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

ВИКОНАВ

студент академічної групи КН-24

Ярослава Дмитренко

ПЕРЕВІРИЛА

викладачка кафедри кібербезпеки

та програмного забезпечення

Анастасія КОВАЛЕНКО

Кропивницький – 2025

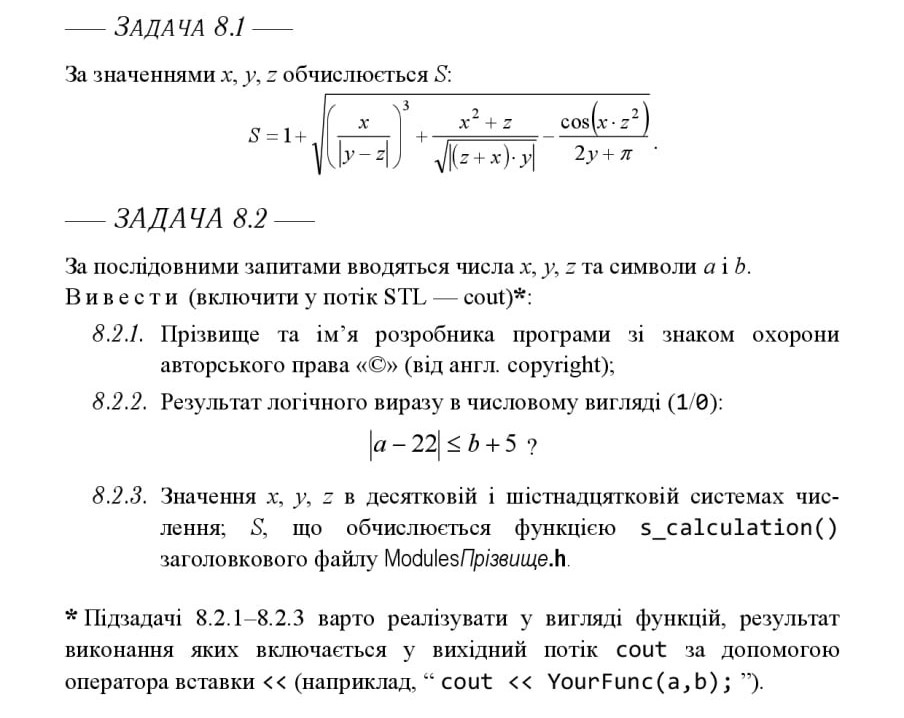
**Тема:** РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ.

**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 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).

**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 14**



Вхідні дані: x, y, z, a, b

Вихідні дані: S, bool\_result(a, b).

a – int;

b – int;

x – double;

y – double;

z – double;

S – double.

**Архітектура**

1. Створення статичної бібліотеки:
   * ModulesDmytrenko.h – заголовковий файл, що містить прототип функції s\_calculation.
   * файл ModulesDmytrenko.cpp – файл реалізації функції s\_calculation.
2. Створення тестового проєкту TestDriver:
   * у файлі main.cpp викликається функція s\_calculation(), що тестує дані.
3. Створення набору тестів TestSuite.

**Висновок:** У ході виконання роботи було досягнуто поставлену мету роботи. У ході роботи ознайомилася з модульним програмування, функціональною декомпозицією та блочним тестування. Набула навичок представлення скалярних типів, арифметичних і логічних операцій, роботу з введенням і виведенням даних у C++.

