Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 8

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ

ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи

КБ-21

Горденко А.В.

ПЕРЕВІРИВ

викладач кафедри

кібербезпеки

та програмного забезпечення

Усік П.С

Кропивницький – 2022

**Тема:** Реалізація статистичних бібліотек модулів лінійних обчислювальних процесів.

**Мета:** полягає у набутті ґрунтовних вмінь і практичних навичок реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування C/C++ 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 libModules*Прізвище* C/C++, яка містить функцію розв’язання задачі 8.1.

2. Реалізувати програмне забезпечення розв’язання задачі 8.2 – консольний застосунок

**ВАРІАНТ 24**



Рисунок 1 – Умова задачі 8.1

**Строга постановка задачі 8.1**

Вхідні дані: x, y, z.

Вихідні дані: S

**Проектування програмного модуля**

Проектування модуля складається з двох частин: перша частина це статична бібліотека (де знаходиться реалізація функції s\_calculation()) та заголовковий файл (де знаходиться прототип функції).

Тестовий драйвер, до нього входять:

* Масив даних: x, y, z;
* Масив даних: очікуваних результатів S.

Усі тестові випадки прописані в додатку Б.



Рисунок 2 – Умова задачі 8.2

**Строга постановка задачі 8.2**

Вхідні дані:

* x, y, z;
* a, b.

Вихідні дані:

* Рядок (із ім’ям та прізвищем автора та символом copyright);
* Результат логічного виразу;
* x, y, z в десятковій та шістнадцятковій формах числення;
* Результат функції s\_calculation().

**Процедурна алгоритмізація задачі 8.2**

* 1. Підключення заголовкового файлу “Modules Gordenko.h”
* 2. Підключити локалізацію.
* 3. Функція, яка виводить інформація про автора та значок copyright.
* 4. Вводимо змінні.
* 5. Функція, що приймає дві символьні змінні і виводить результат логічного виразу.
* 6. Функція, що приймає три цілочисельні змінні і виводить їх у десятковій та шістнадйятковій формах числення.
* 7. Виведення результату функції s\_calculation().

**Висновок:**

У ході лабораторної роботи №8 були набуті ґрунтовні вміння та практичні навички реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування C/C++ 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).

Для виконання лабораторної роботи потрібно таке обладнення:

* персональний комп’ютер з операційною системою Windows;
* вільне кросплатформове Code::Blocks IDE ([www.codeblocks.org](http://www.codeblocks.org/));
* текстовий редактор (OpenOffice Writer, Microsoft Word або ін.);
* файл-шаблон тестового набору Artifact\_TEST\_SUITE\_lab.doc;
* власні обліковий запис на GitHub <https://github.com/>

В ході виконання лабораторної роботи,складнішим виявилося виконання першого завдання.Адже спочатку, коли читав методичні рекомендації не зміг реалізувати завдання,але потім детально прочитавши інтернет-джерела - почав виконувати.Завдання вимагало створення статистичної бібліотеки,в якій знаходилася реалізація функція.Далі потрібно було створити заголовковий файл,в якому знаходився прототип функції s\_calculation().Заголовковий файл .h складається з препроцессорних макропідстановок та оголошених прототипів функцій. #ifndef - макрос, що перевіряє визначеність файлу. Якщо файл не визначено - виконуються #define макропідстановка, що задає файлу визначеності. Внаслідок цього, виконується наступний набір операторів, а в кінці вихідного файлу, перевірка на визначеність файлу припиняється макросом #endif.

Файл формату .cpp містить ініціалізацію функцій, раніше оголошених в .h файлі. Потрібне підключення .h файлу через #include для належної реалізації функцій.Далі розробив TestSuite.

Розробив консольний проект(TestDriver) на мові С++.Підключивши ModulesGordenko.h повинен був переверіти результати TestSuite,якщо дані та результат співпадає – Passed,інакше – Failed.Запускаю проект,а виходить помилка “No such a directory”,дивлюсь методичку забув підключити статистичну бібілотеку: Settings -> Compiler -> Linker setting -> Add directory.

Далі почав перевіряти тестові кейси через лістинг програмного забезпечення,результати яких були Passed.

В задачі 8.2 було деякі труднощі,але швидко виправив .Спочатку розробив свій copyright,далі почав розбирати логічній вираз,що і як повинно працювати використовув функцію Expression і далі вираз: a + 10 >= b.Виконання завдання з шістнадцятковую системою використовув функцію DecHexNumber(),яка виводила змінні х, y, z в десятковій та шістнадцятковій системах числення.До завдань було розроблені TestSuites.Виводив всі функції через сout.Перевіривши результати – працювало все без помилок.

Отже, найбільше труднощів було в задачі 8.1, а саме в заголовковому файлі та статистичній бібліотеці. Треба виділяти більше часу на підготовку до лабораторної роботи,методичні вказівки і на пошук певної інформації. Проектування програмних модулі допомагає зберігати реалізацію функцій в інших файлах, це дуже зручно і ефективно. Дуже сподобалося виконання завдання,для себе відкрив нову можливість використовувати заголовкові файлі та модулі.

