Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Звіт

Про виконання Лабораторої роботи №8

з дисципліни

“Базові методології та технології програмування”

на тему

«Реалізація статичних бібліотек модулів лінійних обчислювальних процесів»

Виконав

Студент академічної групи КН-22

Гончарук О.С.

Перевірив

викладач

Собінов О.Г

Кропивницький – 2022

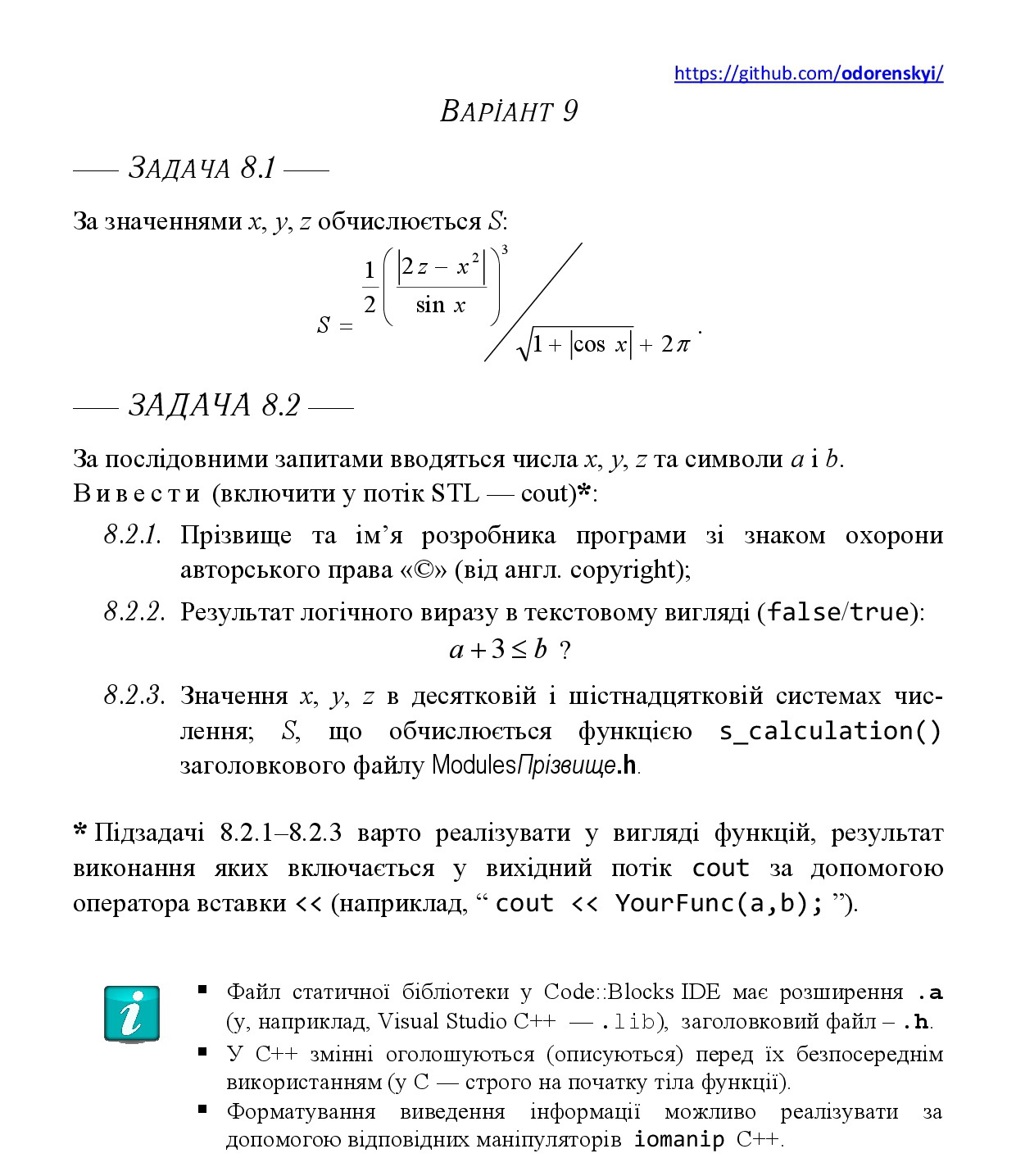
Мета роботи: Набуття ґрунтовних вмінь і практичних навичок застосування теоретичних положень методології модульного програмування, реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування С++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, розроблення програмних модулів та засобів у кросплатформовому середовищі Code::Blocks (GNU GCC Compiler).

