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 Бідненко П.М.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_  П.С. Усік

Кропивницький – 2022

# ЛАБОРАТОРНА РОБОТА №8

**Тема:** Реалізація статичних бібліотек модулів лінійних обчислювальних процесів

**Мета роботи:** Набуття 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).

## Варіант 3

**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.

# ХІД РОБОТИ

## 

Рисунок 1.1 - Умова задачі 8.1

### **Строга постановка задачі:**

Вхідні дані: x, y - дійсні знакові числа;

Вихідні дані: А, B - дійсні числа (результат розрахунків);

### **Проектування програмного модуля:**

Модуль ModulesBidnenko, що складається з заголовкового файлу (містить оголошення прототипу функції) та файлу вихідного коду (в которому пристуня реалізація функції s\_calculate).

Функція s\_calculate приймає три аргументи в якості вхідних даних, та повертає значення, розраховане за виразом, зазначеним в умові.

Тестовий драйвер TestDriver - функція, що містить в собі:

* масиви для вхідних значень x, y та z відповідно;
* масив значень очікуваних результатів;
* цикл, для виведення еталонних вхідних значень, очікуваного результату та результату розрахунку модуля ModuleBidnenko.

Для вихідного коду, див. ДОДАТОК В.

## 

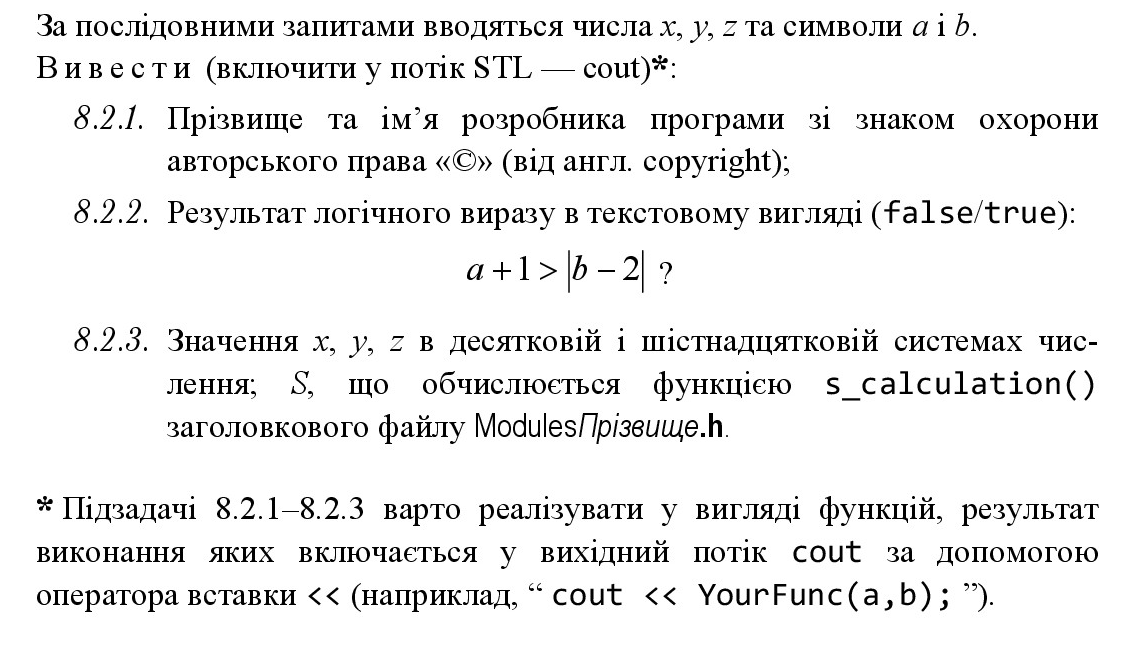


Рисунок 1.2 - Умова задачі 8.2

### **Строга постановка задачі:**

Вхідні дані:

* x, y, z - цілочисельні знакові числа;
* a, b - символьні літерали;

Вихідні дані:

