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 бібліотек модулів лінійних обчислювальних процесів”

ВИКОНАВ

студент академічної групи КБ-21

\_\_\_\_\_\_\_\_\_ А.С. Долинко

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_ О.П. Доренський

м. Кропивницький 2023

**Мета роботи :**

набуття 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 :** №7

**ЗАДАЧА 8.1**

За значеннями x, y, z обчислюється Ѕ:

S = √(|z\*z\*1/2\*y|)+(п+х+е\*\*|y|)/y

**Аналіз умови до задачі 8.1**

Користувач вводить 3 числа, з плаваючою комою і йому виводиться число, підрахунок якого відбувається за формулою S = √(|z\*z\*1/2\*y|)+(п+х+е\*\*|y|)/y

**Початок**

1. Вивести в консоль “Enter x”;
2. Чекати поки користувач введе x;
3. Вивести в консоль “Enter y”;
4. Чекати поки користувач введе y;
5. Вивести в консоль “Enter z”
6. Чекати поки користувач введе z;
7. S = √(|z\*z\*1/2\*y|)+(п+х+е\*\*|y|)/y
8. Вивести на екран S = √(|z\*z\*1/2\*y|)+(п+х+е\*\*|y|)/у

**Кінець**

**Задача 8.2**

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

Вивести (включити у потік STL — cout)\*:

* + 1. Прізвище та ім'я розробника програми зі знаком охорони авторського права «С» (від англ. copyright);

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

a+5>=b ?

8.2.3. Значення x, y, z в десятковій і шістнадцятковій системах числення; Ѕ, що обчислюється функцією обчислюється функцією s\_calculation() заголовкового файлу ModulеѕПрізвище.h.

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

**Аналіз задачі 8.2**

Числа x, y, z повинні бути типу int, a і b повинні бути типу float.

**Аналіз до пункту 8.2.1**

Потрібно створити функцію copyright(), яка повертає прізвище та ім’я розробника програми, в даному випадку це «Долинко Антон (с)».

**Аналіз до пункту 8.2.2**

Потрібно реалізувати 2 функції. Перша функція буде приймати тип bool і повертати текст типу string «true» або «false» залежачи від змінної яку вона прийняла. (назва функції - boolToString())

Наступна функція returnExpression() буде повертати текст типу string «a+5>=b ?» + текст із тим чи правдиве це твердження чи ні(для цього нам було потрібно реалізувати boolToString)

**Аналіз до пункту 8.2.3**

Для цього завдання треба реалізувати 4 функції.

Перша функція буде приймати 3 змінні типу інт та буде повертати числа в десятковій системі числення.

Друга функція буде приймати 3 змінні з типом інт та буде повертати числа в шістнадцятковій системі числення (Для того щоб вертати числа в шістнадцятковій системі числення нам треба створити функцію decToHex(), яка буде приймати тип інт та повертати число в шістнадцятковій системі числення).

Четверта функція приймати 3 числа та повертати тип стрінг, із таким повідомленням «(√(|z\*z\*1/2\*y|)+(п+х+е\*\*|y|)/у» + s\_calculation() в яку ми будемо передавати 3 числа, які ввів користувач.

**Алгоритм до задачі 8.2**

**Початок**

1. Вивести в консоль “Enter x”;
2. Чекати поки користувач введе x;
3. Вивести в консоль “Enter y”;
4. Чекати поки користувач введе y;
5. Вивести в консоль “Enter z”
6. Чекати поки користувач введе z;
7. Вивести в консоль “Enter a”;
8. Чекати поки користувач введе a;
9. Вивести в консоль “Enter b”;
10. Чекати поки користувач введе b;
11. Вивести в консоль «Долинко Антон (с)»;
12. Якщо a+5>=b ? то вивести в консоль «a+5>=b true» та перейти на крок 14
13. Вивести в консоль «a+5>=b ? false»
14. Вивести в консоль числа x, y, z в десятковій системі числення ;
15. Вивести в консоль числа x, y, z в шістнадцятковій системі числення ;
16. S = √(|z\*z\*1/2\*y|)+(п+х+е\*\*|y|)/у;
17. Вивести на екран “(√(|z\*z\*1/2\*y|)+(п+х+е\*\*|y|)/у “ S;

**Кінець**

**Висновок**

Під час виконання лабораторної роботи № 8 на тему **«**Реалізація статичних бібліотек модулів лінійних обчислювальних процесів**»** з предмету «Базові Методології та Технології Програмування» я відпрацював на практиці створення статичних бібліотек та власних модулів і заголовних файлів, написання власних текстових драйверів для своїх модулів. Набув 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).

