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

ВИКОНАВ

студент академічної групи КІ-21-2

Мельник Анна

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Павло УСІК

Кропивницький – 2022

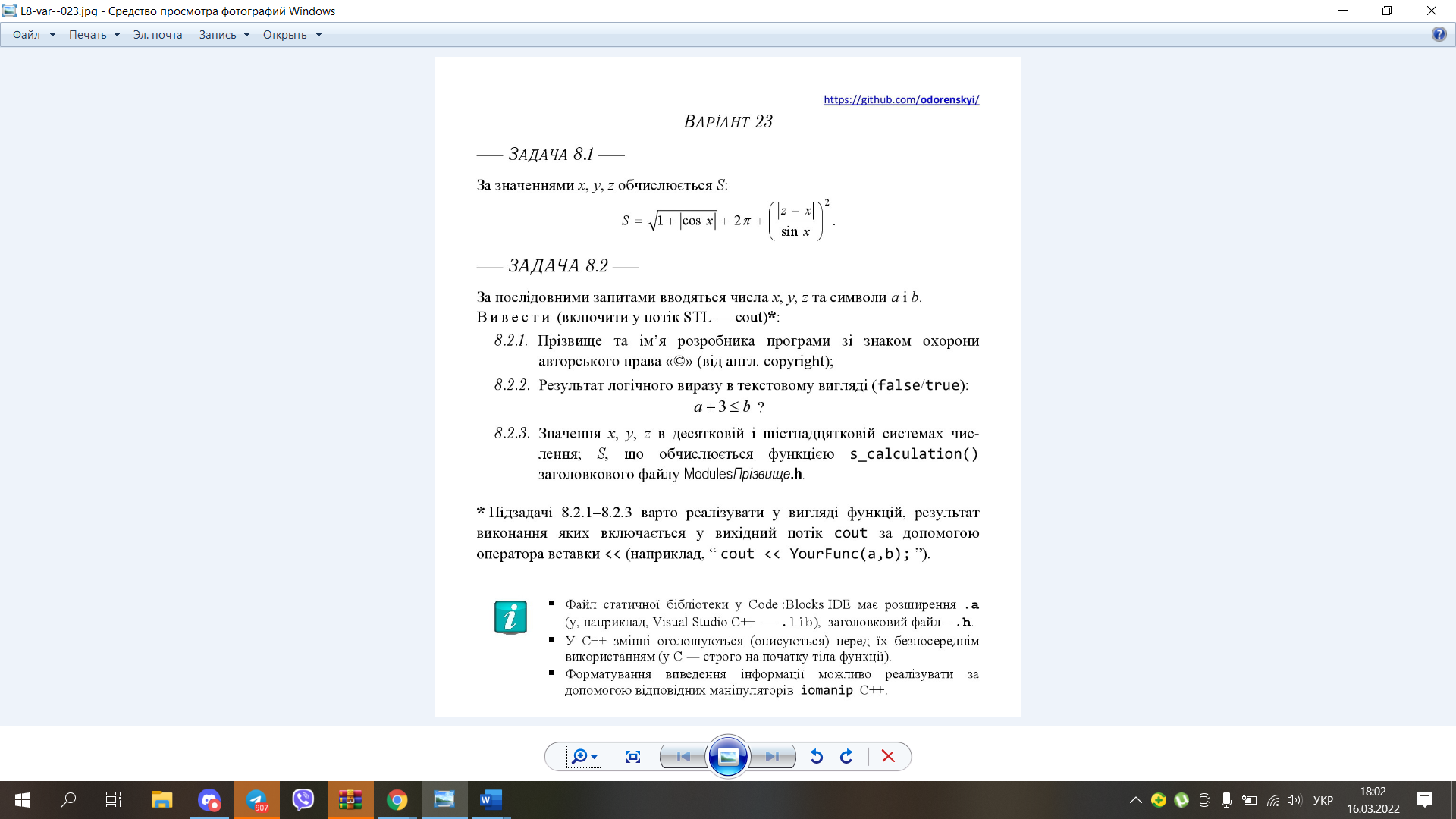
**Мета:** полягає у набутті ґрунтовних вмінь і практичних навичок реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування C/C++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, 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 23



**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.

**Лістинг статичної бібліотеки:**

#include <iostream>

#include <math.h>

#include <tgmath.h>

#include <windows.h>

#include <string>

using namespace std;

float s\_calculation(int x, int z)

{

return (sqrt ( 1 + abs ( cos (x))) +(2 \* M\_PI)+ (pow(abs(z-x) / sin (x), 2)));

}

**Лістинг ModulesMelnik.h:**

#ifndef MODULESMELNIK\_H\_INCLUDED

#define MODULESMELNIK\_H\_INCLUDED

float s\_calculation(int x, int z);

