Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 2

з навчальної дисципліни “Модульне програмування”

РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ВИКОНАВ

студент академічної групи КІ-17

\_\_\_\_\_\_\_\_\_\_ В.О. Дощенко

ПЕРЕВІРИВ

старший викладач кафедри кібербезпеки та програмного забезпечення

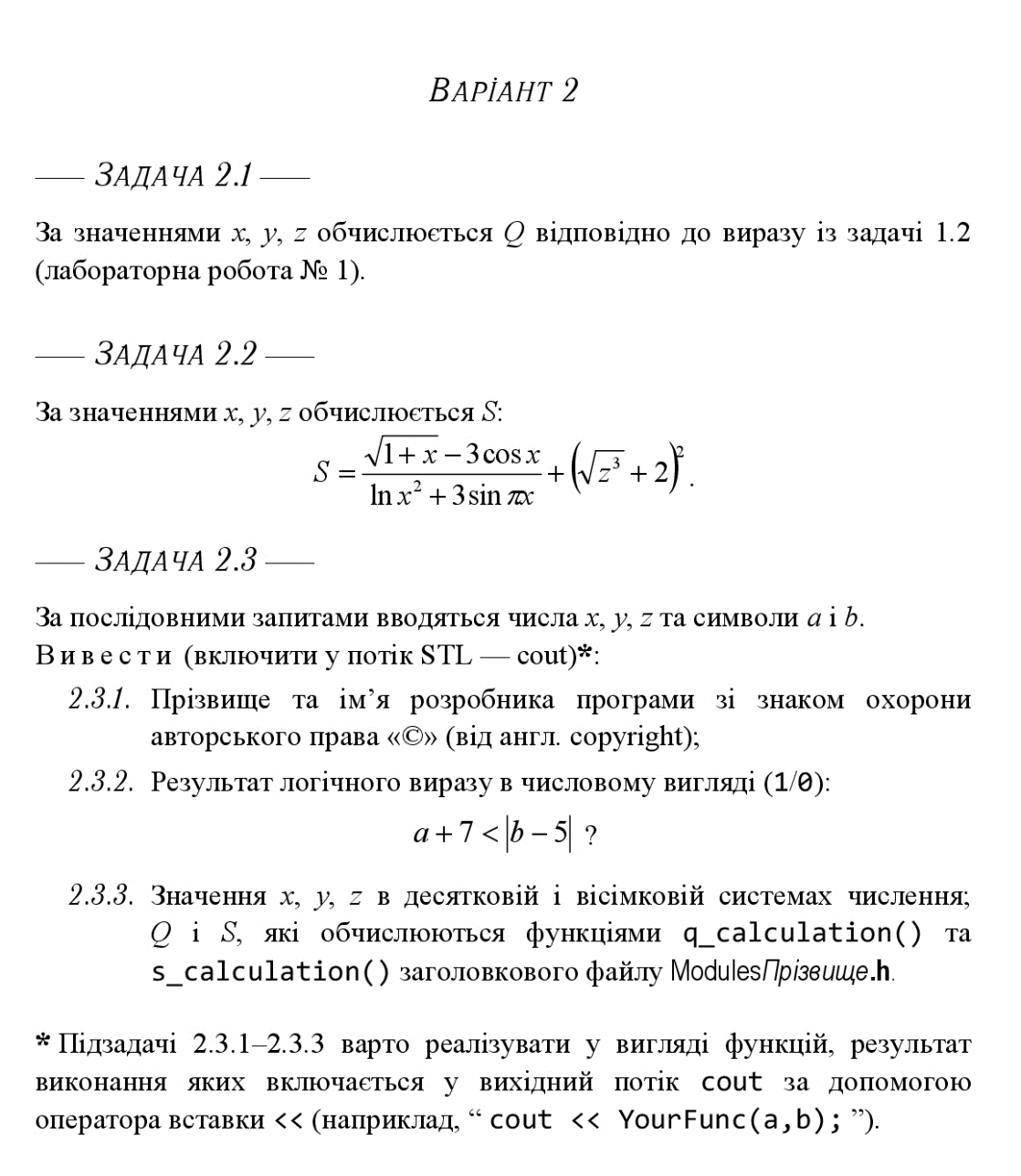
\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.І. Поліщук

