Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Звіт

З лабораторної роботи № 8

з дисципліни “Базові методології та технології проектування ”

на тему

“Реалізація статичних бібліотек модулів

лінійних обчислювальних процесів”

Виконав

студент академічної групи КІ-20

Небесний А.В.

Перевірив

Викладач

Поліщук Л.І.

Кропивницький – 2021

**Зміст**

1. Мета роботи – с. 3
2. Завдання 8.1 – с. 4
3. Завдання 8.1 – с. 5
4. Висновки – с. 6
5. Додатки – с. 8

**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 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 до лабораторної роботи**

1. Реалізувати статичну бібліотеку модулів libModulesПрізвищеC/C++, яка містить функцію розв’язування задачі 8.1.

2. Реалізувати програмне забезпечення розв’язування задачі 8.2 — консольний застосунок.

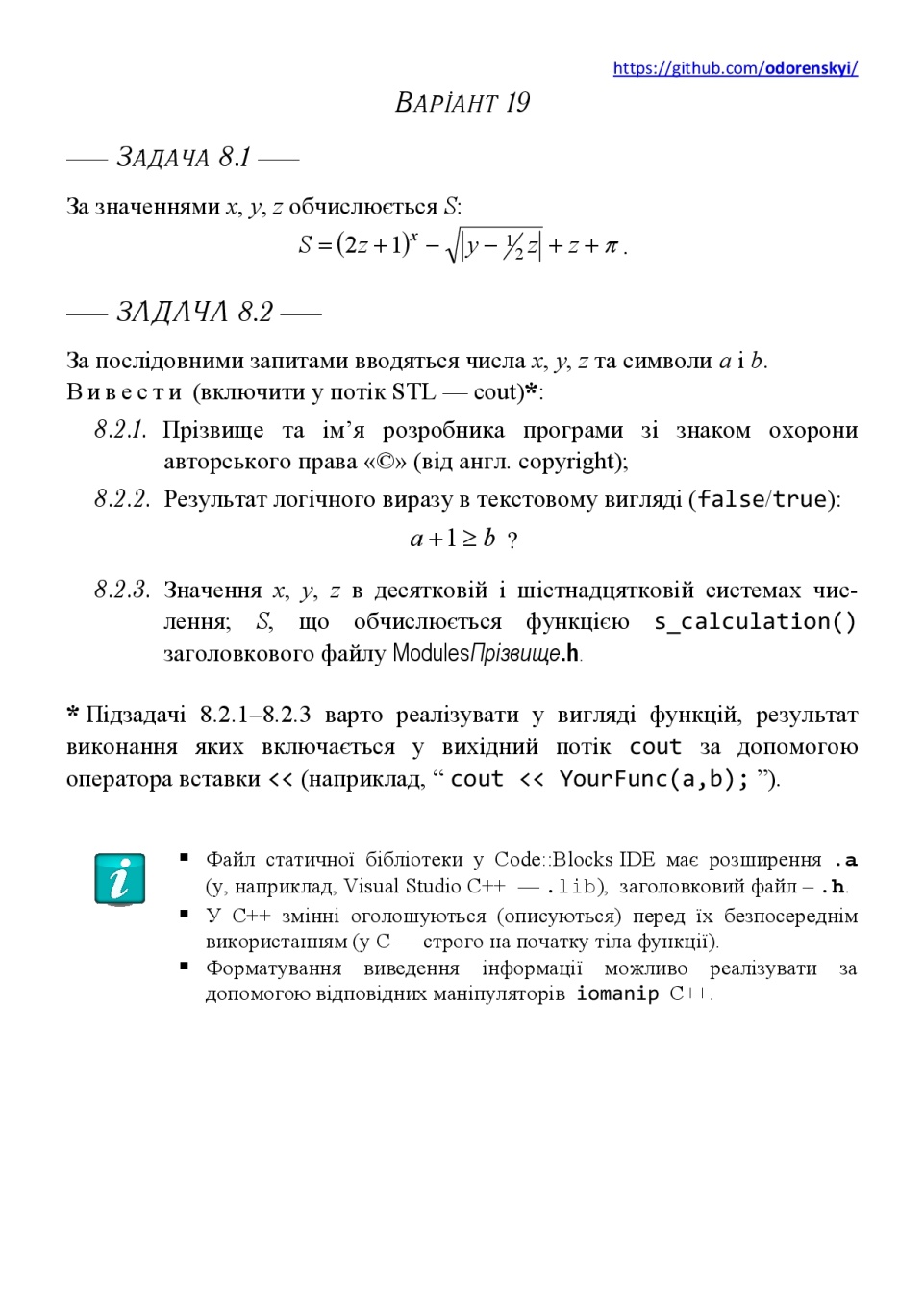


Рисунок 8.1 – Завдання з лабораторної роботи № 8

**Задача 8.1**

**Постановка задачі:**

Вихідні дані:

