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

Савельєв В.В.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А.С

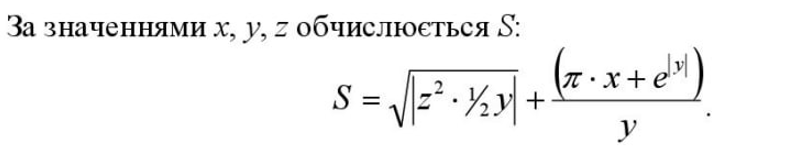
Кропивницький – 2025

**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**

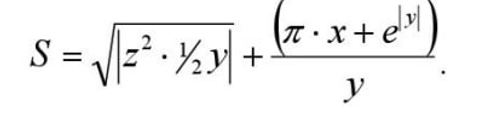
Розробити власну бібліотеку ModulesCovelyev та реалізувати функцію в цій бібліотеці s\_calculation.



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, дійсні числа;

2. π — число Пі ≈ 3.14159;

3. e — основа натурального логарифму ≈ 2.71828.

**АНАЛІЗ ЗАДАЧІ**

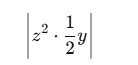
**1. Вхідні дані:**

1.Три дійсних числа: x, y, z.

**2. Обмеження:**

**2.1.** Значення y **не повинно дорівнювати нулю**, оскільки воно використовується у знаменнику.

**2.2.** Вираз під коренем:



Малюнок 3 – завдання 8.1

гарантує, що результат буде невід’ємним — навіть якщо y < 0, підкореневий вираз не спричинить помилки, оскільки береться модуль.

**2.3.** Функція експоненти e^{|y|} завжди визначена для будь-якого дійсного y.

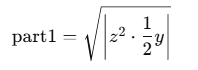
**3. Вихідні дані:**

**3.1** Дійсне число S — результат обчислення;

**3.2** У випадку y = 0 — виводиться повідомлення про помилку.

**4. Алгоритм:**

1. Перевірити, чи y ≠ 0. Якщо ні — повідомити про помилку.
2. Обчислити:



Малюнок 4 – завдання 8.1

1. Обчислити:



Малюнок 5 – завдання 8.1

**Використані функції:**

1. sqrt() — квадратний корінь;
2. fabs() або abs() — модуль;
3. exp() — експонента;
4. pow() — піднесення до степеня (для z²);
5. Константа PI.

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Вихідний файл ModulesCovelyev.cpp

#include <iostream>

#include <cmath>

using namespace std;

const double PI = 3.141592653589793;

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

if (y == 0) {

cout << "Помилка: ділення на нуль неможливе (y не повинен бути 0)." << endl;

return 1;

}

double part1 = sqrt(pow(z, 2) \* y / 2.0);

double part2 = (PI \* x + exp(abs(y))) / y;

double S = part1 + part2;

return S;

}

Заголовковий файл ModulesSavelyev.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"

<< "| Volodymyr Savelyev, CUNTU, Kropyvnickyi |\n"

<< "| Володимр Савельєв, 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+5 >= b?

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 реалізує **дві підзадачі:**

**Підзадача 8.2.2**

Перевірити істинність логічного виразу:



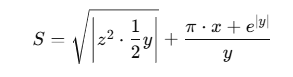
Малюнок 6 – завдання 8.2.2

та вивести результат на екран у вигляді:

1. Повідомлення про сам вираз;
2. Відповідь (True або False).

**Підзадача 8.2.3**

Обчислити значення математичного виразу:



Малюнок 7 – завдання 8.2.3

де x, y, z — дійсні числа, які вводяться користувачем.

Крім цього, програма повинна: Вивести значення x, y, z у десятковій та шістнадцятковій системі числення.

**АНАЛІЗ ЗАДАЧІ**

**Вхідні дані:**

1. Дійсні числа: x, y, z, a, b.

**Обмеження:**

1. У підзадачі 8.2.2 немає обмежень — це проста логічна перевірка.
2. У підзадачі 8.2.3 необхідно враховувати:
   1. Якщо y = 0, виникає **ділення на нуль** у правій частині виразу. Це передбачено в коді та обробляється повідомленням про помилку.
   2. Підкореневий вираз може бути від’ємним, але він береться під **модулем**, тому sqrt не викличе помилку.