Кропивницький – 2018

**Мета лабораторної 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/С++ даних скалярних типів, арифметичних і логічних операцій, потокового введення і виведення інформації, розроблення програмних модулів та засобів у кросплатформовому середовищі Code::Blocks (GNU GCC Compiler).

**Завдання до лабораторної роботи**

1. Реалізувати статичну бібліотеку модулів libModulesDoshchenko С/С++, яка містить функції розв’язування задач 2.1 та 2.2.
2. Реалізувати програмне забезпечення розв’язування задачі 2.3 – консольний додаток.

**Задача 2.1**

Аналіз задачі:

Дана задача має виконувати наступні функції:

-При введенні значення x,y,z виконується обчислення значення Q за формулою:

Строга постановка задачі:

Вихідні дані: змінна result;

Вхідні дані: змінні x,y,z;

Обмеження та допущення: уся введена і виведена інформація є типом Float; Змінна z не може дорівнювати нулю.

Аналіз вимог до програмного забезпечення:

- Модуль перевіряє вхідні дані на валідність.

- Функція повинна обчислювати формулу:

Проектування архітектури програмного забезпечення:

Даний модуль включає в себе перевіряючу частину, а також робочу частину.

Детальне проектування програмного забезпечення:

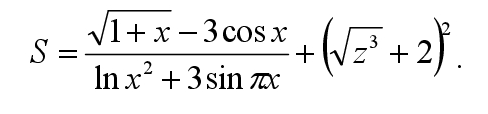
Перевіряюча чистина має виконувати перевірку вхідних даних на валідність.

Робоча частина має виконувати необхідні обчислення.

**Задача 2.2**

Аналіз задачі:

Дана задача має виконувати наступні функції:

-При введенні значення x,y,z виконується обчислення значення S за формулою:

Строга постановка задачі:

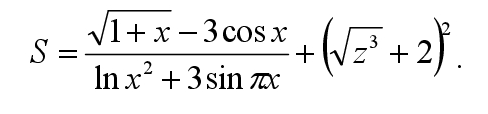
Вихідні дані: змінна result1;

Вхідні дані: змінні x,y,z;

Обмеження та допущення: уся введена і виведена інформація є типом Float; Змінна z та х не можуть бути меншими за нуль, змінна х не може дорівнювати нулю.

Аналіз вимог до програмного забезпечення:

- Модуль перевіряє вхідні дані на валідність.

- Функція повинна обчислювати формулу:

Проектування архітектури програмного забезпечення:

Даний модуль включає в себе перевіряючу частину, а також робочу частину.

Детальне проектування програмного забезпечення:

Перевіряюча чистина має виконувати перевірку вхідних даних на валідність.

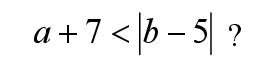
Робоча частина має виконувати необхідні обчислення.

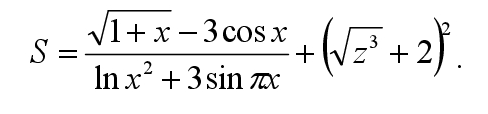
**Задача 2.3**

Аналіз задачі:

Дана задача має виконувати наступні функції:

-Прізвище, ім’я, по-батькові розробника, а також знак охорони авторського права.

- При введені параметрів a та b обчислюється логічний вираз:

 -При введенні значення x,y,z виконується обчислення значення S за формулами:

Строга постановка задачі:

Вихідні дані: потокове виведення значення логічного виразу, а також результатів обчислення задач 2.1 та 2.2;

Вхідні дані: змінні x,y,z;

Обмеження та допущення: відсутні.

