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”

ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРОБЛЕННЯ МАСИВІВ ДАНИХ ТА СИМВОЛЬНОЇ ІНФОРМАЦІЇ

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 КІ - 24

Федотов А. О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А.С

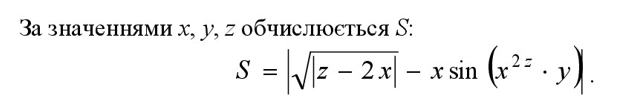
Кропивницький – 2025

**ТЕМА:** Реалізація статичних бібліотек модулів лінійних обчислювальних процесів

**МЕТА:** 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 №8.1**

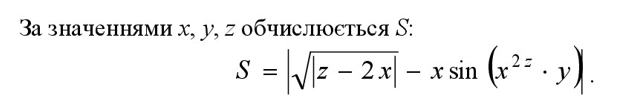
Розробити власну бібліотеку ModulesFedotov та реалізувати функцію в цій бібліотеці s\_calculation.



Малюнок 1 – завдання 8.1

**ПОСТАНОВА ЗАДАЧІ**

Розробити функцію s\_calculation, яка обчислює значення виразу:



Малюнок 2 – завдання 8.1

де:

1.x, y, z — вхідні параметри, що є дійсними числами.

2.Функція повинна повернути абсолютне значення різниці між двома членами виразу:

3.Перший член: квадратний корінь з абсолютного значення (z−2x).

4.Другий член: добуток xxx на синус від виразу x2z \* y

**АНАЛІЗ ЗАДАЧІ**

1. **Вхідні дані**: Три дійсних числа x, y, z.
2. **Обмеження та можливі проблеми**:

2.1 Вираз z − 2x може бути від’ємним, але оскільки ми беремо модуль перед добуванням квадратного кореня, проблем із визначеністю цієї операції не буде.

2.2 Вираз x2z потребує уваги:

2.3Якщо x < 0x і 2z не є цілим числом, операція піднесення до степеня може бути некоректною (у стандартному визначенні дійсних чисел).

2.4 Якщо x = 0x, то при z > 0 результатом буде 0, а при z < 0 виникне помилка через ділення на нуль.

1. **Вихідні дані**:

3.1Дійсне число — значення S, яке є абсолютним значенням різниці між двома членами виразу.

1. **Очікуваний алгоритм роботи**:

1. Обчислення першого члена:

2. Обчислення другого члена:

3. Визначення абсолютної різниці між цими двома значеннями.

1. **Використані математичні функції**:

5.1 sqrt() — для обчислення квадратного кореня.

5.2 abs() — для знаходження модуля.

5.3 sin() — для обчислення синуса.

5.4 pow() — для піднесення до степеня.

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Вихідний файл ModulesFedotov.cpp

#include <cmath>

using namespace std;

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

double term1 = sqrt(abs(z - 2 \* x));

double term2 = x \* sin(pow(x, 2 \* z) \* y);

return abs(term1 - term2);

}

Заголовковий файл ModulesFedotov.h

#pragma once

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

**ЗАВДАННЯ №8.2**

За послідовними запитами вводяться числа х, у, г та символи а і b.

Вивести (включити у потік STL cout)\*:

**ЗАДАЧА №8.2.1**

Прізвище та ім'я розробника програми зі знаком охорони авторського права «©» (від англ. copyright);

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Вихідний файл OutputName.cpp

#include <iostream>

#include <windows.h>

using namespace std;

void Information() {

SetConsoleOutputCP(1251);

cout << "\n ----------------------------------------\n"

<< "| Andrey Fedotov, CUNTU, Kropyvnickyi |\n"

<< "| Андрій Федотов, ЦНТУ, Кропивницький |\n"

<< "\n -------- (c) All Rights Reserved --------\n\n";

}

Заголовковий файл OutputName.h

#pragma once

void Information();

**ЗАДАЧІ №8.2.2 та №8.2.3**

Результат логічного виразу в текстовому вигляді (false/true): a+1=b+2?

Значения х, у, г в десятковій і шістнадцятковій системах чис-лення, Ѕ, що обчислюється функцією s\_calculation() заголовкового файлу Modules Прізвище.һ.

**ПОСТАНОВА ЗАДАЧІ**

Розробити програму, яка виконує три основні підзадачі:

**1.Перевірка логічного виразу:** a + 1 = b + 2.

**2.Обчислення математичного виразу:** за допомогою функції s\_calculation(x, y, z).

**3.Виведення значень** змінних x, y, z у десятковій та шістнадцятковій системах числення.

**АНАЛІЗ ЗАДАЧІ**

**1. Перевірка логічного виразу** a+1=b+2

1.1.**Вхіднідані**: два дійсні числа a та b.

1.2.**Опис виразу**: Вираз a+1=b+2 є логічним порівнянням, яке може приймати значення true або false.

1.3. **Результат роботи програми**:Програма повинна вивести результат порівняння.

