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”

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки

та програмного забезпечення

Доренський О. П.

<https://github.com/odorenskyi/>

ВИКОНАВ

студент академічної групи КН-24

Куртяк К.В

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки

та програмного забезпечення

Коваленко Анастасія Сергіївна

Кропивницький – 2025

**Тема:** Реалізація статичних бібліотек модулів лшншйних 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:** 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.

**Варіант 25**

**Аналіз та постановка задачі 8.1**

1. Аналіз задачі

На основі заданих значень потрібно обчислити значення функції:

дійсні числа, які вводить користувач.

Операції, які потрібно виконати:

Віднімання:

Добування кореня:

Піднесення до квадрата:

Множення:

Обчислення синуса:

Остаточні математичні операції.

2. Обмеження і особливі випадки

1. Корінь визначений лише для .

Якщо , програма повинна повідомити про помилку.

2. Значення можуть бути будь-якими дійсними числами.

3. Вхідні дані можуть бути введені некоректно (наприклад, літери замість чисел), що також потребує перевірки.

3. Вхідні та вихідні дані

Вхідні дані:

Три дійсні числа , які вводить користувач.

Вихідні дані:

Обчислене значення .

Якщо значення від’ємне → вивести повідомлення про помилку.

Якщо введені некоректні значення → повідомлення про помилку.

4. Постановка задачі

Вхідні дані: (дійсні числа).

Результат: обчислене значення або повідомлення про помилку.

Умови коректності: .

Обробка помилок:

Некоректний ввід (букви, символи) → вивести "Помилка: введіть коректні числа."

Від’ємний аргумент кореня → вивести "Помилка: z має бути не менше ніж 2x."

**Аналіз вимог, проектування архітектури, детальне проектування програмного модуля розв’язування задачі 8.1**

1. Аналіз вимог

Функціональні вимоги:

Програма повинна:

1. Приймати на вхід три дійсні числа .

2. Перевіряти умову , щоб уникнути помилки добування квадратного кореня.

Нефункціональні вимоги:

Програма повинна бути простою у використанні.

Повинна мати ефективне обчислення з використанням математичних бібліотек.

Повинна коректно обробляти помилки введення.

2. Проектування архітектури

Архітектурний стиль:

Програма матиме модульний підхід із розділенням функцій на окремі блоки.

Компоненти системи:

1. Модуль введення даних:

Зчитує значення від користувача.

Перевіряє, чи введені значення є числами.

2. Модуль перевірки коректності введених даних:

Перевіряє, чи виконується умова .

3. Модуль обчислення значення :

Виконує основні математичні операції.

4. Модуль виводу результату:

Відображає знайдене значення або повідомлення про помилку.

3. Детальне проектування програмного модуля

Вибір мови програмування

C++ використовується через високу продуктивність і підтримку математичних операцій через бібліотеку <cmath>.

Структура програми

Програма містить такі функції:

getInput() – отримує введені користувачем дані.

validateInput(double x, double z) – перевіряє, чи можливо виконати операцію добування квадратного кореня.

calculateS(double x, double y, double z) – обчислює значення .

main() – координує виконання програми.

Код програми

#include <iostream>

#include <cmath>

#include <limits>

using namespace std;

// Функція отримання вхідних даних

bool getInput(double &x, double &y, double &z) {

cout << "Введіть x: ";

if (!(cin >> x)) return false;

cout << "Введіть y: ";

if (!(cin >> y)) return false;

cout << "Введіть z: ";

if (!(cin >> z)) return false;

return true;

}

// Функція перевірки коректності вхідних даних

bool validateInput(double x, double z) {

if (z < 2 \* x) {

cout << "Помилка: z має бути не менше ніж 2x." << endl;

return false;

}

return true;

}

// Функція обчислення значення S

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

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

double trigPart = x \* sin(x \* x \* y);

return abs(sqrtPart - trigPart);

}

int main() {

double x, y, z;

// Отримання введених користувачем значень

if (!getInput(x, y, z)) {

cout << "Помилка: введіть коректні числові значення." << endl;

return 1;

}

// Перевірка коректності вхідних значень

if (!validateInput(x, z)) {

return 1;

}

// Обчислення S

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

// Виведення результату

cout << "Результат: S = " << S << endl;

return 0;

}

4. Опис роботи модулів

1. Модуль введення (getInput)

Отримує значення від користувача.

Використовує cin >> для зчитування введених даних.

Якщо користувач вводить некоректне значення (наприклад, текст замість числа), cin повертає помилку, і функція повертає false.

