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 Ожеховський Владислав

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_  П.С. Усік

Кропивницький – 2022

# Лабораторна робота №9

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

**Мета роботи:** полягає у набутті 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.

## Варіант 1

**Завдання:**

1. Реалізувати функції розв’язування задач 9.1–9.3 як складових статичної 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

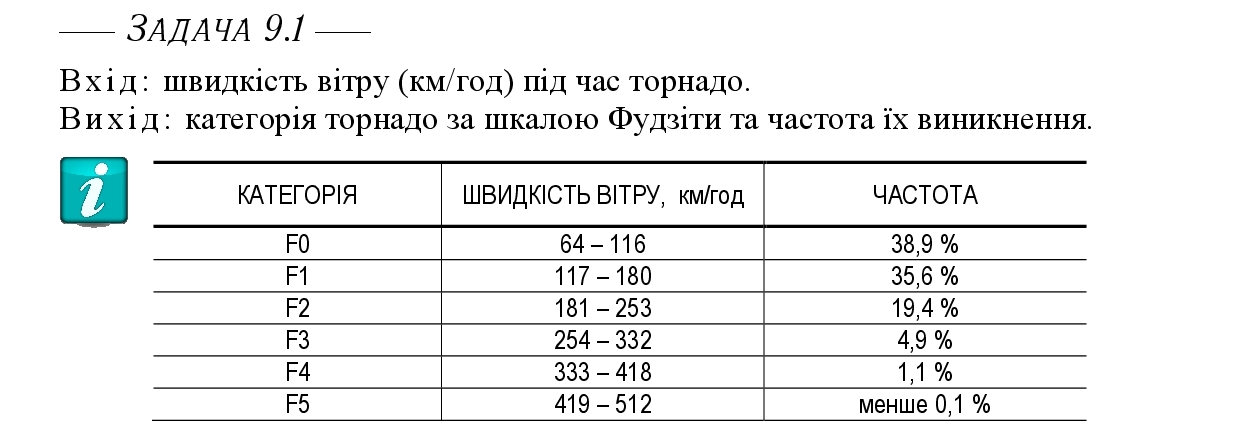


Рисунок 1.1 - Умова задачі 9.1

### **Строга постановка задачі:**

Вхідні дані: швидкість вітру, дійсне число;

Вихідні дані: категорія торнадо за шкалою Фудзіти та частота їх виникнення;

### **Проектування програмного модуля:**

Модуль ModulesOzhekhovskyi, що складається з заголовкового файлу (містить оголошення прототипу функції) та файлу вихідного коду (в котрому присутня реалізація функції s\_calculate та функції з завдань 9.1-2-3).

Функція з задачі 9.1 приймає один аргумент в якості вхідних даних, та повертає структуру, категорію торнадо та частота його виникнення.

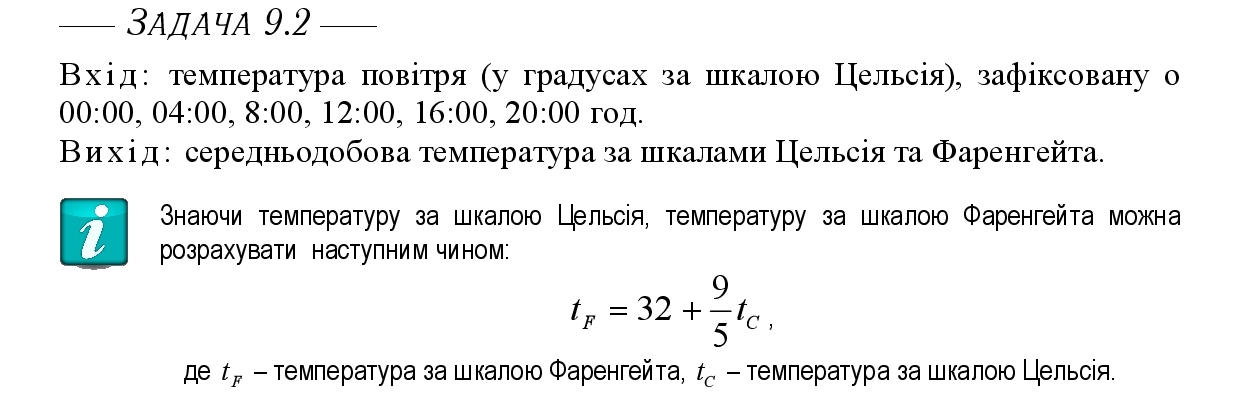


Рисунок 1.2 - Умова задачі 9.2

### **Строга постановка задачі:**

Вхідні дані:

* Температура повітря (у градусах за шкалою Цельсія), зафіксовану о 00:00, 04:00, 08:00, 12:00, 16:00, 20:00.

