Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

# ЗВІТ

Про виконання лабораторної роботи № 8

з навчальної дисципліни “Базові методології та технології програмування”

РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИХ

ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ВИКОНАВ

студент академічної групи

КІ-21-1 Василенко Д.М.

Перевірив асистент

Усік П.С \_\_\_\_\_\_\_\_\_\_\_

Кропивницький – 2022

**Лабораторна робота № 8**

**Варіант № 5**

**Мета 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++ 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:**

1. Реалізувати статичну бібліотеку модулів libModulesПрізвище C/C++, яка містить функцію розв’язування задачі 8.1.
2. Реалізувати програмне забезпечення розв’язування задачі 8.2 — консольний застосунок.

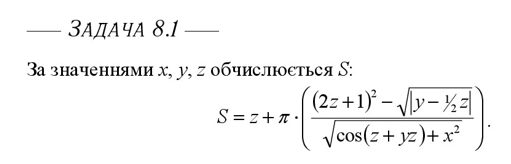
**** **Завдання № 8.1:**

Рисунок 1.1 – Завдання 8.1

**Вхідні дані:**

* x - раціональне число, тип int
* y - раціональне число, тип int
* z - раціональне число, тип int

**Вихідні дані:**

* S - раціональне число, тип double

**Постановка задачі:**

**ПОЧАТОК**

1)Ввести x

2)Ввести y

3)Ввести z

4)Вивести S

**КІНЕЦЬ**

**Результати тестування s\_calculation:**

||=======================||

|| TestDriver was created||

|| by Vasylenko Dima © ||

||=======================||

TC#0:||| passed

TC#1:||| passed

TC#2:||| passed

TC#3:||| passed

TC#4:||| passed

TC#5:||| passed

TC#6:||| passed

TC#7:||| passed

TC#8:||| passed

TC#9:||| passed

**Лістинг завдання № 8.1**

#include <iostream>

#include "ModulesVasylenko.h"

#include <locale>

#include <locale.h>

#include <windows.h>

using namespace std;

void Begin()

{

cout << "|====================|" << endl

<< "|Vasylenko Dima ";

wcout << L"\x00A9 |";

cout << endl

<< "|====================|" << endl;

}

double S(int x, int y, int z)

{

double x1 = x/1;

double y1 = y/1;

double z1 = z/1;

double S;

S = s\_calculation(x1, y1, z1);

return S;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int x, y, z;

cout << "Введіть x: ";

cin >> x;

cout << "Введіть y: ";

cin >> y;

cout << "Введіть z: ";

cin >> z;

Begin();

Changed(x);