Під час лабораторної роботи було виконано такі пункти:

* Отримано новий репозиторій із моїми завданнями до лабораторної роботи №8
* Скопійовано посилання на репозиторій та було клоновано репозиторій на свій локальний диск D . Реалізовано це було за допомогою команди:

git clone https://github.com/odorenskyi/Dolynko-Anton-KN22.git

* Після цього було створено папки prj, Software, Test Suite, та Report і заповнено README.md файл.
* Після цього було відправлено ці дані за допомогою команд :

git add -A

git commit -m “Add folder prj, Software, Test Suite, Report and feeling README file in folder lab08”

git push

* Прочитано завдання та виконано декомпозицію задачі 8.1
* Виконано аналіз першого завдання, в результаті чого, створено файл–шаблон тестового набору (Unit testing) із назвою US\_8\_1.
* Розроблено 7 Test Case
* Та завантажено на репозиторій в паку lab08
* У Code::Blocks створено статичну бібліотеку із назвою ModulesDolynko
* За допомогою визначених вимог до задачі та розробленого алгоритму було реалізовано функцію s\_calculation() і поміщено прототип функції в заголовний файл ModulesDolynko.h.
* Потім було написаний тестовий драйвер функції s\_calculation() та протестовано із всіма позначками passed
* Визначено вимоги до задачі 8.2
* Виконано аналіз задачі 8.2*.*
* Розроблено Test Case для майбутнього програмного застосунку.
* Створено алгоритм виконання задачі 8.2
* Створено лістинг до задачі 8.2.
* Було додано додаткові функції в статичну бібліотеку які використовувалися в реалізації задачі 8.2
* Після компіляції Dolynko\_task було створено Dolynko\_task.exe . Цей файл був переміщенний в папку Software.
* Виконано системне тестування ПЗ застосунку Dolynko\_task.exe, результати записані у TS\_8\_2, всі тести були пройдені із результатом passed.
* Виконано системне тестування ПЗ застосунку Dolynko\_task.exe, результати записані у TS\_8\_2, всі тести були пройдені із результатом passed.
* Всі данні було відправлено на репозиторій в папку lab08

Було цікаво створювати свою статичну бібліотеку та підключати її до проекту. Також було вивчено як можна конвертувати тип int в тип string, для повернення коректних даних із функцій. В ДОДАТКУ А указані лістинг програм, статичної бібліотеки та тестового драйвера. В ДОДАТКУ Б указані знаходяться Unit test до функції s\_calculation() та Test Suite до задачі 8.2

**ДОДАТОК А**

**Лістинг вихідного коду із ModulesDolynko**

#include <iostream>

#include <math.h>

#include <windows.h>

#include <string>

using namespace std;

float s\_calculation(int x, int y, int z)

{

return (pow(abs(z \* z \* 1 / 2 \* y), 1 / 2) + (3.14 \* x + exp(y)) / y);

}

string decToHexa(int n)

{

char hexaDeciNum[100];

string str = "";

int i = 0;

while (n != 0) {

int temp = 0;

temp = n % 16;

// check if temp < 10

if (temp < 10) {

hexaDeciNum[i] = temp + 48;

i++;

}

else {

hexaDeciNum[i] = temp + 55;

i++;

}

n = n / 16;

}

for (int j = i - 1; j >= 0; j--)

str += hexaDeciNum[j];

return str;

}

string boolToString(bool b){

return b ? "true" : "false";

}

string copyright(){

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

return "Dolynko Anton ©";

}

string trueOrFalse(int a, int b){

return "a + 1 > |b - 2| ? " + boolToString(a + 1 > abs(b - 2));

}

string decimal(int x, int y, int z){

return "x = " + to\_string(x) +

", y = " + to\_string(y) +

", z = " + to\_string(z);

}

string hexADecimal(int x, int y, int z){

return "x = " + decToHexa(x) +

", y = " + decToHexa(y) +

", z = " + decToHexa(z);

}

string s\_calculation\_to\_string(int x, int y, int z){

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

return "(pow(abs(z \* z \* 1 / 2 \* y), 1 / 2) + (3.14 \* x + exp(y)) / y);" - 3)))^" +

to\_string(y) + " + " + to\_string(z) + "/" + to\_string(x) + " = " + to\_string(s\_calculation(x, y, z));

}

**Лістинг вихідного коду із TestDriver**

#include <iostream>

#include <ModulesDolynko.h>

using namespace std;

int main()

{

int x [7]= {120, 30, 90, 7, 10, 10, 1};

int y [7] = {2, 5, -2, -10, 4, 4, 0};

int z [7] = {1000, 60, 6, 89, 1000, -1000, 1};

float result [7] = {8.84, 2.10, 0.11, 408.25 , 180.64, -19.36, 2.00};

for(int i = 0; i < 7; i ++){

if(round(s\_calculation(x[i], y[i], z[i]) \* 100) / 100 == result[i] ){

cout << "Test is passed" << endl;

}

else{

cout << "Test failed" << endl;