Вихідні дані:

* Середньодобова температура за шкалами Цельсія та Фаренгейта.

### **Проектування програмного модуля:**

Функція приймає шість аргументів – температура повітря в певні години дня та повертає структуру середньодобової температури за шкалами Цельсія та Фаренгейта.

Всі введені температури сумують та діляться на кількість зафіксованих температур.

Для визначення середньодобової температури за шкалою Фаренгейта використовуємо рівняння.

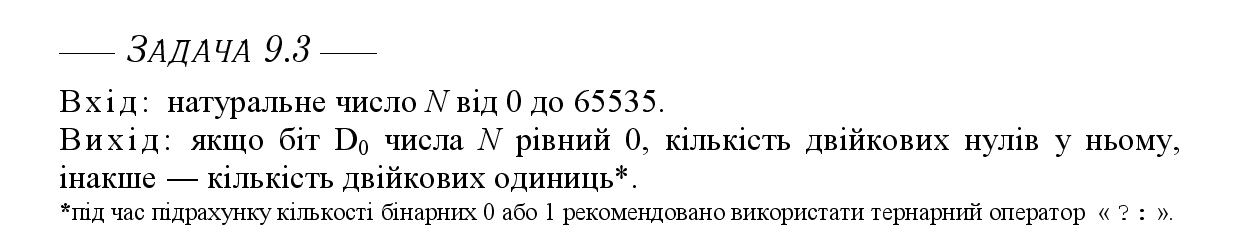


Рисунок 1.3 - Умова задачі 9.3

### **Строга постановка задачі:**

Вхідні та вихідні дані вичерпно описані в умові завдання.

### **Проектування програмного модуля:**

Функція приймає ціле число від 0 до 65535 та повертає кількість «1» або «0» у двійковому представленні цього числа в залежності від того, яким буде біт D0. Для реалізації функції потрібно представити число у бітовому вигляді, перевірити статус D0 та в залежності від нього повернути кількість одиниць або нулів.

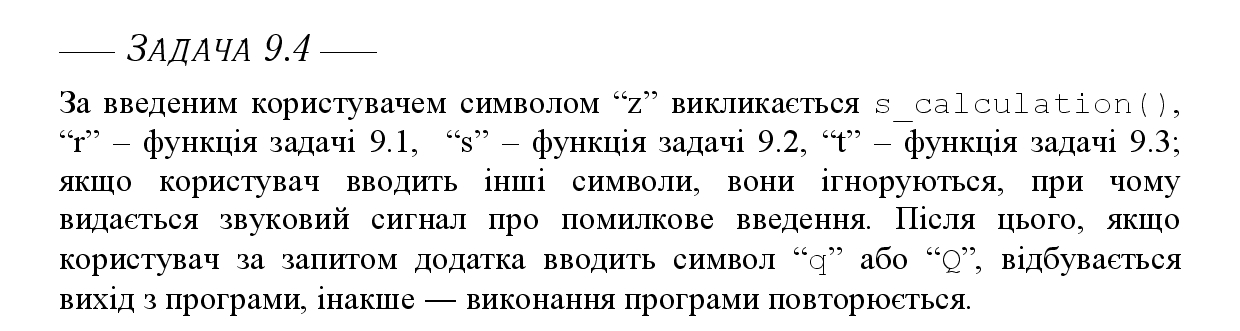


Рисунок 1.4 - Умова задачі 9.4

### **Строга постановка задачі:**

Вхідні дані:

* символьний літерал (із зазначеного переліку), що вводить користувач.

Вихідні дані:

* результат виконання, відповідної до виклику літералом, функції.

### **Проектування програмного модуля:**

Оголошуємо змінну, що буде записувати символ який ми введемо.

Використовуючи оператор множинного вибору, що обробляє значення, введене з клавіатури, реалізуємо різноманітні шляхи виконання функції. Шляхи виконання визначені в умові, як і літерал, введення якого обумовлює вихід з вічного циклу.

**Висновок:**

Дана лабораторна робота була націлена на набуття навичок

у реалізації технології модульного програмування, застосуванні операторів C/С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

До вихідного коду статичної бібліотеки, створеної під час виконання лабораторної роботи №8, було додано реалізації функцій для розв’язування задач 9.1-9.3 та створено нові структури для реалізації повернення функціями даних складного типу. Після компіляції проекту бібліотеки в теці \obj оновився файл з розширенням .а, що і є файлом статичної бібліотеки. Наступним кроком було додавання до заголовкового файлу бібліотеки прототипів функцій та нових структур, які повертаються функціями.