Аналіз вимог до програмного забезпечення:

- Програма має містити інформацію про автора, а також знак захисту інформації.

-Програма повинна виводити результат логічного виразу у числовому форматі, а також результати обчислень функцій 2.1 та 2.2 у десятковій, шістнадцядковій, та вісімковій системах числення.

Проектування архітектури програмного забезпечення:

## Дана програма включає в себе робочу частину,підключений модуль з функціями q\_calcunation і s\_calculation, а також перевіряючу частину.

Детальне проектування програмного забезпечення:

Вихідний код умовно розділений на три частини(вивід інформації про розробника, вивід результату обчислення логічного виразу та вивід результатів обчислення задач 2.1 та 2.2).

Вихідний код проекта Doshchenko\_task:

#include <iostream>

#include <windows.h>

#include <string>

#include "ModulesDoshchenko.h"

#include <stdio.h>

#include <stdlib.h>

#include <cstring>

#include <cmath>

#include <iomanip>

using namespace std;

bool Logic(double a, double b){

bool c = false;

if ((a+7) < (abs(b+5))) {

c = true;}

else

c = false;

return c;

}

int main()

{

setlocale(LC\_CTYPE,"ukr");

cout << "Vladislav Doshchenko (c)" << endl;

int sw;

cout << "Натиснувши 1 ви отримаєте логiчний результат: a+7<|b-5|!" << endl << "Натиснувши 2 ви отримаєте розрахунки попереднiх задач!" << endl;

cin >> sw;

switch (sw) {

case 1: {

double a,b;

bool c;

cout << "Введiть перший параметр\n";

while (!(cin >> a)){

cout << "Некоректнi данi!\n";

cin.clear();

fflush(stdin);}

cout << "Введiть другий переметр\n";

while (!(cin >> b)){

cout << "Некоректнi данi!\n";

cin.clear();

fflush(stdin);}

cout << "Логiчний результат: ";

if (Logic(a,b)==true){

cout << 1 << endl;

}

if (Logic(a,b)==false){

cout << 0 << endl;

}

break;}

case 2: {

double x, y, z;

cout << "Введiть X (при вводi дробу використовуйте КРАПКУ): ";

while (!(cin >> x)){

cout << "Некоректнi данi!\n";

cin.clear();

fflush(stdin);}

cout << "Введiть Y (при вводi дробу використовуйте КРАПКУ): ";

while (!(cin >> y)){

cout << "Некоректнi данi!\n";

cin.clear();

fflush(stdin);}

cout << "Введiть z(при вводi дробу використовуйте КРАПКУ): ";

while (!(cin >> z)){

cout << "Некоректнi данi!\n";

cin.clear();

fflush(stdin);}

if (valid2(z,x)==1){

cout<<"Змiннi z та x не можуть дорiвнювати нулю! Змiннi z та x не можуть бути меншими за нуль!"<<endl;

system("pause");

return 0;

}

if (valid3(x,y,z) == 0)

{

cout<<"Обчислення неможливi,введiть менше значення!"<<endl;;

system("pause");

return 0;

}

cout<<"Q= " << q\_calculation(x,y,z)<<endl;

cout<<"S= " << fixed<<(s\_calculation(x,y,z))<<endl;

int a, b, c,l;

cout << "Надалi будуть використовуватися цiлi значення" << endl;

c = round(q\_calculation(x,y,z));

l = round(s\_calculation(x,y,z));

a = int(c);

b = int(l);

cout << "Числа у вiсiмковiй системi числення:" << endl;

cout << std::oct << "Q = " << a << " S = " << b << endl;

cout << "Числа у шiстнадцятковiй системi числення:" << endl;

cout << std::hex << "Q = " << a << " S = " << b << endl;

break;}

}

system("pause");

}

**Висновки**

Перед тим, як виконувати лабораторну роботу №2 були оброблені теоретичні відомості, підготовані відповіді на контрольні запитання, та опрацьована самостійна підготовка до виконання лабораторної роботи №1, а також інформація подану на лекції.

