

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення
Дисципліна: Базові методології та технології програмування

Лабораторна робота №8
Тема: «РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ
ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ»

Виконав: ст. гр. КН-24
Куріщенко П. В.
Перевірив: викладач
Коваленко А.С.

Кропивницький 2024

Варіант - 10

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 C++ 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 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.

ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Завдання 1.

Аналіз вимог:

Вхідні дані: 3 цілих числа(x, y, z).

Вихідні дані: ціле число(S).

ОДЗ: $\{ x > y \text{ та } x + (z / 2) * y^2 \neq 0$

Результати тестування s_calculation:

```
Enter three numbers: 23 11 78
Your result is 3.07691
```

```
-----  
Enter three numbers: 1 23 1  
Error: log(x - y) is undefined (x must be greater than y).  
-----  
  
Enter three numbers: 4 1 -8  
Error: Division by zero.
```

Вихідний код (TestDriver):

```
1  #include <iostream>  
2  #include <math.h>  
3  #include "sample_module.h"  
4  
5  using namespace std;  
6  
7  float s_calculator(float x, float y, float z);  
8  
9  int main()  
10 {  
11     float x, z, y;  
12     cout << "Enter three numbers: ";  
13     cin >> x >> y >> z;  
14     s_calculator(x, y, z);  
15     return 0;  
16 }
```

Вихідний код (ModulesKurishchenko):

```
1  #include <iostream>  
2  #include <math.h>  
3  
4  using namespace std;  
5  
6  float s_calculator(float x, float y, float z)  
7  {  
8      float S;  
9  }
```

```

10     if(x <= y){
11         cout << "Error: log(x - y) is undefined (x must be
           greater than y).\n";
12     }
13     else if(x + (z / 2) * pow(y, 2) == 0){
14         cout << "Error: Division by zero.\n";
15     }
16     else{
17         S = log(x - y) + sqrt((M_PI * pow(x, 2))/(x + (z / 2 *
           pow(y, 2))));
18         cout << "Your result is " << S;
19     }
20 }

```

TestSuite:

Назва тестового набору Test Suite Description	TS_8_1
Назва проекту / ПЗ Name of Project / Software	Курищенко-task_8_1.exe
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сюита Test Suite Author	Павло Курищенко
Виконавець Implementer	Павло Курищенко

[Розрив обтікання текстом]

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC-01	1. Ввести 23 12 34	Вікно застосунку: Your result is 3.21799	passed
TC-02	1. Ввести -1 -12 23	Вікно застосунку: Your result is 2.44146	passed

TC-03	1. Ввести 3.2 -23 100	Вікно застосунку: Your result is 3.30063	passed
TC-04	1. Ввести 23 23 23	Вікно застосунку: Error: log(x - y) is undefined (x must be greater than y).	passed
TC-05	1. Ввести -78 1 0	Вікно застосунку: Error: log(x - y) is undefined (x must be greater than y).	passed
TC-06	1. Ввести 4 1 -8	Вікно застосунку: Error: Division by zero.	passed

Завдання 2.

Аналіз вимог:

Вхідні дані: 3 цілих числа(x, y, z), 2 символи(a, b).

Вихідні дані: логічний вираз 1 або 0(answer), 3 цілих числа(x, y, z) у десятковій та шістнадцятковій системі числення, ціле число(S).

ОДЗ: $\{ x > y \text{ та } x + (z / 2) * y^2 \neq 0$

Вихідний код (Kurishchenko_task):

```
#include <iostream>
#include <iomanip>
#include "sample_module.h"

using namespace std;

bool log_eq(char a, char b)
{
    bool answer;
    if (a + 10 >= b) {
        answer = true;
    }
    else{
        answer = false;
    }
}
```

```

    }
    return answer;
}

float transf(float x, float y, float z)
{
    cout << "Your numbers in dec: " << x << ", " << y << ", " << z
<< endl;
    cout << "Your numbers in hex: " << hex << showbase <<
static_cast<int>(x) << ", " << static_cast<int>(y) << ", " <<
static_cast<int>(z) << endl;
}

float s_calculator(float x, float y, float z);

int main()
{
    Dev_info();

    float x, y, z;
    char a, b;

    cout << "Enter 2 symbols: ";
    cin >> a >> b;
    cout << "Enter 3 whole numbers: ";
    cin >> x >> y >> z;

    cout << "The result of equation is " << log_eq(a, b) << endl;
    transf(x, y, z);
    cout << "The result of calculation \"S\" is " << s_calculator(x,
y, z) << endl;
    return 0;
}

