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

ВИКОНАВ

студент академічної групи КБ-24

Радченко М. Д.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А. С.

Кропивницький – 2025

**Мета роботи :** 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).

**Завдання :**

Реалізувати статичну бібліотеку модулів libModulesПрізвище C/C++, яка містить функцію розв’язування задачі 8.1.

Реалізувати програмне забезпечення розв’язування задачі 8.2 — консольний застосунок.

**Варіант :** №22

**ЗАДАЧА 8.1**

За значеннями x, y, z обчислюється Ѕ:

S = - In|sin z|

#include <iostream>

#include <cmath>

using namespace std;

int main() {

double x, y, z, S;

// Введення значень x, y, z

cout << "Введіть значення x, y, z: ";

cin >> x >> y >> z;

// Перевірка на коректність значень (щоб уникнути логарифма та синуса від нуля)

if (sin(z) == 0) {

cout << "Помилка: sin(z) дорівнює нулю, логарифм не визначений!" << endl;

return 1;

}

// Обчислення виразу

S = (0.5 \* x \* x) - sqrt(abs(pow(y + z, 2) - pow(x, 5))) - log(abs(sin(z)));

// Виведення результату

cout << "Значення S: " << S << endl;

return 0;

}

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_8\_1 |
| Назва проекта / ПЗ  Name of Project / Software | Radchenko\_task.exe |
| Рівень тестування  Level of Testing | модульний / System Testing |
| Автор тест-сьюта  Test Suite Author | Радченко Максим |
| Виконавець  Implementer | Радченко Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TC-01 | Введення  X = 12 ;  Y = 2;  Z = 10; | = -426 | PASSED |
| TC-02 | Введення  X = 30;  Y = 60;  Z = 5; | = -4479 | PASSED |
| TC-03 | Введення  X = 1;  Y = 1;  Z = 1; | = -1.05 | PASSED |
| TC-04 | Введення  X = 7;  Y = -10;  Z = 89; | = 13.46949 | PASSED |
| TC-05 | Введення  X = 10;  Y = 4;  Z = 1000; | = -902 | PASSED |
| TC-06 | Введення  X = 10;  Y = 4;  Z = -1000; | = 894.276 | PASSED |
| TC-07 | Введення  X = 0;  Y = 0;  Z = 0; | Помилка | PASSED |
| TC-08 | Введення  X = -10;  Y = 4;  Z = 3; | - 264 | PASSED |
| TC-09 | Введення  X = 6;  Y = 2;  Z = 1; | = -69,958 | PASSED |
| TC-10 | Введення  X = 1;  Y = 0;  Z = 1; | = 0.6726 | PASSED |

Задача 8.2

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

// Функція для виводу імені розробника

void printDeveloper() {

cout << "Розробник: Прізвище Ім'я ©" << endl;

}

// Функція для обчислення логічного виразу (1 або 0)

int logicalExpression(char a, char b) {

return (a + 1 <= b) ? 1 : 0;

}

// Функція для обчислення S

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

if (sin(z) == 0) {

cerr << "Помилка: sin(z) = 0, логарифм не визначений!" << endl;

return NAN;

}

return (0.5 \* x \* x) - sqrt(abs(pow(y + z, 2) - pow(x, 5))) - log(abs(sin(z)));

}

// Функція для виводу чисел у двох системах числення

void printNumbers(double x, double y, double z) {

cout << "Числа у десятковій системі: x = " << x << ", y = " << y << ", z = " << z << endl;

cout << hex << uppercase;

cout << "Числа у шістнадцятковій системі: x = " << static\_cast<int>(x)

<< ", y = " << static\_cast<int>(y)

<< ", z = " << static\_cast<int>(z) << dec << endl;

}

int main() {

double x, y, z;

char a, b;

// Введення даних

cout << "Введіть x, y, z: ";

cin >> x >> y >> z;

cout << "Введіть два символи a і b: ";

cin >> a >> b;

// Вивід результатів

printDeveloper();

cout << "Результат логічного виразу: " << logicalExpression(a, b) << endl;

printNumbers(x, y, z);

cout << "Значення S: " << s\_calculation(x, y, z) << endl;

return 0;