Першою проблемою стала робота зі статичними бібліотеками, а саме з їх налаштуваннями. Витративши трохи часу все ж таки налаштування були виконаними.

Надалі розпочалося виконання завдань лабораторної роботи. Спочатку був виконаний аналіз поставлених задач, були вирішені вихідні та вхідні дані, а також обмеження та допущення. Після виконання задач, було проведено мудульне, а потім і системне тестування.

У ході виконання лабораторної роботи були здобуті навички створювати та використовувати модулі на мові С++. Застосування модулів виявилося дуже зручним, адже їх можна використовувати багатократно у різних програмних засобах. Дана лабораторна робота подарувала безцінний досвід роботи з модулями.

Загалом, завдання до цієї лабораторної роботи були досить цікавими, адже вони провели ознайомлення з модульним підходом реалізації ПЗ, зауваження відсутні.

Додаток №1

Тест-сьют для модульного тестування

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_lab2\_Modules |
| Назва проекта / ПЗ  Name of Project / Software | libModulesDoshchenko.a |
| Рівень тестування  Level of Testing | модульне / Unit Testing |

| Ід-р тест-кейса | Вхідні дані | Очікуваний  результат | Вихідні дані | Результат тестування  (passed/failed/ blocked) |
| --- | --- | --- | --- | --- |
| Q#1 | 1; 1; 1. | -0.172604 | -0.172604 | **passed** |
| Q#2 | 1;2;1. | -0.236266 | -0.236266 | **passed** |
| Q#3 | 1.05; 1; 1. | -0.172604 | -0.172604 | **passed** |
| Q#4 | -1; 1; 1. | -0.172604 | -0.172604 | **passed** |

| Ід-р тест-кейса | Вхідні дані | Очікуваний  результат | Вихідні дані | Результат тестування  (passed/failed/ blocked) |
| --- | --- | --- | --- | --- |
| S#1 | 1; 1; 1. | -562611690377528.562500 | -562611690377528.562500 | **passed** |
| S#2 | 1; 2; 1. | -562611690377528.562500 | -562611690377528.562500 | **passed** |
| S#3 | 1.05; 1; 1. | 3.16392 | 3.16392 | **passed** |
| S#4 | 1; -1; 1. | -562611690377528.562500 | -562611690377528.562500 | **passed** |
| S#5 | 1;0;0. | -562611690377528.562500 | -562611690377528.562500 | **passed** |

Додаток №2

Результати виконання тестового драйвера

Task 1

TestCase #1 Passed

TestCase #2 Passed

TestCase #3 Passed

TestCase #4 Passed

Task 2

TestCase #1Passed

TestCase #2Passed

TestCase #3Passed

TestCase #4Passed

TestCase #5Passed

Додаток №3

Лістинг тестового драйвера

#include <iostream>

#include"ModulesDoshchenko.h"

#include<cmath>

#include<cstring>

using namespace std;

int main()

{

setlocale(LC\_ALL,"ukr");

double massiv[5];

massiv[0]=q\_calculation(1,1,1);

massiv[1]=q\_calculation(1,2,1);

massiv[2]=q\_calculation(1.05,1,1);

massiv[3]=q\_calculation(-1,1,1);

massiv[4]=q\_calculation(0,0,1);

cout << "Task 1" << endl;

if (massiv[0] == -0,168056462180752)

cout<<"TestCase #1"<<' '<<"Passed"<<endl;

if (massiv[2] == -0,167829097976335)

cout<<"TestCase #2"<<' '<<"Passed"<<endl;

if (massiv[3] == -0,177151030357432)

cout<<"TestCase #3"<<' '<<"Passed"<<endl;

if (massiv[4] == -0,199887450799131)

