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

\_\_\_\_\_\_\_\_\_\_ С.О. Мельник

ПЕРЕВІРИВ

старший викладач кафедри кібербезпеки та програмного забезпечення

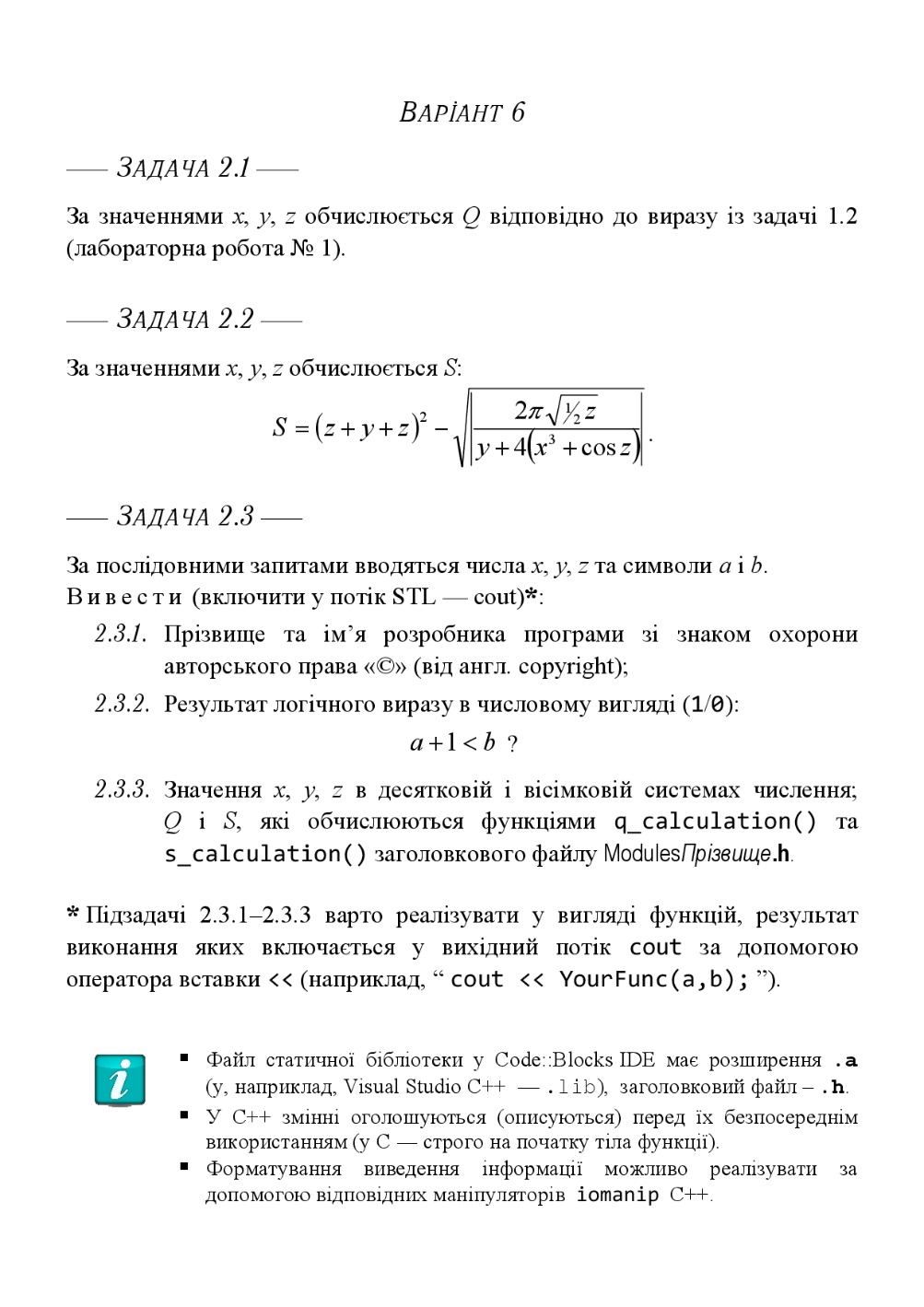
\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.І. Поліщук

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).

