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”**

ВИКОНАВ

студент академічної групи КН-24

Червонець А.В

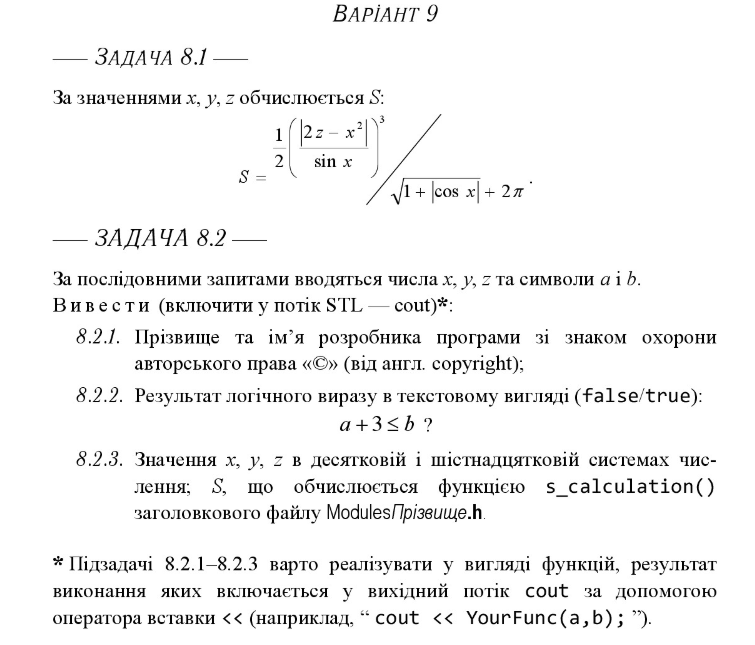
ПЕРЕВІРИВ ВИКЛАДАЧ

**ДОРЕНСЬКИЙ А.П**

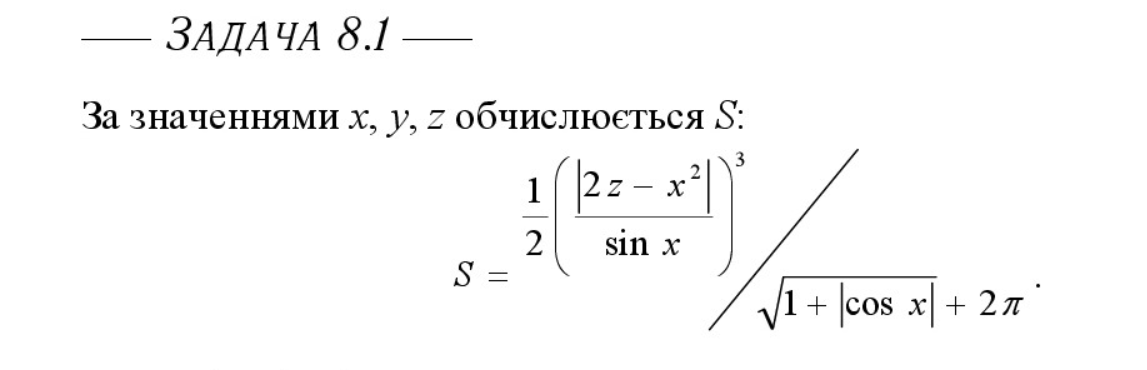
Тема: «РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ»

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).

Варіант 9



Задача 8.1



Лістинг програми ModulesChervonets

#include "ModulesChervonets.h"

#include <cmath> // для sin, cos, sqrt, fabs, pow, nan

#include <limits> // для numeric\_limits

double calculateS(double x, double /\*y\*/, double z) {

constexpr double pi = 3.14159265358979323846;

double sinx = std::sin(x);

if (sinx == 0.0) {

// якщо потрібно повернути NaN при діленні на нуль

return std::numeric\_limits<double>::quiet\_NaN();

}

double a = std::fabs(2.0 \* z - x \* x) / sinx;

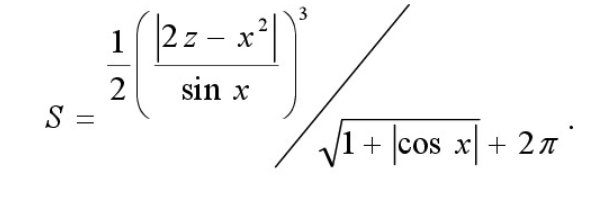
double numerator = 0.5 \* std::pow(a, 3);

double denominator = std::sqrt(1.0 + std::fabs(std::cos(x))) + 2.0 \* pi;

return numerator / denominator;