cout<<"TestCase #4"<<' '<<"Passed"<<endl;

cout << "Task 2" << endl;

double s1, s2, s3, s4, s5;

string tsa1,tsa2,tsa3,tsa4,tsa5;

s1 = s\_calculation(1, 1, 1);

s2 = s\_calculation(1, 2, 1);

s3 = s\_calculation(1.05, 1, 1);

s4 = s\_calculation(1, -1, 1);

s5 = s\_calculation(1, 0, 0);

tsa1 = (abs(abs(-562611690377528.562500) - abs(s1)) < 0.00000000000001 ? "Passed" : "Failed");

tsa2 = (abs(abs(-562611690377528.562500) - abs(s2)) < 0.00000000001 ? "Passed" : "Failed");

tsa3 = (abs(3.16392 - abs(s3)) < 0.0000000000001 ? "Passed" : "Failed");

tsa4 = (abs(abs(-562611690377528.562500) - abs(s4)) < 0.00000000000001 ? "Passed" : "Failed");

tsa5 = (abs(abs(-562611690377528.562500) - abs(s5)) < 0.0000000000000001 ? "Passed" : "Failed");

cout << "TestCase #1"<<tsa1<<endl;

cout << "TestCase #2"<<tsa1<<endl;

cout << "TestCase #3"<<tsa1<<endl;

cout << "TestCase #4"<<tsa1<<endl;

cout << "TestCase #5"<<tsa1<<endl;

return 0;

}

Додаток №4

Лістинг створеного модуля

#include<iostream>

#include<cstring>

#include<cmath>

#include<clocale>

#include<cstdlib>

using namespace std;

short valid3(double x,double y,double z)

{

if (x>100 && y>100 && z>100){

return 0;

}

if (x>400 || y>400 || z>400){

return 0;

}

return 1;

}

short valid2(double z, double x)

{

if (z <= 0)

return 1;

if (x<=0)

return 1;

return 2;

}

double q\_calculation (double x, double y, double z)

{

setlocale(LC\_CTYPE,"ukr");

double p,q,f,n,k;

k = 0;

const double pi = 3.1415926535897932384626433832795;

if ((log(abs(sin(z))))>pow(10,38)){

cout<<"Обчислення неможливi,введiть менше значення!";

return 0;

}

else p=log(abs(sin(z)));

if (((1/7\*pow(x,y))-(sqrt(abs(y-z)))) >pow(10,38)){

cout<<"Обчислення неможливi,введiть менше значення!";

return 0;

}

else q=(1/7\*pow(x,y))-(sqrt(abs(y-z)));

if((sqrt(abs(y-z))) >pow(10,38)){

cout<<"Обчислення неможливi,введiть менше значення!";

return 0;

}

else f =(sqrt(abs(y-z)));

if ((10\*pi\*pow(z,4)) >pow(10,38)){

cout<<"Обчислення неможливi,введiть менше значення!";

return 0;

}

else n =10\*pi\*pow(z,4);

if ((p+((q-f)/n)) >pow(4.54728408833987,2985)){

cout<<"Обчислення неможливi,введiть менше значення!";

return 0;

}

else k = (p+((q-f)/n));

return k;

}

double s\_calculation (double x, double y, double z)

{

setlocale(LC\_CTYPE,"ukr");

double k = 0;

double a,s,d,f,g;

const double pi = 3.1415926535897932384626433832795;

if (abs(((((sqrt(1+x)-3\*cos(x))/(log(x\*x)+3\*sin(pi\*x)))+((sqrt(pow(z,3)+2))\*(sqrt(pow(z,3)+2))))>pow(10,38))))

{

cout<<"Обчислення неможливi,введiть менше значення!";

return 0;

}

else k =((((sqrt(1+x))-(3\*cos(x)))/((log(x\*x))+(3\*sin(pi\*x))))+((sqrt(pow(z,3))+2)\*(sqrt(pow(z,3))+2)));

return k;

}