Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

# ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9

з навчальної дисципліни “Базові методології та технології програмування”

“Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів”

ВИКОНАВ

студент академічної групи

КІ-21-1 Назаревський М. В.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

Усік П. С.

Кропивницький – 2022

# ЛАБОРАТОРНА РОБОТА №9

**Тема:** Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів

**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).

## Варіант 3

**Завдання:** 1. Реалізувати функції розв’язування задач 9.1–9.3 як складовихстатичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище, створений під час виконання лабораторної роботи №8).

2. Реалізувати програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesПрізвище.а.

# ХІД РОБОТИ

**Завдання 1**

## 

Рисунок 1.1 - Умова задачі 9.1

### **Аналіз задачі**

Користувач вводить бал Бофорта – число від 0 до 12.

Застосунок повинен вивести швидкість вітру у м/с та його характеристика.

### **Проектування програмного модуля:**

За даними задачі, користувач вводить число, що має відобразити швидкість вітру та його характеристику.

Потрібно реалізувати систему switch, case, або if, else.

Я обрав систему switch, case.

Також потрібно виконати обмеження вхідного числа (від 0 до 12), якщо число не входить у цей діапазон, то видати помилку – невірний бал Бофорта. Цю умову я виконав, задавши “дефолт” у системі switch, case.

**Завдання 1**

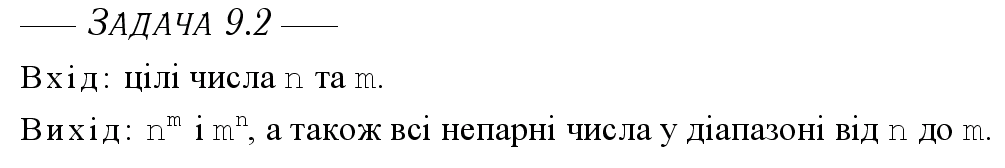


Рисунок 1.2 - Умова задачі 9.2

### **Аналіз задачі:**

n, m - цілі числа, які потрібно ввести користувачу;

Застосунок повинен виводити число n у степені m та n у степені m, а також всі непарні числа у діапазоні від n до m.

Тобто, якщо число n більше за m, то остання умова не буде виконуватись.

### **Процедурна алгоритмізація:**

Розробляємо структуру з n та m, у якій змінні p.n та p.m приймають значення n^m та m^n.

Розробляємо цикл, в котрому буде реалізуватись виведення непарних чисел у діапазоні від n до m.

