

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”

На тему: “Реалізація статичних модулів Лінійних обчислювальних
процесів.”

ВИКОНАВ

студент академічної групи

КБ-22-1

Паращенко Д.С

ПЕРЕВІРИВ

Викладач

Собінов.О.Г

Кропивницьк - 2022

ТЕМА: Реалізація статичних 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 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:

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

ВАРІАНТ 3

— ЗАДАЧА 8.1 —

За значеннями x, y, z обчислюється S :

$$S = \left(3 \sin \sqrt{\frac{12x^2}{\lg(x-3)}} \right)^2 + \frac{1}{2}z.$$

— ЗАДАЧА 8.2 —

За послідовними запитами вводяться числа x, y, z та символи a і b .

В и в е с т и (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`):

$$a + 1 > |b - 2| ?$$

8.2.3. Значення x, y, z в десятковій і шістнадцятковій системах числення; S , що обчислюється функцією `s_calculation()` заголовкового файлу `ModulesПрізвище.h`.

* Підзадачі 8.2.1–8.2.3 варто реалізувати у вигляді функцій, результат виконання яких включається у вихідний потік `cout` за допомогою оператора вставки `<<` (наприклад, “`cout << YourFunc(a,b);`”).



- Файл статичної бібліотеки у Code::Blocks IDE має розширення **.a** (у, наприклад, Visual Studio C++ — **.lib**), заголовковий файл — **.h**.
- У C++ змінні оголошуються (описуються) перед їх безпосереднім використанням (у C — строго на початку тіла функції).
- Форматування виведення інформації можливо реалізувати за допомогою відповідних маніпуляторів **iomanip** C++.

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.

Задача 8.1

Файл `main.cpp`

#include <iostream> // Директивою підключення заголовочного файлу, яка дозволяє використовувати стандартний ввід та вивід даних на консоль.

#include <windows.h> // Директива підключення заголовочного файлу, яка дозволяє використовувати функції та інші елементи API операційної системи Windows.

#include <cmath> //Включає заголовочний файл для роботи з математичними функціями.

using namespace std; //Директива імпорту простору імен. Ця директива дозволяє використовувати ідентифікатори з простору імен std (стандартна бібліотека C++) без префіксу "std::".

int main() {

SetConsoleOutputCP(1251); // Директива підключення заголовочного файлу, яка дозволяє використовувати функції та інші елементи API операційної системи Windows.

SetConsoleCP(1251); //Ця функція встановлює кодову сторінку (code page) консолі Windows для зчитування символів у відповідності з кирилицею (CP1251).

double x, z, S;

// Введення значень змінних x, z.

cout << "Введіть значення x: ";

cin >> x;

cout << "Введіть значення z: ";

cin >> z;

// Викликаємо функції для виконання підзадач та виводимо результат у вихідний потік.

cout << "Значення x: " << x << " (0x" << hex << (int)x << ")" << endl;

cout << "Значення z: " << z << " (0x" << hex << (int)z << ")" << endl;

S = s_calculation(x, z);

```
cout << "Значення виразу S: " << S << endl;  
}
```

Файл s_calculation.h

```
#ifndef S_CALCULATION_H_INCLUDED
```

```
#define S_CALCULATION_H_INCLUDED
```

```
double s_calculation(double x, double z) {
```

```
    // виконання розрахунків
```

```
    double S = pow(3 * sin(sqrt((12 * pow(x, 2)) / log10(x - 3))), 2) + (0.5 * z);
```

```
    return S;
```

```
}
```

```
#endif // S_CALCULATION_H_INCLUDED
```

Задача 8.2

```
#include <iostream> // Директивою підключення заголовочного файлу, яка  
дозволяє використовувати стандартний ввід та вивід даних на консоль.
```

```
#include <windows.h> // Директива підключення заголовочного файлу, яка  
дозволяє використовувати функції та інші елементи API операційної  
системи Windows.
```

```
#include "ModulesParashchenko.h" // Підключаємо заголовочний файл з  
функціями.
```

```
#include "s_calculation.h" // Підключаємо заголовочний файл з функціями.
```

```
int a, b;
```

```
#include <cmath> //Включає заголовочний файл для роботи з  
математичними функціями.
```

using namespace std; //Директива імпорту простору імен. Ця директива дозволяє використовувати ідентифікатори з простору імен std (стандартна бібліотека C++) без префіксу "std::".

```
int main() {
```

```
SetConsoleOutputCP(1251); // Директива підключення заголовочного файлу, яка дозволяє використовувати функції та інші елементи API операційної системи Windows.
```

```
SetConsoleCP(1251); //Ця функція встановлює кодову сторінку (code page) консолі Windows для зчитування символів у відповідності з кирилицею (CP1251).
```

```
int a, b;
```

```
// Введення значень змінних x, z, a та b.
```

```
cout << "Введіть символ a: ";
```

```
cin >> a;
```

```
cout << "Введіть символ b: ";
```

```
cin >> b;
```

```
// Викликаємо функції для виконання підзадач та виводимо результат у вихідний потік.
```

```
cout << "Розробник програми: " << author_name() << " ©" << endl;
```

```
cout << "Результат виразу  $a+1>|b-2|$ : " << boolalpha << expression_result(a, b) << endl;
```

```
S = s_calculation(x, z);
```

```
cout << "Значення виразу S: " << S << endl;
```

```
cout << DecHex(x, z) << endl;
```

```
}
```

#Для зручності перевірки коду вони з'єднанні в одному файлі (main.cpp)

#include <iostream> // Директивою підключення заголовочного файлу, яка дозволяє використовувати стандартний ввід та вивід даних на консоль.