Завдання 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.

Варіант 9



Хід роботи

Завдання 1

Завдання 8.1 Здійснено аналіз і постановку задачі 8.1. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного модуля розв’язування задачі 8.1. Алгоритмізувавши задачу отримаємо:

Вхідні данні: x, y, z.

Вихідні данні: S.

Обмеження, допущення: x, y, z, s — Дійсні числа

Блок схема

Початок

Введення

x, y, z

S = (1.0/2.0)\*pow(fabs(2\*z-pow(x,2))/sin(x),3)/sqrt(1+fabs(cos(x))+2\*3.14\*y)

Виведення

S

Кінець

Лістинг задачі 8.1

#include <iostream>

#include <math.h>

#include "ModulesGoncharuk.h"

#include <windows.h>

using namespace std;

int main()

{

SoftwareDeveloper();

double x, y, z, S;

cout << "Введіть значення для x, y, та z, натискаючи Enter: ";

cin >> x >> y >> z;

S = (1.0/2.0)\*pow(fabs(2\*z-pow(x,2))/sin(x),3)/sqrt(1+fabs(cos(x))+2\*3.14\*y);

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

system("pause");

return 0;

}

Завдання 8.2

Здійснено аналіз і постановку задачі 8.2. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного забезпечення розв’язування задачі 8.2. Алгоритмізувавши задачу отримаємо:

Вхідні данні: x, y, z, a, b

Вихідні данні: анотація розробника(8.2.1), результат логічного виразу(8.2.2), x, y, z в десятковій та шістнадцятковій системі числення та значення s.

Блок схема

Початок

float x, y, z,

char a, b

Введення

x, y, z, a, b

Виведення результатів функцій

System

(“pause”)

Кінець

Лістинг задачі 8.2

#include <iostream>

#include <math.h>

#include "ModulesGoncharuk.h"

#include <windows.h>

using namespace std;

int main()

{

SoftwareDeveloper();

int a, b;

std::cout << "Введіть значення для a: ";

std::cin >> a;

std::cout << "Введіть значення для b: ";

std::cin >> b;

if (a + 3 <= b) {

std::cout << "Результат: a+3<=b is True." << endl;

}

else {

std::cout << "Результат: a+3<=b is False." << endl;

}

double x, y, z, S

cout << "Введіть значення для x, y, та z, натискаючи Enter: ";

cin >> x >> y >> z;

S = (1.0/2.0)\*pow(fabs(2\*z-pow(x,2))/sin(x),3)/sqrt(1+fabs(cos(x))+2\*3.14\*y);

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

cout << "Введіть значення для x: ";

cin >> x;

cout << "Введіть значення для y: ";

cin >> y;

cout << "Введіть значення для z: ";

cin >> z;

cout << endl;

cout << "Десяткове значення x: " << x << " | " << hex << x << " (Шістнадцяткове значення)" << endl;

cout << "Десяткове значення y: " << y << " | " << hex << y << " (Шістнадцяткове значення)" << endl;

cout << "Десяткове значення z: " << z << " | " << hex << z << " (Шістнадцяткове значення)" << endl;

system("pause");

return 0;

}

Додатки

Додаток 1-TestSuite8.1

|  |  |
| --- | --- |
| Назва тестового набору | TestSuite8.1 |
| Рівень тестування | Модульний(Unit-Testing) |
| Автор | Гончарук Олександр Святославович |
| Виконавець | Гончарук Олександр Святославович |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Введені дані | Виведені дані | Результат |
| 1 | 3, 4, 5 | S = 34.1696 | passed |
| 2 | 4, 32, 6 | S = -5.18642 | passed |
| 3 | 34, 43, 343 | S = 2.12568e+07 | passed |

