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

\_\_\_\_\_\_\_\_\_\_ М.С. Задубяк

ПЕРЕВІРИВ

старший викладач кафедри кібербезпеки та програмного забезпечення

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

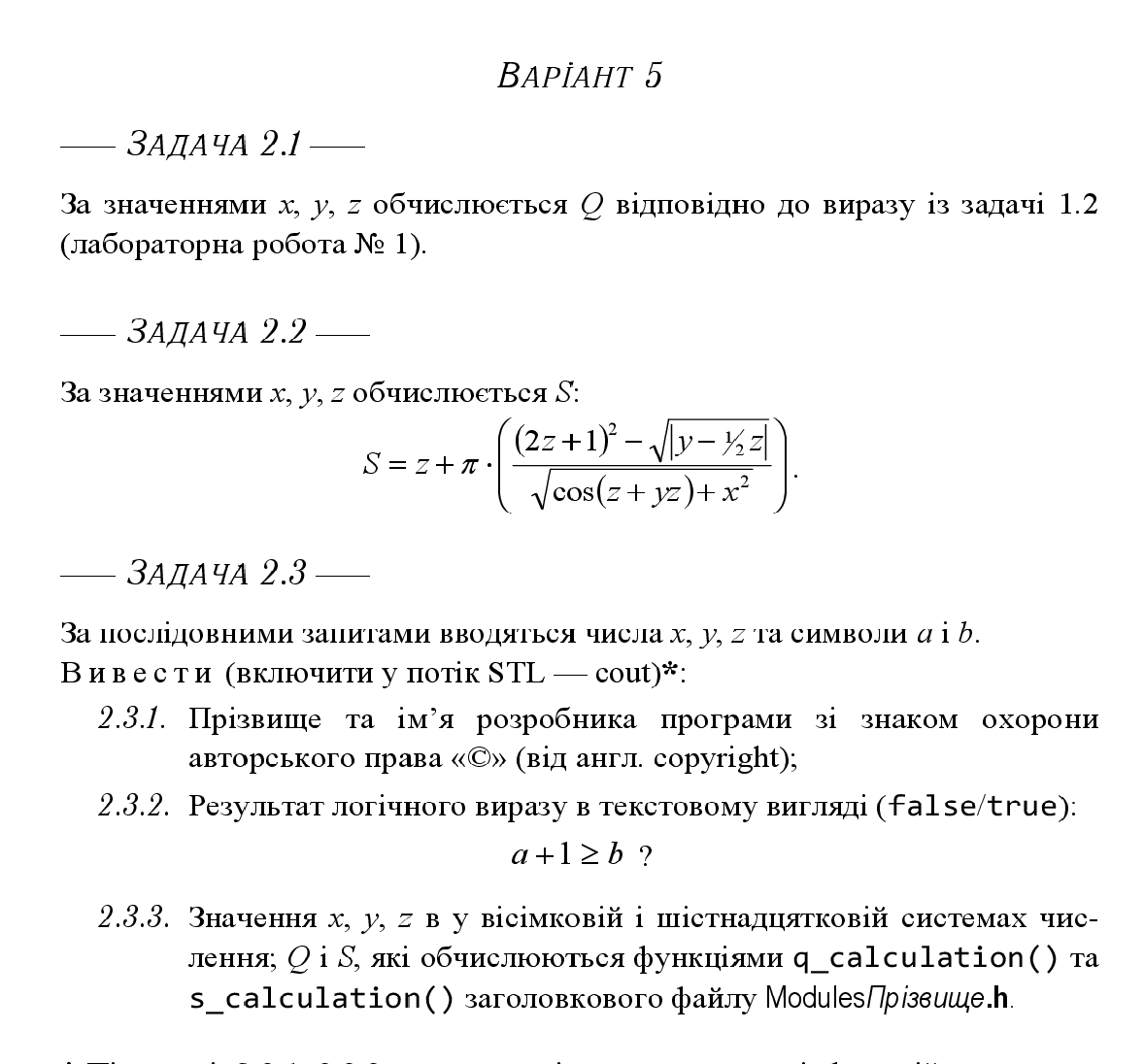
Кропивницький – 2018

**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 libModulesZadubniak С/С++, яка містить функції розв’язування задач 2.1 та 2.2.
2. Реалізувати програмне забезпечення розв’язування задачі 2.3 – консольний додаток.

