

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
КН 22

_____ Червоний Є. В.

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 10

<https://github.com/odorenskyi/>

ВАРІАНТ 10

— ЗАДАЧА 8.1 —

За значеннями x, y, z обчислюється S :

$$S = \ln(x - y) + \sqrt{\frac{\pi \cdot x^2}{x + \frac{z}{2y^2}}}.$$

— 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):

$$a + 10 \geq b ?$$

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);”).



- Файл статичної бібліотеки у Code::Blocks IDE має розширення .a (у, наприклад, Visual Studio C++ — .lib), заголовковий файл — .h.
- У C++ змінні оголошуються (описуються) перед їх безпосереднім використанням (у C — строго на початку тіла функції).
- Форматування виведення інформації можливо реалізувати за допомогою відповідних маніпуляторів iomanip C++.

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:

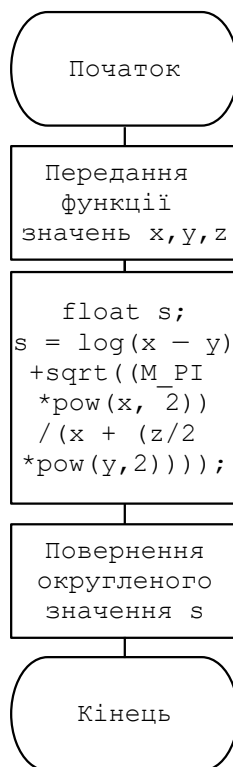
Вхідні данні: x, y, z .

Вихідні данні: s .

Обмеження, допущення:

x, y, z, s — Дійсні числа

Блок схема



Лістинг задачі 8.1

```
#include <iostream>
#include <math.h>
#define _USE_MATH_DEFINES
using namespace std;

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

```

float s;
s = log(x - y)+sqrt((M_PI * pow(x, 2))/(x + (z/2 * pow(y,2))));
return roundf(s * 100000.0f) / 100000.0f;
}

```

Завдання 2

Завдання 8.2

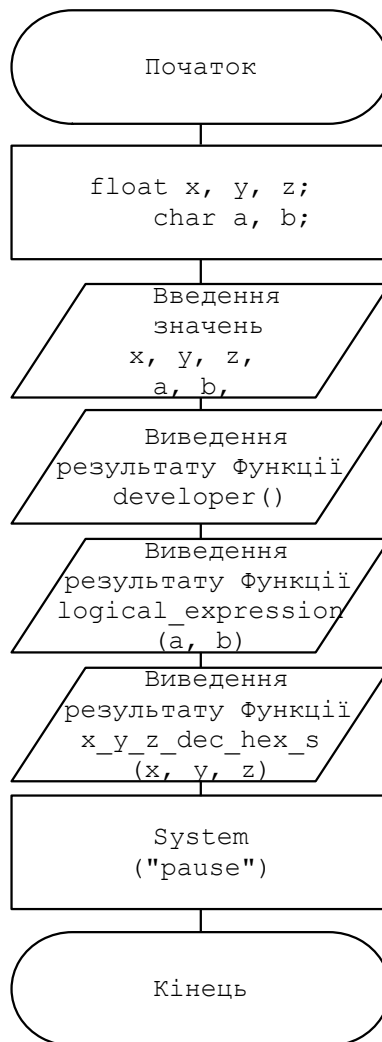
Здійснено аналіз і постановку задачі 8.2. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного забезпечення розв'язування задачі 8.2.

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 <windows.h>
#include "ModulesChervonyi.h"

using namespace std;

string developer(){
    string anotation = "Червоний Єгор ©";
    return anotation;
}

int logical_expression(char a, char b){
    if (a+10>=b){
        return 1;
    }
    else{
        return 0;
    }
}

string x_y_z_dec_hex_s(float x, float y, float z){
    cout << "Десяткова система числення" << endl
    << "x: " << x << endl
    << "y: " << y << endl
    << "z: " << z << endl;
    cout << "Шістнадцяткова система числення" << endl
    << "x: " << hex << (int)x << endl
    << "y: " << hex << (int)y << endl
    << "z: " << hex << (int)z << endl;
    if (x>y){
        cout << "S= " << s_calculation(x, y, z) <<endl;
    }
    else{
        cout << "Значення x має бути більше значення y" << endl;
    }
    return "";
}

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    float x, y, z;
```

```

char a, b;
cout << "Введіть значення x: ";
cin >> x;
cout << "Введіть значення y: ";
cin >> y;
cout << "Введіть значення z: ";
cin >> z;
cout << "Введіть символ a: ";
cin >> a;
cout << "Введіть символ b: ";
cin >> b;
cout << developer() << endl;
cout << logical_expression(a, b) << endl;
cout << x_y_z_dec_hex_s(x, y, z) << endl;
system("pause");
return 0;
}