**2. Обчислення виразу S за функцією** s\_calculation(x, y, z)

2.1. **Формула обчислення**:



Малюнок 3 – завдання 8.2.3

**Вхідні дані**: три дійсні числа x, y, z

**3. Виведення чисел у різних системах числення**

3.1. **Вхідні дані**: x, y, z.

3.2. **Типи виводу**: Виведення значень у десятковій та шістнадцятковій системах.

3.3. **Спосіб реалізації**: Десяткова система: стандартний cout. Шістнадцяткова система: використання hex для конвертації чисел.

3.4. **Перетворення дійсних чисел у цілі**: (int)x, (int)y, (int)z перед передачею в hex, щоб уникнути неправильного форматування.

**ПРОГРАМНА РЕАЛІЗАЦІЯ**

#include <iostream>

#include <cmath>

#include <windows.h>

#include "ModulesFedotov.h"

#include "OutputName.h"

using namespace std;

//Функція для підзадачі 8.2.2 (Результат логічного виразу a + 1 = b + 2)

void LogicaLexpression(double a, double b) {

bool result = (a + 1 == b + 2);

if ((a + 1 == b + 2) == true) {

cout << "Значення " << a << " + 1 = " << b << " + 2" << endl;

cout << "True" << endl;

}

else {

cout << "Значення " << a << " + 1 = " << b << " + 2" << endl;

cout << "False" << endl;

}

}

// Основна функція

int main() {

SetConsoleOutputCP(1251);

Information();

double x, y, z;

double a, b;

//Введення значень для задачі

cout << "Введіть x, y, z числа: ";

cin >> x >> y >> z;

cout << "Введіть числа a і b щоб перевірити логічний вираз (a + 1 = b + 2): ";

cin >> a >> b;

//Підзадача 8.2.2

LogicaLexpression(a, b);

//Підзадача 8.2.3

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

cout << "Результат функції s\_calculation: " << result << endl;

// Виведення значень x, y, z у десятковій та шістнадцятковій системах

cout << "x: " << dec << x << " (десяткова), 0x" << hex << (int)x << " (шістнадцяткова)" << endl;

cout << "y: " << dec << y << " (десяткова), 0x" << hex << (int)y << " (шістнадцяткова)" << endl;

cout << "z: " << dec << z << " (десяткова), 0x" << hex << (int)z << " (шістнадцяткова)" << endl;

system("pause");

return 0;

}

**РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ:**

----------------------------------------

| Andrey Fedotov, CUNTU, Kropyvnickyi |

| Андрій Федотов, ЦНТУ, Кропивницький |

-------- (c) All Rights Reserved --------

Введіть x, y, z числа: 1 2 3

Введіть числа a і b щоб перевірити логічний вираз (a + 1 = b + 2): 2 3

Значення 2 + 1 = 3 + 2

False

Результат функції s\_calculation: 0.0907026

x: 1 (десяткова), 0x1 (шістнадцяткова)

y: 2 (десяткова), 0x2 (шістнадцяткова)

z: 3 (десяткова), 0x3 (шістнадцяткова)

**ВИСНОВОКИ**

1. Методологія модульного програмування дозволяє розбивати код на логічні блоки.
2. Функціональна декомпозиція спрощує розв’язання складних задач.
3. Модульне тестування допомагає виявляти помилки на ранніх етапах.
4. Використання функцій підвищує повторне використання коду.
5. Скалярні типи даних є основою для обробки числової та логічної інформації.
6. Арифметичні операції дозволяють виконувати математичні обчислення.
7. Логічні операції є ключовими для реалізації умовних конструкцій.
8. Потокове введення спрощує отримання даних від користувача.
9. Потокове виведення використовується для представлення результатів.
10. Використання cin і cout робить введення та виведення зручним.
11. Code::Blocks забезпечує зручне середовище для розробки на C++.
12. GNU GCC Compiler є потужним компілятором для C++.
13. Модульний підхід полегшує налагодження коду.
14. Функціональна декомпозиція робить код більш читабельним.
15. Розробка програмних модулів сприяє структурованості проєкту.
16. Кросплатформеність дозволяє запускати код на різних ОС.
17. Розбиття задачі на підзадачі підвищує ефективність програмування.
18. Використання заголовкових файлів допомагає в організації коду.
19. Макроси можуть спростити виконання повторюваних операцій.
20. Використання #define дозволяє створювати символічні константи.
21. Вбудовані функції прискорюють виконання простих обчислень.
22. Динамічне виділення пам’яті дозволяє ефективно працювати з даними.
23. Вказівники є важливими для роботи з пам’яттю.
24. Використання new і delete допомагає керувати пам’яттю.
25. Робота з масивами потребує обережного використання індексів.
26. Використання vector спрощує управління динамічними масивами.
27. Структури (struct) дозволяють групувати дані.
28. Використання класів сприяє об’єктно-орієнтованому програмуванню.
29. Наслідування дозволяє повторно використовувати код.
30. Поліморфізм підвищує гнучкість програм.
31. Оператор if використовується для умовного виконання коду.
32. Цикли (for, while) дозволяють реалізовувати повторювані операції.
33. Використання switch спрощує перевірку кількох значень.
34. Робота з файлами дозволяє зберігати дані між запусками програми.
35. ofstream використовується для запису у файл.
36. ifstream використовується для зчитування з файлу.
37. Використання sstream допомагає працювати з текстовими потоками.
38. string є зручнішим за char[] для роботи з рядками.
39. Функція stoi() допомагає конвертувати рядки в числа.
40. Обробка винятків підвищує надійність коду.
41. Використання try-catch дозволяє керувати помилками.
42. Правильне форматування коду робить його читабельним.
43. Документування коду допомагає іншим розробникам.
44. Оптимізація алгоритмів підвищує продуктивність програми.
45. Використання STL (vector, map, set) спрощує роботу з даними.
46. Шаблони функцій дозволяють працювати з різними типами даних.
47. Рекурсія може бути зручною, але потребує контролю глибини викликів.
48. Лямбда-вирази допомагають створювати компактний код.
49. Використання auto спрощує роботу з типами змінних.
50. Якісне тестування коду знижує кількість помилок у програмі.