#include <windows.h> // Директива підключення заголовочного файлу, яка дозволяє використовувати функції та інші елементи API операційної системи Windows.

#include <cmath> //Включає заголовочний файл для роботи з математичними функціями.

#include "ModulesParashchenko.h" // Підключаємо заголовочний файл з функціями.

#include "s_calculation.h" // Підключаємо заголовочний файл з функціями.

#include <iomanip>

using namespace std; //Директива імпорту простору імен. Ця директива дозволяє використовувати ідентифікатори з простору імен std (стандартна бібліотека C++) без префіксу "std::".

int main() {

SetConsoleOutputCP(1251); // Директива підключення заголовочного файлу, яка дозволяє використовувати функції та інші елементи API операційної системи Windows.

SetConsoleCP(1251); //Ця функція встановлює кодову сторінку (code page) консолі Windows для зчитування символів у відповідності з кирилицею (CP1251).

double x, z, S;

int a, b;

// Введення значень змінних x, z, a та b.

cout << "Введіть значення x: ";

```

cin >> x;

cout << "Введіть значення z: ";
cin >> z;

cout << "Введіть символ a: ";
cin >> a;

cout << "Введіть символ b: ";
cin >> b;

// Викликаємо функції для виконання підзадач та виводимо результат у
вихідний потік.
cout << "Розробник програми: " << author_name() << " ©" << endl;
cout << "Результат виразу a+1>|b-2|: " << boolalpha << expression_result(a,
b) << endl;
cout << "Значення x: " << x << " (0x" << hex << (int)x << ")" << endl;
cout << "Значення z: " << z << " (0x" << hex << (int)z << ")" << endl;
S = s_calculation(x, z);
cout << "Значення виразу S: " << S << endl;
cout << DecHex(x, z) << endl;
return 0;
}

```

Назва тестового набору	BMTP_LW8_TS8
Test Suite Description	
Назва проекту / ПЗ	main.exe

Name of Project / Software	
Рівень тестування Level of Testing	Модульний
Автор тест-сююта Test Suite Author	Паращенко Д. С.
Виконавець Implementer	Паращенко Д. С.

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)
ТС-01	Вхід x=2,z=5,a=6,b=7	<p>Вихід Розробник програми: Паращенко Денис ©</p> <p>Результат виразу $a+1 > b-2$: true</p> <p>Значення x: 2 (0x2)</p> <p>Значення z: 5 (0x5)</p> <p>Значення виразу S: nan</p> <p>Десяткова : 'x' -> 2</p>	passed

		'z' -> 5 Шістнадцяткова: 'x' -> 2 'z' -> 5	
ТС-02	Вхід x=0,z=0,a=0,b=0	Вихід Введіть значення x: 0 Введіть значення z: 0 Введіть символ a: 0 Введіть символ b: 0 Розробник програми: Паращенко Денис © Результат виразу $a+1> b-2 $: false Значення x: 0 (0x0) Значення z: 0 (0x0) Значення виразу S: nan Десяткова : 'x' -> 0 'z' -> 0 Шістнадцяткова: 'x' -> 0 'z' -> 0	passed
ТС-03	Вхід x= -90,z= -25, a=-25,b=30	Вихід Введіть значення x: -90 Введіть значення z: -70 Введіть символ a: -25	passed

		<p>Введіть символ b: 30</p> <p>Розробник програми: Паращенко Денис ©</p> <p>Результат виразу $a+1> b-2$: false</p> <p>Значення x: -90 (0xfffffa6)</p> <p>Значення z: -70 (0xfffffba)</p> <p>Значення виразу S: nan</p> <p>Десяткова :</p> <p>'x' -> -90</p> <p>'z' -> -70</p> <p>Шістнадцяткова:</p> <p>'x' -> ffffffa6</p> <p>'z' -> ffffffba</p>	
ТС-04	Вхід x=130 ,z=150 , a=227,b=228	<p>Вихід Введіть значення x: 130</p> <p>Введіть значення z: 150</p> <p>Введіть символ a: 227</p> <p>Введіть символ b: 228</p> <p>Розробник програми: Паращенко Денис ©</p> <p>Результат виразу $a+1> b-2$: true</p> <p>Значення x: 130 (0x82)</p> <p>Значення z: 150 (0x96)</p> <p>Значення виразу S: 77.3734</p>	passed

		Десяткова : 'x' -> 130 'z' -> 150 Шістнадцяткова: 'x' -> 82 'z' -> 96	
--	--	--	--

Висновок:

Під час виконання лабораторної роботи було отримано багато корисних навичок і вмінь з розробки модульних програм з використанням мови програмування C++. Було вивчено методологію модульного програмування, метод функціональної декомпозиції задач та метод модульного (блочного) тестування.

Було ознайомлено зі засобами мови програмування C++, включаючи роботу зі скалярними типами даних, арифметичними і логічними операціями, потоковим введенням та виведенням інформації. Було проведено практичні вправи з розробки програмних модулів та засобів у кросплатформовому середовищі Code::Blocks (GNU GCC Compiler).

Отже, можна зробити висновок, що виконання лабораторної роботи дало можливість поглибити знання з програмування та розробки програм, заснованих на модульній архітектурі. Крім того, отримані вміння та навички дозволять ефективніше розв'язувати завдання, пов'язані з програмуванням, в тому числі у кросплатформових середовищах.