```

Додатки:

Додаток 1-TestSuite8.1

Назва тестового набору	TestSuite8.1
Рівень тестування	Модульний(Unit-Testing)
Автор	Червоний Єгор Владиславович
Виконавець	Червоний Єгор Владиславович

ID	Введені дані	Виведені дані	Результат
1	x = 3, y = 2, z = 1	s= 2.378	PASSED
2	x = 5, y = 1, z = 6	s= 4.51958	PASSED
3	x = 9, y = 4, z = 7	s= 3.58805	PASSED
4	x = 7, y = 1, z = 5	s= 5.81718	PASSED
5	x = 1, y = 0.2, z = 0.5	s= 1.54051	PASSED

Додаток 2-TestSuite8.2

Назва тестового набору	TestSuite8.2
Рівень тестування	Системний
Автор	Червоний Єгор Владиславович
Виконавець	Червоний Єгор Владиславович

ID	Дії	Очікуваний результат	Результат
1	Запустити програму	Введіть значення x:	PASSED
2	Ввести значення x: 31	Введіть значення y:	PASSED
3	Ввести значення y: 25	Введіть значення z:	PASSED
4	Ввести значення z: 4	Введіть символ a:	PASSED
5	Ввести значення a: h	Введіть символ b:	PASSED
6	Ввести значення b: q	Червоний Єгор © 1 Десяткова система числення x: 31 y: 25 z: 4 Шістнадцяткова система числення x: 1f y: 19 z: 4 S= 3.32695	PASSED

Додаток 3-Лістинг вихідного коду ModulesChervonyi

```
#include <iostream>
#include <math.h>
#define _USE_MATH_DEFINES
using namespace std;

float s_calculation(float x, float y, float z)
{
    float s;
    s = log(x - y)+sqrt((M_PI * pow(x, 2))/(x + (z/2 * pow(y,2))));
    return roundf(s * 100000.0f) / 100000.0f;
}
```

Додаток 4-Лістинг вихідного коду TestDriver

```
#include <iostream>

#include "ModulesChervonyi.h"

using namespace std;

int main()
{
    float test_input[5][3] = {{3, 2, 1}, {5, 1, 6}, {9, 4, 7}, {7, 1, 5}, {1, 0.2, 0.5}};

    float test_output[5] = {2.378, 4.51958, 3.58805, 5.81718, 1.54051};

    for (int i=0;i<sizeof(test_input)/sizeof(test_input[0]); i++){
        if(test_input[i][0]<test_input[i][1]){
            cout << "X must be more then Y";
        }
        else{
            float s = s_calculation(test_input[i][0], test_input[i][1], test_input[i][2]);

            if(s == test_output[i]){
                cout << "Test #" << i+1 << " PASSED" << endl;
            }
            else{
                cout << "Test #" << i+1 << " FAILED" << endl;
            }
        }
    }
}
```


Висновок

Завантажен власний Git-репозиторій <https://github.com/odorenskyi/Chervonyi-Yehor-KN22>. 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 ModulesChervonyi, 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 ModulesChervonyi 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 libModulesChervonyi.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 Chervonyi_task у теці \prj. Мовою програмування C++ реалізовано результати проектування програмного забезпечення розв'язування задачі 8.2. Консольний додаток Chervonyi_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