}

}

return 0;

}

**Лістинг вихідного коду із Zadorozhny\_task**

#include <iostream>

#include <cmath>

#include <ModulesDolynko.h>

#include <windows.h>

using namespace std;

int main()

{

int x = 0.0;

int y = 0.0;

int z = 0.0;

float a = 0.0;

float b = 0.0;

cout << "Enter x" << endl;

cin >> x;

cout << "Enter y" << endl;

cin >> y;

cout << "Enter z" << endl;

cin >> z;

cout << "Enter a" << endl;

cin >> a;

cout << "Enter b" << endl;

cin >> b;

cout << copyright() << endl;

cout << trueOrFalse(a, b) << endl;

cout << decimal(x, y, z) << endl;

cout << hexADecimal(x, y, z) << endl;

cout << s\_calculation\_to\_string(x, y, z) << endl;

return 0;

}

**Додаток Б**

**Модульне тестування функції s\_calculation()**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_8\_1 |
| Назва модуля  Name of Modules/ prj / ModulesDolynko | s\_calculation(); |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Долинко Антон |
| Виконавець  Implementer | Долинко Антон |

|  |  |  |
| --- | --- | --- |
| № | Значення | Очікуваний результат |
| 1 | X = 120 ;  Y = 2;  Z = 1000; | 8.84 |
| 2 | X = 30;  Y = 5;  Z = 60; | 2.10 |
| 3 | X = 90;  Y = -2;  Z = 6; | 0.11 |
| 4 | X = 7;  Y = -10;  Z = 89; | 408.25 |
| 5 | X = 10;  Y = 4;  Z = 1000; | 180.64 |
| 6 | X = 10;  Y = 4;  Z = -1000; | -19.36 |
| 7 | X = 1;  Y = 0;  Z = 1; | 2.00 |

**Результати тестування**

Test is passed

Test is passed

Test is passed

Test is passed

Test is passed

Test is passed

Test is passed

Process returned 0 (0x0) execution time : 0.009 s

Press any key to continue.

**Системне тестування Dolynko\_task.exe**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_8\_2 |
| Назва проекта / ПЗ  Name of Project / Software | Dolynko\_task.exe |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Долинко Антон |
| Виконавець  Implementer | Долинко Антон |

|  |  |  |  |
| --- | --- | --- | --- |
| 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) |
| TS-1 | 1. Введіть 12 2. Введіть 2 3. Введіть 24 4. Введіть 12.5 5. Введіть 12.6 | Enter x  12  Enter y  2  Enter z  24  Enter a  12.5  Enter b  12.6  Dolynko Anton ©  a + 1 > |b - 2| ? true  x = 12, y = 2, z = 24  x = C, y = 2, z = 18  (3 \* sin \* √(12 \* 12 + lg(12 - 3)))^2 + 24/12 = 3.883643 | passed |
| TS-2 | 1. Введіть -12 2. Введіть 2 3. Введіть 10 4. Введіть 8 5. Введіть 13 | Enter x  -12  Enter y  2  Enter z  10  Enter a  8  Enter b  13  Dolynko Anton ©  a + 1 > |b - 2| ? false  x = -12, y = 2, z = 10  x = $, y = 2, z = A  (3 \* sin \* √(12 \* -12 + lg(-12 - 3)))^2 + 10/-12 = nan | passed |
| TS-3 | 1. Введіть 40 2. Введіть 5 3. Введіть 80 4. Введіть 12 5. Введіть 6 | Enter x  40  Enter y  5  Enter z  80  Enter a  12  Enter b  6  Dolynko Anton ©  a + 1 > |b - 2| ? true  x = 40, y = 5, z = 80  x = 28, y = 5, z = 50  (3 \* sin \* √(12 \* 40 + lg(40 - 3)))^5 + 80/40 = 2.000000 | passed |
| TS-4 | 1. Введіть 12 2. Введіть -10 3. Введіть 48 4. Введіть -69 5. Введіть 69 | Enter x  12  Enter y  -10  Enter z  48  Enter a  -69  Enter b  69  Dolynko Anton ©  a + 1 > |b - 2| ? false  x = 12, y = -10, z = 48  x = C, y = &, z = 30  (3 \* sin \* √(12 \* 12 + lg(12 - 3)))^-10 + 48/12 = 4.042171 | passed |
| TS-5 | 1. Введіть 1 2. Введіть 0 3. Введіть 0 4. Введіть 69 5. Введіть -666 | Enter x  1  Enter y  0  Enter z  0  Enter a  69  Enter b  -666  Dolynko Anton ©  a + 1 > |b - 2| ? false  x = 1, y = 0, z = 0  x = 1, y = , z =  (3 \* sin \* √(12 \* 1 + lg(1 - 3)))^0 + 0/1 = 1.000000 | passed |