Додаток 2-TestSuite8.2

|  |  |
| --- | --- |
| Назва тестового набору | TestSuite8.2 |
| Рівень тестування | Модульний(Unit-Testing) |
| Автор | Гончарук Олександр Святославович |
| Виконавець | Гончарук Олександр Святославович |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Дії | Очікуваний результат | Результат |
| 1 | Запустити програму | Введіть значення для a: | passed |
| 2 | Ввести значення а: 4 | Введіть значення для b: | passed |
| 3 | Ввести значення b: 3 | Введіть значення для х: | passed |
| 4 | Ввести значення x: 4 | Введіть значення для y: | passed |
| 5 | Ввести значення y: 3 | Введіть значення для z: | passed |
| 6 | Ввести значення z: 7 | Введіть значення для х: | passed |
| 7 | Ввести значення x: 3 | Введіть значення для y: | passed |
| 8 | Ввести значення y: 2 | Введіть значення для z: | passed |
| 9 | Ввести значення z: 1 | Гончарук Олександр ©  a+3<=b is False  Десяткове значення х: 4 | Шістнадцяткове значення х: 4  Десяткове значення у: 3 | Шістнадцяткове значення у: 3  Десяткове значення z: 7 | Шістнадцяткове значення z: 7  Результат: S = 15998 | passed |

Додаток 3-Лістинг вихідного коду ModulesGoncharuk

#include <iostream>

#include <windows.h>

void SoftwareDeveloper() {

using namespace std;

system(“chcp 1251”)

setlocale(LC\_ALL, "ukr");

cout << "\n ------------------------------------------------\n"

<< " | Oleksandr Goncharuk |"

<< " | Олександр Гончарук | "

<< "\n ----------- **©** All Rights Reserved ---------------\n\n\n\n";

}

int main(){

SoftwareDeveloper();

system(“pause”);

return 0;

}

Додаток 4-Лістинг вихідного коду TestDriver

#include

#include

"ModulesGoncharuk.h"

using namespace std;

int main()

{

float test\_input[5][3] = {{4, 3}, {4, 3, 7}, {9, 7, 2}, {3, 3, 3}, {2, 5, 43}};

float test\_output[5] = {3.51, 4.5532, 6.00, 3.00, 16.667};

for (int i=0;

i<< "X must be more then Y";

}

else {

float s = s\_calculation(test\_input[i][0], test\_input[i][1], test\_input[i][2]);

if(s == test\_output[i]){ cout << "Test #" << i+1 << " PASSED" << endl;

}

else{ cout << "Test #" << i+1 << " FAILED" << endl;

}

}

}

}

Висновок

Завантажений власний Git-репозиторій https://github.com/odorenskyi/GoncharukOleksandr-KN22. У \Lab8 заповнено файл README.md, створено теки prj, Software, TestSuite, Report; отриманий вміст теки \Lab8 завантажено до Git-репозиторію. Здійснено аналіз і постановку задачі 8.1. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного модуля розв’язування задачі 8.1. Розроблено набір контрольних прикладів до задачі 8.1 задля виконання модульного тестування (Unit testing) модулів С++. В Code::Blocks IDE створено проект статичної бібліотеки ModulesChervonyi, збережено у \Lab8\prj. На основі результатів проектування модуля, реалізовано мовою програмування С++ функцію s\_calculation, яка за належним інтерфейсом реалізовує розв’язування задачі 8.1. Скомпільовано. В Code::Blocks IDE створено проект заголовкового файлу ModulesGoncharuk в \Lab8\prj та описано в ньому прототип функції s\_calculation. В Code::Blocks IDE у \prj створено проект консольного додатка С++, під назвою TestDriver. Реалізовано тестовий драйвер для виконання розроблених тестових наборів (\Lab8\TestSuite) і за його допомогою виконано модульне тестування функції s\_calculation зі статичної бібліотеки libModulesGoncharuk.a. Здійснено аналіз і постановку задачі 8.2. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного забезпечення розв’язування задачі 8.2. Розроблено тест-сьют для виконання системного тестування ПЗ розв’язування задачі 8.2.В Code::Blocks IDE створено проект консольного додатка Goncharuk\_task у теці \prj. Мовою програмування С++ реалізовано результати проектування програмного забезпечення розв’язування задачі 8.2.Консольний додаток Goncharuk task.ехе скопійовано у \Software. В ході роботи над лабораторною роботою набув грунтових вмінь та 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.