Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9

з навчальної дисципліни

“Базові методології та технології програмування”

Реалізація статичних бібліотек модулів

лінійних обчислювальних процесів

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КН-23

Гончаренко В.В.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Ганна Дрєєва.

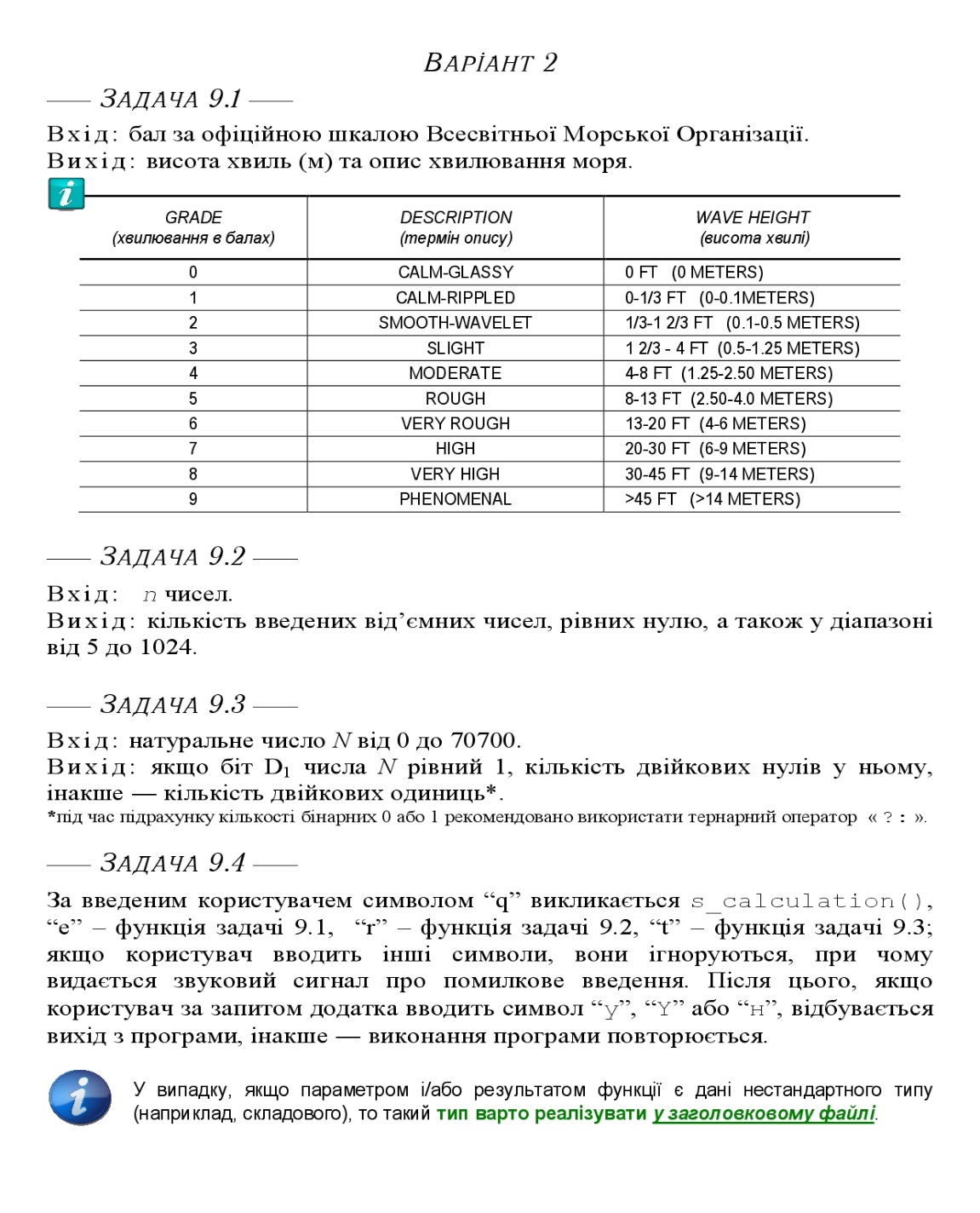
Кропивницький – 2024

ЛАБОРАТОРНА РОБОТА № 9

**Реалізація програмних модулів та ітервційних обчислювальних процесів**

Мета: 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.

**ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ**

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****

**-- Задача 9.1 –**

Вхід: бал за офіційною шкалою Всесвітньої Морської Організації.

Вихід: висота хвиль (м) та опис хвилювання моря.

**-- Лістининг –**

void calcWave(double grade, double& waveHeight, std::string& seaDescription) {

if (grade < 0 || grade > 9) {

waveHeight = 0;

seaDescription = "Некоректний бал за шкалою ВМО.";

} else {

if (grade == 0) {

waveHeight = 0;

seaDescription = "CALM-GLASSY";

} else if (grade == 1) {

waveHeight = 0.1;

seaDescription = "CALM-RIPPLED";

} else if (grade == 2) {

waveHeight = 0.5;

seaDescription = "SMOOTH-WAVELET";

} else if (grade == 3) {

waveHeight = 1.25;

seaDescription = "SLIGHT";

} else if (grade == 4) {

waveHeight = 2.50;

seaDescription = "MODERATE";

} else if (grade == 5) {

waveHeight = 4.0;

seaDescription = "ROUGHT";

} else if (grade == 6) {

waveHeight = 6.0;

seaDescription = "VERY ROUGHT";

} else if (grade == 7) {

waveHeight = 9.0;

seaDescription = "HIGHT";

} else if (grade == 8) {

waveHeight = 14.0;

seaDescription = "VERY HIGHT";

} else if (grade == 9) {

waveHeight = 14.0;

seaDescription = "PHENOMENAL";

}

}

}

//Task 9.1

double grade, waveHeight;

std::string seaDescription;

cout << "Введіть бал ( 0 - 9 ): ";

cin >> grade;

calcWave(grade, waveHeight, seaDescription);

cout << "Висота хвилі: " << waveHeight << "метрів\n";

cout << "Опис: " << seaDescription << "\n\n";

**-- Задача 9.2 –**

Вхід: n чисел.

Вихід: кількість введених від’ємних чисел, рівних нулю, а також у діапазоні від 5 до 1024.

**-- Лістининг –**

void countNumbers(int n, int& negativeCount, int& zeroCount, int& rangeCount) {

negativeCount = 0;

rangeCount = 0;

zeroCount = 0;

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

int num;

cout << "Введіть число " << (i + 1) << ": ";

cin >> num;

if (num < 0) {

negativeCount++;

} else if (num == 0) {

zeroCount++;

} else if (num >= 5 && num <= 1024) {

rangeCount++;

}

}

}

//Task 9.2

int n, negativeCount, zeroCount, rangeCount;

cout << "Введіть n кількість чисел: ";

cin >> n;

countNumbers(n, negativeCount, zeroCount, rangeCount);

cout << "Від'ємні числа: " << negativeCount << endl;

cout << "Числа які = нулю: " << zeroCount << endl;

cout << "Діапазон від [5 - 1024]: " << rangeCount << endl;

**-- Задача 9.3 –**

Вхід: натуральне число N від 0 до 70700.

Вихід: якщо біт D1 числа N рівних 1, кількість двійкових нулів у ньому, інакше - кількість двійкових одиниць.

**-- Лістининг –**

void calcBinaryInfo(unsigned int N, bool& D1, int& binaryCount) {

D1 = (N & 1) == 1;

binaryCount = 0;

while (N > 0) {

if (D1) {

binaryCount += (N % 2 == 0) ? 1 : 0;

} else {

binaryCount += (N % 2 == 1) ? 1 : 0;

}

N /= 2;

}

}

//Task 9.3

unsigned int N;

bool D1;

int binaryCount;

cout << "Введіть число N: ";

cin >> N;

calcBinaryInfo(N, D1, binaryCount);

if (D1) {

cout << "Біт D1 числа N рівний 1." << endl;

cout << "Кількість двійкових нулів у числі N: " << binaryCount << endl;

} else {

cout << "Біт D1 числа N рівний 0." << endl;

cout << "Кількість двійкових одиниць у числі N: " << binaryCount << endl;

} return 0;

**-- Задача 9.4 –**

За введеним користувачем символом «q» викликається s\_caclulation(), “e” – функція задачі 9.1, «r» - фунцыя задачі 9.2, «t» - функція задачі 9.3; якщо користувач вводить інші символи, вони ігноруються, при чому видається звуковий сигнал про помилкове введеня. Після цього, якщо користувач за запитом додатка вводить символ «y», «Y» або «н», відбувається вихід з програми, інакше – виконання програми повторюється..