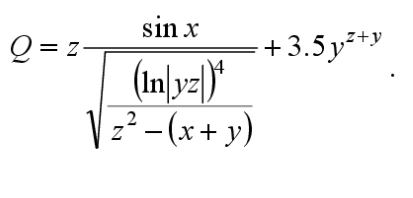
**Завдання до лабораторної роботи**

1. Реалізувати статичну бібліотеку модулів libModulesMelnyk 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:

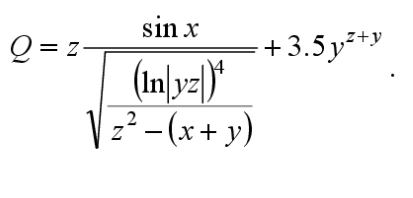
Строга постановка задачі:

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:

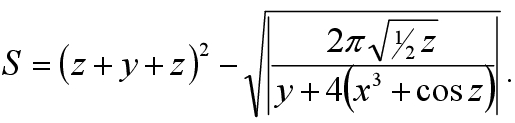
Перевіряюча чистина має виконувати перевірку вхідних даних на валідність.

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:

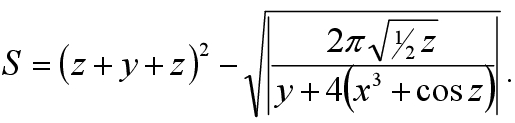
Вихідні дані: змінна result1;

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**

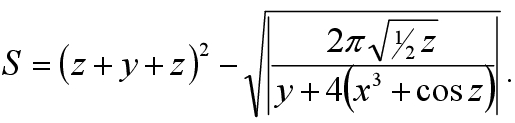
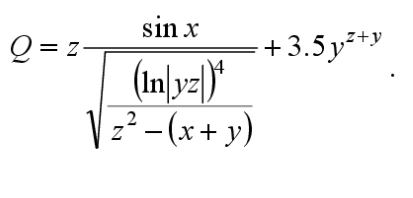
Аналіз задачі:

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 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 Melnyk\_task:

#include <iostream>

#include <windows.h>

#include <string>

#include <stdio.h>

#include <stdlib.h>

#include <cstring>

#include <cmath>

#include <iomanip>

#include "MelnykModules.h"

using namespace std;

void Logic(){

int a=0,b=0;

bool s;

cout << "Введiть перший параметр: ";

cin>>a;

cout << "Введiть другий параметр: ";

cin>>b;

s=a+1<b;

cout << "Логічний результат: "<<s<<endl;

}

void Name()

{

system("chcp 1251");

system("cls");

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

cout<<"|Melnyk Sergey CNTU |"<<endl;

cout<<"|Мельник Сергій ЦНТУ|"<<endl;

cout<<" ---------©---------"<<endl;

}

int main()

{

Name();

int sw;

while(sw!=0)

{

cout << "Натиснувши 1 ви отримаєте логiчний результат: a+1<b" << endl << "Натиснувши 2 ви отримаєте розрахунки попереднiх задач!" << endl<<"Натиснувши iнший символ програма завершиться!"<<endl;;

cin >> sw;

switch (sw)

{

case 1:

{

Logic();

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 (valid(z,x,y)==1)

{

cout<<"Неможливо ділити на нуль, введіть інші значення!"<<endl;

system("pause");

return 0;

}

if (valid2(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 s,c,l;

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

c = round(x);

l = round(y);

s = round(z);

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

cout << std::oct << "X = " << c<<" " << " Y = " << l <<" "<<"Z = "<<s<<" "<< endl;

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

cout << std::dec << "X = " << c <<" "<< " Y = " << l<<" " <<"Z = "<<s<<" "<< 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 №2, 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 | libMelnykModules.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; 2; 3 | 3,63105 | 3,63105 | **passed** |
| Q#2 | 1.2;1.3;1.4 | -3,238324 | -3,238324 | **passed** |
| Q#3 | 2,3,4 | 3,553549 | 3,553549 | **passed** |
| Q#4 | 45,45,45 | 3,500341 | 3,500341 | **passed** |

| Ід-р тест-кейса | Вхідні дані | Очікуваний  результат | Вихідні дані | Результат тестування  (passed/failed/ blocked) |
| --- | --- | --- | --- | --- |
| S#1 | 1,1,0 | 1.000000 | 1.000000 | **passed** |
| S#2 | 1; 2; 1 | 15.262172 | 15.262172 | **passed** |
| S#3 | 1.5,2,1 | 15.498441 | 15.498441 | **passed** |
| S#4 | 1; -1; 1 | 0.072195 | 0.072195 | **passed** |
| S#5 | 1;2;0 | 4.000000 | 4.000000 | **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 "MelnykModules.h"