2. Модуль перевірки коректності (validateInput)

Перевіряє, чи z >=2x

Якщо ні, виводить повідомлення про помилку і повертає false.

3. Модуль обчислення (calculateS)

Обчислює значення за формулою.

Використовує функції sqrt() і sin() з бібліотеки <cmath>.

Використовує abs() для взяття модуля.

4. Головний модуль (main)

Координує весь процес. Викликає функції введення, перевірки, обчислення та виводу результату. Використовує перевірку на помилки для уникнення некоректного введення.

**Аналіз та постановка задачі 8.2**

**Аналіз задачі**

Завдання 8.2 передбачає розробку програми, яка:

1. **Отримує на вхід** числові значення x, y, z та символи a і b.
2. **Виводить** у консоль:
   * Ім'я розробника зі знаком авторського права (©);
   * Результат логічного виразу a + 1 = b + 2 у текстовому форматі (false/true);
   * Значення x, y, z у десятковій (dec) та шістнадцятковій (hex) системах числення;
   * Значення S, яке обчислюється за допомогою функції s\_calculation() із заголовкового файлу Modules/Прізвище.h.

**Постановка задачі**

**Вхідні дані:**

* Цілі числа a та b.
* Дійсні числа x, y, z.

**Вихідні дані:**

* Прізвище та ім’я розробника у форматі © Прізвище Ім'я.
* Результат перевірки логічного виразу a + 1 = b + 2 у форматі false або true.
* Вивід x, y, z у десятковій та шістнадцятковій формах.
* Обчислене значення S (за допомогою s\_calculation()).

**Обмеження та припущення:**

* Числа x, y, z можуть бути будь-якими дійсними числами.
* Функція s\_calculation(x, y, z) уже реалізована у файлі Modules/Прізвище.h.
* Формат виведення має відповідати вимогам завдання (чітка структура, використання cout).

**Підходи до реалізації**

1. **Функція для виведення імені розробника**
2. void printDeveloperInfo() {
3. cout << "© Прізвище Ім'я" << endl;
4. }
5. **Функція для перевірки логічного виразу**
6. bool checkLogicalExpression(int a, int b) {
7. return (a + 1) == (b + 2);
8. }
9. **Функція main() для обробки вводу/виводу**
   * Отримати x, y, z, a, b від користувача.
   * Використати checkLogicalExpression(a, b) для виводу true/false.
   * Вивести x, y, z у dec і hex.
   * Використати s\_calculation(x, y, z) для обчислення S.

**Тест-сьют до задачі 8.1**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | BMTP-Lab8 |
| Назва проекта / ПЗ  Name of Project / Software | Lab8 |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Куртяк К.В. |
| Виконавець  Implementer | Куртяк К.В. |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | 1;2;6. | 1.08368 | Passed |
| 2 | 0;0;5 | 2,23607 | Passed |
| 3 | -2;2;7 | 3.06428 | Passed |
| 4 | 1.5;-2;5 | 0.92829 | Passed |
| 5 | 3;0.5;10 | 0.71864 | Passed |
| 6 | 1;1;2 | 0.0 | Passed |
| 7 | 1;1;1 | Помилка | Passed |
| 8 | 0;1;0 | 0.0 | Passed |
| 9 | 1;100;10000 | 99.98333 | Passed |
| 10 | 1;3.14159;5 | 1.69006 | Passed |
| 11 | 0;100;100 | 10.0 | Passed |
| 12 | -1;-1;3 | 2.38177 | Passed |
| 13 | 1;0.01;10 | 2.806994 | Passed |

**Додатки**

ModulesKurtiak

#include <iostream>

#include <cmath>

#include <stdexcept>

// Функція для обчислення S відповідно до формули варіанту 25

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

return fabs(sqrt(fabs(z - 2 \* x)) - x \* sin(pow(x, 2 \* z) \* y));

}

ModulesK (headers)

#ifndef MODULESK\_H\_INCLUDED

#define MODULESK\_H\_INCLUDED

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

#endif // MODULESK\_H\_INCLUDED

TestDriver

#include <iomanip>

#include <locale>

#include <windows.h>

#include <iostream>

#include "C:\Users\Asus\Desktop\BMTP\_LAB8\_Kurtiak\prj\ModulesKurtiak\ModulesK.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "Ukrainian");

double x, y, z;

cout << "Введіть x, y, z: ";

cin >> x >> y >> z;

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

cout << "S = " << result << endl;

return 0;