Далі виводимо на екран значення змінних p.n та p.m та повертаємо нашу структуру

**Завдання 3**

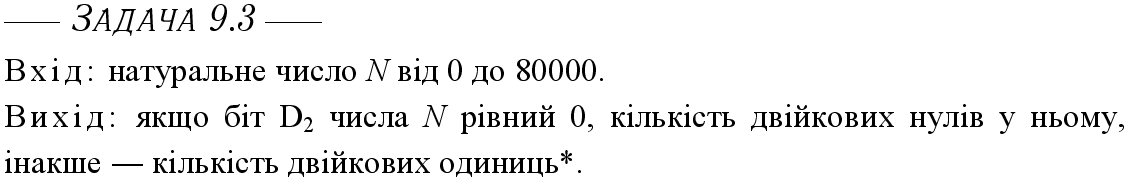


Рисунок 1.3 - Умова задачі 9.3

### **Аналіз задачі:**

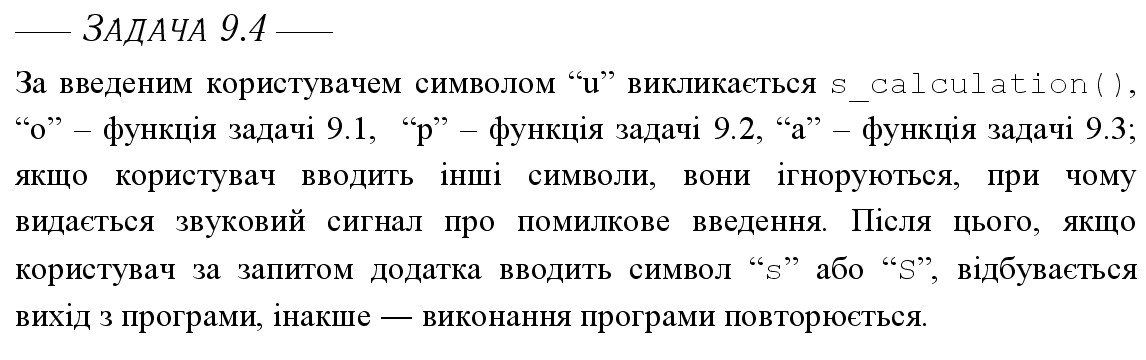
Користувач вводить число від 0 до 80000. Спочатку число потрібно перетворити в двійкову систему числення. Потім після цього порівняти 2 біт, якщо він дорівнює 0, то вивести кількість двійкових нулів у ньому, інакше – кількість двійкових одиниць.

### **Процедурна алгоритмізація:**

Створити умову, що перевіряє число на заданий діапазон у задачі (від 0 до 80000), якщо число не задовільняє умову - видати помилку.

Перетворити десяткове число на двійкове. Створити умову, якщо біт D2 числа рівний 0, то вивести кількість двійкових нулів у ньому, у випадку якщо біт D2 числа рівний 1, то вивести кількість двійкових одиниць.

**Завдання 4**

 Рисунок 1.4 - Умова задачі 9.4

**Алгоритм до задачі 9.4**

**START**

1. DO, WHILE symbol != ‘S’ symbol != ‘s’
2. PRINT “Введіть символ”;
3. INPUT symbol;
4. IF symbol = ‘u’

s\_calculation();

1. ELSE IF symbol = ‘o’

bofort(bal);

1. ELSE IF symbol = ‘p’

p = powAndNum(n, m);

1. ELSE IF symbol = ‘a’

numBit(num);

1. ELSE

PRINT Звуковий сигнал

1. ENDWHILE

**END**

**Результати виконання тестового модуля**

Тест завдання 9.1 Тест завдання 9.2 Тест завдання 9.3

Passed Passed Passed

Passed Passed Passed

Passed Passed Passed

Passed Passed Passed

Passed Passed Passed

# ВИСНОВКИ

Під час виконання лабораторної роботи № 9 на тему **«**Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів**»** з предмету «Базові Методології та Технології Програмування» я відпрацював на практиці створення статичних бібліотек та власних модулів і заголовних файлів, написання власних текстових драйверів для своїх модулів. Набув ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів С/С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

Завдання 1 передбачало виконання застосунку, котрий повинен видавати значення сили вітру (у метрах за секунду) та характеристику вітру, маючи у вхідних даних число – бал за таблицею Бофорта.

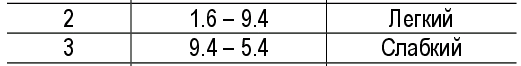
Під час ознайомлення з цим завданням я виявив помилку, а саме фрагмент у таблиці швидкості вітру при балі за Бофортом 2 та 3 -

Рисунок 2.1 – Помилка у завданні 1

Повідомивши про помилку викладачу кафедри кібербезпеки та програмного забезпечення Усіку П.С., мені оперативно була надіслана табличка з коректними значеннями, за що окремо вдячний.

Під час планування завдання 1, я обрав спосіб виконання методом switch, case, а не if, else, спонукаючись на те, що цей метод є більш підходящим за розумінням та об'ємом написання і має чудову функцію – default, що присвоює значення, якщо жоден з “кейсів” не спрацював.

Також, обов'язково, після кожного “кейсу”, я не забував вставляти break, така помилка є дуже розповсюдженною серед починаючих студентів.

Завдання 2 представило для мене труднощі через неоднозначність у його перевірці…Після довгих міркувань мені довелось звернутися до свого товариша - Владислава Стрюка, котрий порадив трохи зайти наперед та ознайомитись з структурами. Саме ознайомлення вимусило мене створити структуру з змінних n та m, тому що, для успішної роботи тест-драйвера потрібно, щоб хоча б один return був у модулі, але моє завдання передбачало два return'и, що є неможливим без використання структури. Я не знайшов іншого методу для виконання даного завдання.

