

ЗВІТ
Основи програмування
Лабораторна робота 7
ПОБУДОВА ТА ВИКОРИСТАННЯ СТРУКТУР ДАНИХ

Виконав: Кісурін Кирило ІІІ-44

Завдання:

№	Тип даних елементів	Тип списку	Спосіб додавання елементу списку	Операції зі списком
11	<u>float</u>	Односпрямований	Включення після другого елементу списку	1. Знайти перший від'ємний елемент списку. 2. Знайти суму елементів більших за середнє значення. 3. Отримати новий список зі значень позитивних елементів поточного списку. 4. Видалити всі від'ємні елементи поточного списку.

Код:

SinglyLinkedListLibrary.cs

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

namespace SinglyLinkedListLibrary
{
    public class Node
    {
        public float Data;
        public Node? Next;

        public Node(float data)
        {
            Data = data;
            Next = null;
        }
    }

    public class SinglyLinkedList : IEnumerable<float>
    {
        private Node? head;

        // Додавання після другого елемента
        public void AddAfterSecond(float value)
        {
            Node newNode = new Node(value);
            if (head == null || head.Next == null)
            {
                // Якщо менше ніж 2 елементи — додаємо в кінець
                if (head == null)
                {
                    head = newNode;
                }
                else
                {
                    head.Next = newNode;
                }
            }
        }
    }
}
```

```

    }
    else
    {
        Node second = head.Next;
        newNode.Next = second.Next;
        second.Next = newNode;
    }
}

public float? FindFirstNegative()
{
    var current = head;
    while (current != null)
    {
        if (current.Data < 0)
            return current.Data;
        current = current.Next;
    }
    return null;
}

private float FindAverage()
{
    float sum = 0;
    int count = 0;
    var current = head;
    while (current != null)
    {
        sum += current.Data;
        count++;
        current = current.Next;
    }
    return count == 0 ? 0 : sum / count;
}

public float SumGreaterThanAverage()
{
    float average = FindAverage();
    float sum = 0;
    var current = head;
    while (current != null)
    {
        if (current.Data > average)
            sum += current.Data;
        current = current.Next;
    }
    return sum;
}

public SinglyLinkedList GetPositiveElements()
{
    var result = new SinglyLinkedList();
    var current = head;
    while (current != null)
    {
        if (current.Data > 0)
        {
            result.AddAfterSecond(current.Data); // або AddToEnd,
якщо хочеш в кінець
        }
        current = current.Next;
    }
    return result;
}

```

```

    public void RemoveNegatives()
    {
        while (head != null && head.Data < 0)
        {
            head = head.Next;
        }

        var current = head;
        while (current != null && current.Next != null)
        {
            if (current.Next.Data < 0)
            {
                current.Next = current.Next.Next;
            }
            else
            {
                current = current.Next;
            }
        }
    }

    public void Print()
    {
        foreach (var item in this)
        {
            Console.Write($"{item} ");
        }
        Console.WriteLine();
    }

    public IEnumerator<float> GetEnumerator()
    {
        var current = head;
        while (current != null)
        {
            yield return current.Data;
            current = current.Next;
        }
    }

    IEnumerator IEnumerable.GetEnumerator() => GetEnumerator();
}

```

Program.cs

```

using System;
using SinglyLinkedListLibrary;
class Program
{
    static void Main()
    {
        var list = new SinglyLinkedList();
        list.AddAfterSecond(2.5f);
        list.AddAfterSecond(-3.1f);
        list.AddAfterSecond(1.2f);
        list.AddAfterSecond(4.8f);
        list.AddAfterSecond(-0.5f);
    }
}

```

```
Console.WriteLine("Original list:");  
list.Print();  
  
Console.WriteLine("First negative: " + list.FindFirstNegative());  
Console.WriteLine("Sum > average: " + list.SumGreaterThanAverage());  
  
Console.WriteLine("Positive elements list:");  
var positives = list.GetPositiveElements();  
positives.Print();  
  
list.RemoveNegatives();  
Console.WriteLine("List after removing negatives:");  
list.Print();  
}  
}
```