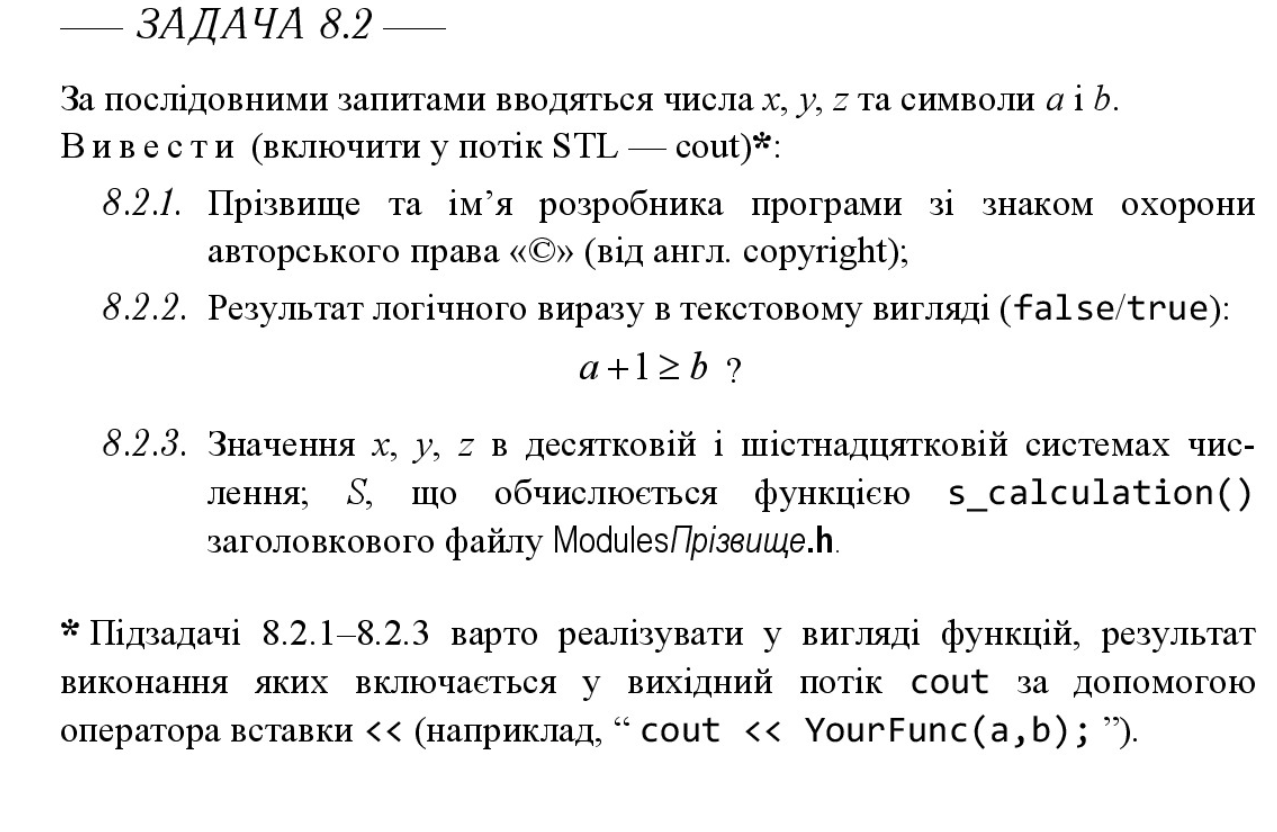
Changed(y);

Changed(z);

cout << "S: " << S(x, y, z);

return 0;

}

**Завдання № 8.2**

**Рисунок 1.2 – Завдання № 8.2**

**Вхідні дані:**

* x - раціональне число, тип int
* y - раціональне число, тип int
* z - раціональне число, тип int
* S - раціональне число, тип double
* a – символ, тип сhar
* b – символ, тип сhar

**Вихідні дані:**

* Ім’я розробника з символом copyright;
* x, y, z в десятковій та шістнадцятковій системі числення;
* Результат функції c\_calculation з модуля ModulesVasylenko.cpp

**Постановка задачі:**

**ПОЧАТОК**

1)Ввести x

2)Ввести y

3)Ввести z

4)Ввести a

5)Ввести b

6)Вивести результат логічного виразу a+1>=b

7)Вивести S

**КІНЕЦЬ**

**Лістинг завдання № 8.2**

#include <iostream>

#include "ModulesVasylenko.h"

#include <locale>

#include <locale.h>

#include <windows.h>

using namespace std;

void Begin()

{

cout << "|====================|" << endl

<< "|Vasylenko Dima ";

wcout << L"\x00A9 |";

cout << endl

<< "|====================|" << endl;

}

void Comparison(char a, char b)

{

if (a+1 >= b)

cout << "a+1 >= b is true" << endl;

else

cout << "a+1 >= b is false" << endl;

}

double Change(int value)

{

int k = 1, b = 0;

while (value != 0)

{

b += value%16\*k;

value/=16;

k\*=10;

}

return b;

}

void Changed(int val)

{

cout << val << "x10: " << val << " " << val << "x16: 0" << Change(val) << endl;

}

double S (int x, int y, int z)

{

double x1 = x/1;

double y1 = y/1;

double z1 = z/1;

double S;

S = s\_calculation(x1, y1, z1);

return S;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int x, y, z;

char a, b;

cout << "Введіть x: ";

cin >> x;

cout << "Введіть y: ";

cin >> y;

cout << "Введіть z: ";

cin >> z;

cout << "Введіть a: ";

cin >> a;

cout << "Введіть b: ";

cin >> b;

Begin();

Comparison(a,b);

Changed(x);

Changed(y);

Changed(z);

cout << "S: " << S(x, y, z)<<endl;

system("pause");

return 0;

}

**Висновок**

Мета лабораторної роботи полягала 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++ 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).