* Ім’я розробника з символом copyright;
* Результат логічного виразу: a + 10 >= b;
* x, y, z в десятковій та шістнадцятковій системі числення;
* Результат функції s\_calculate з модуля ModulesBidnenko;

### **Процедурна алгоритмізація:**

Модуль ModulesBidnenko використовуємо з попереднього проекту.

inputVariables - функція, що приймає 5 аргументів (x, y, z, a, b) та використовує їх для запиту на введення користувачем.

authorCopyright - виводить прізвище ім’я по-батькові розробника, з використанням символу copyright.

logicalExpression - виводить у потік числове логічне значення (1/0) виразу a + 10 >= b;

decimalHexNumbersOutput - слугує для виведення значень x, y, z у десятковій та шістнадцятковій системі числення.

Вищеперераховані функції викликаються в main, а також - до потоку включається функція s\_calculate що приймає аргументи x, y, z та повертає результат обчислень.

# ВИСНОВКИ

Завдяки цій лабораторній роботі я здобув навичок у використанні більш упорядкованої парадигми програмування. Заголовковий файл . h формату 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. Інакше кажучи - дані макроси дозволяють, або забороняють компіляцію виділеної ними частини коду. Файл вихідних кодів формату . cpp містить ініціалізацію функцій, раніше оголошених в . h файлі. Потребує підключення директиви . h файлу через макропідстановку #include для належної реалізації функцій. Робота була виконана за допомогою дистрибутиву C++ та кроссплатформенного IDE - QtCreator 5. Дане програмне забезпечення дозволяє виконати лінкування файлів за допомогою відповідних команд системи збору QML. В корінному каталозі проекту міститься файл формату . pro, в якому були вказані всі заголовкові файли та файли вихідного коду (зокрема і main. Рішення про зміну середовища розробки аргументується тим, що лінкування QtCreator на відміну від лінкування, реалізованому в Code: :Blocks, є більш наочним та визначеним, що дозволяє безпосередньо в редакторі коду налаштовувати базові поля збірки проекту системою QML При виконанні завдань 8. 1 та 8. 2 був задіяний метод висхідного проектування програмного забезпечення, що полягав у початковій реалізації менш значимих (або допоміжних) функцій нижнього рівня. Такою функцією виявилась s\_calculate, з модуля ModulesStriuk, розробленим на етапі задачі 8. До даного застосовувались всі правила побудови файлів вихідного коду та заголовкового. Надалі, модуль ModulesStriuk використовувався в main функціях обох задач. Також, необхідно виділити увагу бібліотеці , що використовувалась для операцій всередині s\_calcuate. Базова математична функція log(int) використовувалась замість ln (натуральный логарифм), адже вона відсутня у стандартній бібліотеці cmath (або math.

**ДОДАТОК А**

(TestSuite до завдання 8.1)

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_8\_1 |
| Date: 18/04/2022 | Назва проекта/ПЗ /  **Name of project** | Bidnenko-task\_8\_1.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Бідненко Павло |
|  | Виконавець /  **Implementer** | Бідненко Павло |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Actions** | **Expected Result** | **Test Result** |
| TS\_01 | X = 22  Y = -0,8 | 1 | Passed |
| TS\_02 | X = 33  Y = 0,8 | 5 | Passed |
| TS\_03 | X = 8  Y = 0 | 0 | Passed |
| TS\_04 | X = 8  Y = -8 | -3 | Passed |
| TS\_05 | X = 1  Y = -4 | 6 | Passed |

**ДОДАТОК Б**

(TestSuite до завдання 8.2)