#endif // MODULESMELNIK\_H\_INCLUDED

**Лістинг тестового драйвера:**

#include <iostream>

#include <iomanip>

#include <conio.h>

#include <clocale>

#include <math.h>

#include "../ModulesMelnik.h"

using namespace std;

int main()

{

char \*locale = setlocale(LC\_ALL, "ukr");

float x[5] = { 1, 9, 7, 8, 10};

float z[5] = { 8, 5, 1, 2, 5};

double expectedResult[5] = {76.7261, 101.871, 91.012, 44.1321, 92.1105};

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

{

if (round(s\_calculation(x[i], z[i])\*10000)/10000.0 == expectedResult [i])

{

cout <<"Результат: "<< s\_calculation(x[i], z[i]) <<endl;

cout << "Очiкуваний результат: " << expectedResult[i] << endl;

cout << "Результат: [" << i + 1 << "] PASSED\n";

}

else

{

cout <<"Результат: "<< s\_calculation(x[i], z[i]) <<endl;

cout <<"Очiкуваний результат: " << expectedResult[i] << endl;

cout << "Результат: [" << i + 1 << "] FAILED\n";

}

}

\_getch();

return 0;

}

**Текст протоколу тестування:**

Результат: 76.7261

Очiкуваний результат: 76.7261

Результат: [1] PASSED

Результат: 101.871

Очiкуваний результат: 101.871

Результат: [2] PASSED

Результат: 91.012

Очiкуваний результат: 91.012

Результат: [3] PASSED

Результат: 44.1321

Очiкуваний результат: 44.1321

Результат: [4] PASSED

Результат: 92.1105

Очiкуваний результат: 92.1105

Результат: [5] PASSED

**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, який містить прототип функції s\_calculation().

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 шістнадцятковій системах числення числення.

7. Виведення результату функції s\_calculation().

**Лістинг коду 8\_2:**

#include <iostream>

#include "ModulesMelnik.h"

using namespace std;

void copyright()

{

cout << "© Мельник Анна" << endl << endl;

}

void res ()

{

int a = 0,b = 0;

if (a + 3 <= b){

cout <<"true"<<endl;

}

else {

cout <<"false"<<endl;

}

}

void DecAndHex(int x, int y, int z){

cout << "'y' в десятковій: " << x << endl

<< "'x' в десятковій: " << y << endl

<< "'z' в десятковій: " << z << endl << endl;

cout << "'y' в шістнацятковій: " << hex << x << endl

<< "'x' в шістнацятковій: " << hex << y << endl

<< "'z' в шістнацятковій: " << hex << z << endl;

}

int main()

{

system("chcp 1251 && cls");

copyright();

int x = 0,y = 0,z = 0;

int a = 0,b = 0;

cout << "Введіть x" << endl;

cin >> x;

cout << "Введіть y" << endl;

cin >> y;

cout << "Введіть z" << endl;

cin >> z;

cout << "Введіть a" << endl;

cin >> a ;

cout << "Введіть b" << endl;

cin >> b;

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

res();

DecAndHex(x,y,z);

return 0;

**Висновок**: з цієї лабораторної роботи я навчилася реалізовувати статичних бібліотек модулів лінійних обчислювальних процесів.

Перше завдання полягало в знаходження S шляхом звичайних математичних дій над двома значеннями: х, у. Але для цього потрібно розробити статичну бібліотеку модулів, в якому міститься функція s\_calculation та помістити туди формулу для вирішення прикладу.

Також потрібно створити hfile – заголовковий файл С++.

На жаль, під час підключення бібліотек виникли помилки, оскільки потрібно враховувати багато факторів, такі як: біля головного файлу краще піключити заголовковий,оскільки в разі не з'являння підказки під час підключення файлу в препроцесорних директивах,можна скопіювати назву,а також швидко редагувати заголовковий файл; ще одна ймовірна причина не підлючення відповідного шляху,де знаходяться бібліотеки, а також підлючення але в Debug, а не в загальному підлюченні до програми.

В другому завданні проблем з підключення не було. Важливим фактором для запобігання незрозумілих символів було підключення локалізації, для якого я підлючила кодування Windows1251.

Під час розробки тестового драйвера виникли проблеми виконанням завдання, а точніше з результатом, оскільки не всі результати виявилися Passed, як виявилося, проблема була в тому, що не всі відповіді містили 4 знаки після кому (деякі і більше), тому при написанні очікуваних результатів я вводила лише чотири знаки після коми, і звичайно ж, результат кардинально відрізнявся від потрібного( річ йшла про одну мільйонну ), але у відповідях не відображалося цієї проблеми. Тому було прийнято рішення змінити цифри х та у, і в результаті змінено очікуваний результат. Я спочатку користувалася вже готовою роботою MelnikTask1, там я вводила значення х та у, і отримувала S, далі в програмі я записувала очікуваний результат, після чого в консольному вікні порівнювалося очікуваний результат з реальним.