#include <cstdlib>

#include <iostream>

#include <windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL,"ukr");

double massiv[9];

massiv[0]=q\_calculation(1,2,3);

cout<<"TestCase Q#1"<<' '<<"(1,2,3)"<<" 3,63105"<<' '

<<q\_calculation(1,2,3)<<endl;

if (3,63105-massiv[0]>=0.000001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

massiv[1]=q\_calculation(1.2,1.3,1.4);

cout<<"TestCase Q#2"<<' '<<"(1.2,1.3,1.4)"<<" -3,238324"<<' '

<<q\_calculation(1.2,1.3,1.4)<<endl;

if (-3,238324-massiv[1]>=0.0001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

massiv[2]=q\_calculation(2,3,4);

cout<<"TestCase Q#3"<<' '<<"(2,3,4)"<<" 3,553549"<<' '

<<q\_calculation(2,3,4)<<endl;

if (3,5535492-massiv[2] >=0.0000000001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

massiv[3]=q\_calculation(45,45,45);

cout<<"TestCase Q#4"<<' '<<"(45,45,45)"<<" 3,500341"<<' '

<<q\_calculation(45,45,45)<<endl;

if (3,500341-massiv[3] >=0.0000001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

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

cout<<"TestCase Q#5"<<' '<<"(0,0,1)"<<" 3,500000"<<' '

<<q\_calculation(0,0,1)<<endl;

if (3,5000002-massiv[4] >=0.000001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

massiv[5]=s\_calculation(1,1,0);

cout<<"TestCase S#1"<<' '<<"(1,1,0)"<<" 1.000000"<<' '

<<s\_calculation(1,1,0)<<endl;

if (massiv[5]-1.000000 <=0.0001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

massiv[6]=s\_calculation(1,2,1);

cout<<"TestCase S#2"<<' '<<"(1,2,1)"<<" 15.262172"<<' '

<<s\_calculation(1,2,1)<<endl;

if (massiv[6]-15.262172<=0.0001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

massiv[7]=s\_calculation(1.5,2,1);

cout<<"TestCase S#3"<<' '<<"(1.5,2,1)"<<" 15.498441"<<' '

<<s\_calculation(1.5,2,1)<<endl;

if (massiv[7]-15.498441<=0.0001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

massiv[8]=s\_calculation(1,-1,1);

cout<<"TestCase S#4"<<' '<<"(1,-1,1)"<<" 0.072195"<<' '

<<s\_calculation(1,-1,1)<<endl;

if (massiv[8]-0.072195<=0.0001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

massiv[9]=s\_calculation(1,2,0);

cout<<"TestCase S#5"<<' '<<"(1,2,0)"<<" 4.000000"<<' '

<<s\_calculation(1,2,0)<<endl;

if (massiv[9]-4.000000<=0.0001)

cout<<"Passed"<<endl;

else

cout<<"Failed"<<endl;

system("pause");

return 0;

}

Додаток №4

Лістинг створеного модуля

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

short valid2(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 valid(double z, double x, double y)

{

if ((z\*z-(x+y))==0)

return 1;

return 2;

}

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

{

setlocale(LC\_CTYPE,"ukr");

double t,q,k;

k = 0;

if ((sin(x))/(sqrt(pow(log(abs(y\*z)),4)))>pow(10,38)){

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

return 0;

}

else t=(sin(x))/(sqrt(pow(log(abs(y\*z)),4)));

if ((z\*z-(x+y)) >pow(10,38)){

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

return 0;

}

else q=z\*z-(x+y);

if (z\*(t/q)+3.5 >pow(4.54728408833987,2985)){

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

return 0;

}

else k = z\*(t/q)+3.5;

return k;

}

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

{

setlocale(LC\_CTYPE,"ukr");

double k = 0;

const double pi = 3.1415926535897932384626433832795;

if (pow(z+y+z,2)- sqrt(abs((2\*pi\*sqrt(z/2))/(y+4\*(pow(x,3)+cos(z)))))>pow(10,38))

{

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

return 0;

}

else k =pow(z+y+z,2)- sqrt(abs((2\*pi\*sqrt(z/2))/(y+4\*(pow(x,3)+cos(z)))));

return k;

}