**Аргументи на користь досягнення мети роботи**

1. Аналіз обчислюваної задачі допоміг у її реалізації.
2. Чітко оформлений тест-сьют спростив аналіз результатів тестування.
3. Попередній досвід у створення тестових випадків допоміг у їх правильному оформленні.
4. Слідування методичним вказівкам допомогло в процесі роботи.
5. Правильно оформлений тестовий набір забезпечив ефективність тестування.
6. Завдяки тестуванню вдалося порівняти очікувані та отримані результати.
7. Проведення відлагодження, при виявленні помилок, забезпечило коректність результатів.
8. Виконання двадцяти тестових випадків дозволило ефективно перевірити роботу програми.
9. Створений список, вхідних і вихідних даних, полегшив написання коду.
10. Виконання аналізу алгоритму допомогло уникнути помилок на ранніх етапах.
11. Документування допомогло не пропустити виконання важливих етапів роботи.
12. Використання кросплатформового середовища Code::Blocks забезпечило якісну компіляцію.
13. Опрацювання стандартів ISO 5807:1985 допомогло у правильності побудови блок-схеми.
14. Створення алгоритму на основі структурного підходу допомогло в реалізації задачі.
15. Завдяки тест-сайсам вдалося перевірити програму на відповідність вимогам.
16. Представлення алгоритму у вигляді блок-схеми полегшило розуміння реалізації програми.
17. Завдяки меті роботи вдалося зосередитися на важливих моментах роботи.
18. Тест-кейси допомогли в стеженні за змінами результатів виконання програми, в залежності від вхідних даних.
19. Використання блок-схеми, як способу формального представлення алгоритму, зменшило ймовірність виникнення помилок у структурі.
20. Структура тест-кейсів допомагає виявити помилки або в роботі програми.
21. Аналіз допоміг у виявленні помилок.
22. Чітка організація тек пришвидшила знаходження потрібних файлів.
23. Завдяки набору тестових випадків вдалося перевірити як програма реагує на введення різних вхідних даних.
24. Успішних результатів вдалося досягти завдяки слідуванню вимогам виконання роботи.
25. Навчальні матеріали допомогли розібратися у поставлених задачах.
26. Тест-сьют дає змогу одразу встановити статус кожного тесту, що допомагає пришвидшує аналіз стану програми.
27. Завдяки уважному опрацюванню умови задачі вдалося успішно її реалізувати.
28. Правильно розподіл часу пришвидшив виконання завдань.
29. Блок-схема дала змогу побачити варіанти розвитку подій.
30. Завдяки формальному представлення алгоритму легше проводити оптимізацію.
31. При внесені зміни в код, тест-кейси дав змогу перевірити, чи не пошкодив новий функціонал уже існуючий.
32. Використання зручного, кросплатформового середовища Code::Blocks полегшило написання програми.
33. Використання тест-сьюту забезпечило комплексне тестування.
34. Чітко описана послідовність створення тек допомогла в їхній організації.
35. Коментування в певних частинах коду полегшує його розуміння.
36. Використання тест-кейсів забезпечило систематичний підхід до тестування.
37. Завдяки систематичному тестуванню вдалося виявити всі можливі помилки.
38. Використання функцій полегшило реалізацію програми.
39. Додаткова література допомогла підготуватися до лабораторної роботи.
40. Лістинг 6.1 допоміг зрозуміти як виконувати завдання.
41. Використання тестових випадків забезпечило перевірку функціональності програми.
42. Створення програм за певною структурою полегшило їх розуміння.
43. Чітко сформована мета допомогла організувати роботу.
44. Виконання вказівок допомогло реалізувати завдання.
45. Побудова блок-схеми допомогла сформувати послідовність побудови коду.
46. Попередній досвід використання Code::Blocks спростив процес роботи.
47. Завдяки тест-сьюту було проведено швидке тестування.
48. Слідування вимогам ISO/IEC 9899:2018 забезпечило правильність синтаксису.
49. Проходження самоконтролю допомогло підготуватися до роботи.
50. Реалізація програми відповідно до стандартів мови С забезпечила її правильність.

**Додатки**

**Лістинг програми ModulesDmytrenko**

#include <iostream>

#include <cmath>

#include <cstdio>

double s\_calculation(double x, double y, double z)

{

double S;

if (y == z || ((z + x) \* y) == 0 || (2 \* y + M\_PI) == 0) {

std::cout << "Помилка: Дiлення на нуль!" << std::endl;

return NAN;

}

double resulte = 1 + sqrt(pow((x / abs(y - z)), 3) + ((pow(x, 2) + z) / sqrt(abs((z + x) \* y))) - cos(x \* pow(z, 2)) / (2 \* y + M\_PI));

return resulte;

}

**Файл ModulesDmytrenko.h**

#ifndef MODULESDMYTRENKO\_H\_INCLUDED

#define MODULESDMYTRENKO\_H\_INCLUDED

double s\_calculation(double x, double y, double z);

#endif // MODULESDMYTRENKO\_H\_INCLUDED

**Лістинг файлу TestDriver**

#include <iomanip>

#include <iostream>

#include "C:\Users\dmytr\Desktop\lab08\prj\ModulesDmytrenko\ModulesDmytrenko.h"

#include <locale>

#include <windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Ukrainian");

double x, y, z, S;

cout << "Введiть числа x, y, z: ";

cin >> x >> y >> z;

S = s\_calculation(x, y, z);

cout << "S = "<< S << endl;

return 0;