Порівнючи з попередніми лабораторними роботами, новим було використання GitHub. Скориставшись допоміжною літературою для вивчення Git, базових проблем було уникнуто.

Перше завдання було заповнити на Github файл README, у якому було записано тему,мету і варіант. Потім було створено теки,необхідні для роботи на Git. Проаналізувавши і зробивши постанову задачі 8.1, було розроблено мат модель. Згодом було розроблено набір контрольних прикладів до задачі 8.1 задля виконання модульного тестування, усі отримані тест-кейси були додані до \lab08\TestSuite та згодом додані до звіту як додаток.

При виконані лабораторної роботи були надані, використані та розширенні знання про статичні бібліотеки(створення, користування). Для коректної роботи статичної бібліотеки потрібно два файли: заголовковий файл типу .h та статична бібліотека типу .a . Та показати Code::Blocks де ці файли зберігаються. Зазвичай щоб не загубити файли бібліотеки створюється окрема папка.

Реалізувавши функцію s\_calculation мовою програмування С++, яка реалізовує умову задачі 8.1, після компіляції цього проекту було створено файл з розширенням .а. Згодом було створено заголовковий файл ModulesVasylenko.h у якому було описано прототип функції s\_calculation.

Створивши проект консольного додатку “TestDriver” було виконано модульне тестування за допомогою тест-сьюта, що був створений раніше. Модульне тестування було виконане для фунції s\_calculation і отримало всі позначки passed.

Результати тест-кейсів повністю задовольняють очікувані результати. За допомогою тестування можна впевнитися, що програмне забезпечення працює правильно. Вихідний код проектів ModulesVasylenko та TestDriver включено до звіту як додаток А та Б. Результати тестування були задокументовані та додані до звіту як додаток В та Г.

Під час опрацювання завдання 8.2 було опрацьовано складові елементи Архітектури ПЗ.

Архітектура програмного забезпечення — спосіб структурування програмної або обчислювальної системи, абстракція елементів системи на певній фазі її роботи. Система може складатись з кількох рівнів абстракції і мати багато фаз роботи, кожна з яких може мати окрему архітектуру.

Дослідження архітектури програмного забезпечення намагається визначити як найкраще розбити систему на частини, як ці частини визначають та взаємодіють одна з одною, як між ними передається інформація, як ці частини розвиваються поодинці і як все вищеописане найкраще записати використовуючи формальну чи неформальну нотацію

Архітектура повинна будуватись щоб найкраще відповідати вимогам до системи що створюється, згідно принципу "форма відповідає функції”.

**Додаток А**

**ModulesVasylenko:**

#ifndef MODULESVASYLENKO\_H\_INCLUDED

#define MODULESVASYLENKO\_H\_INCLUDED

#include <cmath>

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

{

double S;

S = z + M\_PI\*((pow((2\*z + 1), 2) - sqrt(abs(y - 0.5\*z))/sqrt(cos(z + y\*z)+ pow(x, 2))));

return S;

}

#endif // MODULESVASYLENKO\_H\_INCLUDED

**Додаток Б**

**TestDriver:**

#include <iostream>

#include <iomanip>

#include <cmath>

#include "ModulesVasylenko.h"

#include <locale.h>

#include <windows.h>

using namespace std;

int main()

