

**ЗВІТ**  
**Основи програмування**  
**Лабораторна робота 7**  
**ПОБУДОВА ТА ВИКОРИСТАННЯ СТРУКТУР ДАНИХ**

Виконав: Чуніхін Вадим Вадимович ІІІ-44

### Завдання:

3	short	Одно спрямова ний	Включення до початку	1.Знайти перше входження елементу, кратного 3. 2.Знайти добуток елементів, значення яких менші за <u>середнє</u> значення. 3.Отримати новий список зі значень елементів кратних 3. 4.Видалити елементи більші за середнє значення.
---	-------	-------------------------	-------------------------	--

### Код:

#### SinglyLinkedList.cs

```
using System.Collections;

namespace SinglyLinkedListLibrary
{
    public class SinglyLinkedList : IEnumerable<short>
    {
        private Node? head;
        public void AddFirst(short value)
        {
            Node newNode = new Node(value);
            newNode.Next = head;
            head = newNode;
        }
        public short? FindFirstMultipleOf(int divisor)
        {
            if(head == null) throw new EmptyListException();
            var current = head;
            while (current != null)
            {
                if (current.Data % divisor == 0)
                {
                    return current.Data;
                }
                current = current.Next;
            }
            return null;
        }

        private decimal FindAverage()
    }
}
```

```

{
    short sum = 0;
    int count = 0;
    var current = head;
    while (current != null)
    {
        sum += current.Data;
        count++;
        current = current.Next;
    }
    return (decimal)sum / count;
}

public int GetMultiplyElementsBelowAverage()
{
    if(head == null) throw new EmptyListException();

    int result = 0;
    decimal average = FindAverage();
    var current = head;
    while (current != null)
    {
        if (current.Data < average)
        {
            if (result == 0)
            {
                result = current.Data;
            }
            else
            {
                result *= current.Data;
            }
        }
        current = current.Next;
    }
    return result;
}

public SinglyLinkedList GetAllMultipleOf(int divisor)
{
    if(divisor == 0) throw new DivideByZeroException();

    var list = new SinglyLinkedList();
    var current = head;
    while (current != null)
    {
        if (current.Data % divisor == 0)
        {

```

```

        list.AddFirst(current.Data);
    }
    current = current.Next;
}
return list;
}

public void RemoveGraterThanAverage()
{
    decimal average = FindAverage();
    Node current = head;
    Node prev = null;

    while (current != null)
    {
        if (current.Data > average)
        {
            if (prev == null)
            {
                head = head.Next;
                current = head;
            }
            else
            {
                prev.Next = current.Next;
                current = current.Next;
            }
        }
        else
        {
            prev = current;
            current = current.Next;
        }
    }
}

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

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

```

## Node.cs

```
namespace SinglyLinkedListLibrary
{
    public class Node
    {
        private short data;
        private Node? next;

        public short Data
        {
            get => data;
            set => data = value;
        }

        public Node Next
        {
            get => next;
            set => next = value;
        }

        public Node(short data)
        {
            this.data = data;
            next = null;
        }
    }
}
```

## EmptyListException.cs

```
namespace SinglyLinkedListLibrary;

public class EmptyListException : Exception
{
    public EmptyListException() : base("This list is empty.") { }
    public EmptyListException(string message) : base(message) { }
    public EmptyListException(string message, Exception innerException) :
base(message, innerException) { }
}
```