Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 10

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ОБРОБЛЕННЯ ДАНИХ СКЛАДОВИХ ТИПІВ З ФАЙЛОВИМ ВВЕДЕННЯМ/ВИВЕДЕННЯМ

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

Гребенюк Д. О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Дрєєва Г. М.

Кропивницький – 2024

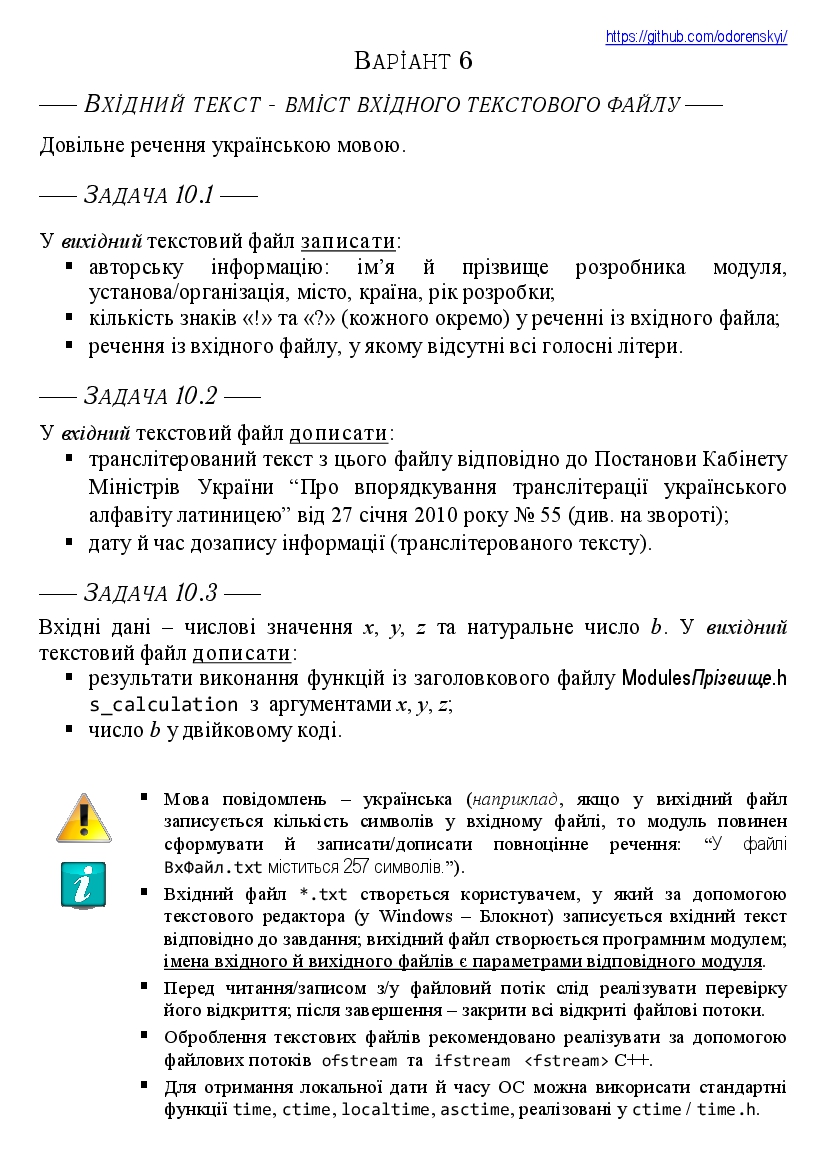
**Тема:** Реалізація програмних модулів оброблення даних складових типів з файловим введенням/виведенням

**Завдання до лабораторної роботи:**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище лабораторних робіт №8–9).

2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3.

**Варіант 6**



**Задача 10.1**

Вхідні дані: inputFileName, outputFileName – назва вхідного та вихідного файлу

Вихідні дані: вихідний файл з вказаною назвою та його зміст: авторська інформація; кількість знаків «!» та «?»; речення, у якому відсутні всі голосні літери.

Для роботи з файлами потрібно використати файлові потоки ofstream та ifstream з <fstream>. Зміст вхідного файлу зчитується в змінну за допомогою функції getline, а потім обробляється відповідно до завдань. Спочатку обраховується кількість знаків за допомогою циклу, а потім в тому ж циклі перевіряється літера з переліку голосних літер. У випадку, якщо літера дійсно голосна – не додавати її в змінну з новим змістом файлу. В кінці подати інформацію українською мовою (за вимогою завдання), закрити файлові потоки.