{

double num[10][3] = {0, 0, 0,

1, 0, 0,

0, 1, 0,

0, 0, 1,

-2.5, 3, 0,

-2.5, 0, -3.12,

0, 4, -1,

4.65, 1.1, 0,

2, 5, 6,

-11, 2.28, -9};

double res[10] = {3.141593, 3.141593, 0.000000, 29.274334, 1.120708,

81.769521, -9.655613, 2.481082, 534.671308, 898.219779};

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "||=======================||" << endl

<< "|| TestDriver was created||" <<endl;

wcout<< L"|| by Dima Vasylenko \x00A9 || ";

cout << endl

<<"||=======================||" << endl;

for (int i = 0; i < 10; i++) {

if((round(s\_calculation(num[i][0], num[i][1], num[i][2])\*1000000)/1000000.0 == res[i]) )

{

cout << "TC#" << '0' << i << setfill(' ')<<" passed"<<endl;

}

else

{

cout << "TC#" << '0' << i << setfill(' ')<<" failed"<<endl;

}

}

system("pause");

return 0;

}

**Додаток В**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | Test-Suite8.1 |
| Назва проекта / ПЗ  Name of Project / Software | Vasylenko-task.exe |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Vasylenko Dmytro |
| Виконавець  Implementer | Vasylenko Dmytro |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | 1. Увести 2 2. Увести 2 3. Увести 3 | |====================|  |Vasylenko Dmytro © |  |====================|  S: 156.938 | PASSED |
| 2 | 1. Увести 3 2. Увести 15 3. Увести 8 | |====================|  |Vasylenko Dmytro © |  |====================|  S: 912.305 | PASSED |
| 3 | 1. Увести 8 2. Увести 3 3. Увести 4 | |====================|  |Vasylenko Dmytro © |  |====================|  S: 258.073 | PASSED |
| 4 | 1. Увести 16 2. Увести 5 3. Увести 7 | |====================|  |Vasylenko Dmytro © |  |====================|  S: 713.662 | PASSED |
| 5 | 1. Увести 12 2. Увести 6 3. Увести 7 | |====================|  |Vasylenko Dmytro © |  |====================|  S: 713.488 | PASSED |

**Додаток Г**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | Test-Suite8.2 |
| Назва проекта / ПЗ  Name of Project / Software | Vasylenko-task.exe |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Vasylenko Dmytro |
| Виконавець  Implementer | Vasylenko Dmytro |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1-A | 1. Увести 2 2. Увести 2 3. Увести 3 4. Увести 5 5. Увести 3 | |====================|  |Vasylenko Dmytro © |  |====================|  a+1 >= b is true  2x10: 2 2x16: 02  2x10: 2 2x16: 02  3x10: 3 3x16: 03  S: 156.938 | PASSED |
| 1-B | 1. Увести 3 2. Увести 15 3. Увести 8 4. Увести 10 5. Увести 20 | |====================|  |Vasylenko Dmytro © |  |====================|  a+1 >= b is false  3x10: 3 3x16: 03  15x10: 15 15x16: 015  8x10: 8 8x16: 08  S: 912.305 | PASSED |
| 1-C | 1. Увести 8 2. Увести 3 3. Увести 4 4. Увести 30 5. Увести 5 | |====================|  |Vasylenko Dmytro © |  |====================|  a+1 >= b is true  8x10: 8 8x16: 08  3x10: 3 3x16: 03  4x10: 4 4x16: 04  S: 258.073 | PASSED |
| 1-D | 1. Увести 16 2. Увести 5 3. Увести 7 4. Увести 6 5. Увести 7 | |====================|  |Vasylenko Dmytro © |  |====================|  a+1 >= b is true  16x10: 16 16x16: 010  5x10: 5 5x16: 05  7x10: 7 7x16: 07  S: 713.662 | PASSED |
| 1-E | 1. Увести 12 2. Увести 6 3. Увести 7 4. Увести 2 5. Увести 5 | |====================|  |Vasylenko Dmytro © |  |====================|  a+1 >= b is false  12x10: 12 12x16: 012  6x10: 6 6x16: 06  7x10: 7 7x16: 07  S: 713.488 | PASSED |