Робота була виконана за допомогою дистрибутиву C++ та кросплатформованого IDE – Code::blocks.

Перша задача потребувала в залежності від вхідного значення за таблицею повертати відповідний бал (ціле число) та опис цього балу (рядок). Вхідне значення береться за модулем через природу показника. Для коректної роботи функції було задекларовано структур даних, яка складається з цілого числа та рядка, що відповідає типу даних, які мають повертатися функцією.

Друга задача складалася в тому, щоб з дійсних чисел, які відображають середньодобові показники температури в певному місяці, знайти середнє арифметичне та виразити його в градусах Цельсія та Фаренгейту (останній розраховується за формулою). Таким чином, новий тип даних, який повертає функція, складається з набору значень у відповідних системах числення (два дійсних числа).

У третій задачі потрібно було у цілому невід’ємному числі, переведенному у двійкову систему, перевірити значення певної за рахунком цифри (у даній реалізації було використано нестандартний тип даних bitset, що може репрезентувати двійковий запис цілого числа; таким чином перевіряється певний біт), і в залежності від цього значення («1» чи «0») потрібно було підрахувати кількість одиниць або нулів у двійковому записі числа (довжина запису бралася фіксована, розрахована для типу даних int – 32 біти). У класичному розв'язку потрібно ітераційно перевіряти ділимість числа на 2 та відповідно ділити його, зберігаючи кількість тих чи інших значень, проте тип даних bitset має відповідний набір функцій, які роблять це автоматично. Так як послідовніть порядків рахується з 0 зправа наліво, операція [ ] відносно bitset знаходить значення елементу у наборі зправа наліво. Також існує функція count, яка для bitset підраховує кількість елементів, які відповідають значенню true («1»), тож щоб підрахувати кількість «0», треба від загальної кількості бітів відняти кількість одиниць.

Було створено проект консольного додатка С++ під назвою TestDriver, метою створення якого є реалізація модульного тестування. Для кожної функції було розраховано еталонні вхідні дані та відповідні еталонні результати, які порівнювалися з результатами, що повертаються функціями.

Результат unit test–у всіх тест кейсів є Passed. Автоматизоване тестування проведено успішно.

Після цього було створено ПЗ з реалізацією функцій з наступного завдання. Важливою умовою було проектування функцій з інтерфейсом, який підходить для виводу у вихідний поток (cout). Задача потребувала при вводі певного символу запустити потрібну функцію (включаючи функцію s\_calculation(), розроблену раніше) або реалізувати завершення роботи програмного засобу. Для кожної з функцій було реалізовано інтерфейс вводу значень для функцій та виведення результату її роботи. Після завершення роботи функції усі попередні дії повторюються.

**ДОДАТОК А**

(TestSuite до завдання 9.1-2-3)

|  |  |  |
| --- | --- | --- |
|  | Назва тестового набору /  **Test Suite Description** | TS\_9\_1 |
|  | Назва проекта/ПЗ /  **Name of project** | TestDriver.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Ожеховський Владислав |
|  | Виконавець /  **Implementer** | Ожеховський Владислав |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Actions** | **Expected Result** | **Test Result** |
| **TS1\_01** | speedWind = 223 | Incorrect value entered | **Passed** |
| **TS1\_02** | speedWind = 78 | Tornado category: F0  Frequency: 38,9% | **Passed** |
| **TS1\_03** | speedWind = 346 | Tornado category: F4  Frequency: 1,1% | **Passed** |
| **TS2\_01** | Enter the wind temperature at: 00:00 = 3  04:00 = 2  08:00 = 4  12:00 = 9  16:00 = 14  20:00 = 10 | Average daily temperature on the Celsius scale: 7, Fahrenheit scale: 44.6 | **Passed** |
| **TS2\_02** | Enter the wind temperature at: 00:00 = 4  04:00 = 2  08:00 = 7  12:00 = 12  16:00 = 16  20:00 = 12 | Average daily temperature on the Celsius scale: 8.83, Fahrenheit scale: 47.9 | **Passed** |
| **TS2\_03** | Enter the wind temperature at: 00:00 = -5  04:00 = -6  08:00 = -2  12:00 = 0  16:00 = -1  20:00 = -2 | Average daily temperature on the Celsius scale: -2.67, Fahrenheit scale: 27.2 | **Passed** |
| **TS3\_01** | num = 13792 | 7 | **Passed** |
| **TS3\_02** | num = 236 | 5 | **Passed** |
| **TS3\_03** | num = 45422 | 9 | **Passed** |

