

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 8
з навчальної дисципліни
“Базові методології та технології програмування”
РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИЙ
ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ВИКОНАВ
студент академічної групи
КІ-22-2

_____ Карась І. М.

ПЕРЕВІРИВ
викладач кафедри кібербезпеки
та програмного забезпечення

_____ Олександр СОБІНОВ

Мета роботи

Полягає у набутті ґрунтовних вмінь і практичних навичок застосування теоретичних положень методології модульного програмування, реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування C++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, розроблення програмних модулів та засобів у кросплатформовому середовищі Code::Blocks (GNU GCC Compiler).

Завдання до лабораторної роботи

1. Реалізувати статичну бібліотеку модулів `libModulesПрізвище` C/C++, яка містить функцію розв'язування задачі 8.1.
2. Реалізувати програмне забезпечення розв'язування задачі 8.2 — консольний застосунок.

Варіант 8

ВАРІАНТ 8

— ЗАДАЧА 8.1 —

За значеннями x, y, z обчислюється S :

$$S = \ln|\sin z| + \frac{\frac{1}{2}x^2 - \sqrt{(y+z)^2 - x^5}}{10\pi \cdot z^4}.$$

— ЗАДАЧА 8.2 —

За послідовними запитами вводяться числа x, y, z та символи a і b .

Вивести (включити у потік STL — `cout`)*:

8.2.1. Прізвище та ім'я розробника програми зі знаком охорони авторського права «©» (від англ. `copyright`);

8.2.2. Результат логічного виразу в числовому вигляді (1/0):

$$a + 1 \leq b ?$$

8.2.3. Значення x, y, z в десятковій і шістнадцятковій системах числення; S , що обчислюється функцією `s_calculation()` заголовкового файлу `ModulesПрізвище.h`.

* Підзадачі 8.2.1–8.2.3 варто реалізувати у вигляді функцій, результат виконання яких включається у вихідний потік `cout` за допомогою оператора вставки `<<` (наприклад, “`cout << YourFunc(a,b);`”).

Хід роботи

Завдання 1

Завдання 8.1

Здійснено аналіз і постановку задачі 8.1. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного модуля розв'язування задачі 8.1.

Алгоритмізувавши задачу отримаємо:

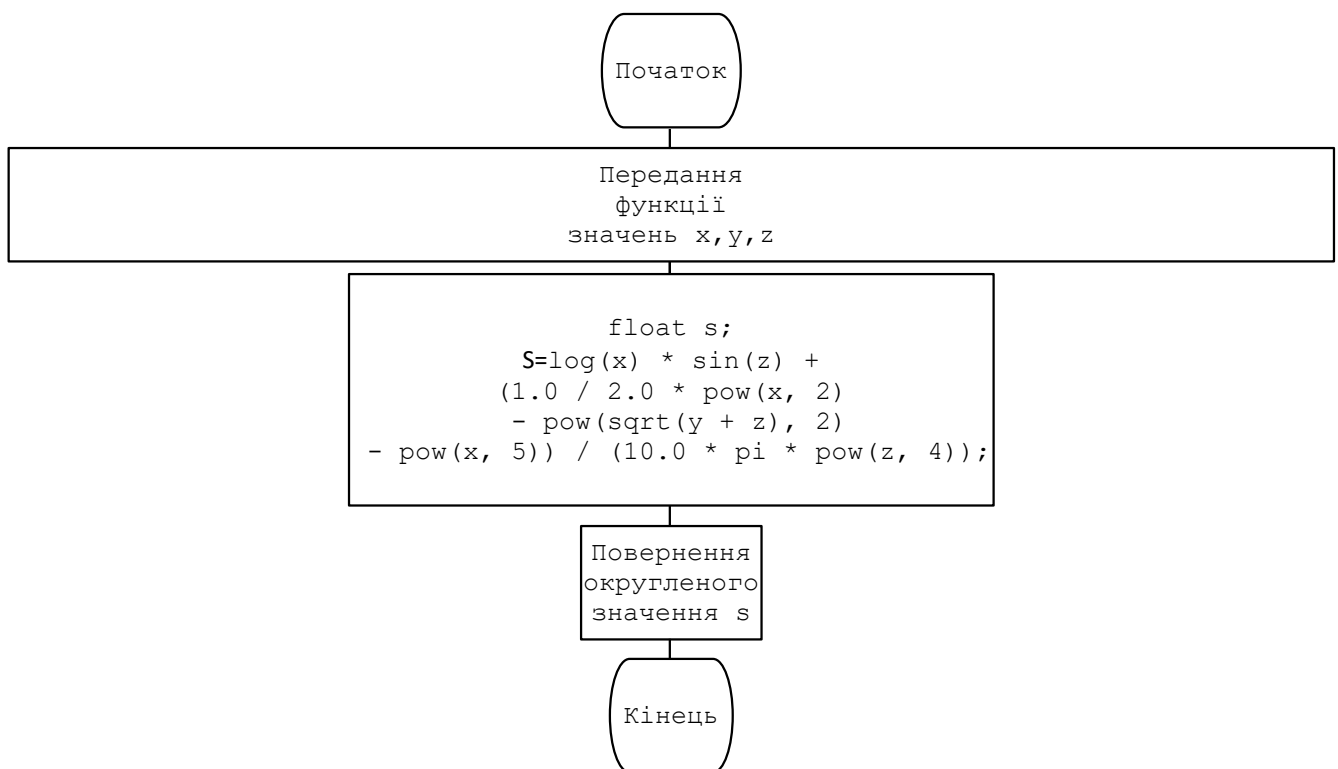
Вхідні данні: x, y, z .