При роботі з github проблем не виникло, оскільки я вже давно працювала з цією програмою. Програма дуже проста у використанні, при додаванні роботи у гіт можна використовувати сайт, або через консольне вікно, перші спроби я вирішила зробити на сайті, але на мою думку простіше використовувати консольне вікно. Те, що дуже тішить в консольному застосунку, це те, що при неправильному написанні слова, у вікні пишеться, в чому помилка, також неодноразово стикалася з проблемою, коли додаю змінення не в один файл, а в декілька, і коміт додається на декілька файлів, де це не дуже доречно.

Щодо методичної книжки, то все чітко написано, але одна проблема, в деяких випадках не додавалися бібліотеки , і доводилося використовувати рохи інші методи, ніж ті, які зазначалися у методичці.

При розробці TestSuit проблем не виникло, але слід пам’ятати, що модульне та системне тестування відрізняються.

Модульне тестування включає в себе вхідні та вихідні дані, а також результат виконання(passed/faled).

Системне тестування – набір дій для виконання, після чого видається результат, який порівнюють з очікуваним.

Про лабораторну роботу загалом, можу сказати, що вона навчила працювати з модулями, на практиці розробляти та підключати бібліотеки.

На мою думку, важливо щоб в коді не було багато інформації, тому розробка модулів дуже допомагає при виправленні помилок, проблем в коді.

**ДОДАТОК А**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_8\_1 |
| Назва проекта / ПЗ  Name of Project / Software | Melnik\_task\_1.exe |
| Рівень тестування  Level of Testing | модульний |
| Автор тест-сьюта  Test Suite Author | Мельник Анна |
| Виконавець  Implementer | Анна |

|  |  |  |  |
| --- | --- | --- | --- |
| 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) |
| TCM\_1 | Вхід: X=1  Y=8 | Вихід: 76.7261 | PASSED |
| TCM\_2 | Вхід: X=9  Y=5 | Вихід: 101.871 | PASSED |
| TCM\_3 | Вхід: X=7  Y=1 | Вихід: 91.012 | PASSED |
| TCM\_4 | Вхід: X=8  Y=2 | Вихід: 44.1321 | PASSED |
| TCM\_5 | Вхід: X=10  Y=5 | Вихід: 92.1105 | PASSED |

**ДОДАТОК Б**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_8\_2 |
| Назва проекта / ПЗ  Name of Project / Software | Melniktask\_2.exe |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Мельник Анна |
| Виконавець  Implementer | Мельник Анна |

|  |  |  |  |
| --- | --- | --- | --- |
| 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) |
| TC\_1 | 1. Введіть x, y, z: 6,5,47  2. Введіть ‘a’ та ‘b’: 7,3  3. Натисніть Enter | © Мельник Анна  S= 21538.8  true  'x' В десятковій: 6 'x' В шістнадцятковій: 6  'y' В десятковій: 5 'y' В шістнадцятковій: 5  'z' В десятковій: 47 'z' В шістнадцятковій: 2f | PASSED |
| TC\_2 | 1. Введіть x, y, z: 10,8,31  2. Введіть ‘a’ та ‘b’: 1,9  3. Натисніть Enter | © Мельник Анна  S= 1497.71  true  'x' В десятковій: 10 'x' В шістнадцятковій: A  'y' В десятковій: 8 'y' В шістнадцятковій: 8  'z' В десятковій: 31 'z' В шістнадцятковій: 1f | PASSED |
| TC\_3 | 1. Введіть x, y, z: 1,3,5  2. Введіть ‘a’ та ‘b’: 4,5  3. Натисніть Enter | © Мельник Анна  S= 30.1208  false  'x' В десятковій: 1 'x' В шістнадцятковій: 1  'y' В десятковій: 3 'y' В шістнадцятковій: 3  'z' В десятковій: 5 'z' В шістнадцятковій: 5 | PASSED |
| TC\_4 | 1. Введіть x, y, z: 25,34,83  2. Введіть ‘a’ та ‘b’: 90,98  3. Натисніть Enter | © Мельник Анна  S= 192050  true  'x' В десятковій: 25 'x' В шістнадцятковій: 19  'y' В десятковій: 34 'y' В шістнадцятковій: 22  'z' В десятковій: 83 'z' В шістнадцятковій: 53 | PASSED |
| TC\_5 | 1. Введіть x, y, z: 95,1,26  2. Введіть ‘a’ та ‘b’: 65,67  3. Натиснітть Enter | © Мельник Анна  S= 10205.8  false  'x' В десятковій: 95 'x' В шістнадцятковій: 5f  'y' В десятковій: 1 ' y' В шістнадцятковій: 1  'z' В десятковій: 26 'z' В шістнадцятковій: 1A | PASSED |