Завдання 3 було важким для мене, я не відразу зрозумів, як його можна виконати. Почитавши документацію, я виявив дуже зручну для цього завдання бібліотеку bitset, яка дає можливість перетворювати числа з десяткової системи числення у двійкову систему. Написавши умову перевірки числа на задані у завданні (число від 0 до 80000), я додав функцію bitset, з допомогою якої успішно виконав завдання 3.

Фінальне завдання 4 ставило мету об'єднати у собі всі виконані модулі, при натиску на певні символи запустити певний модуль та організувати нескінченну роботу застосунку, з якого можна вийти, також натиснувши деякий символ. Використавши систему do, while, мені вдалося організувати нескінченну роботу та використавши велику кількість умов if, else, (хоча можна було застосувати систему switch, case, як у першому завданні) я без проблем виконав завдання 4.

Я розробив TestDriver, котрий мав автоматично перевірити працездатність створених мною модулів, заздалегідь створивши unit test'и та test suite, я записав до тест-драйверу значення з unit test'ів . У результаті виконання, помилок не виявлено, тобто модулі справні.

Після того, як TestDriver підтвердив справність модулів, я почав тестувати завдання 4. Перевіривши застосунок з допомогою test suite, аналогічно з модулями – помилок не виявлено.

Також мені довелось змінити формат текстових файлів з ANSI на UTF-8, задля коректного відображення на сайті Github.

Лабораторна робота №9 була успішно виконана мною та завантажена до власного репозитарію.

В ДОДАТКУ А вказані unit test до функцій bofort(bal), powAndNum(n,m), numBit(num). А також test suite до задачі 9.4.

В ДОДАТКУ Б вказані лістинг програм, статичної бібліотеки, тестового драйвера та Nazarevskyi\_task.

**ДОДАТОК А**

**Лістинг ModulesNazarevskyi**

#include <iostream>

#include <cmath>

#include <bitset>

#include <string>

#include "ModulesNazarevskyi.h"

using namespace std;

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

{

return z\*sin(pow(x,2) \* y) + sqrt(abs(z - x\*12))/pow(y,3);

}

double bofort(int bal)

{

switch (bal)

{

case 0:

cout << "< 0.3 - Штиль";

return 0.3;

break;

case 1:

cout << "0.3 - 1.5 - Тихий";

return 1.5;

break;

case 2:

cout << "1.5 - 3.3 - Легкий";

return 3.3;

break;

case 3:

cout << "3.3 - 5.4 - Слабкий";

return 5.4;

break;

case 4:

cout << "5.4 - 7.9 - Помірний";

return 7.9;

break;

case 5:

cout << "7.9 - 10.7 - Свіжий";

return 10.7;

break;

case 6:

cout << "10.7 - 13.8 - Сильний";

return 13.8;

break;

case 7:

cout << "13.8 - 17.1 - Міцний";

return 17.1;

break;

case 8:

cout << "17.1 - 20.7 - Дуже міцний";

return 20.7;

break;

case 9:

cout << "20.7 - 24.4 - Шторм";

return 24.4;

break;

case 10:

cout << "24.4 - 28.4 - Сильний шторм";

return 28.4;

break;

case 11:

cout << "28.4 - 32.6 - Шквальний шторм";

return 32.6;

break;

case 12:

cout << ">32.6 - Ураган(Буревій)";

return 32.7;

break;

default:

cout << "Невірний бал Бофорта, введіть число від 0 до 12";

return -1;

}

}

powers powAndNum(int n, int m)

{

powers p;

p.n = pow(n, m);

p.m = pow(m, n);

cout << "Непарні числа у проміжку: ";

for (int i = n; i <= m; ++i){

if (i%2){

cout << i <<" ";

}

}

cout << endl;

cout << "Число n в степені m = " << p.n << endl;

cout << "Число m в степені n = " << p.m << endl;

return p;

}

int numBit(int num)

{

if(num < 0 || num > 80000){

cout << "Невірне число, введіть число від 0 до 80000 " << endl;

return -1;

}

bitset<32> b\_number(num);

if(b\_number[2]){

return b\_number.count();