}

**Лістинг Dmytrenko\_task**

**#include <locale>**

**#include <windows.h>**

**#include "C:\Users\dmytr\Desktop\lab08\prj\ModulesDmytrenko\ModulesDmytrenko.h"**

**using namespace std;**

**void Name()**

**{**

**cout << "Дмитренко Ярослава ©" << endl;**

**}**

**bool bool\_result(int a, int b) {**

**return abs(a - 22) <= (b + 5);**

**}**

**void oct\_hex(double x, double y, double z) {**

**cout << "Десяткове x = " << dec << x << endl;**

**cout << "Десяткове y = " << dec << y << endl;**

**cout << "Десяткове z = " << dec << z << endl;**

**cout << "Шiстнадцяткове x = " << hex << static\_cast<int>(x) << endl;**

**cout << "Шiстнадцяткове y = " << hex << static\_cast<int>(y) << endl;**

**cout << "Шiстнадцяткове z = " << hex << static\_cast<int>(z) << endl;**

**}**

**int main() {**

**setlocale(LC\_ALL, "Ukrainian");**

**Name();**

**int a, b;**

**cout << "Введiть значення a i b: ";**

**cin >> a >> b;**

**cout << "Результат: " << bool\_result(a, b) << endl;**

**double x, y, z, S;**

**cout << "Введiть числа x, y, z: ";**

**cin >> x >> y >> z;**

**oct\_hex(x, y, z);**

**S = s\_calculation(x, y, z);**

**cout << "S = "<< S << endl;**

**return 0;**

**}**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TestSuite\_8.1 |
| Назва проекта / ПЗ  Name of Project / Software | TestDriver\_8\_1.ехе |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Ярослава Дмитренко  телеграм: t.me/y\_aradm  e-пошта: dmytrenkovika4@gmail.com |
| Виконавець  Implementer | Ярослава Дмитренко |

Artifact: Test Suite

Date: 3/18/2025

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TC-1 | 1. Запустити застосунок | Вікно застосунку:  Введіть числа x, y, z: | passed |
| TC-2 | 1. Запустити застосунок 2. Увести 1 2 3 | Вікно застосунку:  S = 2.5943 | passed |
| TC-3 | 1. Запустити застосунок 2. Увести 35 57 89 | Вікно застосунку:  S = 5.11603 | passed |
| TC-4 | 1. Запустити застосунок 2. Увести 1 0 5 | Вікно застосунку:  Помилка: Дiлення на нуль!  S = nan | passed |
| TC-5 | 1. Запустити застосунок 2. Увести 1 5 0 | Вікно застосунку:  S = 1.61573 | passed |

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TestSuite\_8.2 |
| Назва проекта / ПЗ  Name of Project / Software | TestDriver\_8\_2.ехе |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Ярослава Дмитренко  телеграм: t.me/y\_aradm  e-пошта: dmytrenkovika4@gmail.com |
| Виконавець  Implementer | Ярослава Дмитренко |

Artifact: Test Suite

Date: 3/18/2025

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TC-1 | 1. Введiть значення a i b: 2. a = 23 3. b = 1 4. Введiть числа x, y, z: 5. x = 2 6. y = 3 7. z = 4 | Вікно застосунку:  Дмитренко Ярослава ©  Введiть значення a i b: 23 1  Результат: 1  Введiть числа x, y, z: 2 3 4  Десяткове x = 2  Десяткове y = 3  Десяткове z = 4  Шiстнадцяткове x = 2  Шiстнадцяткове y = 3  Шiстнадцяткове z = 4  S = 4.12959 | passed |
| TC-2 | 1. Введiть значення a i b: 2. a = 34 3. b = 27483 4. Введiть числа x, y, z: 5. x = 23 6. y = 45 7. z = 67 | Вікно застосунку:  Дмитренко Ярослава ©  Введiть значення a i b: 34 27483  Результат: 1  Введiть числа x, y, z: 23 45 67  Десяткове x = 23  Десяткове y = 45  Десяткове z = 67  Шiстнадцяткове x = 17  Шiстнадцяткове y = 2d  Шiстнадцяткове z = 43  S = 4.2418 | passed |
| TC-3 | 1. Введiть значення a i b: 2. a = 12 3. b = 4 4. Введiть числа x, y, z: 5. x = 40 6. y = 26 7. z = 16 | Вікно застосунку:  Дмитренко Ярослава ©  Введiть значення a i b: 12 4  Результат: 0  Введiть числа x, y, z: 40 26 16  Десяткове x = 40  Десяткове y = 26  Десяткове z = 16  Шiстнадцяткове x = 28  Шiстнадцяткове y = 1a  Шiстнадцяткове z = 10  S = 11.3127 | passed |
| TC-4 | 1. Введiть значення a i b: 2. a = 34 3. b = 27483 4. Введiть числа x, y, z: 5. x = 23 6. y = 45 7. z = 67 | Вікно застосунку:  Дмитренко Ярослава ©  Введiть значення a i b: 54 3  Результат: 0  Введiть числа x, y, z: 2 0 6  Десяткове x = 2  Десяткове y = 0  Десяткове z = 6  Шiстнадцяткове x = 2  Шiстнадцяткове y = 0  Шiстнадцяткове z = 6  Помилка: Д?лення на нуль!  S = nan | passed |
| TC-5 | 1. Введiть значення a i b: 2. a = 0 3. b = 0 4. Введiть числа x, y, z: 5. x = 23 6. y = 55555 7. z = 0 | Вікно застосунку:  Дмитренко Ярослава ©  Введiть значення a i b: 0 0  Результат: 0  Введiть числа x, y, z: 23 55555 0  Десяткове x = 23  Десяткове y = 55555  Десяткове z = 0  Шiстнадцяткове x = 17  Шiстнадцяткове y = d903  Шiстнадцяткове z = 0  S = 1.68409 | passed |