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_8\_2 |
| Date: 18/04/2022 | Назва проекта/ПЗ /  **Name of project** | Bidnenko-task\_8\_2.ехе |
|  | Рівень тестування /  **Level of testing** | Системний |
|  | Автор тест-сьюта /  **Test Suite Author** | Бідненко Павло |
|  | Виконавець /  **Implementer** | Бідненко Павло |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест кейса /  **Test Case ID** | Дії (кроки)  /  **Action (Test Steps)** | Очікуваний результат  /  **Expected Result** | Результат тестування  /  **Test Result** |
| TS\_01 | 1. Відкрити застосунок | © Бідненко Павло Максимович | Passed |
| TS\_02 | 1. Ввести число 1 = 4  2. Ввести число 2 = 5  3. Ввести символ А = G  4. Ввести символ B = B | Результат G + 10 >= B : true  'x' в десятковій: 4  'z' в десятковій: 5  'x' в шістнацятковій: 4  'z' в шістнацятковій: 5 | Passed |
| TS\_03 | 1. Ввести число 1 = 3  2. Ввести число 2 = 6  3. Ввести символ А = R  4. Ввести символ B = Y | Результат R + 10 >= Y : false  'x' в десятковій: 3  'z' в десятковій: 6  'x' в шістнацятковій: 3  'z' в шістнацятковій: 6 | Passed |
| TS\_04 | 1. Ввести число 1 = 1  2. Ввести число 2 = 3  3. Ввести символ А = Q  4. Ввести символ B = Z | Результат Q + 10 >= Z : false  'x' в десятковій: 1  'z' в десятковій: 3  'x' в шістнацятковій: 1  'z' в шістнацятковій: 3 | Passed |
| TS\_05 | 1. Ввести число 1 = 5  2. Ввести число 2 = 8  3. Ввести символ А = H  4. Ввести символ B = K | Результат H + 10 >= K : false  'x' в десятковій: 5  'z' в десятковій: 8  'x' в шістнацятковій: 5  'z' в шістнацятковій: 8 | Passed |

**ДОДАТОК В**

(Вихідний код завдання 8.1)

#include <iostream>

#include <iomanip>

#include <windows.h>

#include "ModulesBidnenko.h"

using namespace std;

int main()

{

system("chcp 65001 & cls");

float X[5] = { 22, 33, 8, 8, 11 };

float Z[5] = { -0.8, 0.8, 0, -8, -4 };

double exResult[5] = { 1, 5, 0, -3, 6 };

double doneResult = 0;

double delta = 0.00005;

bool testResult = false;

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

doneResult = s\_calculation(X[i], Z[i]);

if (abs(exResult[i]) >= abs(doneResult)) { testResult = abs(exResult[i]) - abs(doneResult) <= delta; }

else { testResult = abs(doneResult) - abs(exResult[i]) <= delta; }

cout << "Тестові значення №" << i + 1 << " : "

<< "X = " << X[i]

<< " Z = " << Z[i]

<< endl;

cout << "Очікуваний результат: " << exResult[i] << endl;

cout << "Отриманий результат: " << doneResult << endl;

cout << boolalpha << "Тестовий результат: " << testResult << endl << endl;

}

getchar();

return 0;

}

**ДОДАТОК Г**

(Вихідний код завдання 8.2)

#include <iostream>

#include <clocale>

#include <iomanip>

#include <cstring>

#include "ModulesBidnenko.h"

using namespace std;

int x = 0, z = 0;

char a = '0', b = '0';

string authorCopyright()

{

return "© Бідненко Павло Максимович";

}

void inputVariables(int X, int Z, char A, char B)

{

cout << "Введіть число X: "; cin >> X; x = X;

cout << "Введіть число Z: "; cin >> Z; z = Z;

cout << "Введіть символ №1: "; cin >> A; a = A;

cout << "Введіть символ №2: "; cin >> B; b = B;

cout << endl;

}

bool logicalExpression(int A, int B)

{

return A + 1 > abs(B - 2);

}

void decimalHexNumbersOutput(int X, int Z)

{

cout << "'x' в десятковій: " << dec << X << endl

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

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

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

}

int main()

{

system("chcp 65001 & cls");

cout << authorCopyright() << endl << endl;

inputVariables(x, z, a, b);

cout << boolalpha

<< "Результат " << a << " + 10 >= " << b << " : " << logicalExpression(a, b) << endl << endl;

decimalHexNumbersOutput(x, z);

cout << endl << "Результат виразу: " << s\_calculation(x, z);

getchar(); return 0;

}