1. S – змінна типу float, що є результатом виконання виразу на рис 8.1

Вхідні дані:

1. x – змінна типу float
2. y – змінна типу float
3. z – змінна типу float

**Математична модель:**

**Алгоритм модуля:**

1. Вивести інформацію про за стосунок та розробника
2. Запитати х
3. Отримати х
4. Запитати у
5. Отримати у
6. Запитати z
7. Отримати z
8. Обчислити S
9. Вивести S

Даний модуль отримує дані у вигляді 3-х чисел, що передаються до нього. Він вводить дані числа в формулу та отримує значення S. Далі модуль повертає значення S.

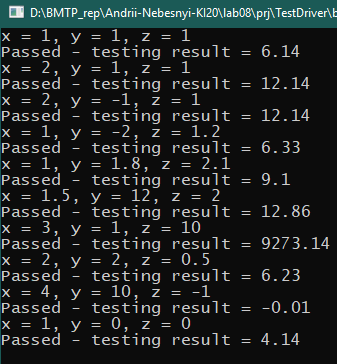
**

Рисунок 8.2 – Результати тестування модуля ModulesNebesnyi

**Задача 8.2**

**Постановка задачі:**

Вихідні дані:

S – змінна типу float, що є результатом виконання функції s\_calculation()

Вхідні дані:

x – змінна типу float, визначається користувачем

y – змінна типу float, визначається користувачем

z – змінна типу float, визначається користувачем

a – змінна типу char

b – змінна типу char

**Вимоги до програмного забезпечення:**

1. Мовний інтерфейс – українська
2. Наявність повідомлення з призначенням додатку
3. Повідомлення з інформацією про розробника додатку зі символом захисту інформації

**Математична модель:**

1. Якщо a+1 > b = {true}

**Алгоритмізація:**

Процедура №1 – виведення інформації про розробника

1. Виведення псевдографіки, що починає таблицю
2. Виведення інформації про розробника з використанням псевдографіки
3. Виведення псевдографіки для закриття таблиці, містить символ (с)

Процедура №2 – виконання логічної операції

1. Отримати значення символів а та b
2. Якщо a + 1 > b, то вивести true
3. Якщо а + 1 < b, то вивести false

Процедура №3 – виконання дій над числами

1. Отримати значення x, y, z
2. Вивести x, y, z у десятковій та шістнадцятковій системах числення
3. Провести виконання функції s\_calculation()
4. Вивести S

Процедура №4 – функція main

1. Викликати процедуру №1
2. Вивести інформацію про за стосунок
3. Запитати та отримати значення змінних x, y, z, a, b
4. Викликати процедуру №2
5. Викликати процедуру №3

**Висновки**

Нещодавно нам випала можливість виконати лабораторну роботу №8. Це перша лабораторна робота, в якій студентам надалася можливість написати програмне забезпечення мовою програмування С++. Метою даної лабораторної роботи є навчити студента створювати статичні бібліотеки та їх заготовочні файли, вміти їх підключати та використовувати на практиці, тестувати модулі за допомогою тестових драйверів. В якості IDE використовувалося програмне забезпечення вільного кросплатформового програмування Code::Blocks.

Метою першого завдання було навчити студента створювати статичні бібліотеки та проводити їх тестування за допомогою Test-Unit. Спершу було проаналізовано задачу, представлено принцип роботи. Далі було проведено строгу постановку задачі та математичну модель. В якості вхідних та вихідних даних було вирішено взяти змінні типу float, так як в виразі зустрілося число Пі, що не є цілим. Далі було представлено принцип роботи модуля й розроблено алгоритм роботи майбутньої статичної бібліотеки. Після проведення алгоритмізації було створено 10 тест-кейсів. Після проведення аналізу задачі та проектування майбутнього проекту, було реалізовано проект статичної бібліотеки та її заголовкового файлу. Наступним кроком було проведення тестування отриманого модуля. Для цього було створено проект консольного застосунку TestDriver, що використовував дані з тест-кейсів та автоматично передавав їх до модуля. Перше тестування не було успішним, так як в тест-кейсах було вказано числа з двома знаками після коми у якості вихідних даних (модуль видавав повні значення). Було проведено перебудову статичної бібліотеки, реалізовано зменшення числа після обчислень до двох знаків після коми. Після від лагодження проекту статичної бібліотеки, тестування було проведено повторно. Усі тест-кейси пройдено успішно.