}

else {

return 32 - b\_number.count();

}

}

string test\_module(bool result)

{

if(result){

return "Test is passed";

}

else{

return "Test failed";

}

}

**Лістинг TestDriver**

#include <iostream>

#include <ModulesNazarevskyi.h>

using namespace std;

int main()

{

powers p;

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

int bal[5] = {0, 1, 2, 3, 4};

double result\_bofort[5] {0.3, 1.5, 3.3, 5.4, 7.9};

cout << "Тест завдання 9.1" << endl;

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

if (result\_bofort[i] == bofort(bal[i])){

cout << endl;

cout << "Passed" << endl;

}

else {cout << endl; cout << "Failed" << endl;}

}

cout << "Тест завдання 9.2" << endl;

p = powAndNum(2, 5);

if (p.n == 32 && p.m == 25){

cout << "Passed" << endl;

}

else {cout << "Failed" << endl;}

p = powAndNum(3, 6);

if (p.n == 729 && p.m == 216){

cout << "Passed" << endl;

}

else {cout << "Failed" << endl;}

p = powAndNum(2, 9);

if (p.n == 512 && p.m == 81){

cout << "Passed" << endl;

}

else {cout << "Failed" << endl;}

p = powAndNum(1, 6);

if (p.n == 1 && p.m == 6){

cout << "Passed" << endl;

}

else {cout << "Failed" << endl;}

p = powAndNum(4, 11);

if (p.n == 4194304 && p.m == 14641){

cout << "Passed" << endl;

}

else {cout << "Failed" << endl;}

cout << "Тест завдання 9.3" << endl;

int num[5] = {22, 28312, 3042, 13, 2270};

int res\_numBit [5] = {3, 24, 25, 3, 7};

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

cout << "Результат: " << numBit(num[i]) << endl << "Очікуваний результат: "<< res\_numBit[i] << endl;

if (res\_numBit[i] == numBit(num[i])){

cout << "Passed" << endl;

}

else {cout << "Failed" << endl;}

}

return 0;}

**Лістинг Nazarevskyi\_task**

#include <iostream>

#include <locale.h>

#include <windows.h>

#include <ModulesNazarevskyi.h>

using namespace std;

int main()

{

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

powers p;

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

int bal = 0;

int n = 0, m = 0;

int num = 0;

char symbol;

do {

cout << endl;

cout << "Натисніть символ" << endl;

cout << " u - для підрахунку чисел x, y, z за формулою" << endl;

cout << " o - для визначення швидкості та характеристики вітру" << endl;

cout << " p - для підрахунку n^m і m^n, всі непарні числа у діапазоні від n до m" << endl;

cout << " a - для визначення кількості двійкових 0 у числі, якщо біт D2 числа = 0, інакше - кількість двійкових 1" << endl;

cout << " s - для завершення роботи" << endl;

cin >> symbol;

if (symbol == 'u'){

cout << "Введіть x: " << endl;

cin >> x;

cout << "Введіть y: " << endl;

cin >> y;

cout << "Введіть z: " << endl;

cin >> z;

cout << "Результат: " << endl;

cout << s\_calculation(x, y, z) << endl;

}

else if(symbol == 'o'){

cout << "Введіть бал Бофорта" << endl;

cin >> bal;

bofort(bal);

}

else if(symbol == 'p'){

cout << "Введіть ціле число n: " << endl;

cin >> n;

cout << "Введіть ціле число m: " << endl;

cin >> m;

p = powAndNum(n, m);

}

else if (symbol == 'a'){

cout << "Введіть натуральне число: " << endl;

cin >> num;

cout << numBit(num) << endl;

}

else if (symbol != 's' && symbol != 'S') {

cout << "\a" << endl;

}

} while (symbol != 's' && symbol != 'S');

}

**ДОДАТОК Б**

**Модульний тест до задачі 9.1**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_9\_1 |
| Назва модуля  Name of Modules/ prj / ModulesNazarevskyi | bofort(bal) |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Назаревський Максим |
| Виконавець  Implementer | Назаревський Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Значення | Очікуваний результат | Результат модульного тестування  (Passed/Failed) |
| 1 | bal = 0 | < 0.3 – Штиль | Passed |
| 2 | bal = 1 | 0.3 – 1.5 – Тихий | Passed |
| 3 | bal = 2 | 1.5 – 3.3 – Легкий | Passed |
| 4 | bal = 3 | 3.3 – 5.4 – Слабкий | Passed |
| 5 | bal = 4 | 5.4 – 7.9 – Помірний | Passed |

