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 СТАТИСТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КН-24

Безкровна В. В.

ПЕРЕВІРИЛА

викладачка кафедри кібербезпеки   
та програмного забезпечення

Анастасія КОВАЛЕНКО

Кропивницький – 2025

**ТЕМА: РЕАЛІЗАЦІЯ СТАТИСТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ**

**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 С++ 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 - 18**

**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.

**Хід роботи**

Завантажили власний Git-репозиторій.

У \lab08 заповнили файл README.md, створили теки prj, Software, TestSuite, Report; отриманий вміст теки \lab08 завантажили до Git-репозиторію.

Аналіз задачі 8.1:

Дано: Значення x, y.

Знайти: Значення S.

Постановка задачі 8.1:

Вхідні дані: x, y.

Вихідні дані: S.

Аналіз вимог програмного модуля розв’язування задачі 8.1:

* Програма проводить розрахунки за формулою , використовуючи значення x та y;
* При введенні некоректних даних, програма повідомляє про помилку.

Проектування архітектури програмного модуля розв’язування задачі 8.1:

1. Підключення потрібних бібліотек;
2. Ініціалізація функції, оголошення x та y.
3. Перевірка чи є значення змінних коректним для рівняння;
4. Прорахунок рівняння;
5. Повернення результату;

Детальне проектування програмного модуля розв’язування задачі 8.1:

1. Підключення бібліотек «c.math» та «iostream», директиви для доступу до математичних констант «\_USE\_MATH\_DEFINES»;
2. Ініціалізація функції, оголошення x та y;
3. Якщо під коренем отримується від’ємне значення, програма повідомляє про помилку;
4. За формулою розраховується значення змінної S – результату роботи функції;
5. Повернення значення S.

Розробили набір контрольних прикладів до задачі 8.1 задля виконання модульного тестування модулів С++ (Див. Додаток А).

В Code::Blocks IDE створили проект статичної бібліотеки ModulesBezkrovna, зберегли його у \lab08\prj, розширення файлу вихідного коду змінили на cpp.

На основі результатів проектування модуля, реалізували мовою програмування С++ функцію s\_calculation.

Скомпілювали проект статичної бібліотеки ModulesBezkrovna.

В Code::Blocks IDE створили проект заголовкового файлу ModulesBezkrovna в \lab08\prj та описали в ньому прототип функції s\_calculation.

В Code::Blocks IDE у \prj створили проект консольного додатка С++, іменували його TestDriver.

Реалізували тестовий драйвер для виконання розроблених тестових наборів і за його допомогою виконали модульне тестування функції s\_calculation зі статичної бібліотеки lib ModulesBezkrovna.а.

Результати тестування s\_calculation зі статичної бібліотеки libModulesBezkrovn.а тестовим драйвером:

Test case #01 PASSED.

Test case #02 PASSED.

Test case #03 PASSED.

Test case #04 PASSED.

Test case #05 PASSED.

Test case #06 PASSED.

Вихідний код проектів ModulesBezkrovna та TestDriver включили до звіту як додатки. (Див. Дотаток Б, Додаток В).

Аналіз задачі 8.2:

Дано: x, y, z, a, b.

Знайти: логічний вираз в числовому вигляді, x, y, z в десятковій і шістнадцятковій системах числення, S.

Постановка задачі 8.2:

Вхідні дані: x, y, z, a, b.

Вихідні дані: x\_ hex, y\_ hex, z\_ hex, S, logical\_expression.

Аналіз вимог програмного забезпечення розв’язування задачі 8.2:

* Прізвище та ім’я розробника програми зі знаком охорони авторського права «©»
* Інформація про розробника та обробка логічного виразу реалізується у вигляді функцій, результат виконання яких включається у вихідний потік.
* S обчислюється функцією s\_calculation.

Проектування архітектури програмного забезпечення розв’язування задачі 8.2:

1. Підключення потрібних бібліотек;
2. Введення x, y, z, a, b.
3. Функція виводить прізвище та ім’я розробника програми
4. Функція обчислює результат логічного виразу
5. Значення x, y, z переводяться в десяткову і шістнадцяткову системи числення
6. Виведення результатів обчислень.

Детальне проектування програмного забезпечення розв’язування задачі 8.2:

1. Підключення потрібних бібліотек для роботи з введенням\виведенням, математичними операціями.
2. Підключення заголовкового файлу «ModulesBezkrovna.h»
3. Реалізується функція author виводить прізвище та ім’я розробника програми
4. Реалізується функція bool\_expression обчислює результат логічного виразу та виводить його у вигляді 1\0
5. Реалізується функція decimal\_to\_hex, яка переводить числа в шістнадцяткову систему
6. Вводяться значення x, y, z, a, b
7. Виклик реалізованих раніше функцій
8. Виклик функції s\_calculation
9. Виведення результатів обчислення та переведення.

Розробили тест-сьют для виконання системного тестування ПЗ розв’язування задачі 8.2. (Див. Додаток Г).

В Code::Blocks IDE створили проект консольного додатка Bezkrovna\_task у теці \prj.

Мовою програмування С++ реалізували результати проектування програмного забезпечення розв’язування задачі 8.2, скомпілювали проект.

Додаток А

Таблиця 1

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_8\_1 |
| Назва проекта / ПЗ  Name of Project / Software | libModulesBezkrovna.а |
| Рівень тестування  Level of Testing | Модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Вероніка Безкровна |
| Виконавець  Implementer | Вероніка Безкровна |

Таблиця 2

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Вхід/Input | Вихід/Output | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TC-1 | x=2  y=3 | S= -20.58 | passed |
| TC-2 | x=6  y=4 | S= 35.98 | passed |
| TC-3 | x=4  y=-7 | S=407.97 | passed |
| TC-4 | x=10  y=2 | S= 123.95 | passed |
| TC-5 | x=8  y=5 | S= 564 | passed |
| TC-6 | x=1  y=12 | S= -1724.02 | passed |

Додаток Б

Лістинг ModulesBezkrovna:

#include <cmath>

#include <iostream>

#define \_USE\_MATH\_DEFINES

float s\_calculation (float x, float y)

{

if ((pow(y,x) - M\_PI) < 0)

{

std::cout << "Помилка: вираз під квадратним коренем від'ємний" << std::endl;

}

float S = pow(x,2) - pow(y, 3) + sqrt(pow(y,x) - M\_PI);

return S;

}

Додаток В

Лістинг TestDriver:

#include <iostream>

#include <iomanip>

#include <locale>

#include <windows.h>

#include <cmath>

#include "ModulesBezkrovna.h"

using namespace std;

void test\_s\_calculation(void)

{

float e[6] = {-20.58, 35.98, 407.97, 123.95, 564, -1724.02};

float n[6] = {2, 6, 4, 10, 8, 1};

float m[6] = {3, 4, -7, 2, 5, 12};

short i;

float result;

for (i=0; i<6; i++)

{

result = s\_calculation(n[i],m[i]);

result = round(result \* 100) / 100.0;

if (result == e[i])

printf ("Test case #%02d PASSED.\n", i+1);

else printf("Test case #%02d FAILED.\n", i+1);

}

}

int main()

{

test\_s\_calculation();

return 0;

}