Метою задачі 8.2 було навчити студента використовувати створені власноруч модулі на практиці. Також це завдання надало можливість навчитися користуватися маніпуляторами С++. Як завжди, першим кроком був аналіз задачі та проведення строгої постановки задачі. Було виділено 3 змінні типу float та 2 змінні типу char. Далі було побудовано математичну модель та проведено алгоритмізацію задачі 8.2. Майбутній код було вирішено розбити на функції, тому а алгоритмі було виділено 4 процедури:

1. Процедура №1 – відображає інформацію про розробника, окреслює виведену інформацію рамкою (використовує псевдографіку.) Разом з інформацією виводиться символ захисту розумової власності.
2. Процедура №2 – отримує два символи, у якості вхідних даних та проводить над ними логічну операцію a+1>b. Виводиться true/false залежно від розміру коду введених символів.
3. Процедура №3 – отримує три раціональних числа, проводить над ними операції переходу з однієї системи числення в іншу, точніше з десяткової в шістнадцятирічну. Також дана процедура повинна викликати функцію s\_calculation та вивести значення S залежно від x, y, z.
4. Процедура №4 є головною, викликає інші процедури. В ній здійснюється введення інформації користувачем та проводиться виведення інформації про застосунок.

Наступним кроком була побудова тест-кейсів для проведення системного тестування майбутнього продукту. Усього було створено 5 тестових випадків. Далі було реалізовано програмне забезпечення з використанням результатів аналізу та проектування. Отриманий програмний продукт було протестовано. Усі тестові артефакти було пройдено успішно.

Дана лабораторна робота була досить корисною для студент. Вона навчила створювати статичні бібліотеки мовою програмування С++ та підключати їх до інших проектів. Також вона дала змогу зрозуміти, що таке маніпулятори та як їх правильно використовувати.

Під час тестування отриманих програмних продуктів виникла невелика проблема. Під час запуску файлів з розширенням ехе, виникали помилки, в контексті яких було описано нестачу динамічних бібліотек для повноцінної роботи програми. Проблему було вирішено шляхом встановлення необхідних динамічних бібліотек в теку з програмою, але дане рішення не було правильним. Пізніше було знайдено спосіб підключити необхідні бібліотеки до проекту під час компіляції, але деякі з них були відсутні. Тому вони збереглися в теці виконавчих файлів для підтримання їх працездатності.

**Додатки**

**Додаток А.1 – Тест-кейси для проекту ModulesNebesnyi**

|  |  |
| --- | --- |
| Назва тестового набору | TS8.1\_LAB\_8\_BMTP |
| Назва проекта | ModulesNebesnyi.a |
| Рівень тестування | Модульне тестування |
| Автор тест-сьюта | Небесний А.В. |

|  |  |  |
| --- | --- | --- |
| Номер | Вхідні дані | Очікуваний результат |
| TestCase\_1 | x = 1 , y = 1, z = 1 | S = 6,14 |
| TestCase\_2 | x = 2 , y = 1, z = 1 | S = 12,14 |
| TestCase\_3 | x = 2 , y = -1, z = 1 | S = 12,14 |
| TestCase\_4 | x = 1 , y = -2, z = 1,2 | S = 6,33 |
| TestCase\_5 | x = 1 , y = 1,8, z = 2,1 | S = 9,10 |
| TestCase\_6 | x = 1,5 , y = 12, z = 2 | S = 12,86 |
| TestCase\_7 | x = 3 , y = 1, z = 10 | S = 9273,14 |
| TestCase\_8 | x = 2 , y = 2, z = 0,5 | S = 6,23 |
| TestCase\_9 | x = 4 , y = 10, z = -1 | S = -0,01 |
| TestCase\_10 | x = 1 , y = 0, z = 0 | S = 4,14 |

**Додаток А.2 – Код модуля (статичної бібліотеки) ModulesNebesnyi.a**

#include <cmath>

#include <iostream>

#include "ModulesNebesnyi.h"

using namespace std;

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

float S = pow((2\*z + 1), x) - sqrt(abs(y - 1/2 \* z)) + 3.14 + z;

S = int(S\*100 + 0.5)/100.0; // two symbols after the coma

return S;

}

**Додаток А.3 – Код тестового драйвера TestDrive.exe**

#include <iostream>

#include "ModulesNebesnyi.h"

using namespace std;

int main()

{

float x[10] = {1, 2, 2, 1, 1, 1.5, 3, 2, 4, 1};

float y[10] = {1, 1, -1, -2, 1.8, 12, 1, 2, 10, 0};

float z[10] = {1, 1, 1, 1.2, 2.1, 2, 10, 0.5, -1, 0};

float S[10] = {6.14, 12.14, 12.14, 6.33, 9.10, 12.86, 9273.14, 6.23, -0.01, 4.14};

for(int i = 0; i < 10; i++){

if(s\_calculation(x[i], y[i], z[i]) == S[i]){

cout << "x = " << x[i] << ", y = " << y[i] << ", z = " << z[i] << endl;

cout << "Passed - testing result = " << S[i] << endl;