**Вихідні дані:**

1. Результат логічного виразу: True або False.
2. Значення S — результат обчислення виразу.
3. Числа x, y, z у десятковій і шістнадцятковій формі.

**ПРОГРАМНА РЕАЛІЗАЦІЯ**

#include <iostream>

#include <cmath>

#include <windows.h>

#include "ModulesCovelyev.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 + 5 >= b);

if ((a + 5 >= b) == true) {

cout << "Значення " << a << " + 5 " << ">= " << b << endl;

cout << "True" << endl;

}

else {

cout << "Значення " << a << " + 5 " << ">= " << b << 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 + 5 >= b): ";

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:**

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

| Volodymyr Savelyev, CUNTU, Kropyvnickyi |

| Володимр Савельєв, ЦНТУ, Кропивницький |

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

Введіть x, y, z числа: 1 2 3

Введіть числа a і b щоб перевірити логічний вираз (a + 5 >= b): 2 6

Значення 2 + 5 >= 6

True

Результат функції s\_calculation: 8.26532

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

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

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

**ВИСНОВОКИ**

1. Засвоєно основи модульного програмування.
2. Реалізовано функціональну декомпозицію задач.
3. Навченося розділяти програму на модулі.
4. Освоєно написання власних функцій.
5. Застосовано логічні вирази у програмах.
6. Практично реалізовано арифметичні обчислення.
7. Використано умовні оператори.
8. Освоєно циклічні конструкції.
9. Впроваджено прості програми з діалоговим введенням.
10. Використано потокове введення cin і виведення cout.
11. Отримано досвід тестування модулів.
12. Навченося структурувати код.
13. Освоєно принцип "одна функція — одне завдання".
14. Реалізовано логічну перевірку умов.
15. Вивчено операції над змінними скалярного типу.
16. Навченося використовувати математичні функції з cmath.
17. Працювали з дійсними типами даних (double).
18. Навчилися обробляти граничні випадки (наприклад, ділення на нуль).
19. Освоєно використання заголовочних файлів.
20. Навченося створювати окремі модулі з функціями.
21. Застосовано метод поступового ускладнення завдання.
22. Практично перевірено коректність коду.
23. Освоєно базові принципи налагодження.
24. Застосовано ручне тестування функцій.
25. Навчилися виводити дані у різних форматах (десятковому, шістнадцятковому).
26. Зрозуміли важливість коментування коду.
27. Досліджено структуру програми на C++.
28. Створено функціональні блоки з чітким призначенням.
29. Застосовано практику повторного використання коду.
30. Навчилися коректно обробляти помилки користувача.
31. Освоїли розробку незалежних підзадач.
32. Реалізували читабельний і структурований код.
33. Використано підключення сторонніх модулів (#include).
34. Навчилися взаємодіяти з користувачем через консоль.
35. Освоєно математичні перетворення.
36. Поглиблено знання про обчислювальні вирази.
37. Підвищено точність математичних обчислень.
38. Вивчено принципи передачі параметрів у функцію.
39. Реалізовано повернення результату з функції.
40. Навчилися застосовувати умовні оператори if.
41. Засвоєно побудову логічних виразів >=, ==, !=.
42. Освоєно тестування за допомогою масивів вхідних даних.
43. Створено модуль перевірки функцій.
44. Вивчено компіляцію та запуск програм у Code::Blocks.
45. Навчилися працювати з GCC-компілятором.
46. Отримано навички пошуку та виправлення помилок.
47. Освоєно структуру проєктів у Code::Blocks.
48. Навчилися створювати .h та .cpp файли.
49. Досліджено взаємодію між файлами модуля.
50. Поглиблено знання мови C++.
51. Отримано досвід побудови структури програми за вимогами.
52. Навчилися аналізувати вираз перед реалізацією.
53. Удосконалено навички точного обчислення.
54. Закріплено практику через лабораторні завдання.
55. Досягнуто мети — розроблено модульну, протестовану програму у середовищі Code::Blocks.