}

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_8\_2 |
| Назва проекта / ПЗ  Name of Project / Software | Radchenko\_task.exe |
| Рівень тестування  Level of Testing | модульний / System Testing |
| Автор тест-сьюта  Test Suite Author | Радченко Максим |
| Виконавець  Implementer | Радченко Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TC-01 | Введіть 12  Введіть 2  Введіть 4  Введіть 12  Введіть 13 | Enter x  12  Enter y  2  Enter z  4  Enter a  12  Enter b  13  Radchenko Maksim ©  a+1 <=b = 1  x = 12, y = 2, z = 4  x = C, y = 2, z = 4  S: -426.516 | passed |
| TC-02 | Введіть -12  Введіть 2  Введіть 10  Введіть 8  Введіть 13 | Enter x  -12  Enter y  2  Enter z  10  Enter a  8  Enter b  13  Radchenko Maksim ©  a+1 <=b = 0  x = -12, y = 2, z = 10  x = FFFFFFF4, y = 2, z = A  S: -426.366 | passed |
| TC-03 | Введіть 40  Введіть 5  Введіть 80  Введіть 12  Введіть 6 | Enter x  40  Enter y  5  Enter z  80  Enter a  12  Enter b  6  Radchenko Maksim ©  a+1 <=b = 1  x = 40, y = 5, z = 80  x = 28, y = 5, z = 50  S: -9318 | passed |
| TC-05 | Введіть 1  Введіть 1  Введіть 1  Введіть 1  Введіть 1 | Enter x  1  Enter y  1  Enter z  1  Enter a  1  Enter b  1  Radchenko Maksim ©  a+1 <=b = 0  x = 1, y = 1, z = 1  x = 1, y = 1, z = 1  S: -1,05 | passed |
| TC-06 | Введіть 0  Введіть 0  Введіть 0  Введіть 0  Введіть 0 | Enter x  0  Enter y  0  Enter z  0  Enter a  0  Enter b  0  Radchenko Maksim ©  a+1 <=b = 0  x = 0, y = 0, z =0  x = 0, y = 0, z = 0  S: nan | passed |
| TC-07 | Введіть -2  Введіть 11  Введіть 22  Введіть 20  Введіть 10 | Enter x  -2  Enter y  11  Enter z  22  Enter a  20  Enter b  10  Radchenko Maksim ©  a+1 <=b = 0  x =-2, y = 11, z =12  x = F, y = B, z = C  S: -21.06 |  |
| TC-08 | Введіть 10  Введіть 50  Введіть 100  Введіть 120  Введіть 1 | Enter x  10  Enter y  50  Enter z  100  Enter a  120  Enter b  1  Radchenko Maksim ©  a+1 <=b = 0  x = 10, y = 50, z =100  x = 0, y = 32, z = 64  S: -227 | passed |
| TC-09 | Введіть 1  Введіть 0  Введіть 1  Введіть 0  Введіть 1 | Enter x  1  Enter y  0  Enter z  1  Enter a  0  Enter b  1  Radchenko Maksim ©  a+1 <=b = 1  x = 1, y = 0, z =1  x = 1, y = 0, z = 1  S: 0.67 | passed |
| TC-10 | Введіть 0  Введіть 1  Введіть 0  Введіть 1  Введіть 0 | Enter x  0  Enter y  1  Enter z  0  Enter a  1  Enter b  0  Radchenko Maksim ©  a+1 <=b = 0  x = 0, y = 1, z =0  x = 0, y = 1, z = 0  S: nan | passed |

Аргументи для досягнення мети лабораторної роботи:

1. Завдання виконано з використанням модульного підходу до програмування, що спрощує налагодження та підтримку програми.
2. Створена статична бібліотека забезпечує можливість повторного використання коду в різних проектах.
3. Прототип функції, вказаний у заголовковому файлі, допомагає чітко розмежувати інтерфейс та реалізацію.
4. Кожен модуль і функція супроводжуються коментарями, що полегшують розуміння логіки коду.
5. Використання стандартних бібліотек (cmath, iostream, iomanip) гарантує кросплатформність програми.
6. Перевірки вхідних даних запобігають арифметичним помилкам, що покращує надійність.
7. Функція s\_calculation використовує математичні функції ,що демонструє ефективне застосування бібліотеки cmath.
8. Розроблений тестовий драйвер перевіряє роботу функції на різних наборах даних, що забезпечує високу якість коду.
9. Перевірка результатів з використанням epsilon гарантує точність обчислень з числами з плаваючою крапкою.
10. Демонструється здатність форматувати вивід, що важливо для точності звітності.
11. Усі обчислення виконуються в окремій функції, що покращує структуру коду.
12. Виконано всі пункти методичних рекомендацій, що підтверджує досягнення поставленої мети.
13. Використання Git-репозиторію забезпечує відстеження змін та історії розробки.
14. Розділення прототипів і реалізації покращує організацію коду та спрощує співпрацю між розробниками.
15. Форматування виводу забезпечує чіткість і зрозумілість результатів.
16. Проект відповідає міжнародним стандартам, що підвищує якість програмного забезпечення.
17. Використання ефективних алгоритмів сприяє зменшенню часу обчислень.
18. Розглянуто крайні значення вхідних даних, що свідчить про уважний підхід до аналізу задачі.
19. Умовні оператори роблять код компактним і легким для розуміння.
20. Обробка випадків з NAN результатами запобігає непередбачуваним помилкам.
21. Тестовий набір містить як позитивні, так і негативні сценарії, що забезпечує повноцінне тестування функцій.
22. Логічні вирази побудовані коректно, що гарантує отримання правильних результатів.
23. Протокол тестування містить докладну інформацію про результати кожного тесту.
24. Тестування кожної функції окремо полегшує пошук і виправлення помилок.
25. Автоматичний запуск тестів підвищує ефективність перевірки програмного коду.
26. Використання математичних операцій, таких як множення, ділення та піднесення до степеня, свідчить про розуміння математичних принципів.
27. Логічні вирази правильно перевіряються, що підтверджує розуміння основ програмування.
28. Виконано всі завдання з методичних вказівок, що підтверджує досягнення мети роботи.
29. Реалізація проекту сприяє розвитку практичних навичок роботи з C++ та Code::Blocks.
30. Точна перевірка функцій забезпечує їх правильну роботу у різних ситуаціях.
31. Застосування математичних функцій гарантує точність обчислень.
32. Архітектура проекту дозволяє легко додавати нові модулі та розширювати функціональність.
33. Чітке виведення числових даних сприяє кращому розумінню результатів.
34. Робота з типами double та char демонструє універсальність розробки.
35. Коректне використання директиви #include гарантує правильну компіляцію проекту.
36. Розділення коду на окремі компоненти (бібліотека, заголовковий файл, тестовий драйвер) покращує організованість роботи.
37. Успішне створення статичної бібліотеки та її інтеграція з тестовим драйвером підтверджує правильність підключення компонентів.
38. Хоча застосовано процедурний підхід, принципи інкапсуляції та абстракції застосовуються у проекті.
39. Використання стандартних бібліотек гарантує переносимість коду між різними платформами.
40. Програма правильно обробляє недопустимі вхідні значення, повертаючи NAN.
41. Умовні оператори коректно обробляють різні логічні сценарії.
42. Розподіл коду на модулі значно спрощує виявлення та виправлення помилок.
43. Тестування охоплює як звичайні, так і граничні ситуації, що підвищує довіру до результатів.
44. Продемонстровано вміння працювати з різними системами числення.
45. Результати програми структуровані і легко інтерпретуються.
46. Протокол тестування містить деталі всіх тестових кейсів.
47. Раннє тестування дозволило виявити та виправити можливі проблеми в функціональності.
48. Детальне тестування забезпечує високий рівень якості кінцевого продукту.
49. Завершення всіх етапів (від аналізу до тестування) підтверджує комплексне розуміння методології розробки.
50. Завдання виконано згідно з методичними рекомендаціями, що підтверджує здобуття практичних навичок та досягнення мети курсу «Базові методології та технології програмування».

**Відповіді на контрольні запитання:**

**1. Мета та задачі процесів проектування ПЗ за стандартами ISO/IEC 12207 / ISO/IEC/IEEE 15288:**

Забезпечення узгодженого підходу до організації процесів розробки, експлуатації, обслуговування та завершення життєвого циклу програмного забезпечення (або систем загалом). Формування системних, повторюваних і контрольованих процесів, що гарантують високу якість кінцевого продукту. Чітке визначення, документація та контроль вимог до програмного забезпечення протягом усього життєвого циклу. Планування циклічних процесів (аналіз, проектування, валідація, тестування) для безперервного вдосконалення продукту. Забезпечення ефективної комунікації між розробниками, замовниками, користувачами та іншими учасниками процесу. Ідентифікація, оцінка та управління ризиками на всіх етапах розробки. Систематичне оформлення документації для забезпечення відтворюваності та підтримки проекту. Гнучке управління процесами для швидкого реагування на зміни вимог та умов проекту.

**2. Відмінність функції від модуля у C/С++:**

1. **Функція:**Це окрема одиниця коду, яка виконує певну операцію, приймаючи вхідні дані та повертаючи результат.Має чітко визначений інтерфейс (прототип) та реалізацію і може бути викликана з будь-якої частини програми. Призначена для виконання конкретного завдання або обчислення.
2. **Модуль:**Це логічний блок або набір пов'язаних функцій, класів і змінних, що об'єднані для виконання певної функціональності системи. Забезпечує інкапсуляцію і відокремлення інтерфейсу від реалізації, що сприяє організації коду та його повторному використанню (часто реалізується як статична або динамічна бібліотека). Має більший масштаб: модуль організовує групу функцій, забезпечуючи цілісність і ізоляцію окремих аспектів програми.