}

Kurtiak\_task

#include <iomanip>

#include <locale>

#include <windows.h>

#include <iostream>

#include <stdexcept>

#include "C:\Users\stasj\Desktop\BMTP\_LAB8\_Kurtiak\prj\ModulesKurtiak\ModulesK.h"

using namespace std;

// Функція виведення імені розробника

void printDeveloperInfo() {

cout << "© Катерина Крутяк" << endl;

}

// Функція перевірки логічного виразу a + 1 = b + 2

bool checkLogicalExpression(int a, int b) {

return (a + 1) == (b + 2);

}

int main() {

setlocale(LC\_ALL, "Ukrainian");

// Виведення інформації

printDeveloperInfo();

int a, b;

double x, y, z;

// Введення даних

cout << "Введіть значення x, y, z: ";

cin >> x >> y >> z;

cout << "Введіть значення a, b: ";

cin >> a >> b;

// Обчислення логічного виразу

cout << "Результат логічного виразу: "

<< (checkLogicalExpression(a, b) ? "1 (true)" : "0 (false)") << endl;

// Виведення результатів у різних системах числення

cout << "x = " << dec << x << " (dec), " << hex << x << " (hex)" << endl;

cout << "y = " << dec << y << " (dec), " << hex << y << " (hex)" << endl;

cout << "z = " << dec << z << " (dec), " << hex << z << " (hex)" << endl;

// Обчислення значення S

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

cout << fixed << setprecision(3) << "S = " << S << endl;

return 0;

}

**Аналіз та постановка задачі 8.2**

**Аналіз задачі**

Завдання 8.2 передбачає розробку програми, яка:

1. **Отримує на вхід** числові значення x, y, z та символи a і b.
2. **Виводить** у консоль:
   * Ім'я розробника зі знаком авторського права (©);
   * Результат логічного виразу a + 1 = b + 2 у текстовому форматі (false/true);
   * Значення x, y, z у десятковій (dec) та шістнадцятковій (hex) системах числення;
   * Значення S, яке обчислюється за допомогою функції s\_calculation() із заголовкового файлу Modules/Прізвище.h.

**Постановка задачі**

**Вхідні дані:**

* Цілі числа a та b.
* Дійсні числа x, y, z.

**Вихідні дані:**

* Прізвище та ім’я розробника у форматі © Прізвище Ім'я.
* Результат перевірки логічного виразу a + 1 = b + 2 у форматі false або true.
* Вивід x, y, z у десятковій та шістнадцятковій формах.
* Обчислене значення S (за допомогою s\_calculation()).

**Обмеження та припущення:**

* Числа x, y, z можуть бути будь-якими дійсними числами.
* Функція s\_calculation(x, y, z) уже реалізована у файлі Modules/Прізвище.h.
* Формат виведення має відповідати вимогам завдання (чітка структура, використання cout).

**Підходи до реалізації**

1. **Функція для виведення імені розробника**
2. void printDeveloperInfo() {
3. cout << "© Прізвище Ім'я" << endl;
4. }
5. **Функція для перевірки логічного виразу**
6. bool checkLogicalExpression(int a, int b) {
7. return (a + 1) == (b + 2);
8. }
9. **Функція main() для обробки вводу/виводу**
   * Отримати x, y, z, a, b від користувача.
   * Використати checkLogicalExpression(a, b) для виводу true/false.
   * Вивести x, y, z у dec і hex.
   * Використати s\_calculation(x, y, z) для обчислення S.