**ДОДАТОК Б**

(TestSuite до завдання 9.4)

|  |  |  |
| --- | --- | --- |
|  | Назва тестового набору /  **Test Suite Description** | TS\_9\_4 |
|  | Назва проекта/ПЗ /  **Name of project** | Ozhekhovskyi\_task.ехе |
|  | Рівень тестування /  **Level of testing** | Системний |
|  | Автор тест-сьюта /  **Test Suite Author** | Ожеховський Владислав |
|  | Виконавець /  **Implementer** | Ожеховський Владислав |

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': | **Passed** |
| **TS\_02** | 1. Ввести ‘r’ 2. Ввести 223 | Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': r  function tornadoCategory():  Enter speed wind: 223  Result of tornadoCategory(): F2 | **Passed** |
| **TS\_03** | 1. Ввести ‘q’ | Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': q  Process returned 0 (0x0) execution time : 87.655 s  Press any key to continue. | **Passed** |
| **TS\_04** | 1. Ввести ‘d’ | Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': d  «звуковий сигнал»  Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': q | **Passed** |
| **TS\_05** | 1. Ввести s 2. Ввести 3 3. Ввести 5 4. Ввести 4 5. Ввести 6 6. Ввести 2 7. Ввести 8 | Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': s  function averageDailyTemp():  Enter the temperature at 0:00 : 3  Enter the temperature at 4:00 : 5  Enter the temperature at 8:00 : 4  Enter the temperature at 12:00 : 6  Enter the temperature at 16:00 : 2  Enter the temperature at 20:00 : 8  Result of averageDailyTemp(): Average daily temperature on the Celsius scale: 4.66, Fahrenheit scale: 40.4 | **Passed** |
| **TS\_06** | 1. Ввести z 2. Ввести 4 3. Ввести 3 4. Ввести -5 | Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': z  function s\_calcuation():  Enter x: 4  Enter y: 3  Enter z: -5  Result of s\_calculation(): 728.483 | **Passed** |
| **TS\_07** | 1. Ввести t 2. Ввести 6742 | Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': t  function bitsNum():  Enter integer number 1 - 65535: 6742  Result of bitsNum(): 7 | **Passed** |

**ДОДАТОК В**

(Вихідний код завдання 9.1-2-3)

**… / ModulesOzhekhovskyi / main.cpp**

#include "ModulesOzhekhovskyi.h"

#include <bitset>

#include <iostream>

#include <sstream>

using namespace std;

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

return ( sin(x) / sqrt( fabs( (y \* z) / (x + y) ) ) ) + (3 \* pow(y, 5));

}

tornado tornadoCategory(double speedWind){

if(speedWind < 64 || speedWind > 512){

return {-1, "Incorrect value entered"};

} else if(speedWind <= 116){

return {0, "Tornado category: F0, Frequency: 38,9%"};

} else if(speedWind <= 180){

return {1, "Tornado category: F1, Frequency: 35,6%"};

} else if(speedWind <= 253){

return {2, "Tornado category: F2, Frequency: 19,4%"};

} else if(speedWind <= 332){

return {3, "Tornado category: F3, Frequency: 4,9%"};

} else if(speedWind <= 418){

return {4, "Tornado category: F4, Frequency: 1,1%"};

} else if(speedWind <= 512){

return {5, "Tornado category: F5, Frequency: <0,1%"};

}

}

averageTemp averageDailyTemp(double sumTempC){

double averageTempC = floor((sumTempC / 6) \* 100) / 100;

double averageTempF = floor(( sumTempC / 6 \* 1.8 + 32 ) \* 100) / 100;

stringstream ss;

ss << "Average daily temperature on the Celsius scale: "

<< averageTempC

<< ", Fahrenheit scale: "

<< averageTempF;

return {averageTempC, averageTempF, ss.str()};

}

int bitsNum(unsigned short int num)

{

if (num < 0 || num > 65535) {

cout << "ERROR: number in invalid range!" << endl;

return -1;

}

bitset<32> bNum{num};

if (bNum[0]) {

return 32 - bNum.count();

}

return bNum.count();

}

**… / ModulesOzhekhovskyi / ModulesOzhekhovskyi.h**

#ifndef MODULESOZHEKHOVSKYI\_H\_INCLUDED

#define MODULESOZHEKHOVSKYI\_H\_INCLUDED

#include <cmath>

#include <string>

struct tornado {

int category = -1;

std::string description = "";

