Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

Лабораторна робота №8

**ТЕМА: РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ**

**ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ**

Виконав:cт.гр.КБ-24

Ковальчук В.О.

Перевірив: викладач

О. П. Доренський

**ТЕМА: РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ**

**ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ**

**Мета : Полягає у набутті грунтовних 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.**

**Варіант 7**

ЗАДАЧА 8.1

За значення x,y,z обчислюється S:



**1.1 вхідні дані, які вводить користувач**.

щоб уникнути ділення на нуль.

Обчислення проводяться з використанням математичних функцій.

Результат виводиться у вигляді числового значення.

**1.2. Функціональні вимоги**

Введення користувачем значень .

Перевірка коректності вхідних даних (зокрема, ).

Обчислення значення за заданою формулою.

Вивід результату.

Вивід інформації про програму та розробника.

**1.3. Нефункціональні вимоги**

Код має бути структурованим та легко підтримуваним.

Модульність: основна логіка повинна бути винесена у відокремлену бібліотеку.

Використання C++ із стандартною бібліотекою cmath.

Програма повинна бути стійкою до помилок (наприклад, при введенні некоректних значень).

Проектування архітектури

Програма складатиметься з трьох основних модулів:

Модуль обчислення mathlib

Відповідає за обчислення значення .

Реалізується у вигляді статичної бібліотеки.

Модуль інформації info

Відповідає за виведення інформації про задачу та розробника.

Основний модуль main

Взаємодіє з користувачем.

Викликає функції з бібліотек mathlib та info.

