

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
ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

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

ВИКОНАВ
студент академічної групи КН-
22
Бездольний К. 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 1

— ЗАДАЧА 8.1 —

За значеннями x, y, z обчислюється S :

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

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 <cmath>
```

```
const double pi = 3.14159;
```

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

```
    return abs(sin(abs(y - pow(z,2))) + sqrt(x) - sqrt(pow(y*z,x) + y/2*pi));  
}
```

Лістинг тест-драйвера

```
#include <iostream>
```

```
#include <cmath>
```

```
#include "ModulesKohan.h"
```

```
#define ARRAY_SIZE 6
```

```
using namespace std;
```

```
int main() {
```

```
    const float x[ARRAY_SIZE] = {2, 5, 4, 4, 4, 4};
```

```
    const float y[ARRAY_SIZE] = {3, 8, -2, 11, -4, 4};
```

```
    const float z[ARRAY_SIZE] = {1, 2, 5, 6, 9, 9};
```

```
    const float expectedResult[ARRAY_SIZE] = {1,1022,97,4354,1294,1293};
```

```
    for (short int i = 0; i < ARRAY_SIZE; i++) {
```

```
        const float currentReasult = round(s_calculation(x[i], y[i], z[i])*10000) / 10000;
```

```
        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):

$$a + 7 = b ?$$

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 $a + 7 = b$.

Лістинг програми до завдання 8.2

```
#include <iostream>
#include <locale>
#include "ModulesKohan.h"

using namespace std;

string copyright() {
    return "(c) Кохан Ігор";
}

bool expression(char a, char b) {
    return a + 7 == b;
}

string DecHexSFunc(int x, int y, int z) {
    cout << "В десятковій: " << endl
        << "'x' -> " << dec << x << endl
        << "'y' -> " << dec << y << endl
        << "'z' -> " << dec << z << endl << endl;
    cout << "В шістнадцятковій: " << endl
        << "'x' -> " << hex << x << endl
        << "'y' -> " << hex << y << endl
        << "'z' -> " << hex << z << endl << endl;
    cout << endl << "Результат обчислення виразу: S = " << s_calculation(x, y, z) << endl;
    return "";
}

int main() {
```

```

setlocale(LC_ALL, "ukr");
cout << copyright() << endl << endl;

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

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

cout << "Результат виразу: a + 7 = b -> " << expression(a, b) << endl << endl;
cout << DecHexSFunc(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.

Головна проблема при виконанні цієї лаб. Роботи це був додаток Code::Blocs код повинен був працювати але виникали дивні помилки. Які вирішувались перезапусками додатками та танцями з бубном. Ця лабораторна робота зайняла багато часу саме через це. Також при виконанні лаб. Роботи активно використовувались додаткові джерела. Проблеми виникли також при створенні тестового драйвера. Спочатку я не правильно зрозумів суть його роботи. Але потім мені підказали як його можна та треба реалізувати. Також при спробах через Git Bash файли додати до Git Hub репозиторію. Перша спроба була успішна а далі виникали помилки при спробі закомітити та запусити до гіт хабу файли. Вирішив я цю проблему так. Клонував репозиторій до нової папки та скинув туди нові файли потім запусив до гіт хаба. Це був неприємний але важливий досвід для становлення як програміста в майбутньому.