Вихідні данні: s .

Обмеження, допущення:

x, y, z, s — Дійсні числа

Блок схема



Лістинг задачі 8.1

```
#define _USE_MATH_DEFINES
#include <math.h>
#include <iostream>
```

```
float s_calculation(float x, float y, float z)
{
```

```

float S;
const double pi = 3.14159265358979323846;
S=log(x) * sin(z) + (1.0 / 2.0 * pow(x, 2) - pow(sqrt(y + z), 2) - pow(x, 5))
/ (10.0 * pi * pow(z, 4));
return roundf(S * 100000.0f) / 100000.0f;

```

Завдання 2

Завдання 8.2

Здійснено аналіз і постановку задачі 8.2. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного забезпечення розв'язування задачі 8.2.

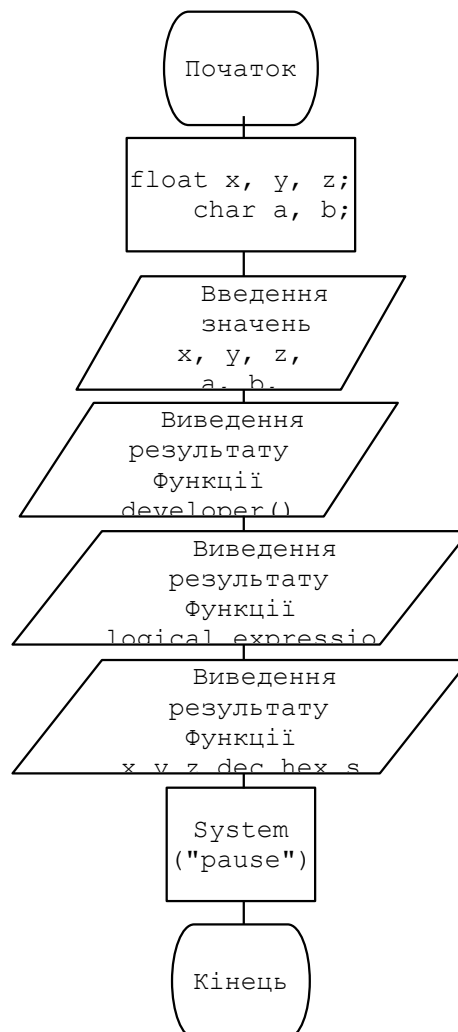
Алгоритмізувавши задачу отримаємо:

Вхідні данні: x, y, z, a, b

Вихідні данні: анотація розробника(8.2.1), результат логічного виразу(8.2.2),

x, y, z в десятковій та шістнадцятковій системі числення та значення s.

Блок схема



Лістинг задачі 8.2

```

#include <iostream>
#include <cmath>
#include <iomanip>
#include <conio.h>

#include "ModulesKaras.h"

void copyright();
void expression(char a, char b);
void DecHexFunc(int x, int y, int z);

using namespace std;

void copyright()
{
    cout << "© Карась Ілля" << endl << endl;
}

void expression(char a, char b)
{
    bool expression = a + 1 <= b;
    cout << noboolalpha << "Результат a + 1 <= b: " << expression << endl << endl;
}

void DecHexFunc(int x, int y, int z)
{
    cout << "'x' В десятковій: " << dec << x << endl;
    cout << "'y' В десятковій: " << dec << y << endl;
    cout << "'z' В десятковій: " << dec << z << endl;
    cout << "'x' В шістнадцятковій: " << hex << x << endl;
    cout << "'y' В шістнадцятковій: " << hex << y << endl;
    cout << "'z' В шістнадцятковій: " << hex << z << endl << endl;
}

int x, y, z;
char a, b;

int main()
{
    system("chcp 65001 && cls");
    copyright();

    cout << "Введіть x: ";
    cin >> x;

```

```

cout << "Введіть y: ";
cin >> y;
cout << "Введіть z: ";
cin >> z;
cout << "Введіть символи 'a' та 'b': ";
cin >> a >> b;

cout << endl;

expression(a, b);
DecHexFunc(x, y, z);

cout << "Результат обчислення виразу: S = " << s_calculation(x, y, z) << endl;
getch();

return 0;
}Додатки:

```

Додаток 1-TestSuite8.1

Назва тестового набору	TestSuite8.1
Рівень тестування	Модульний(Unit-Testing)
Автор	Карась Ілля Миколайович
Виконавець	Карась Ілля Миколайович