};

struct averageTemp{

double averageTempC;

double averageTempF;

std::string description = "";

};

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

tornado tornadoCategory(double speedWind);

averageTemp averageDailyTemp(double sumTempC);

int bitsNum(unsigned short int num);

#endif // MODULESOZHEKHOVSKYI\_H\_INCLUDED

**ДОДАТОК Г**

**… / TestDriver / main.cpp**

#include <iostream>

#include "ModulesOzhekhovskyi.h"

#include <conio.h>

using namespace std;

int main()

{

int speedWindInput[3] = {223, 78, 346};

int expectedResult[3] = {2, 0, 4};

for(int i = 0; i < 3; i++){

cout << "TC01\_0" << i + 1 << ":" << endl;

cout << "Speed wind = " << speedWindInput[i] << endl;

cout << "Expected result: F" << expectedResult[i] << endl;

cout << "The result obtained: " << tornadoCategory(speedWindInput[i]).description << endl;

cout << "Test result: "

<< ((expectedResult[i] == tornadoCategory(speedWindInput[i]).category)? "Passed" : "Failed")

<< endl << endl;

}

int tempInput[3][6] = {3, 2, 4, 9, 14, 10,

4, 2, 7, 12, 16, 12,

-5, -6, -2, 0, -1, -2};

double expectedResult2[3] = {7, 8.83, -2.67};

for(int i = 0; i < 3; i++){

double sumTemp = tempInput[i][0] + tempInput[i][1] + tempInput[i][2] +

tempInput[i][3] + tempInput[i][4] + tempInput[i][5];

cout << "TC02\_0" << i + 1 << ":" << endl;

cout << "Sum temperature = " << sumTemp << endl;

cout << "Expected result on the Celsius scale: " << expectedResult2[i] << endl;

cout << "The result obtained: " << endl << averageDailyTemp(sumTemp).description << endl;

cout << "Test result: "

<< ((expectedResult2[i] == averageDailyTemp(sumTemp).averageTempC)? "Passed" : "Failed")

<< endl << endl;

}

unsigned short int numInput[3] = {13792, 236, 45422};

unsigned short int expectedResult3[3] = {7, 5, 9};

for(int i = 0; i < 3; i++){

cout << "TC03\_0" << i + 1 << ":" << endl;

cout << "Num input = " << numInput[i] << endl;

cout << "Expected result: " << expectedResult3[i] << endl;

cout << "The result obtained: " << bitsNum(numInput[i]) << endl;

cout << "Test result: "

<< ((expectedResult3[i] == bitsNum(numInput[i]))? "Passed" : "Failed")

<< endl << endl;

}

getch();

}

**ДОДАТОК Д**

(Вихідний код завдання 9.4)

**… / Ozhekhovskyi\_task / main.cpp**

#include <iostream>

#include <conio.h>

#include "ModulesOzhekhovskyi.h"

using namespace std;

int main()

{

char sym;

do{

cout << endl << "Enter one of the symbols 'z', 'r', 's', 't', 'q', 'Q': ";

cin >> sym;

switch (sym) {

case 'z':

cout << endl << "function s\_calcuation(): " << endl;

double x, y, z;

cout << "Enter x: ";

cin >> x;

cout << "Enter y: ";

cin >> y;

cout << "Enter z: ";

cin >> z;

cout << "Result of s\_calculation(): "

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

break;

case 'r':

cout << endl << "function tornadoCategory():" << endl;

double speedWind;

cout << "Enter speed wind: ";

cin >> speedWind;

cout << "Result of tornadoCategory(): F"

<< tornadoCategory(speedWind).category << endl;

break;

case 's':

cout << endl << "function averageDailyTemp():" << endl;

double temp;

unsigned short int time;

double sumTemp;

for (int i = 0; i < 6; i++) {

cout << "Enter the temperature at " << time << ":00 : ";

cin >> temp;

sumTemp += temp;

time += 4;

}

cout << "Result of averageDailyTemp(): "

<< averageDailyTemp(sumTemp).description << endl;

break;

case 't':

cout << endl;

cout << "function bitsNum():" << endl;

unsigned short int num;

cout << "Enter integer number 1 - 65535: ";

cin >> num;

cout << "Result of bitsNum(): " << bitsNum(num) << endl;

break;

default:

cout << "\a";

}

}while (sym != 'q' && sym != 'Q');

}