**Задача 10.2**

Вхідні дані: inputFileName – назва вхідного файлу

Вихідні дані: транслітерований текст з файлу, дата й час дозапису тексту

В цій задачі потрібно дописати дані у вхідний файл, а не окремо у вихідний. Для реалізації транслітерації можна використати рядковий масив, індекси якого будуть відповідати коду символу в кодуванні Windows-1251. Відповідно, аналог на латині записуєтсья під індексом (кодом) кожної букви українського алфавіту. Для визначення дати можна застосувати <ctime>, а саме функції time та ctime.

**Задача 10.3**

Вхідні дані: x, y, z – числові значення; b – натуральне число; outputFileName – назва вихідного файлу

Вихідні дані: результат виконання функції s\_calculation з модуля ModulesHrebeniuk та число b у двійковому коді

Так само, як в попередніх задачах, використовується файловий потік ofstream для дозапису у файл. З модуля ModulesHrebeniuk використовується функція s\_calculation. Для переводу числа b у двійковий код використовується клас bitset, що надається заголовковим файлом <bitset>.

*Реалізація завдань 10.1-10.3 знаходиться у лабораторній роботі №8, в теці prj/ModulesHrebeniuk.*

*Результат виконання тестового драйвера:*

=== TS\_10\_1 ===

Тест-кейс №1

Вхідні дані:

inputFileName: input\_1.txt

outputFileName: output\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

inputFileName: input\_2.txt

outputFileName: output\_2.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

inputFileName: input\_3.txt

outputFileName: output\_3.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

inputFileName: input\_1.txt

outputFileName: output\_4.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

inputFileName: input\_4.txt

outputFileName: output\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

=== TS\_10\_2 ===

Тест-кейс №1

Вхідні дані:

inputFileName: input\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

inputFileName: input\_2.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

inputFileName: input\_3.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

inputFileName: input\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

inputFileName: input\_4.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

=== TS\_10\_3 ===

Тест-кейс №1

Вхідні дані:

x: 5

y: -27

z: -21

b: 90

outputFileName: output\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

x: -20

y: 28

z: -24

b: 2024

outputFileName: output\_2.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

x: 27

y: 21

z: 15

b: 912012

outputFileName: output\_3.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

x: 15

y: 4

z: 14

b: 1007

outputFileName: output\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

x: 27

y: -7

z: -25

b: 128

outputFileName: output\_4.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

*Лістинг проекту ModulesHrebeniuk:*

#include <cmath>

#include <fstream>

#include <ctime>

#include <bitset>

#include "ModulesHrebeniuk.h"

using namespace std;

// Laboratory work №8

float s\_calculation(int x, int y, int z) {

float S = M\_PI \* x + pow(M\_E, abs(y)) - sqrt(abs(pow(z, 2) - y));

return S;

}

// Laboratory work №9

float calculateElectricityConsume(int kWh) {

float sum = 0.0;

if (kWh > 0 && kWh <= 150) {

sum = kWh \* 130.843;

} else if (kWh > 150 && kWh <= 800) {

sum = kWh \* 241.945;

} else if (kWh > 800) {

sum = kWh \* 534.047;

}

return sum;

}

Temperature averageYearTemperature(int averageMonthTemperature[MONTH\_IN\_YEAR]) {

Temperature averageYearTemperature;

int totalMonthTemperature = 0;

for (int i = 0; i < MONTH\_IN\_YEAR; ++i) {

totalMonthTemperature += averageMonthTemperature[i];

}

averageYearTemperature.celsius = (float)totalMonthTemperature / MONTH\_IN\_YEAR;

averageYearTemperature.fahrenheit =

32 + (9.0 / 5) \* averageYearTemperature.celsius;

return averageYearTemperature;

}

unsigned int countBinaryZerosOrOnes(unsigned int N) {

if (N > 0 && N < 5740500) {

unsigned int tenthBit = (N >> 9) & 1;

unsigned int result = 0;

for (int i = 0; i < sizeof(N) \* 8; i++) {

result += ((N & 1) == tenthBit) ? 1 : 0;

N = N >> 1;

}

return result;

} else {

return -1;

}

}

// Laboratory work №10

void taskTenthPointOne(string inputFileName, string outputFileName) {

ifstream inputFile(inputFileName);

if (!inputFile) {

return;

}

ofstream outputFile(outputFileName, ios::out | ios::trunc);

outputFile << " ---------------------------------------------------------- " << endl;

outputFile << "| Гребенюк Денис |" << endl;

outputFile << "| Центральноукраїнський національний технічний університет |" << endl;

outputFile << "| м. Кропивницький, Україна |" << endl;

outputFile << "| 2024 |" << endl;

outputFile << " ------------------ © All rights reserved ----------------- " << endl;

outputFile << endl;

int exclamationMarkCount = 0, questionMarkCount = 0;

string vowels = "аеиіоуяюєї", line, inputFileContent, newOuputFileContent;

while (getline(inputFile, line)) {

inputFileContent += line + "\n";