**Тест-сьют до задачі 8.2**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | BMTP-Lab8 |
| Назва проекта / ПЗ  Name of Project / Software | Lab8 |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Куртяк К.В. |
| Виконавець  Implementer | Куртяк К.В. |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | 1.Увести значення  x = 2, y = 47, z = 7  a = 1, b = 2 | © Катерина Крутяк  Введіть значення x, y, z: Введіть значення a, b:  Результат логічного виразу: 1 (true)  x = 2 (dec), 2 (hex)  y = 47 (dec), 2f (hex)  z = 7 (dec), 7 (hex)  S = 0.985 | Passed |
| 2 | 1.Увести значення  x, y, z: 3 57 10  a, b: 7 1 | © Катерина Крутяк  Введіть значення x, y, z: Введіть значення a, b:  Результат логічного виразу: 0 (false)  x = 3 (dec), 3 (hex)  y = 57 (dec), 39 (hex)  z = 10 (dec), a (hex)  S = 0.500 | Passed |
| 3 | 1.Увести значення  a = 5  b = 0  x = 11  y = 5  z = 1 | © Катерина Крутяк  Введіть значення x, y, z: Введіть значення a, b:  Результат логічного виразу: 0 (false)  x = 11 (dec), b (hex)  y = 5 (dec), 5 (hex)  z = 1 (dec), 1 (hex)  S = 2.322 | Passed |
| 4 | 1.Увести значення  x, y, z: 1 11 6  a, b: 12 2 | © Катерина Крутяк  Введіть значення x, y, z: Введіть значення a, b:  Результат логічного виразу: 0 (false)  x = 1 (dec), 1 (hex)  y = 11 (dec), b (hex)  z = 6 (dec), 6 (hex)  S = 3.000 | Passed |
| 5 | 1.Увести значення  a = 4, b = 9, x = 3, y = 1, z = 7 | © Катерина Крутяк  Введіть значення x, y, z: Введіть значення a, b:  Результат логічного виразу: 0 (false)  x = 3 (dec), 3 (hex)  y = 1 (dec), 1 (hex)  z = 7 (dec), 7 (hex)  S = 3.715 | Passed |

**Архітектура**

**1)** Спершу створюється статична бібліотека ModulesKurtiak та заголовковий файл ModulesK.h, в якому міститься прототип s\_calculation в файлі ModulesKurtiak.cpp реалізація функції s\_calculation.

2) Текстовий проект під назвою TestDriver містить файл main.cpp, який викликає функцію s\_calculation і тестує вхідні данні.

3) Створюємо текстовий набір TestSuite у файл text.txt, в якому міститься вхідні данні та очікувані результати.

Висновок:

1. Практика написання коду дозволяє закріпити теоретичні знання на практиці. 2. Допомагає зрозуміти реальні виклики та труднощі програмування. 3. Покращує навички розв’язання алгоритмічних задач. 4. Розвиває логічне мислення та аналітичні здібності. 5. Дозволяє ефективніше використовувати мову програмування. 6. Покращує знання про структури даних та алгоритми. 7. Вчить писати чистий та зрозумілий код. 8. Допомагає виявити та виправити власні помилки. 9. Розвиває здатність до оптимізації коду. 10. Дає змогу отримати реальний досвід роботи з мовою C++. 11. Дозволяє засвоїти метод функціональної декомпозиції. 12. Вчить розділяти великий код на незалежні модулі. 13. Дає змогу створювати масштабовані та зручні для підтримки проєкти. 14. Допомагає правильно використовувати інтерфейси між модулями. 15. Покращує розуміння принципів інкапсуляції. 16. Дозволяє ефективно використовувати бібліотеки та зовнішні модулі. 17. Вчить реалізовувати принцип повторного використання коду. 18. Покращує навички написання та розуміння API. 19. Дозволяє краще структурувати програмний код. 20. Вчить уникати надмірної залежності між модулями. 21. Дає змогу навчитися тестувати код. 22. Дозволяє засвоїти методи модульного тестування. 23. Вчить ефективно використовувати відладчик. 24. Дозволяє виявляти та виправляти помилки. 25. Допомагає навчитися працювати з винятками. 26. Вчить писати тести для власного коду. 27. Покращує розуміння межових випадків у програмуванні. 28. Дозволяє краще зрозуміти внутрішню логіку програми. 29. Навчає використовувати інструменти автоматизованого тестування. 30. Знижує ймовірність появи критичних помилок у коді. 31. Допомагає підготуватися до технічних співбесід. 32. Вчить працювати з реальними проєктами. 33. Покращує здатність швидко адаптуватися до нових технологій. 34. Дозволяє швидше писати код. 35. Збільшує шанси отримати хорошу роботу. 36. Розширює можливості для роботи в команді. 37. Допомагає стати фахівцем у сфері програмування. 38. Дозволяє брати участь в open-source проєктах. 39. Дає змогу створювати власні проєкти та стартапи. 40. Розширює можливості кар'єрного зростання. 41. Допомагає навчитися працювати з документацією. 42. Вчить читати та розуміти чужий код. 43. Розвиває навички алгоритмічного мислення. 44. Допомагає стати кращим програмістом. 45. Розвиває увагу до деталей. 46. Дозволяє краще зрозуміти принципи роботи комп’ютерів. 47. Покращує здатність працювати з новими технологіями. 48. Дає змогу отримати задоволення від створення власних програм. 49. Розширює світогляд у сфері IT. 50. Дозволяє досягти професійного успіху в програмуванні.