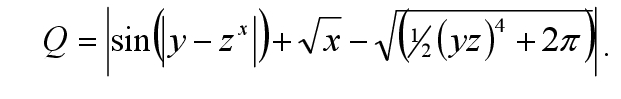


**Задача 2.1**

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;

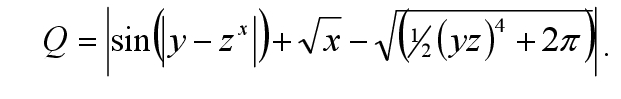
Вхідні дані: змінні x,y,z;

Обмеження та допущення: уся введена і виведена інформація є типом Float; Змінна xбільша або рівна 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.

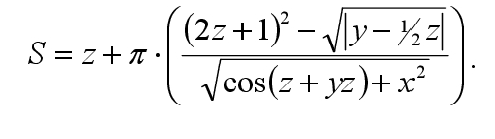
Робоча частина має виконувати необхідні обчислення.

**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;

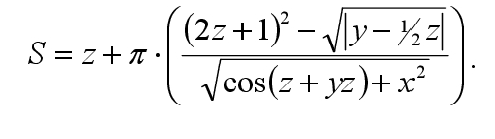
Вхідні дані: змінні x,y,z;

Обмеження та допущення: уся введена і виведена інформація є типом Float; вираз повинен бути більшим або рівним 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:

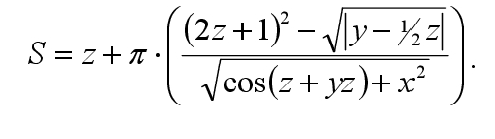
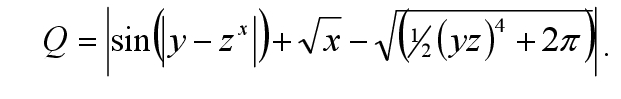
Дана задача має виконувати наступні функції:

-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 Zadubniak\_task:

#include <iostream>

#include <cmath>

#include "test.h"

using namespace std;

float S\_calculation( int r,int t,int g) {

float S, pi = 3.14;

S = r+pi\*((pow((2\*r+1),2))-(sqrt(fabs(g-r/2)))/(sqrt(cos(r+g\*r)+pow(t,2))));

return S;

}

float Q\_calculation( int r,int t,int g) {

float Q, pi = 3.14;

Q = fabs(sin(fabs(g-pow(r,t)))+sqrt(t)-sqrt(pow(g\*r,4)/2+2\*pi));

return Q;

}

int main()

{

setlocale(LC\_ALL,"Ukr");

char a, b;

int z , x, y;

cout << 0x00A9 << "Никита";

cout << "Введіть ціле число z:";

cin >> z;

cout << "Введіть ціле число x:";

cin >> x;

cout << "Введіть ціле число y:";

cin >> y;

cout << "Введіть а:";

cin >> a;

cout << "Введіть b:";

cin >> b;

if ((a+1)>b) {

cout << "true" << endl;}

else {cout << "false" << endl;}

cout << "Число z у вісімковій системі числення = "<< oct << z << endl;

cout << "Число x у вісімковій системі числення = "<< oct << x << endl;

cout << "Число y у вісімковій системі числення = "<< oct << y << endl;

cout << "Число z у шістнадцятковій системі числення = "<< hex << z << endl;

cout << "Число x у шістнадцятковій системі числення = "<< hex << x << endl;

cout << "Число y у шістнадцятковій системі числення = "<< hex << y << endl;

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

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

return 0;

}

**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.