}

**Функція calculateS обчислює значення математичного виразу, перевіряючи можливість ділення на нуль. Вона використовує стандартні математичні операції: модуль, синус, косинус, піднесення до степеня та добування квадратного кореня. У випадку, якщо sin(x) = 0, функція виводить повідомлення про помилку і повертає спеціальне значення NaN (не число). Якщо ж значення допустимі, функція виконує обчислення виразу:**



Файл modulesChervonets.h

#ifndef MODULESCHERVONETS\_H\_INCLUDED

#define MODULESCHERVONETS\_H\_INCLUDED

double calculateS(double x, double y, double z);

#endif // MODULESCHERVONETS\_H\_INCLUDED

Лістинг Файлу TestDriver

#include <iostream>

#include <cmath>

#include "ModulesChervonets.h"

int main() {

double x, y, z;

std::cout << "x y z? ";

std::cin >> x >> y >> z;

double S = calculateS(x, y, z);

if (std::isnan(S)) {

std::cout << "S = NaN (sin(x) == 0)\n";

} else {

std::cout << "S = " << S << "\n";

}

return 0;

1. Налаштовано консоль на UTF-8 для коректного відображення українського тексту.

2. Програма приймає від користувача три дійсні числа: x, y, z.

3. Викликається функція calculateS(x, y, z) з модуля ModulesChervonets.h для обчислення виразу.

4. На основі перевірки sin(x), виводиться або результат, або повідомлення про помилку (NaN).

Лістинг програми Chervonets\_task

include "ModulesChervonets.h"

#include <windows.h>

#include <iostream>

#include <cmath> // sin, cos, fabs, sqrt, pow, isnan

#include <limits> // numeric\_limits

#include <iomanip> // fixed, setprecision, hex, dec, nouppercase

#include <cstdint> // uint64\_t

namespace {

constexpr double pi = 3.14159265358979323846;

}

void developer\_information() {

std::cout << "Артем Червонець ©, Усі права захищені." << std::endl;

}

double logical(int a, int b) {

bool res = (a + 1 <= b);

std::cout << "Результат логічного виразу: " << (res ? '1' : '0') << std::endl;

return res ? 1.0 : 0.0;

}

void numbers\_in\_systems(double x, double y, double z) {

// Для бітового представлення double використовуємо union

auto print\_pair = [&](double v, const char\* name) {

union {

double d;

uint64\_t u;

} cv{v};

std::cout << name << " = "

<< std::fixed << std::setprecision(2) << v

<< " (десяткова) = 0x"

<< std::hex << std::uppercase << cv.u

<< std::dec << std::nouppercase

<< std::endl;

};

std::cout << "\nЗначення змінних в десятковій та шістнадцятковій системах числення:\n";

print\_pair(x, "x");

print\_pair(y, "y");

print\_pair(z, "z");

double s = calculateS(x, y, z);

std::cout << "\nРезультат обчислення функції calculateS: "

<< std::fixed << std::setprecision(6)

<< s << std::endl;

}

int main() {

SetConsoleOutputCP(65001);

SetConsoleCP(65001);

developer\_information();

// Введення даних для функції logical

int a, b;

std::cout << "Введіть два цілих числа для логічної функції (a b): ";

std::cin >> a >> b;

logical(a, b);

// Введення даних для функції numbers\_in\_systems

double x, y, z;

std::cout << "Введіть три дійсних числа (x y z): ";

std::cin >> x >> y >> z;

numbers\_in\_systems(x, y, z);

return 0;

}

**Принцип роботи програми:**

1. **Налаштування середовища:**  
   Програма встановлює кодування UTF-8 для коректного виведення українських символів у консолі.
2. **Виведення інформації про розробника:**  
   Викликається функція developer\_information(), яка виводить ім’я автора.
3. **Логічна перевірка:**  
   Користувач вводить два цілих числа a і b.  
   Функція logical(a, b) перевіряє умову (a + 1 <= b) і виводить результат:  
   1 — якщо умова істинна, 0 — якщо хибна.
4. **Робота з числами:**  
   Користувач вводить три дійсних числа x, y, z.  
   Функція numbers\_in\_systems(x, y, z):
   * Виводить кожне з чисел у десятковій та шістнадцятковій системах (у вигляді бітового представлення).
   * Обчислює значення виразу через calculateS(x, y, z) та виводить результат.
5. **Обробка виняткових ситуацій:**  
   Функція calculateS перевіряє, чи sin(x) не дорівнює нулю, щоб уникнути ділення на нуль. У разі помилки виводиться NaN.

