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

Кондратенко Дмитро

ПЕРЕВІРИВ

викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А. С.

Кропивницький – 2025

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 №1**

**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

**Аналіз вимог та проектування архітектури для задачі 8.1**

1. **Аналіз вимог:**

Обчислення виразу для змінних x,y,z.

Виведення результатів у потік cout.

Реалізація функції s\_calculation(), яка буде обчислювати значення S.

Робота з десятковими та шістнадцятковими представленнями чисел.

Проектування архітектури:

Виділити функцію s\_calculation() у заголовковому файлі Modules/Прізвище.h.

Окремі функції для:

Виведення інформації про автора.

Логічного порівняння символів.

Форматованого виведення значень у різних системах числення.

Детальне проектування:

Написати реалізацію кожної функції у відповідних .cpp файлах.

Забезпечити коректне введення та обробку даних.

Лістинг файлу main.cpp проєкту ModulesKondratenko:

#include <iostream>

#include <cmath>

using namespace std;

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

return fabs(abs(sin(y - pow(z, 2 ))) + sqrt(x) - sqrt(pow((y \* z), x) + y / (2 \* M\_PI)));

}

Лістинг файлу ModulesKondratenko.h:

#ifndef MODULESKONDRATENKO\_H\_INCLUDED

#define MODULESKONDRATENKO\_H\_INCLUDED

#include <cmath>

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

#endif // MODULESKONDRATENKO\_H\_INCLUDED

Лістинг файлу main.cpp проєкту TestDrive:

#include <iostream>

#include <clocale>

#include "../ModulesKondratenko.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "ukr");

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

cout << " Dmytro Kondratenko, yeq29880@gmail.com " << endl;

cout << " Дмитро Кондратенко, yeq29880@gmail.com " << endl;

cout << "==========(c)All Rights Reserved=========" << endl;

double x, y, z;

cout << "Введіть значення x, y, z: " << endl;

cin >> x >> y >> z;

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

cout << "Значення S = " << S << endl;

return 0;

}  
  
  
**Аналіз вимог та проектування архітектури для задачі 8.2**

1. **Аналіз вимог**

**1.1 Вхідні дані**

Вводяться числа x, y, z.

Вводяться символи a, b.

**1.2 Функціональні вимоги**

Виведення авторської інформації:

Функція має вивести прізвище та ім’я розробника разом із знаком охорони авторського права ©.

Перевірка логічного виразу:

Визначення та виведення результату логічного виразу a + 7 == b у вигляді false або true.

Обчислення значення S:

Використання функції s\_calculation() із заголовкового файлу Modules/Прізвище.h.

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

**1.3 Нефункціональні вимоги**

Реалізація всіх підзадач у вигляді окремих функцій.

Використання потоку виводу cout для виведення результатів.

Використання маніпуляторів iomanip для форматування виводу.

1. **Проектування архітектури**

**2.1** **Структура модулів**

main.cpp – головний модуль програми, який ініціалізує введення даних, викликає функції та виводить результати.

Modules/Прізвище.h – заголовковий файл, що містить оголошення функції s\_calculation().

Modules/Прізвище.cpp – реалізація функції s\_calculation().

**2.2 Опис основних функцій**

void SW\_Developer();

Виводить інформацію про автора із символом ©.

bool check\_expression(char a, char b);

Виконує перевірку a + 7 == b і повертає логічне значення.

void calculate (int x, int y, int z);

Виводить значення x, y, z у десятковій та шістнадцятковій формах.

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

Обчислює значення S відповідно до формули та повертає його.