}

else{

cout << "Testing result - " << s\_calculation(x[i], y[i], z[i]) << endl;

cout << "Failed - testing result != " << S[i] << endl;

}

}

return 0;

}

**Додаток В.1 – Тест-сьют для системного тестування за стосунку Nebesnyi-task.exe**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS8.2\_BMTP-lab8 |
| Назва проекта / ПЗ  Name of Project / Software | Nebesnyi-tasl.exe |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Небесний А.В. |
| Виконавець  Implementer | Небесний А.В. |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TestCase\_1 | - | Вивести інформацію про розробника з використанням псевдографіки та символом (с).  Вивести інформацію про розробника  Запитати користувача ввести x, y, z, a, b | Passed |
| TestCase\_2 | 1. x = 1, y = 1, z = 1, a = a, b = z | …  Повідомлення з виразом a + 1 > b з результатом false.  Повідомлення (зі значеннями x, y, z у десятковій та шістнадцятко вій системах числення )  1 = 0x8p-3  1 = 0x8p-3  1 = 0x8p-3  Повідомлення зі значенням S = 6.14 | Passed |
| TestCase\_3 | 1. x = 1.5 , y = 12 , z = 2 , a = b , b = a | …  Повідомлення з виразом a + 1 > b з результатом true.  Повідомлення (зі значеннями x, y, z у десятковій та шістнадцятко вій системах числення )  1.5 = 0xcp-3  12 = 0xcp+0  2 = 0x8p-2  Повідомлення зі значенням S = 12.86 | Passed |
| TestCase\_4 | 1. x = 3 , y = 1 , z = 10 , a = f , b = a | …  Повідомлення з виразом a + 1 > b з результатом true.  Повідомлення (зі значеннями x, y, z у десятковій та шістнадцятко вій системах числення )  3 = 0xcp-2  1 = 0x8p-3  10 = 0xap+0  Повідомлення зі значенням S = 9273.14 | Passed |
| TestCase\_5 | 1. x = 1 , y = 0 , z = 0 , a = а , b = к | …  Повідомлення з виразом a + 1 > b з результатом false.  Повідомлення (зі значеннями x, y, z у десятковій та шістнадцятко вій системах числення )  1 = 0x8p-3  0 = 0x0p+0  0 = 0x0p+0  Повідомлення зі значенням S = 4.14 | Passed |

**Додаток В.2 – Код програмного забезпечення Nebesnyi-task.exe**

#include <iostream>

#include <iomanip>

#include "ModulesNebesnyi.h"

using namespace std;

void dev\_information()

{

cout << " ---------------------------------------------------------------------------- " << endl;

cout << "| Даний застосунок розроблено студентом центральноукраїнського національного |" << endl;

cout << "| технічного університету Небесним Андрієм Вадимовичем 24 березня 2021 року. |" << endl;

cout << " ---------------------------©ALL\_RIGHTS\_ARE\_RESERVED-------------------------" << endl;

}

int bool\_operation(char a, char b)

{

cout << "Результат виконання логічної операції a+1>b ---> " << boolalpha << bool(a + 1 > b) << endl;

}

float num\_operations(float x, float y, float z, float S)

{

cout << fixed << setprecision(2) << "x = " << x << "->" << hexfloat << x << endl;

cout << fixed << setprecision(2) << "y = " << y << "->" << hexfloat << y << endl;

cout << fixed << setprecision(2) << "z = " << z << "->" << hexfloat << z << endl;

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

cout << fixed << setprecision(2) << "S = " << S << endl;

getchar();

}

int main()

{

system(chcp 1251 & cls);

char a, b;

float x = 0;

float y = 0;

float z = 0;

float S = 0;

dev\_information();

cout << " ---------------------------------------------------------------------------- " << endl;

cout << "Завдання даної програми - провести деякі логічні та арефметичні операції." << endl;

cout << "Користувач вводить два символи та три раціональні числа. Над символами " << endl;

cout << "проводиться логічна операція a+1>b, результатом якої є істина чи хиба." << endl;

cout << "Раціональні числа переводяться в шістнадцятирічну систему числення," << endl;

cout << "також виконується арифметичний вираз :" << endl;

cout << "S = (2\*z - 1)^x - √(|y - 0.5 \* z|) + z + π ." << endl;

cout << " ---------------------------------------------------------------------------- " << endl;

cout << "Введіть символ №а" << endl;

cin >> a;

cout << "Введіть символ №b" << endl;

cin >> b;

bool\_operation(a, b);

cout << " ---------------------------------------------------------------------------- " << endl;

cout << "Введіть х: ";

cin >> x;

cout << "Введіть y: ";

cin >> y;

cout << "Введіть z: ";

cin >> z;

num\_operations(x, y, z, S);

getchar();

return 0;

}