50 Аргументів

1. Програма використовує модульний підхід.

2. Всі функції винесені в окремий cpp-файл.

3. Для зручності створено заголовковий файл.

4. Функція calculateS обчислює складний математичний вираз.

5. В обчисленні використовується модуль.

6. Обчислення включає піднесення до степеня.

7. У виразі використовується квадратний корінь.

8. Застосовується косинус числа.

9. Використовується синус числа.

10. Значення π задане як константа.

11. Перед діленням перевіряється sin(x).

12. При діленні на нуль повертається NaN.

13. Значення NaN перевіряється через std::isnan.

14. Програма виводить повідомлення при помилці обчислення.

15. Всі математичні функції взяті з бібліотеки cmath.

16. Функція logical перевіряє умову a + 1 <= b.

17. Результат логічної функції виводиться як 1 або 0.

18. Повертається числовий результат логічного виразу.

19. Ввід a і b здійснюється з клавіатури.

20. Усі введення супроводжуються підказками.

21. Функція numbers\_in\_systems виводить x, y, z.

22. Вивід значень здійснюється у десятковій системі.

23. Також значення виводяться у шістнадцятковій системі.

24. Для представлення double у hex використовується union.

25. Використовується тип uint64\_t для представлення бітів.

26. Для виводу використовується std::fixed.

27. Встановлюється точність два знаки після коми.

28. Вивід шістнадцяткових значень здійснюється великими літерами.

29. Після hex вивід повертається до десяткового формату.

30. Користувач вводить x, y, z вручну.

31. У програмі виводиться ім’я автора.

32. Використовується функція developer\_information.

33. Авторські права виводяться українською.

34. Програма підтримує українську мову.

35. Установлено кодування UTF-8.

36. Підключено windows.h для роботи з консоллю.

37. Дані вводяться в консоль у зручному форматі.

38. Усі повідомлення зрозумілі та лаконічні.

39. Програма легко розширюється.

40. Кожна функція має чітке призначення.

41. Всі дані опрацьовуються локально.

42. Немає складної залежності між модулями.

43. Структура проєкту зручна для подальшої підтримки.

44. Програма компілюється без помилок.

45. Кожна функція має відповідну сигнатуру.

46. Всі заголовки підключені правильно.

47. Для математичних операцій підключено cmath.

48. Для перевірки NaN використано limits.

49. Стиль коду витриманий та охайний.

50. Програма відповідає вимогам лабораторної роботи.

Відповіді на контрольні запитання:

1. У відповідності до міжнародного стандарту ISO/IEC 12207 (або ISO/IEC/IEEE 12207:2016) мета і задачі процесів проектування програмного забезпечення полягають у:

Розробці детального проекту програмного забезпечення, що реалізує визначені вимоги.

Створенні чіткої архітектури програмного продукту з визначеними компонентами.

Визначенні інтерфейсів між компонентами програми та із зовнішніми системами.

Встановленні основи для верифікації та валідації програмного забезпечення.

Створенні основи для подальшої реалізації, тестування та підтримки програмного продукту.

Оптимізації проектних рішень з урахуванням технічних та економічних обмежень.

2. Функція мови програмування C/C++ відрізняється від модуля тим, що:

Функція є базовою одиницею програмного коду, яка виконує певну операцію або набір операцій і може повертати значення, тоді як модуль є більшою структурною одиницею, що об'єднує логічно пов'язані функції, класи, дані та інші елементи програм.

Функція має чітко визначений інтерфейс з параметрами та типом повернення, а модуль забезпечує інкапсуляцію та контроль доступу до групи взаємопов'язаних функцій та даних.

Функція зазвичай виконує одну конкретну задачу, тоді як модуль реалізує більш високорівневу функціональність через набір взаємодіючих функцій.

У C/C++ функція має локальну область видимості для своїх змінних, а модуль може створювати власний простір імен для уникнення конфліктів імен.

Модуль сприяє повторному використанню коду на більш високому рівні абстракції, ніж окрема функці.