList.First;
        int index = 0;

        while (current != null)
        {
            LinkedListNode<double> next = current.Next;
            if (index % 2 != 0)
            {
                originalList.Remove(current);
            }
            index++;
            current = next;
        }
    }

```

```

        return originalList;
    }

    public LinkedList<double> CreateList()
    {
        var list = new LinkedList<double>();
        Console.WriteLine("\nВведіть значення для списку (нечислові значення — завершити введення списку):");

        while (true)
        {
            Console.Write("> ");
            string input = Console.ReadLine();
            if (double.TryParse(input, NumberStyles.Any, CultureInfo.InvariantCulture, out double value))
            {
                list.AddLast(value);
            }
            else
            {
                break;
            }
        }
        return list;
    }

    public double GetNodeValueByIndex(LinkedList<double> list, int index)
    {
        if (list == null || list.Count == 0)
            throw new InvalidOperationException("Список порожній.");
        if (index < 0 || index >= list.Count)
            throw new ArgumentOutOfRangeException(nameof(index), "Індекс поза межами списку.");

        var current = list.First;
        for (int i = 0; i < index; i++)
        {
            current = current.Next;
        }
        return current.Value;
    }

    public LinkedList<double> DeleteNodeByIndex(LinkedList<double> list, int index)
    {
        if (list == null || list.Count == 0)
            throw new InvalidOperationException("Список порожній.");
        if (index < 0 || index >= list.Count)

```



```
        throw new ArgumentOutOfRangeException(nameof(index), "Індекс поза  
межами списку.");
```

```
        var current = list.First;  
        for (int i = 0; i < index; i++)  
        {  
            current = current.Next;  
        }  
        list.Remove(current);  
        return list;  
    }  
}
```

Введені та одержані результати:

```
Microsoft Visual Studio Debug Console

Меню:
1. Створити новий список
2. Вийти
Ваш вибір: 1

Введіть значення для списку (нечислове значення – завершити введення списку):
> 12
> 34
> 0.67
> -45
> -6.67
> 52
> ghj3
Список: [12.00; 34.00; 0.67; -45.00; -6.67; 52.00]

Меню роботи зі списком:
1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад
Ваш вибір: !
Некоректний вибір. Спробуйте ще раз.

Меню роботи зі списком:
1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад
Ваш вибір: 1
Перший елемент більший за середнє значення = 12

Меню роботи зі списком:
1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад
Ваш вибір: 2
Введіть значення порогу: 3
Сума елементів більших за 3 = 98

Меню роботи зі списком:
1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад
Ваш вибір: 5
Введіть індекс елемента: 4
Елемент за індексом 4: -6,67
```

Меню роботи зі списком:

1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад

Ваш вибір: 5

Введіть індекс елемента: 4

Елемент за індексом 4: -6,67

Меню роботи зі списком:

1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад

Ваш вибір: 6

Введіть індекс елемента для видалення: 7

Індекс поза межами списку. (Parameter 'index')

Меню роботи зі списком:

1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад

Ваш вибір: 6

Введіть індекс елемента для видалення: 4

Елемент за індексом 4 видалено.

Список: [12.00; 34.00; 0.67; -45.00; 52.00]

Меню роботи зі списком:

1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад

Ваш вибір: 3

Оновлений список (елементи менші за середнє значення оригіналу):

Список: [0.67; -45.00]

Меню роботи зі списком:

1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад

Ваш вибір: 4

Список після видалення елементів на парних позиціях:

Список: [0.67]

```

Меню роботи зі списком:
1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад
Ваш вибір: 4
Список після видалення елементів на парних позиціях:
Список: [0.67]

Меню роботи зі списком:
1. Знайти перший елемент > середнього
2. Знайти суму елементів > заданого
3. Отримати новий список зі значень < середнього
4. Видалити елементи на парних позиціях
5. Отримати елемент за індексом
6. Видалити елемент за індексом
7. Видалити список та повернутись назад
Ваш вибір: 7

Меню:
1. Створити новий список
2. Вийти
Ваш вибір: 2

A:\Універ\ООП\Laba7\ООП_Laba7\bin\Debug\net8.0\ООП_Lab
To automatically close the console when debugging stop
nsole when debugging stops.
Press any key to close this window . . .

```

Теоретичні розрахунки:

C6	fx = AVERAGE(A6:A11)		
	A	B	C
5			
6	12		7,833333333
7	34		
8	0,67		
9	-45		
10	-6,67		
11	52		

- 1) $12 > 7,8(3)$
- 2) $12 + 34 + 52 = 98$
- 3) $0,67 < 7,8(3)$
 $-45 < 7,8(3)$
- 4) $0,67, -45$
- 5) $12; 34; 0,67, -45; \underline{-6,67}; 52$
- 6) $12; 34; 0,67, -45; \underline{-6,67}; 52$
- 7) $12; 34; 0,67, -45; 52$

Висновки: Теоретичні розрахунки відповідають отриманим. Програма працює коректно. Програма вирішує поставлене завдання .

