

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
КІ22-2
Ткаченко О. Ю.

ПЕРЕВІРИВ
ст. викладач
кафедри кібербезпеки
та програмного забезпечення
Собінов О. Г.

Мета: Набуття 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).

Варіант №19

Завдання: 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.

	Вхід	Вихід
1	1, 2, 2	9.14
2	0, 0, 0	4.14
3	-1, -2, -2	-0.19

Набір контрольних прикладів задачі 8.1

Результати тестування s_calculation зі статичної бібліотеки

libModulesTkachenkoOY.a

(9.14=9.14) 1 case status: passed

(4.14=4.14) 2 case status: passed

(-0.19=-0.19) 3 case status: passed

Test driver status: passed

Результати системного тестування: passed.

Висновок: : Виконавши цю лабораторну роботу з теми “ Реалізація статичних бібліотек модулів лінійних обчислювальних процесів” я набув 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).

В основному в мене не виникало проблем з ініціалізацією роботи над цією лабораторною, так як вже більш-менш знав основи Git та GitHub.

Для мене створення статичних бібліотек є новим, того це було доволі цікавим завданням, націлене на розвиток моїх навичок, як програміста, та поліпшило загальні навички пошуку рішення задач.

Статичні бібліотеки дозволяють забезпечити швидкий доступ до функціональності модулів, що входять до складу бібліотеки, що дозволяє розробникам працювати над окремими функціональними модулями, що спрощує розробку та підтримку коду. Крім того, статичні бібліотеки дозволяють використовувати готовий код з інших проектів, що зменшує час розробки та покращує якість коду.

Підсумовуючи, реалізація статичних бібліотек модулів лінійних обчислювальних процесів є важливим етапом у розробці програмного забезпечення, який дозволяє забезпечити швидкий доступ до функціональності та спрощує розробку та підтримку коду в команді.

Додатки

Додаток А – Тестовий артефакт

Назва тестового набору Test Suite Description	TS_lab8
Назва проекту / ПЗ Name of Project / Software	TkachenkoOY-task.exe
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сьюта Test Suite Author	Олексій Ткаченко
Виконавець Implementer	Олексій Ткаченко

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC-01	1. Запустити застосунок.	Вікно застосунку: ----- Ткаченко Олексій Юрійович (©) ----- Введіть послідовно числа x, y, z та символи a, b, розділяючи їх пробілами або Enter:	Passed
TC-02	1. Запустити застосунок. 2. Увести 1 2 2 a z	Вікно застосунку: ----- Ткаченко Олексій Юрійович (©) ----- Введіть послідовно числа x, y, z та символи a, b, розділяючи їх пробілами або Enter: 1 2 2 a z (a + 1 >= b) результат: false x, y, z у десятичній системі числення: 1 2 2 x, y, z у шістнадцятковій системі числення: 1 2 2 s = 9.14 Натисніть Enter для завершення...	Passed
TC-03	1. Запустити застосунок. 2. Увести 0 0 0 z a	Вікно застосунку: ----- Ткаченко Олексій Юрійович (©) ----- Введіть послідовно числа x, y, z та символи a, b, розділяючи їх пробілами або Enter: 0 0 0 z a (a + 1 >= b) результат: true x, y, z у десятичній системі числення:	Passed

		0 0 0 x, y, z у шістнадцятковій системі числення: 0 0 0 s = 4.14 Натисніть Enter для завершення...	
TC-04	1. Запустити застосунок. 2. Увести -1 -2 -2 а а	Вікно застосунку: <pre> ----- Ткаченко Олексій Юрійович (©) ----- </pre> Введіть послідовно числа x, y, z та символи a, b, розділяючи їх пробілами або Enter: -1 -2 -2 а а (a + 1 >= b) результат: true x, y ,z у десятковій системі числення: -1 -2 -2 x, y, z у шістнадцятковій системі числення: ffffff ffffffe ffffffe s = -0.19 Натисніть Enter для завершення...	Passed
TC-05	1. Запустити застосунок. 2. Увести 10,99.	<pre> ----- Ткаченко Олексій Юрійович (©) ----- </pre> Введіть послідовно числа x, y, z та символи a, b, розділяючи їх пробілами або Enter: 5 5000 5000 я я (a + 1 >= b) результат: true x, y ,z у десятковій системі числення: 5 5000 5000 x, y, z у шістнадцятковій системі числення: 5 1388 1388 s = 1.0005e+20 Натисніть Enter для завершення...	Passed

Додаток Б – Лістинг ModulesTkachenkoOY та TestDriver

ModulesTkachenkoOY:

```
#include <cmath>
```

```
#define PI 3.14
```

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

```

    float s = round(pow( 2*z+1 , x) * 100)/100 - round(sqrt(abs(y - z/2))*100)/100 +
z + PI;

    return s;
}

```

TestDriver:

```

#include <iostream>
#include <ModulesTkachenkoOY.h>
#include <cmath>
#define CASES_COUNT 3

using namespace std;

int main()
{
    bool passed = true;
    float output[CASES_COUNT] = {9.14, 4.14, -0.19};
    float input[CASES_COUNT] = {s_calculation(1, 2, 2), s_calculation(0,0,0),
s_calculation(-1,-2,-2)};
    for(int i=0;i<CASES_COUNT;i++){
        if (output[i] == input[i]){
            cout << "(" << output[i] << "=" << input[i] << ")" " << i+1 << " case status:
passed" << endl;
        }
        else{
            passed = false;
            cout << "(" << output[i] << "=" << input[i] << ")" " << i+1 << " case status:
failed" << endl;
        }
    }
}

```

```
if (passed){  
    cout << endl << "Test driver status: passed"<< endl;  
}  
else{  
    cout << endl << "Test driver status: failed"<< endl;  
}  
return 0;  
}
```