}

inputFile.close();

for (int i = 0; i < inputFileContent.length(); i++) {

if (inputFileContent[i] == '!') {

exclamationMarkCount++;

} else if (inputFileContent[i] == '?') {

questionMarkCount++;

}

if (vowels.find(inputFileContent[i]) == string::npos) {

newOuputFileContent += inputFileContent[i];

}

}

outputFile << "У файлі " << inputFileName << " міститься " << exclamationMarkCount << " знаків оклику." << endl;

outputFile << "У файлі " << inputFileName << " міститься " << questionMarkCount << " знаків запитання." << endl;

outputFile << "Речення з файлу " << inputFileName << " без голосних літер:" << endl << newOuputFileContent << endl;

outputFile.close();

}

void taskTenthPointTwo(string inputFileName) {

string transliteration[256];

transliteration[192] = "A";

transliteration[193] = "B";

transliteration[194] = "V";

transliteration[195] = "H";

transliteration[165] = "G";

transliteration[196] = "D";

transliteration[197] = "E";

transliteration[170] = "Ye";

transliteration[198] = "Zh";

transliteration[199] = "Z";

transliteration[200] = "Y";

transliteration[178] = "I";

transliteration[175] = "Yi";

transliteration[201] = "Y";

transliteration[202] = "K";

transliteration[203] = "L";

transliteration[204] = "M";

transliteration[205] = "N";

transliteration[206] = "O";

transliteration[207] = "P";

transliteration[208] = "R";

transliteration[209] = "S";

transliteration[210] = "T";

transliteration[211] = "U";

transliteration[212] = "F";

transliteration[213] = "Kh";

transliteration[214] = "Ts";

transliteration[215] = "Ch";

transliteration[216] = "Sh";

transliteration[217] = "Shch";

transliteration[222] = "Yu";

transliteration[223] = "Ya";

transliteration[224] = "a";

transliteration[225] = "b";

transliteration[226] = "v";

transliteration[227] = "h";

transliteration[180] = "g";

transliteration[228] = "d";

transliteration[229] = "e";

transliteration[186] = "ie";

transliteration[230] = "zh";

transliteration[231] = "z";

transliteration[232] = "y";

transliteration[179] = "i";

transliteration[191] = "i";

transliteration[233] = "i";

transliteration[234] = "k";

transliteration[235] = "l";

transliteration[236] = "m";

transliteration[237] = "n";

transliteration[238] = "o";

transliteration[239] = "p";

transliteration[240] = "r";

transliteration[241] = "s";

transliteration[242] = "t";

transliteration[243] = "u";

transliteration[244] = "f";

transliteration[245] = "kh";

transliteration[246] = "ts";

transliteration[247] = "ch";

transliteration[248] = "sh";

transliteration[249] = "shch";

transliteration[254] = "iu";

transliteration[255] = "ia";

ifstream inputFile(inputFileName);

if (!inputFile) {

return;

}

string line, inputFileContent, newOuputFileContent;

while (getline(inputFile, line)) {

inputFileContent += line + "\n";

}

inputFile.close();

for (int i = 0; i < inputFileContent.length(); i++) {

if (inputFileContent[i] == 'Ь' || inputFileContent[i] == 'ь' || inputFileContent[i] == '\'') {

continue;

}

if ((i - 1) >= 0) { // перевірка, чи існує попередня літера

if (

(inputFileContent[i] == 'г' || inputFileContent[i] == 'Г') &&

(inputFileContent[i - 1] == 'з' || inputFileContent[i - 1] == 'З')

) {

newOuputFileContent += 'g';

}

}

string transliteratedLetter = transliteration[(unsigned char)inputFileContent[i]];

if (transliteratedLetter.empty()) {

newOuputFileContent += inputFileContent[i];

} else {

newOuputFileContent += transliteratedLetter;

}

}

ofstream outputFile(inputFileName, ios::out | ios::app);

outputFile << endl << newOuputFileContent;

time\_t curr\_time;

curr\_time = time(NULL);

outputFile << "Дата й час дозапису інформації: " << ctime(&curr\_time);

outputFile.close();

}

void taskTenthPointThree(int x, int y, int z, unsigned int b, string outputFileName) {

ofstream outputFile(outputFileName, ios::out | ios::app);

outputFile << "Результат виконання функції s\_calculation: " << s\_calculation(x, y, z) << endl;

outputFile << "Число b у двійковому коді: " << bitset<32>(b).to\_string() << endl << endl;

outputFile.close();

}

*Лістинг проекту TestDriver:*

#include <iostream>

#include <conio.h>

#include <string>

#include "ModulesHrebeniuk.h"

using namespace std;