```

TestSuite:

Назва тестового набору	TS_8_2
Test Suite Description	

Назва проекту / ПЗ Name of Project / Software	Kurishchenko_task.exe
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сюита Test Suite Author	Павло Курищенко
Виконавець Implementer	Павло Курищенко

[Розрив обтікання текстом]

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC-01	1. Ввести A z 2. Ввести 4 1 -8	Вікно застосунку: The result of equation is 0 Your numbers in dec: 4, 1, -8 Your numbers in hex: 0x4, 0x1, 0xffffffff8 The result of calculation "S" is Error: Division by zero.	passed
TC-02	1. Ввести # & 2. Ввести 2.34 23 12	Вікно застосунку: The result of equation is 1 Your numbers in dec: 2.34, 23, 12 Your numbers in hex: 0x2, 0x17, 0xc The result of calculation "S" is Error: log(x - y) is undefined (x must be greater than y). 0	passed
TC-03	1. Ввести 0 9 2. Ввести 1 0 0	Вікно застосунку: The result of equation is 1 Your numbers in dec: 1, 0, 0 Your numbers in hex: 0x1, 0, 0 The result of calculation "S" is Your result is 1.772450	passed
TC-04	1. Ввести ** 2. Ввести -1 -2 -3	Вікно застосунку: The result of equation is 1 Your numbers in dec: -1, -2, -3 Your numbers in hex: 0xfffffffff, 0xfffffffffe, 0xfffffffffd The result of calculation "S" is Your result is nan0.	passed

TC-05	1. Ввести (: 2. Ввести 34 45 56	Вікно застосунку: The result of equation is 0 Your numbers in dec: 34, 45, 56 Your numbers in hex: 0x22, 0x2d, 0x38 The result of calculation "S" is Error: log(x - y) is undefined (x must be greater than y).	passed
-------	-------------------------------------	--	--------

Висновок:

У ході виконання лабораторної роботи №8 я отримав цінний досвід та навчився:

1. Аналізувати умови завдань і формувати алгоритми розв'язку.
2. Використовувати математичні формули у програмному коді.
3. Реалізовувати логічні вирази та оператори порівняння.
4. Використовувати бібліотеки STL для форматowanego виводу.
5. Виводити текст та значення змінних за допомогою cout.
6. Реалізовувати функції для обчислення значень.
7. Підключати заголовкові файли у проєкті.
8. Використовувати зовнішні модулі для розділення коду.
9. Декларувати та використовувати функції у заголовкових файлах.
10. Перетворювати числові значення між десятковою та шістнадцятковою системами.
11. Реалізовувати обчислення з використанням логарифмічних функцій.
12. Використовувати математичні константи (π , e).
13. Опрацьовувати квадратні корені у виразах.
14. Застосовувати оператори +, -, *, / у складних математичних виразах.
15. Створювати функції, що повертають числові значення.
16. Використовувати булеві значення у програмуванні.
17. Виводити булеві значення у вигляді 0 та 1.
18. Опрацьовувати вхідні дані від користувача.
19. Впорядковувати код для зручності читання.
20. Дотримуватися стилю написання коду.

21. Використовувати оператори namespace std.
22. Оголошувати змінні перед їх використанням.
23. Відокремлювати оголошення змінних від їх ініціалізації.
24. Додавати коментарі для покращення розуміння коду.
25. Відлагоджувати код для усунення помилок.
26. Створювати трасувальну таблицю для перевірки роботи алгоритму.
27. Реалізовувати тестові кейси для перевірки правильності виконання алгоритму.
28. Використовувати умовні вирази if для перевірки коректності вхідних даних.
29. Опрацьовувати помилки введення.
30. Використовувати iomanip для форматування виводу.
31. Відображати числа з заданою кількістю знаків після коми.
32. Відокремлювати логіку програми на окремі функції.
33. Реалізовувати оператор return для повернення значень.
34. Додавати обробку винятків у коді.
35. Аналізувати вихідні дані на коректність.
36. Перевіряти рівність двох значень у коді.
37. Використовувати операції >=, <=, ==, !=.
38. Реалізовувати програму з розділенням функцій між файлами.
39. Перевіряти граничні випадки у тестових прикладах.
40. Оптимізувати алгоритми для швидшого виконання.
41. Впорядковувати змінні у коді.
42. Використовувати const для незмінних значень.
43. Оголошувати глобальні та локальні змінні.
44. Розбиратися в основах роботи зі змінними типу char.
45. Опрацьовувати символьні значення у коді.
46. Використовувати bool для логічних змінних.
47. Реалізовувати програму з урахуванням читаємості коду.
48. Писати документацію до функцій.