**Аналіз вимог до ПЗ:**

**Функціональні вимоги:**

* Користувач може вводити символи «q», «e», «r», «t».
* При введенні символу «q» викликається функція s\_calculation().
* При введенні символу «e» викликається функція задачі 9.1.
* При введенні символу «r» викликається функція задачі 9.2.
* При введенні символу «t» викликається функція задачі 9.3.
* В іншому випадку символи ігноруються, а користувач отримує звуковий сигнал.
* Після обробки введеного символу, користувач має можливість вийти з програми, введіть символи «y», «Y» або «н».

**-- Лістининг –**

int main() {

setlocale(LC\_ALL, "ukr");

char choice;

bool exit = false;

while (!exit) {

cout << "Введіть символ ('q' - s\_caclulation, 'e' - задача 9.1, 'r' - задача 9.2, 't' - задача 9.3, 'y' або 'н' - вихід): ";

cin >> choice;

switch (choice) {

case 'q': {

double x, y, z;

cout << "Введіть значення x, y та z: ";

cin >> x >> y >> z;

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

break;

}

case 'e': {

double grade, waveHeight;

string seaDescription;

cout << "Введіть бал (0 - 9): ";

cin >> grade;

calcWave(grade, waveHeight, seaDescription);

cout << "Висота хвилі: " << waveHeight << " метрів\n";

cout << "Опис: " << seaDescription << "\n\n";

break;

}

case 'r': {

int n, negativeCount, zeroCount, rangeCount;

cout << "Введіть n кількість чисел: ";

cin >> n;

countNumbers(n, negativeCount, zeroCount, rangeCount);

cout << "Від'ємні числа: " << negativeCount << endl;

cout << "Числа, які = нулю: " << zeroCount << endl;

cout << "Діапазон від [5 - 1024]: " << rangeCount << endl;

break;

}

case 't': {

unsigned int N;

bool D1;

int binaryCount;

cout << "Введіть число N: ";

cin >> N;

calcBinaryInfo(N, D1, binaryCount);

cout << "Біт D1 числа N: " << (D1 ? "1" : "0") << endl;

cout << "Кількість двійкових " << (D1 ? "нулів" : "одиниць") << " у числі N: " << binaryCount << endl;

break;

}

case 'y':

case 'н': {

exit = true;

break;

}

default: {

Beep(500, 300); // Звуковий сигнал у разі неправильного введення

cout << "Неправильний ввід! Будь ласка, введіть 'q', 'e', 'r', 't', 'y' або 'н'.\n";

break;

}

}

}

return 0;

}

**Висновок:** 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.

Функція calcWave (задача 9.1):

* Ця функція визначає висоту хвиль та описує морський стан в залежності від введеного балу.
* Результат роботи функції залежить від вхідного значення параметру grade.
* Функція працює із рядками для надання опису морського стану.
* Виклик функції передбачає взаємодію з користувачем, який вводить бал.

Функція countNumbers (задача 9.2):

* Ця функція рахує кількість введених від'ємних чисел, кількість нулів та кількість чисел у заданому діапазоні.
* Виклик функції передбачає взаємодію з користувачем, який вводить числа.

Взаємодія з користувачем (задача 9.3):

* Програма реалізує можливість вибору виконання певної функції в залежності від введених користувачем символів.
* Обробка неправильного вводу реалізована шляхом ігнорування неправильних символів та видачі звукового сигналу про помилку.
* Користувач може вибрати завершення програми або продовження виконання через введення відповідних символів.

Ці задачі демонструють різні аспекти розробки програмного забезпечення, включаючи роботу з функціями, обробку введення користувача та взаємодію з ним, а також створення тест-сьютів для перевірки правильності роботи програми.