**Модульний тест до задачі 9.2**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_9\_2 |
| Назва модуля  Name of Modules/ prj / ModulesNazarevskyi | PowAndNum(n,m) |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Назаревський Максим |
| Виконавець  Implementer | Назаревський Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Значення | Очікуваний результат | Результат модульного тестування  (Passed/Failed) |
| 1 | n = 2  m = 5 | Непарні числа у проміжку: 3 5  Число n в степені m = 32  Число m в степені n = 25 | Passed |
| 2 | n = 3  m = 6 | Непарні числа у проміжку: 3 5  Число n в степені m = 729  Число m в степені n = 216 | Passed |
| 3 | n = 2  m = 9 | Непарні числа у проміжку: 3 5 7 9  Число n в степені m = 512  Число m в степені n = 81 | Passed |
| 4 | n = 1  m = 6 | Непарні числа у проміжку: 1 3 5  Число n в степені m = 1  Число m в степені n = 6 | Passed |
| 5 | n = 4  m = 11 | Непарні числа у проміжку: 5 7 9 11  Число n в степені m = 4194304  Число m в степені n = 14641 | Passed |

**Модульний тест до задачі 9.3**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_9\_3 |
| Назва модуля  Name of Modules/ prj / ModulesNazarevskyi | numBit(num) |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Назаревський Максим |
| Виконавець  Implementer | Назаревський Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Значення | Очікуваний результат | Результат модульного тестування  (Passed/Failed) |
| 1 | num = 22 | 3 | Passed |
| 2 | num = 28312 | 24 | Passed |
| 3 | num = 3042 | 25 | Passed |
| 4 | num = 13 | 3 | Passed |
| 5 | num = 2270 | 7 | Passed |

**Системний тест до задачі 9.4**

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_9\_4 |
| Date: 03/04/2022 | Назва проекта/ПЗ /  **Name of project** | Nazarevskyi\_task.ехе |
|  | Рівень тестування /  **Level of testing** | Системний |
|  | Автор тест-сьюта /  **Test Suite Author** | Назаревський Максим |
|  | Виконавець /  **Implementer** | Назаревський Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест кейса /  **Test Case ID** | Дії (кроки)  /  **Action (Test Steps)** | Очікуваний результат  /  **Expected Result** | Результат тестування  /  **Test Result** |
| TS\_01 | 1. Ввести “u”  2. Ввести 14  3. Ввести 12  4. Ввести 8 | Натисніть символ  u - для підрахунку чисел x, y, z за формулою  o - для визначення швидкості та характеристики вітру  p - для підрахунку n^m і m^n, всі непарні числа у діапазоні від n до m  a - для визначення кількості двійкових 0 у числі, якщо біт D2 числа = 0, інакше - кількість двійкових 1  s – для завершення роботи  u  Введіть x:  14  Введіть y:  12  Введіть z:  8  Результат:  6.95821  … | Passed |
| TS\_02 | 1. Ввести “o”  2. Ввести 7  3. Ввести “s” | …  Введіть бал Бофорта  7  13.8 – 17.1 – Міцний  …  s | Passed |
| TS\_03 | 1. Ввести ”p”  2. Ввести 5  3. Ввести 8  4. Ввести “s” | …  Введіть ціле число n:  5  Введіть ціле число m:  8  Непарні числа у проміжку: 5 7  Число n в степені m = 390625  Число m в степені n = 32768  …  s | Passed |
| TS\_04 | 1. Ввести “a”  2. Ввести 12  3. Ввести “s” | …  Введіть натуральне число:  12  2  …  s | Passed |
| TS\_05 | 1. Ввести ”w”  2. Ввести “q”  3. Ввести “s” | …  w  \*Звуковий сингал\*  …  q  \*Звуковий сингал\*  …  s | Passed |