49. Використовувати коментарі для пояснення коду.
50. Розуміти принципи роботи арифметичних операцій.
51. Використовувати змінні `int`, `double` у різних контекстах.
52. Аналізувати та тестувати різні підходи до реалізації алгоритму.
53. Покращувати навички роботи у середовищі розробки.
54. Використовувати оператор `cout` для зручного відображення даних.

Ця лабораторна робота дозволила значно покращити навички написання коду, реалізації математичних операцій та перевірки алгоритмів на коректність.

КОНТРОЛЬНІ ЗАПИТАННЯ

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

Стандарт визначає процеси життєвого циклу ПЗ, включаючи вимоги, архітектуру, дизайн, тестування та супровід. Його мета – створення якісного та надійного ПЗ, забезпечення його сумісності та керованості.

2. Різниця між функцією та модулем у C/C++

Функція – це блок коду, що виконує певне завдання. Модуль – це файл або частина програми, що містить пов’язані функції та змінні й використовується для організації коду.

3. Відмінність `main` від інших функцій у C/C++

`main` – це точка входу у програму, з неї починається виконання. Інші функції викликаються з `main` або одна з одної.

4. Призначення маніпуляторів у C++

Маніпулятори (`setw`, `setprecision`, `hex`, `endl` тощо) змінюють формат виводу даних через `cout`, наприклад, встановлюючи точність чисел чи змінюючи систему числення.

5. Призначення заголовкового файлу у препроцесінгу

Заголовкові файли містять оголошення функцій, макроси, константи й

структури. Вони дозволяють розділити код на частини та використовувати функції з інших модулів (`#include <iostream>`).

6. Використання `iostream` у лабораторній роботі

Використано `cout` та `cin` для виводу і введення даних, а також `endl` для переходу на новий рядок.

7. Стандартний простір імен у C++

Простір імен `std` містить стандартні об'єкти (наприклад, `cout`, `cin`). Визначається директивою `using namespace std;`.

8. Приклад оголошення змінної з ініціалізацією

```
int          x          =          5;
double       pi         =          3.14;
char         c          =          'A';
```

9. Заборонені символи в ідентифікаторах

Не можна використовувати пробіли, спеціальні символи (`@`, `#`, `%`), починати ім'я з цифри або використовувати ключові слова мови (`int`, `return` тощо).

10. Перетворення десяткового літерала у шістнадцятковий

```
int          num        =          255;
cout << hex << num;    // Виведе "ff"
```

11. Підключення власної бібліотеки

Необхідно написати заголовковий файл (`.h`), реалізувати його у `.cpp`, а потім підключити:

```
#include "my_library.h"
```

12. Різниця між заголовковим та об'єктним файлом

Заголовковий (`.h`) містить оголошення, об'єктний (`.o` або `.obj`) – скомпільований код.

13. Виведення нестандартних символів

```
cout << "\u03C0"; // Виведе символ π
```

14.Різниця між символьним та рядковим літералом

Символьний ('A') зберігається як один символ, рядковий ("ABC") – як масив символів.

15.Синтаксис прототипу функції

```
int add(int a, int b); // Оголошення  
int add(int a, int b) { return a + b; } // Реалізація
```

Прототип записується у заголовковому файлі або перед main().

16.Оператор повернення значення у C++

```
return x + y;
```

17.ESC-послідовності у C++

\n – новий рядок
\t – табуляція
\\ – зворотний слеш

18.Функції cmath та їх відмінність від math.h

cmath є стандартом C++, підтримує перевантаження. Функції:

sqrt(x) – квадратний корінь
pow(x, y) – зведення у ступінь
sin(x), cos(x), tan(x) – тригонометрія

19.Призначення тестових драйверів

Вони використовуються для перевірки окремих функцій або модулів у процесі розробки.

20.Відмінність тестування модулів від тестування exe-файлу

Тестування модулів (unit testing) перевіряє окремі компоненти, а тестування exe-файлу – поведінку всієї програми.