

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 БІБЛОТЕК МОДУЛІВ
ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ
доцент кафедри кібербезпеки
та програмного забезпечення
Доренський О. П.
<https://github.com/odorenskyi/>

ВИКОНАВ
студент академічної групи КІ-
22-2
Макаренко М. О

ПЕРЕВІРИВ
викладач кафедри кібербезпеки
та програмного забезпечення
Олександр Собінов

Мета: полягає у набутті ґрунтовних вмінь і практичних навичок реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування C/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'язання задачі 8.1.
2. Реалізувати програмне забезпечення розв'язання задачі 8.2 – консольний застосунок

Варіант 5

— ЗАДАЧА 8.1 —

За значеннями x, y, z обчислюється S :

$$S = z + \pi \cdot \left(\frac{(2z+1)^2 - \sqrt{|y - \frac{1}{2}z|}}{\sqrt{\cos(z + yz) + x^2}} \right).$$

Вербальний опис програми:

Вхідні дані: числа x, y, z

Вихідні дані: обчислена формула S

Створити статичну бібліотеку де буде знаходитися математична формула та обчислювати її. Створити тест-драйвер для перевірки роботи функції та перевірити її 6 разів. З'єднати статичну бібліотеку з тест-драйвером та перевірити роботу модуля. Тест-драйвер видає результат Passed або Failed. За допомогою тест драйвера є можливість швидко перевірити результат.

Лістинг модуля

```
#include <iostream>
#include <cmath>
using namespace std;

double s_calculation(int x, int y, int z){
```

```

        return z + M_PI * (((pow((2 * z + 1), 2) - sqrt(abs((y -
(1/2) * z)))))/(cos(z + y * z) + pow(x, 2))));

```

}Лістинг тест-драйвера

```

#include <iostream>
#include <cmath>
#include "ModulesMakarenko.h"
#define ARRAY_SIZE 6
using namespace std;
int main() {
const float x[ARRAY_SIZE] = {1, -4, 0, -1, 100, -100};
const float y[ARRAY_SIZE] = {1, 2, 0, -1, 50, -50};
const float z[ARRAY_SIZE] = {1, 5, 0, -1, 100, -100};
const float expectedResult[ARRAY_SIZE] = {44,30,3,-1,113,-88};

for (short int i = 0; i < ARRAY_SIZE; i++) {
const float currentReasult = round(s_calculation(x[i], y[i],
z[i]));
if (currentReasult == expectedResult[i])
    cout << "Test #" << i+1 << " - PASSED\n";
else
    cout << "Test #" << i+1 << " - FAILED\n";
}
}

```

Результат компілювання тест-драйверу

Test #1 - PASSED

Test #2 - PASSED

Test #3 - PASSED

Test #4 – PASSED

Test #5 - PASSED

Test #6 – PASSED

Умова задачі 8.2

— 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 текстовому вигляді (`false/true`):

$$a + 1 \geq b ?$$

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

Вербальний опис програми до задачі 8.2

Вхідні дані: числа x , y , z та символи a, b

Вихідні дані: обчислена функція `s_calculation`, переведені числа

x, y, z у шістнадцяткову та десяткову систему числення, результат виразу $a + 1 \geq b$.

Лістинг програми до завдання 8.2

```
#include <iostream>
#include "ModulesMakarenko.h"
using namespace std;

void makar_copyright() {
    cout << "Makarenko Maxim ©" << endl;
}

void check(int a, int b) {
    if (a + 1 >= b)
    {
        cout << "True" << endl;
    }
    else
    {
        cout << "False" << endl;
    }
}

void hexadecimal(int x, int y, int z) {
    cout << "Decimal x-> ";
    cout << dec << x << endl;
    cout << "Hexadecimal x-> ";
    cout << hex << x << endl;
    cout << "Decimal y-> ";
```

```

        cout << dec << y << endl;
        cout << "Hexadecimal y-> ";
        cout << hex << y << endl;
        cout << "Decimal z-> ";
        cout << dec << z << endl;
        cout << "Hexadecimal z-> ";
        cout << hex << z << endl;
    }

int main()
{
    system("chcp 1251");
    system("cls");

    int x, y, z;
    double s, a, b;

    cout << "Enter integer x: " << endl;
    cin >> x;

    cout << "Enter integer y: " << endl;
    cin >> y;

    cout << "Enter integer z: " << endl;
    cin >> z;

    cout << "Enter a: " << endl;
    cin >> a;

    cout << "Enter b: " << endl;
    cin >> b;

    cout << "s = " << s_calculation(x, y, z) << endl;

    makar_copyright();

    check(a, b);

    hexadecimal(x, y, z);

    return 0;
}

```

Висновок: На цій лабораторній роботі я здобув навички створення статичної бібліотеки, лінування її з різними програмами та вивчив новий метод тестування програмного забезпечення який називається модульним. При виконанні лаб. роботи активно використовувались додаткові джерела. Проблеми виникли при створенні заголовкового файлу, але з допомогою додаткових джерел я зміг вирішити ці проблеми. Завдання були цікавими та корисними для здобуття практичних навичок з реалізації бібліотек та роботою з Git та GitHub.