Міністерство освіти та науки  
Центрально-Український Національний Технічний Університет  
Механіко-технічний факультет  
Кафедра Кібербезпеки

ЗВІТ

З лабораторної роботи №8  
З навчальної дисципліни «Базові технології та техніки програмування»  
З теми «Реалізація статичних бібліотек, модулів лінійних обчислювальних процесів»

Виконав   
Студент Академічної групи  
КБ-21  
Крюков К. Є.  
  
  
Перевірив  
Асистент   
Усік П. С.

Кропивницький – 2022р.

Лабораторна робота №8

Тема: Реалізація статичних бібліотек, модулів лінійних обчислювальних процесів

Мета: набути грунтовні вміння і практичні навички застосування теоретичних положень методології модульного програмування, реалізація 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).

Варіант №3

Задача 8.1   
За значеннями x, y, z обчислюється S.

Виконання завдання 8.1

1. Спочатку створюємо модуль s\_calculation, який буде вираховувати формулу, яка задана вище

#include <cmath>

#include <math.h>

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

float S;

S = pow((3 \* sin(sqrt((12 \* pow(x, 2)) / (log(x - 3))))), 2) + 1/2 \* z;

return S;

}

1. Далі створюємо статичну бібліотеку та вставляємо цей модуль

#ifndef MODULEKRIUKOV\_H\_INCLUDED

#define MODULEKRIUKOV\_H\_INCLUDED

#include <cmath>

#include <math.h>

#include <iostream>

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

float S;

S = pow((3 \* sin(sqrt((12 \* pow(x, 2)) / (log(x - 3))))), 2) + 1/2 \* z;

return S;

}

int main()

{

int x = 010;

std::cout << x;

return 0;

}

#endif // MODULEKRIUKOV\_H\_INCLUDED

1. Створюємо тестовий драйвер з статичною бібліотекою (але не забувати, щоб все працювало, треба підключити .h файл та лінкер (Build options… \ Search directory \ Compiler <Шлях до .h файлу>) (Build options… \ Search directory \ Linker <Шлях до лінкера>))
2. Створюємо тест-сьюд та перевіряємо, чи все вірно чи ні. Якщо вірно, записуємо в результаті passed, інакше – fail (див. ДОДАТОК)

#include <iostream>

#include <math.h>

#include "ModuleKriukov.h"

using namespace std;

int main()

{

system("chcp 1251 && cls");

float x[5] = {50, 30, 10, 90, 10};

float y[5] = {60, 20, 5, 70, 15};

float z[5] = {60, 10, 3, 30, 60};

float result[5] = {0.822, 3.692, 0.784, 0.141, 0.784};

float SS;

for(int i = 0; i<5; i++){

SS = (floor(s\_calculation(x[i], y[i], z[i]) \* 1000))/1000;

if (SS == result[i]){

cout << "TestCase ¹" << i+1 << " passed" << endl;

}

else cout << "TestCase ¹" << i+1 << " falled" << endl;

}

}

Задача 8.2

За послідовними запитами вводяться числа x, y, z та символи a і b.  
Вивести (включити у потік STL – cout) :

8.2.1. Прізвище та ім’я розробника програми зі значком охорони авторського права «©» (від англ. Copyright);

8.2.2. Результат логічного виразу в текстовому вигляді (false/true):

8.2.3. Значення x, y, z в десятковій і в шістнадцятковій системах числення; S, що обчислюється функцією s\_calculation() заголовкового файлу ModuleПрізвище.h

Виконання завдання 8.2

1. Спочатку пишемо прізвище та ім’я розробника цього коду зі значком охорони авторського права.

void Autor(){

cout << "© Крюков Костянтин" << endl;

}

1. Далі створюємо функцію, яка повинна рахувати та видавати результат в текстовому вигляді.

void logic(int a, int b){

if((a + 1) > (abs(b - 2)))

{

cout << "True" << endl;

}

else cout << "False" << endl;

}

int main()

{

system("chcp 1251 && cls");

int a, b;

cout << "Введіть а : ";

cin >> a;

cout << "Введіть b : ";

cin >> b;

Autor();

logic(a, b);

return 0;

}

Далі робимо щоб працювало в десятковій та в шістнадцятковій системах числення.

int main(){

int x = 010;

int y = 010;

int z = 010;

std::cout << x;

std::cout << y;

std::cout << z;

return 0;

}

int main(){

int x = 0xF;

int y = 0xF;

int z = 0xF;

std::cout << x;

std::cout << y;

std::cout << z;

return 0;

}

1. Отримуємо такий код (не забуваємо підключити статичну бібліотеку)

#include <iostream>

#include <cmath>

#include <math.h>

#include <ModuleKriukov.h>

using namespace std;

bool isEqual(int a, int b)

{

return (a > b);

}

void Autor(){

cout << "© Крюков Костянтин" << endl;

}

void logic(int a, int b){

if((a + 1) > (abs(b - 2)))

{

cout << "True" << endl;

}

else cout << "False" << endl;

}

int main()

{

system("chcp 1251 && cls");

int a, b;

cout << "Введіть а : ";

cin >> a;

cout << "Введіть b : ";

cin >> b;

Autor();

logic(a, b);

return 0;

}

Далі створюємо тест кейси та перевіряємо чи вірно працює чи ні.