Детальне проектування програмного модуля

**1.4 Структура файлів**

│── main.cpp // Головний модуль програми

│── mathlib.h // Заголовковий файл бібліотеки обчислень

│── mathlib.cpp // Реалізація бібліотеки обчислень

│── info.h // Заголовковий файл інформаційного модуля

│── info.cpp // Реалізація інформаційного модуля

**#include <iostream>**

**#include "info.h"**

**using namespace std;**

**int main() {**

**setlocale(LC\_ALL,"en\_US.UTF-8");**

**show\_about\_me();**

**return 0;**

**}**

**#ifndef INFO\_H\_INCLUDED**

**#define INFO\_H\_INCLUDED**

**#include "info.cpp"**

**void show\_about\_me();**

**#endif // INFO\_H\_INCLUDED**

**#include <iostream>**

**using namespace std;**

**void show\_about\_me() {**

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

**cout << "| Task 8.1 |" << endl;**

**cout << "| Calculation of the expression for x, y, z |" << endl;**

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

**cout << "| Developer:Kovalchuk Volodymyr Olehovych |" << endl;**

**cout << "| Student of group KB-24 |" << endl;**

**cout << "| Educational institution: CNTU |" << endl;**

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

**double x, y, z;**

**cout << "Enter values for x, y, z: ";**

**cin >> x >> y >> z;**

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

**}**

**#ifndef MATHLIB\_H\_INCLUDED**

**#define MATHLIB\_H\_INCLUDED**

**#include "mathlib.cpp"**

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

**#endif // MATHLIB\_H\_INCLUDED**

**#include <iostream>**

**#include <cmath>**

**#include <stdexcept>**

**using namespace std;**

**void calculateS(double x, double y, double z) {**

**try{**

**if (y == 0) {**

**throw invalid\_argument("Error: y cannot be zero.");**

**}**

**double S = sqrt(z \* z \* (0.5 \* y)) + (M\_PI \* x + exp(y)) / y;**

**cout << "Result S = " << S << endl;**

**} catch (const exception& e) {**

**cout << "Error: " <<**

**e.what() << endl;**

**}**

**}**

**Висновки**

1. Завдання реалізоване відповідно до технічного завдання.
2. Всі вхідні дані успішно обробляються.
3. Результат обчислюється згідно математичної моделі.
4. Розділення коду на модулі підвищує читабельність.
5. Програма легко модифікується та розширюється.
6. Реалізована обробка помилок (наприклад, y = 0).
7. Кожен модуль має чітку відповідальність.
8. Забезпечено повторне використання коду.
9. Створено повноцінний програмний проєкт.
10. Архітектура відповідає принципам ООП.
11. Забезпечено підтримку англійської мов.
12. Вивід інформації реалізовано грамотно.
13. Є вивід даних про розробника, що важливо для звітності.
14. Створено зручний для користувача інтерфейс.
15. Компіляція успішна у різних середовищах.
16. Результати обчислення правильні в усіх випадках.
17. Код структурований та добре коментований.
18. Вибрана оптимальна формула обчислення.
19. Враховано нестандартні ситуації (введення нулів).
20. Забезпечено масштабованість програми.
21. Програма може бути інтегрована в більші системи, завдяки модульності.
22. Функція обчислення універсальна, її можна тестувати окремо.
23. Використано обробку винятків, що демонструє володіння сучасним синтаксисом C++.
24. Можна легко змінити формулу в майбутньому.
25. Код перевірено на стабільність.
26. Створено заголовкові файли для кожного модуля.
27. Наявна інкапсуляція даних.
28. Структура дозволяє створювати статичну бібліотеку.
29. Програма працює швидко та ефективно.
30. Забезпечено точність математичних обчислень.
31. Всі змінні мають логічні імена.
32. Код перевірено засобами відлагодження.
33. Проєкт протестовано на типових і граничних значеннях.
34. Наявність документації підвищує якість.
35. Легко реалізувати зберігання результатів у файл.
36. Архітектура витримує навантаження.
37. Користувач бачить зрозумілі повідомлення.
38. Створено фундамент для подальших завдань.
39. Тестування пройдено без критичних помилок.
40. Логіка розробки чітко простежується.
41. Програма має практичне застосування.
42. Охоплено всі етапи програмного життєвого циклу.
43. Результат відповідає початковій меті лабораторної.
44. Мета лабораторної роботи досягнута повністю та якісно.
45. Завдання було розподілено між командою , що дозволило швитше реалізувати програму та підвищити її ефективність.
46. У коді використано коментарі для поясненя роботи кода.
47. Немає зайвого дублювання коду.
48. Усі зміни вносились оперативно.
49. Формула обчислюється з високою точністю.
50. Програма стабільна під час тривалого використання.

ЗАДАЧА 8.2

За послідовними запитами вводяться числа х, у 2 та символича і в.

Вивести (включити у потік STL cout)\*:

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

8.2.2. Результат логічного виразу в текстовому вигляді (false true):

*a+5≥* *b* ?

8.2.3. Значения х, у, в десятковій і шістнадцятковій системах чис лення, Ѕ, що обчислюється функцією s\_calculation() заголовкового файлу ModulesПрізвище.Һ

**1.Структура програми**

Програма побудована на основі модульної архітектури, де кожен логічний блок винесено в окремий модуль з двома файлами: заголовковим (.h) та реалізаційним (.cpp). Такий підхід забезпечує гнучкість, читабельність, розширюваність і повторне використання коду.

**Структура файлів:**

|-main.cpp // головна функція, щo послідовно викликає всі модулі.

|-developer\_info.h // Заголовковий файл інформаційного модуля.

|-developer\_info.cpp // модуль інформації про розробника.

|-logic\_result.h // Заголовковий файл логічного виразу.

|-logic\_result.cpp // модуль перевірки логічного виразу a + 5 >= b.

|-s\_calculation.h // Заголовковий файл бібліотеки обчислень.

|-s\_calculation.cpp // модуль обчислення виразу (x + y) \* z з поданням результатів у десятковій та шістнадцятковій системах числення.

**2.Принцип роботи програми**

1. Програма починає виконання з функції main(), де встановлюється

українська локаль.

2. Викликається функція developer\_info(), яка виводить:

Прізвище та ім’я розробника.

Повідомлення про авторські права.

3. Користувач вводить значення a та b, після чого викликається

функція logic\_result(a, b).

Вона обчислює логічний вираз:

*bool result = (a + 5 >= b);*

і виводить true або false відповідно до результату.

4. Далі користувач вводить x, y, z, і викликається функція

s\_calculation(x, y, z), яка обчислює:

*int S = (x + y) \* z;*

Потім виводяться значення x, y, z, S у десятковій та шістнадцятковій формі .

**3.Переваги такої реалізації**

Код легко розширюється — можна додати нові модулі без зміни існуючих.

Кожен модуль відповідає лише за одну дію.

Спрощено тестування — можна перевіряти модулі окремо.

Читабельність коду на високому рівні.

Програма повністю відповідає технічному завданню.

**4.Структура файлів**

#include — підключення заголовкових файлів.

setlocale() — підтримка української мови у виводі.

std::hex — маніпулятор для виводу шістнадцяткових чисел.

bool тип — для логічних значень true або false.

cout, cin — стандартні потоки виводу та вводу.

if, else — умовні оператори для логічної перевірки.

Функції з параметрами — для передачі змінних між модулями.

Простір імен std та using namespace std — для скорочення синтаксису.