**Аргументи досягнення мети лабораторної роботи:**

1. Розвиток навичок програмування на C++.
2. Закріплення знань з використання математичних функцій.
3. Ознайомлення з бібліотекою cmath для математичних обчислень.
4. Використання операторів cout і cin для введення та виведення даних.
5. Форматування вихідних даних за допомогою iomanip.
6. Робота з заголовковими файлами (.h).
7. Використання модульної структури програм.
8. Відпрацювання навичок роботи з функціями.
9. Закріплення вмінь перетворення десяткової системи в шістнадцяткову.
10. Вміння працювати з операціями округлення значень.
11. Використання бібліотеки cmath для обчислення тригонометричних функцій.
12. Відпрацювання роботи з абсолютними значеннями чисел.
13. Визначення та використання змінних у програмі.
14. Використання #include для підключення заголовкових файлів.
15. Робота з логічними операціями та перевірками умов.
16. Вивчення основ математичної логіки.
17. Використання математичних операторів (+, -, \*, /, %).
18. Опрацювання концепції інкапсуляції у функціях.
19. Навчання роботи з бібліотеками стандартної бібліотеки C++.
20. Використання namespace std для спрощення синтаксису.
21. Робота зі змінними різних типів (int, double, char).
22. Використання маніпуляторів setprecision і fixed для точності обчислень.
23. Відпрацювання концепції "чорного ящика" у функціях.
24. Структуроване програмування та розбиття коду на модулі.
25. Аналіз результатів логічних операцій.
26. Відпрацювання роботи з операторами порівняння (==, !=, <, >).
27. Використання return для повернення значення з функцій.
28. Аналіз поведінки арифметичних операцій над різними типами даних.
29. Вміння налагоджувати програму через cout (відлагоджувальні виводи).
30. Опрацювання константних значень у програмі.
31. Використання параметрів у функціях для передачі значень.
32. Розробка заголовкових файлів для багаторазового використання коду.
33. Вміння працювати з дробовими числами (double).
34. Використання модуляційного підходу до програмування.
35. Відпрацювання способів передачі змінних у функції.
36. Використання бібліотеки iomanip для управління форматуванням.
37. Вміння створювати коментарі (// та /\* ... \*/).
38. Практика в написанні зрозумілого та коментованого коду.
39. Ознайомлення з правильним стилем написання коду.
40. Закріплення принципів побудови простих математичних виразів.
41. Відпрацювання механізмів обчислення за формулою.
42. Використання механізму абсолютного значення (fabs()).
43. Вивчення правильного оголошення та виклику функцій.
44. Використання операторів &&, ||, ! у логічних виразах.
45. Вивчення основ структурного програмування.
46. Опрацювання алгоритму роботи функцій з різними типами даних.
47. Закріплення знань про оператори введення/виведення.
48. Використання операторів sizeof для аналізу пам’яті.
49. Практика у правильному виборі типів змінних.
50. Навчання написання ефективного та продуктивного коду

**Відповіді до контрольних запитань:**

1. **Мета й задачі процесів проектування ПЗ відповідно до ISO/IEC 12207 або ISO/IEC/IEEE 15288:2016**

**Мета** – встановити стандартизовані процеси для життєвого циклу програмного забезпечення, забезпечити якість і ефективність розробки.

**Задачі**:

Визначення вимог до ПЗ

Проектування архітектури

Реалізація, тестування та інтеграція

Супровід і модернізація

1. **Чим функція в С/С++ відрізняється від модуля?**

**Функція** – це блок коду, що виконує певне завдання і може повертати значення.

**Модуль** – це окремий файл або набір функцій, що містять певну логіку (наприклад, заголовковий файл .h разом з .cpp файлом реалізації).

1. **Відмінність функції main від інших функцій у С/С++**

Це точка входу у програму.

Завжди має стандартний формат: int main() або int main(int argc, char\* argv[]).

Може повертати код завершення return 0;.

1. **Призначення маніпуляторів у C++ і їх використання**

**Маніпулятори** (setw, fixed, setprecision, hex) змінюють формат виводу в cout.

1. **Роль заголовкового файлу під час препроцесингу**

Містить оголошення функцій та змінних, які можуть бути використані в інших файлах.

Підключається через #include "filename.h" або #include <iostream>.

1. **Що було використано зі стандартного заголовкового файлу <iostream>?**

Основні елементи: std::cout, std::cin, std::endl.

Використовуються для введення та виведення даних у програмі.

**Що таке стандартний простір імен у C++ і як він визначається?**

* std – це стандартний простір імен, що містить стандартні функції C++ (cout, cin).
* Визначається директивою:

using namespace std;

**Висновок:** Виконуючи цю лабораторну роботу, я закріпив свої знання з роботи з функціями, математичними операціями та заголовковими файлами у C++. Було цікаво розібратися, як передавати параметри у функції, працювати з форматуванням виводу та конвертувати числа в різні системи числення.

Також я зрозумів, наскільки важливо правильно структурувати код і використовувати заголовкові файли для зручності. Загалом, ця робота допомогла мені покращити практичні навички програмування та навчитися ефективніше писати код.



Додаток 1 – тест-сьют задачі 8.1



Додаток 2 – тест-сьют задачі 8.2