Висновок:

Під час виконання лабораторної роботи №8, були деякі проблеми, стосовно створення статистичної бібліотеки та з 2 завданням. Але все таки зрозумів як треба зробити.

В першому завданні треба було створити код, який повинен розв’язати функцію та треба щоб задавалися числа.   
Що ми робимо? Спочатку створюємо модуль, який повинен працювати разом із кодом. Це виглядає ось так:

#include <cmath>

#include <math.h>

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

float S;

S = pow((3 \* sin(sqrt((12 \* pow(x, 2)) / (log(x - 3))))), 2) + 1/2 \* z;

return S;

}

Далі цей код ми записуємо в бібліотеку, яку потрібно створити.

#ifndef MODULEKRIUKOV\_H\_INCLUDED

#define MODULEKRIUKOV\_H\_INCLUDED

#include <cmath>

#include <math.h>

#include <iostream>

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

float S;

S = pow((3 \* sin(sqrt((12 \* pow(x, 2)) / (log(x - 3))))), 2) + 1/2 \* z;

return S;

}

int main()

{

int x = 010;

std::cout << x;

return 0;

}

#endif // MODULEKRIUKOV\_H\_INCLUDED

Створюємо тестовий драйвер, і підключаємо цю бібліотеку. Але є питання, як ми будемо підключати її? Натискаємо на проєкт, далі натискаємо на Build option… Вибираємо Search directory та Compiler. Туди ми додаємо шлях до нашої статистичної бібліотеки. Далі натискаємо Linker і туди додаємо шлях до лінкера, який повинен лінкувати. Пишемо програму. Після цього ми перевіряємо, чи правильно записано чи ні.

#include <iostream>

#include <math.h>

#include "ModuleKriukov.h"

using namespace std;

int main()

{

system("chcp 1251 && cls");

float x[5] = {50, 30, 10, 90, 10};

float y[5] = {60, 20, 5, 70, 15};

float z[5] = {60, 10, 3, 30, 60};

float result[5] = {0.822, 3.692, 0.784, 0.141, 0.784};

float SS;

for(int i = 0; i<5; i++){

SS = (floor(s\_calculation(x[i], y[i], z[i]) \* 1000))/1000;

if (SS == result[i]){

cout << "TestCase ¹" << i+1 << " passed" << endl;

}

else cout << "TestCase ¹" << i+1 << " falled" << endl;

}

}

(Див. ДОДАТОК)

В другому завданні, треба було створити код, який повинен видавати відповідь тільки в текстовому вигляді (false/true). Вийшло це принаймі так

#include <iostream>

#include <cmath>

#include <math.h>

#include <ModuleKriukov.h>

using namespace std;

bool isEqual(int a, int b)

{

return (a > b);

}

void Autor(){

cout << "© Крюков Костянтин" << endl;

}

void logic(int a, int b){

if((a + 1) > (abs(b - 2)))

{

cout << "True" << endl;

}

else cout << "False" << endl;

}

int main()

{

system("chcp 1251 && cls");

int a, b;

cout << "Введіть а : ";

cin >> a;

cout << "Введіть b : ";

cin >> b;

Autor();

logic(a, b);

return 0;

}

ДОДАТОК

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_8\_1 |
| Назва модуля  Name of Modules/ prj / ModulesZadorozhny | s\_calculation(); |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Крюков Костянтин / Kriukov Kostiantin |
| Виконавець  Implementer | Крюков Костянтин / Kriukov Kostiantin |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Значення | Очікуваний результат | Результат модульного тестування  (passed/fail) |
| 1 | X=50  Y=60  Z=60 | S=0.822 | Passed |
| 2 | X=30  Y=20  Z=10 | S=3.692 | Passed |
| 3 | X=10  Y=5  Z=3 | S=0.784 | Passed |
| 4 | X=90  Y=70  Z=30 | S=0.141 | Passed |
| 5 | X=10  Y=15  Z=60 | S=0.784 | Passed |

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_8\_2 |
| Назва модуля  Name of Modules/ prj / ModulesZadorozhny | s\_calculation(); |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Крюков Костянтин / Kriukov Kostiantin |
| Виконавець  Implementer | Крюков Костянтин / Kriukov Kostiantin |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Значення | Очікуваний результат | Результат модульного тестування  (passed/fail) |
| 1 | A = 60  B = 30 | True | Passed |
| 2 | A = 50  B = 40 | True | Passed |
| 3 | A = 30  B = 40 | False | Passed |
| 4 | A = 80  B = 60 | True | Passed |
| 5 | A = 80  B = 90 | False | Passed |