**ДОДАТОК A**

**(Вихідний код завдань 8.1-8.2.3)**

**Лістинг статистичної бібліотеки main.cpp**

#include <math.h>

using namespace std;

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

{

return (z + 2 \* pow(y, 2)) / (2.72 \* (x - y)) + sqrt((3.14 \* x));

}

**Лістинг ModulesGordenko.h**

#ifndef MODULESGORDENKO\_H\_INCLUDED

#define MODULESGORDENKO\_H\_INCLUDED

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

#endif // MODULESGORDENKO\_H\_INCLUDED

**Лістинг TestDriver**

#include <iostream>

#include "ModulesGordenko.h"

#include <math.h>

using namespace std;

void test\_driver(){

system("chcp 65001 & cls");

int x [5] = {6, 9, 7, 8, 9};

int y [5] = {5, 5, -15, -5, -7};

int z [5] = {9, -10, 8, 6, 10};

int result [5] = {26, 9, 12, 7, 8};

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

{

if(round(s\_calculation( x[i], y[i], z[i])) == result[i])

{

cout << "Passed" << endl;

}

else

{

cout << "Failed" << endl;

}

}

}

int main()

{

test\_driver();

}

**Лістинг GordenkoTask**

#include <iostream>

#include <cmath>

#include <locale>

#include <windows.h>

#include <string>

#include "ModulesGordenko.h"

using namespace std;

string authorCopyright()

{

return "© Gordenko Andrey";

}

bool Expression(double a, double b)

{

return (a + 10 >= b);

}

float DecHexNumber(int x,int y,int z)

{

double S;

cout << endl << "'X' in decimal: " << dec << x << endl

<< "'X' in the hexadecimal: " << hex << x << endl;

cout <<"'Y' in decimal = " << dec << y << endl

<< "'Y' in the hexadecimal: " << hex << y << endl;

cout << "'Z' in decimal = " << dec << z << endl

<< "'Z' in the hexadecimal: " << hex << z << endl;

S = s\_calculation(x,y,z);

cout << endl << "Result s\_calculation: ";

return S;

}

int main()

{

setlocale(LC\_ALL,"UKR");

double x,y,z,a,b;

cout << authorCopyright();

cout << endl << "Enter a value x:";

cin >> x;

cout << "Enter a value y:";

cin >> y;

cout << "Enter a value z:";

cin >> z;

cout << "Enter a value a: ";

cin >> a;

cout << "Enter a value b: ";

cin >> b;

cout << endl << "Result of a boolean expression: " << Expression(a,b) << endl;

cout << DecHexNumber(x, y, z) << endl << endl;

system("pause");

return 0;

}

**ДОДАТОК А**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_8\_1 |
| Назва проекта / ПЗ  Name of Project / Software | TS\_8\_1.exe |
| Рівень тестування  Level of Testing | Модульне |
| Автор тест-сьюта  Test Suite Author | Горденко Андрiй |
| Виконавець  Implementer | Горденко Андрiй |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TS\_1 | X = 6  Y = 5  Z = 9 | 26 | Passed |
| TS\_2 | X = 9  Y = 5  Z = -10 | 9 | Passed |
| TS\_3 | X = 7  Y = -15  Z = 8 | 12 | Passed |
| TS\_4 | X = 8  Y = -5  Z = 6 | 7 | Passed |
| TS\_5 | X = 9  Y = -7  Z = 10 | 8 | Passed |

**ДОДАТОК В**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_8\_2 |
| Назва проекта / ПЗ  Name of Project / Software | GordenkoTask.exe |
| Рівень тестування  Level of Testing | Модульне |
| Автор тест-сьюта  Test Suite Author | Горденко Андрiй |
| Виконавець  Implementer | Горденко Андрiй |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TS\_1 | 1.Відкрити GordenkoTask. | Виведення знаку «©» (від -англ. copyright). | Passed |
| TS\_2 | * Відкрити GordenkoTask * Ввести x = 10 * Ввести y = 4 * Ввести z = 1 * Ввести a = 5 * Ввести b = 11 | * X (10) в шістнадцятковій системі числення = a * Y(4) в шістнадцятковій системі числення = 4 * Z(1) в шістнадцятковій системі числення = 1   Функція s\_calculation:7.62563 | Passed |
| TS\_3 | * ВідкритиGordenkoTask * Ввести x = 15 * Ввести y = 20 * Ввести z = 9 * Ввести a = 5 * Ввести b = 8 | * X(15) в шістнадцятковій системі числення = f * Y(20) в шістнадцятковій системі числення = 14 * Z(9) в шістнадцятковій системі числення = 9   Функція s\_calculation:  -52.6223 | Passed |
| TS\_4 | * Відкрити GordenkoTask * Ввести x = 66 * Ввести y = 50 * Ввести z = 5 * Ввести a = 6 * Ввести b = 7 | * X(66) в шістнадцятковій системі числення = 42 * Y(50) в шістнадцятковій системі числення = 32 * Z(5) в шістнадцятковій системі числення = 5   Функція s\_calculation:  129.4 | Passed |
| TS\_5 | * Відкрити GordenkoTask * Ввести x = 9 * Ввести y = 9 * Ввести z = 8 * Ввести a = 7 * Ввести b = 6 | * X(9) в шістнадцятковій системі числення = 9 * Y(9) в шістнадцятковій системі числення = 9 * Z(8) в шістнадцятковій системі числення = 8   Функція s\_calculation: inf | Passed |
| TS\_6 | * Відкрити GordenkoTask * Ввести x =15 * Ввести y = 21 * Ввести z = 13 * Ввести a = 12 * Ввести b = 5 | * X(15) в шістнадцятковій системі числення = f * Y(21) в шістнадцятковій системі числення = 15   3.Y(13) в шістнадцятковій системі числення = d  Функція s\_calculation:  -47.9777 | Passed |