**ВІДПОВІДЬ НА КОНТРОЛЬНІ ЗАПИТАННЯ**

1. **У чому полягають мета й задачі процесів проектування ПЗ відповідно до міжнародного стандарту ISO/IEC 12207 (або ISO/IEC/IEEE 15288:2016)?**  
   ISO/IEC 12207 визначає процеси життєвого циклу ПЗ, спрямовані на якісну розробку, тестування та супровід.
2. **Обґрунтовано поясніть, чим функція мови програмування С/С++ відрізняється від модуля.**  
   Функція виконує конкретну задачу, а модуль є набором функцій та інших елементів, що забезпечують розширену функціональність.
3. **Яка відмінність функції main від решти функцій С/С++?**  
   main() є точкою входу в програму, а решта функцій викликаються у її межах.
4. **Яке призначення маніпуляторів і яким чином вони використовуються під час реалізації ПЗ мовою програмування С++?**  
   Маніпулятори (endl, hex, setw) змінюють формат виводу даних у cout.
5. **Як і для чого використовується заголовковий файл в процесі препроцесинга програми?**  
   Заголовковий файл містить оголошення функцій та макросів, що компілюються перед основним кодом.
6. **Що під час виконання лабораторної роботи Вами використано зі стандартного заголовкового файлу iostream та задля реалізації яких функцій?**  
   Використано cin, cout з iostream для введення та виведення даних.
7. **Що розуміють під стандартним простором імен у С++ і якою директивою він визначається?**  
   Простір імен std визначається директивою using namespace std;.
8. **Наведіть приклади одночасного оголошення й ініціалізації початковим значенням змінної (об’єкта).**  
   int x = 5;, double y(3.14);, std::string str = "Hello";.
9. **Перелічіть символи, використання яких у ідентифікаторах не допускаються відповідно до синтаксису мови С/С++.**  
   Заборонено @, #, !, пробіли, цифри на початку ідентифікатора.
10. **Яким чином у С++ можливо вивести десятковий літерал у шістнадцятковій системі числення?**  
    Використовується маніпулятор hex: cout << hex << 255; // 0xff.
11. **Що необхідно виконати для підключення й використання функцій нестандартної (наприклад, власної) бібліотеки?**  
    Підключити #include "myLib.h" і зкомпілювати .cpp файл бібліотеки.
12. **Чим відрізняється заголовковий файл від об’єктного?**  
    Заголовковий файл містить оголошення, а об’єктний – скомпільований код.
13. **Яким чином мовою програмування С/С++ можна реалізувати консольне виведення нестандартних символів (наприклад, ©)?**  
    Використовуються кодування символів (setlocale, wcout).
14. **В чому полягає відмінність між записом символьного і рядкового константного літерала в С/С++?**  
    Символьний літерал 'a', рядковий – "abc" (масив символів).
15. **Яким є синтаксис запису прототипа функції у С/С++? Де він записується у програмі та яке його призначення?**  
    int func(int); записується перед main(), визначає сигнатуру функції.
16. **Який оператор С/С++ призначений для повернення функцією значення - результату і який синтаксис його запису?**  
    Оператор return, наприклад: return x + y;.
17. **Перелічіть ключові символи (ESC-послідовності) мови С, їх призначення, синтаксис запису та спосіб використання у С++.**  
    \n – новий рядок, \t – табуляція, \\ – зворотний слеш.
18. **Перелічіть відомі Вам функції заголовкового файла cmath та їх призначення? В чому полягає їх відмінність від функцій бібліотеки math.h?**  
    sqrt(), pow(), sin(), cos() – функції cmath, що підтримують C++.
19. **Яке призначення тестових драйверів?**  
    Тестові драйвери перевіряють функції без повної програми.
20. **Яким чином здійснюється тестування модулів (функцій бібліотек) і чим означений процес відрізняється від тестування програмного засобу (файла ехе)?**  
    Тестування модулів перевіряє окремі компоненти, тестування .exe – всю програму.