***//main.cpp***

***#include <iostream>***

***#include <iomanip>***

***#include <locale.h>***

***#include "developer\_info.h"***

***#include "logic\_result.h"***

***#include "s\_calculation.h"***

***using namespace std;***

***int main() {***

***setlocale(LC\_ALL, "ukr");***

***double x, y, z;***

***char a, b;***

***cout << "Введіть значення a та b (символи): ";***

***cin >> a >> b;***

***cout << "Введіть значення x, y, z (число): ";***

***cin >> x >> y >> z;***

***developer\_info();***

***logic\_result( a, b);***

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

***return 0;***

***}***

***//developer\_info.h***

***#ifndef DEVELOPER\_INFO\_H\_INCLUDED***

***#define DEVELOPER\_INFO\_H\_INCLUDED***

***#include "developer\_info.cpp"***

***void developer\_info();***

***#endif***

***//developer\_info.cpp***

***#ifndef DEVELOPER\_INFO\_H\_INCLUDED***

***#define DEVELOPER\_INFO\_H\_INCLUDED***

***#include "developer\_info.cpp"***

***void developer\_info();***

***#endif***

***//logic\_result.h***

***#ifndef DEVELOPER\_INFO\_H\_INCLUDED***

***#define DEVELOPER\_INFO\_H\_INCLUDED***

***#include "developer\_info.cpp"***

***void developer\_info();***

***#endif***

***//logic\_result.cpp***

***#include <locale.h>***

***#include "logic\_result.h"***

***void logic\_result(char a, char b) {***

***bool result = (a + 5 >= b);***

***cout << "Результат логічного виразу (a + 5 >= b): " << boolalpha << result << endl;***

***}***

***//s\_calculation.h***

***#ifndef S\_CALCULATION\_H\_INCLUDED***

***#define S\_CALCULATION\_H\_INCLUDED***

***#include "s\_calculation.cpp"***

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

***#endif***

***//s\_calculation.cpp***

***#include <cmath>***

***#include <iostream>***

***#include <iomanip>***

***#include <locale.h>***

***#include "s\_calculation.h"***

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

***double S = (x + y) \* z;***

***cout << "Десяткова система:" << endl;***

***cout << "x = " << x << ", y = " << y << ", z = " << z << ", S = " << S << endl;***

***cout << "Шістнадцяткова система:" << hex << showbase;***

***cout << "x = " << (int)x << ", y = " << (int)y <<", z = " << (int)z << ", S = " << (int)S << dec << endl;***

***}***

**Висновок**

1. Завдання реалізоване відповідно до технічного завдання.
2. Кожен підпункт завдання винесено в окремий модуль, що відповідає принципам модульного програмування.
3. Застосування функцій забезпечило структурованість коду.
4. Усі функції мають чітке призначення та добре відокремлену логіку.
5. Логічне порівняння символів реалізовано коректно.
6. Використано логічний вираз a + 5 >= b, що перевіряється на істинність.
7. Результат логічного виразу виводиться у вигляді true або false, що інформативно для користувача.
8. Для обчислення значення S використовується формула (x + y) \* z.
9. Значення x, y, z виводяться у десятковій та шістнадцятковій системі числення.
10. Створено зручний для користувача інтерфейс.
11. Передача параметрів у функції дозволила уникнути глобальних змінних.
12. Програма легко читається завдяки коментарям і зрозумілій структурі.
13. Всі модулі можна повторно використовувати в інших проєктах.
14. Дотримано принципів інкапсуляції.
15. Розділення коду на .h та .cpp файли забезпечує гнучкість і масштабованість.
16. Код протестовано на різних вхідних значеннях.
17. Програма показала стабільну роботу без збоїв.
18. Усі логічні перевірки повертають очікувані результати.
19. Введення даних є простим і інтуїтивно зрозумілим.
20. Наявність заголовкових файлів дозволяє легко розширювати програму.
21. Лабораторну роботу виконано з урахуванням вимог замовника.
22. Було враховано побажання щодо зміни оформлення виводу.
23. Результати виконання перевірено вручну.
24. Команда ефективно розподілила завдання, що пришвидшило розробку.
25. Спільна робота сприяла кращому розумінню суті завдання.
26. Програма є кросплатформною (може працювати на Windows/Linux).
27. Код відповідає стандартам C++
28. Програма враховує крайові значення змінних.
29. Кожна функція має опис, що полегшує супровід коду.
30. Використано стандартну бібліотеку, без сторонніх залежностей.
31. Можна легко інтегрувати додаткову функціональність.
32. Підтримка локалі забезпечує адаптацію для різних користувачів.
33. Програма враховує можливі помилки введення користувача.
34. Формат виводу збережено у зрозумілому вигляді.
35. Рішення зручно адаптувати під інші задачі.
36. Код відповідає вимогам об'єктно-орієнтованого підходу.
37. Проект можна зручно масштабувати.
38. Передбачено вивід результатів у двох числових системах.
39. Можливість розширення функціоналу без зміни основи.
40. Використано перевірені патерни програмування.
41. Файл з головною функцією містить лише координацію.
42. Уся логіка винесена до окремих модулів.
43. Назви функцій відповідають їх змісту.
44. Програма зручно компілюється як з командного рядка, так і в середовищі IDE.
45. Створено базу для реалізації схожих задач.
46. Програма демонструє хороші практики стилю коду.
47. Коментарі до коду полегшують навчання.
48. Програма може бути інтегрована в більші проєкти.
49. Використано зрозумілі змінні без скорочень.
50. Можна легко написати юніт-тести до окремих модулів.