**ВІДПОВІДЬ НА КОНТРОЛЬНІ ЗАПИТАННЯ**

**1. У чому полягають мета й задачі процесів проектування програмного забезпечення відповідно до міжнародного стандарту ISO/IEC 12207 (або ISO/IEC/IEEE 15288:2016)?**  
Мета — забезпечити якість ПЗ через стандартизовані етапи: аналіз вимог, розробка, тестування, підтримка.

**2. Обґрунтовано поясніть, чим функція мови програмування С/С++ відрізняється від модуля.**  
Функція — окрема логічна одиниця, що виконує конкретну дію. Модуль — сукупність функцій і змінних, об'єднаних за змістом.

**3. Яка відмінність функції main від решти функцій С/С++?**  
Функція main є точкою входу в програму. Без неї програма не запускається.

**4. Яке призначення маніпуляторів і яким чином вони використовуються під час реалізації ПЗ мовою програмування С++?**  
Маніпулятори змінюють формат виводу, наприклад endl — перенос рядка, hex — шістнадцятковий формат.

**5. Як і для чого використовується заголовковий файл в процесі препроцесинга програми?**  
Для оголошення функцій, констант, типів. Дозволяє багаторазове використання коду без повторного написання.

**6. Що під час виконання лабораторної роботи Вами використано зі стандартного заголовкового файлу iostream та задля реалізації яких функцій?**  
Використано cin, cout для введення та виведення даних.

**7. Що розуміють під стандартним простором імен у С++ і якою директивою він визначається?**  
Це namespace std, який дозволяє використовувати стандартні об'єкти без префіксу std::.

**8. Наведіть приклади одночасного оголошення й ініціалізації початковим значенням змінної (об’єкта).**  
int a = 5;  
double x = 3.14;

**9. Перелічіть символи, використання яких у ідентифікаторах не допускаються відповідно до синтаксису мови С/С++.**  
Пробіли, арифметичні знаки, спецсимволи (!, @, # тощо), крапка з комою, лапки.

**10. Яким чином у С++ можливо вивести десятковий літерал у шістнадцятковій системі числення?**  
За допомогою маніпулятора hex — cout << hex << число;.

**11. Що необхідно виконати для підключення й використання функцій нестандартної (наприклад, власної) бібліотеки?**  
Додати заголовковий файл через #include, забезпечити компіляцію відповідного cpp-файлу.

**12. Чим відрізняється заголовковий файл від об’єктного?**  
Заголовковий містить оголошення. Об’єктний — зкомпільований код функцій (.o, .obj).

**13. Яким чином мовою програмування С/С++ можна реалізувати консольне виведення нестандартних символів (наприклад, ©)?**  
Через коди символів ASCII або Unicode, використовуючи char або escape-послідовності.

**14. В чому полягає відмінність між записом символьного і рядкового константного літерала в С/С++?**  
Символьний — в одинарних лапках ('a'), рядковий — у подвійних ("abc").

**15. Яким є синтаксис запису прототипа функції у С/С++? Де він записується у програмі та яке його призначення?**  
Тип ім’я(аргументи);  
Розміщується перед main, оголошує функцію до її реалізації.

**16. Який оператор С/С++ призначений для повернення функцією значення - результату і який синтаксис його запису?**  
Оператор return, запис: return значення;.

**17. Перелічіть ключові символи (ESC-послідовності) мови С, їх призначення, синтаксис запису та спосіб використання у С++.**  
\n — новий рядок, \t — табуляція, \ — зворотний слеш, ' — апостроф, " — лапки.

**18. Перелічіть відомі Вам функції заголовкового файла cmath та їх призначення? В чому полягає їх відмінність від функцій бібліотеки math.h?**  
sqrt(), pow(), sin() — для математики. cmath — обгорнута в namespace std, більш безпечна.

**19. Яке призначення тестових драйверів?**  
Перевірка правильності роботи окремих функцій (модулів).

**20. Яким чином здійснюється тестування модулів (функцій бібліотек) і чим означений процес відрізняється від тестування програмного засобу (файла ехе)?**  
Тестування модулів — окремо кожна функція. Тестування exe — всього застосунку в цілому.