ID	Введені дані	Виведені дані	Результат
1	x = 5, y = 10, z = 3	s= -0.453831	PASSED
2	x = 5, y = 10, z = 3	s= -0.161521	PASSED
3	x = 2.5, y = 1.2, z =1.5	s= -0.0885473	PASSED
4	x = 0, y = 0, z = 0	s= Некоректні вхідні данні	PASSED
5	x = 5, y = 10, z = 3	s= -0.000104685	PASSED

Додаток 2-TestSuite8.2

Назва тестового набору	TestSuite8.2
Рівень тестування	Системний
Автор	Карась Ілля Миколайович
Виконавець	Карась Ілля Миколайович

ID	Дії	Очікуваний результат	Результат
1	Запустити програму	Введіть значення x:	PASSED
2	Ввести значення x: 5	Введіть значення y:	PASSED
3	Ввести значення y: 10	Введіть значення z:	PASSED
4	Ввести значення z: 3	Введіть символ a:	PASSED
5	Ввести значення a: h	Введіть символ b:	PASSED
6	Ввести значення b: q	Карась Ілля © 1 Десяткова система числення x: 5 y: 10 z: 3 Шістнадцяткова система числення x: 5 y: a z: 3 S= -0.453831	PASSED

Додаток 3-Лістинг вихідного коду ModulesKaras

```
#define _USE_MATH_DEFINES
#include <math.h>
#include <iostream>
float s_calculation(float x, float y, float z)
{
    float S;
    const double pi = 3.14159265358979323846;
    S=log(x) * sin(z) + (1.0 / 2.0 * pow(x, 2) - pow(sqrt(y + z), 2) - pow(x, 5))
/ (10.0 * pi * pow(z, 4));
    return roundf(S * 100000.0f) / 100000.0f;
```

Додаток 4-Лістинг вихідного коду TestDriver

```
#include <iostream>
#include <cmath>
#include "ModulesKaras.h"

using namespace std;

void test_s_calculation()
{
    system("chcp 1251 && cls");

    float possiblresult[7] = {-0.453831, -0.161521, -0.0885473, 0,
                               0.078321, -0.000104685, 0};

    float x[7] = {5, 5, 2.5, 0, 1000000, -5, 1.79769e+10};
    float y[7] = {10, 10, 1.2, 0, 1000000, 10, 1.79769e+10};
    float z[7] = {3, 3, 1.5, 0, 1000000, 10, 1.79769e+10};

    for (short int i = 0; i < 7; i++)
    {
        cout << round(s_calculation(x[i], y[i], z[i])*10000)/10000.0 << " = " <<
        possiblresult[i] << endl;

        if (round(s_calculation(x[i], y[i], z[i])*10000)/10000.0 ==
        round(possiblresult[i]*10000)/10000.0)
        {
            cout << "Test #" << i + 1 << ": PASSED\n";
        }
        else
        {
            cout << "Test #" << i + 1 << ": FAILED\n";
        }
    }
}

int main()
{
    test_s_calculation();
}
```


Висновок

Завантажен власний Git-репозиторій <https://github.com/odorenskyi/Karas-Illia-KI222>
У \Lab8 заповнено файл README.md, створено теки prj, Software, TestSuite, Report;
отриманий вміст теки \Lab8 завантажено до Git-репозиторію.

Здійснено аналіз і постановку задачі 8.1. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного модуля розв'язування задачі 8.1. Розроблено набір контрольних прикладів до задачі 8.1 задля виконання модульного тестування (Unit testing) модулів C++. В Code::Blocks IDE створено проект статичної бібліотеки ModulesKaras, збережено у \Lab8\prj. На основі результатів проектування модуля, реалізовано мовою програмування C++ функцію s_calculation, яка за належним інтерфейсом реалізовує розв'язування задачі 8.1. Скомпільовано. В Code::Blocks IDE створено проект заголовкового файлу ModulesKaras в \Lab8\prj та описано в ньому прототип функції s_calculation. В Code::Blocks IDE у \prj створено проект консольного додатка C++, під назвою TestDriver. Реалізовано тестовий драйвер для виконання розроблених тестових наборів (\Lab8\TestSuite) і за його допомогою виконано модульне тестування функції s_calculation зі статичної бібліотеки libModulesKaras.a. Здійснено аналіз і постановку задачі 8.2. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного забезпечення розв'язування задачі 8.2. Розроблено тест-сьют для виконання системного тестування ПЗ розв'язування задачі 8.2. В Code::Blocks IDE створено проект консольного додатка Karas_task у теці \prj. Мовою програмування C++ реалізовано результати проектування програмного забезпечення розв'язування задачі 8.2. Консольний додаток Karas_task.exe скопійовано у \Software. В ході роботи над лабораторною роботою набув ґрунтових вмінь та практичних навичок застосування теоретичних положень методології модульного програмування, реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування C++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, розроблення програмних модулів та засобів у кросплатформовому середовищі Code::Blocks