# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №11**

# з курсу

**«Алгоритми і структури даних»**

*Студента 1 курсу*

*групи ПП-11 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

д.е.н., к.т.н., проф. Плескач В.Л.

к.ф.-м.н., доц. Шолохов О.В.

## Київ – 202

**1.Назва роботи**

Рекурсивні алгоритми

1. **Тема роботи**

Рекурсивні алгоритми

1. **Мета роботи**

Засвоєння теоретичного матеріалу та набуття практичного досвіду

створення рекурсивних алгоритмів та написання відповідних їм програм.

1. **Умова завдання**

(4 варіант)

Дано натуральне число n. Знайти суму перших n членів ряду чисел,

заданого рекурентною формулою. Розв’язати задачу трьома способами:

1. в програмі використати рекурсивну процедуру або функцію,яка виконує

обчислення і членів ряду, і суми на рекурсивному спуску;

2. в програмі використати рекурсивну процедуру або функцію,яка виконує

обчислення і членів ряду, і суми на рекурсивному поверненні;

3. в програмі використати рекурсивну процедуру або функцію,яка виконує

обчислення членів ряду на рекурсивному спуску, а обчислення суми на

рекурсивному поверненні.

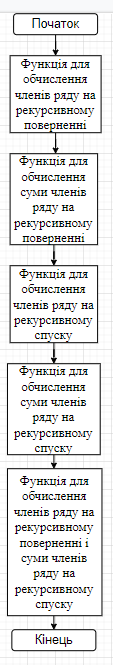
Програми повинні працювати коректно для довільного натурального n включно з n = 1



1. **Рішення**

Описуємо функцію для обчислення членів ряду на рекурсивному поверненні, описуємо функцію для обчислення суми членів ряду на рекурсивному поверненні, функцію для обчислення членів ряду на рекурсивному спуску, функцію для обчислення суми членів ряду на рекурсивному суми, функцію для обчислення членів ряду на рекурсивному спуску та обчислення суми членів ряду на рекурсивному поверненні.

Блок-схема:



Код:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Лаба11 {

class Program {

static double RecursiveAscentFunc(int N) {

if (N == 1) {

return 1;

}

double recAscVal = 2 \* (RecursiveAscentFunc(N - 1)) - Math.Cos(N);

return recAscVal;

}

static double RecursiveAscentSum(int N) {

if (N == 0) {

return 0;

}

double recAscSum = RecursiveAscentSum(N - 1) + RecursiveAscentFunc(N);

return recAscSum;

}

static void RecursiveDescentFunc(int N, ref double recDescVal) {

if (N != 0) {

RecursiveDescentFunc(N - 1, ref recDescVal);

if (N == 1) {

recDescVal = 1;

}

else

{

recDescVal = 3 \* Math.Sin(recDescVal) + Math.Log(N);

}

}

}

static void RecursiveDescentSum(int N, ref double recDescSum) {

double recDecVal = 0;

if (N != 0) {

RecursiveDescentSum(N - 1, ref recDescSum);

RecursiveDescentFunc(N, ref recDecVal);

recDescSum = recDescSum + recDecVal;

}

}

static double RecursiveAscentDescentSum(int N) {

if (N == 0) {

return 0;

}

double recDescVal = 0;

RecursiveDescentFunc(N, ref recDescVal);

double recAscSum = RecursiveAscentSum(N - 1) + recDescVal;

return recAscSum;

}

static void Main(string[] args) {

Console.WriteLine("Recursive ascention: " + RecursiveAscentFunc(5));

double drval = 0;

RecursiveDescentFunc(5, ref drval);

Console.WriteLine("Recursive descention: " + drval);

double loopsum = 0;

for (int i = 1; i <= 5; i++)

{

loopsum += RecursiveAscentFunc(i);

}

Console.WriteLine("The sum of five elements in a loop: " + loopsum);

Console.WriteLine("The sum of five elements on recursive ascent: " + RecursiveAscentSum(5));

double dessum = 0;

RecursiveDescentSum(5, ref dessum);

Console.WriteLine("The sum of five elements on recursive descent: " + dessum);

Console.WriteLine("The sum of five elements on recursive ascent-descent: " + RecursiveAscentDescentSum(5));

}

}

}

Результат роботи програми:

[screenshot]

**Контрольні питання**

1. Визначення рекурсивного об’єкту

Об’єкт, що визначається сам через себе (або ж сам себе викликає)

1. Визначення глибини та поточного рівня рекурсії

Глибина рекурсії - глибина дерева рекурсивних викликів (довжина максимального шляху з вершини до одного з базових значень функції)

1. Форма виконання рекурсивних дій на рекурсивному спуску

Дії виконуються до рекурсивного виклику

1. Форма виконання рекурсивних дій на рекурсивному поверненні

Дії виконуються після рекурсивного виклику

Форма виконання рекурсивних дій на як рекурсивному спуску, так і на рекурсивному поверненні

Дії виконуються як до, так і після рекурсивного виклику

1. **Висновки**

В результаті виконання даної лабораторної роботи я оволодів практичними навичками використання рекурсивних алгоритмів, навчився складати програми для виконання операцій з використанням даних алгоритмів. Вважаю дану лабораторну роботу виконаною в повному обсязі.