#define TEST\_CASES\_COUNT 5

void testFirstTask(int test\_case, string inputFileName, string outputFileName) {

cout << "Тест-кейс №" << test\_case << endl;

cout << "Вхідні дані:\ninputFileName: " << inputFileName << "\noutputFileName: " << outputFileName << endl;

taskTenthPointOne(inputFileName, outputFileName);

cout << "Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n): " ;

char checkResult = getch();

if (checkResult == 'y' || checkResult == 'Y') {

cout << endl << "Статус тест-кейса: passed" << endl;

} else {

cout << endl << "Статус тест-кейса: failed" << endl;

}

cout << endl;

}

void testSecondTask(int test\_case, string inputFileName) {

cout << "Тест-кейс №" << test\_case << endl;

cout << "Вхідні дані:\ninputFileName: " << inputFileName << endl;

taskTenthPointTwo(inputFileName);

cout << "Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n): " ;

char checkResult = getch();

if (checkResult == 'y' || checkResult == 'Y') {

cout << endl << "Статус тест-кейса: passed" << endl;

} else {

cout << endl << "Статус тест-кейса: failed" << endl;

}

cout << endl;

}

void testThirdTask(int test\_case, int x, int y, int z, unsigned int b, string outputFileName) {

cout << "Тест-кейс №" << test\_case << endl;

cout << "Вхідні дані: " << endl;

cout << "x: " << x << endl;

cout << "y: " << y << endl;

cout << "z: " << z << endl;

cout << "b: " << b << endl;

cout << "outputFileName: " << outputFileName << endl;

taskTenthPointThree(x, y, z, b, outputFileName);

cout << "Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n): " ;

char checkResult = getch();

if (checkResult == 'y' || checkResult == 'Y') {

cout << endl << "Статус тест-кейса: passed" << endl;

} else {

cout << endl << "Статус тест-кейса: failed" << endl;

}

cout << endl;

}

int main()

{

system("chcp 1251 & cls");

// Task 10.1

cout << "=== TS\_10\_1 ===" << endl;

const string firstTaskTCData[TEST\_CASES\_COUNT][2] = {

{"input\_1.txt", "output\_1.txt"},

{"input\_2.txt", "output\_2.txt"},

{"input\_3.txt", "output\_3.txt"},

{"input\_1.txt", "output\_4.txt"},

{"input\_4.txt", "output\_1.txt"}

};

for (int i = 0; i < TEST\_CASES\_COUNT; i++) {

testFirstTask(i + 1, firstTaskTCData[i][0], firstTaskTCData[i][1]);

}

// Task 10.2

cout << "=== TS\_10\_2 ===" << endl;

const string secondTaskTCData[TEST\_CASES\_COUNT] = {

"input\_1.txt",

"input\_2.txt",

"input\_3.txt",

"input\_1.txt",

"input\_4.txt"

};

for (int i = 0; i < TEST\_CASES\_COUNT; i++) {

testSecondTask(i + 1, secondTaskTCData[i]);

}

// Task 10.3

cout << "=== TS\_10\_3 ===" << endl;

const string thirdTaskTCData[TEST\_CASES\_COUNT][5] = {

{"5", "-27", "-21", "90", "output\_1.txt"},

{"-20", "28", "-24", "2024", "output\_2.txt"},

{"27", "21", "15", "912012", "output\_3.txt"},

{"15", "4", "14", "1007", "output\_1.txt"},

{"27", "-7", "-25", "128", "output\_4.txt"}

};

for (int i = 0; i < TEST\_CASES\_COUNT; i++) {

testThirdTask(i + 1, stoi(thirdTaskTCData[i][0]), stoi(thirdTaskTCData[i][1]), stoi(thirdTaskTCData[i][2]), stoi(thirdTaskTCData[i][3]), thirdTaskTCData[i][4]);

}

system("pause");

return 0;

}

**Висновок**

Виконуючи цю лабораторну роботу, я отримав ґрунтовні знання та практичний досвід роботи зі створенням та обробленням різноманітних типів даних, такі як масиви, структури, об'єднання, множини та переліки. Особливу увагу у цій лаборіторній роботі було приділено роботі з файловими потоками, що дало мені змогу попрацювати зі збереженням та зчитуванням даних з файлів.

Згідно методичних рекомендацій потрібно було почергово проаналізувати і постановити задачі 10.1, 10.2 та 10.3, проаналізувати вимоги до ПЗ та до вмісту вхідного файлу, спроектувати архітектуру та задокументувати результати у звіт. В цій лабораторній було додатково вказано, що під час проектування слід враховувати вхідні дані модуля, а саме ім’я вхідного та/або вихідного файлу.

Під час розробки тест-сьютів було обрано напівавтоматизоване unit-тестування програмного модуля. Як й завжди було розроблено три тестових наборів для задач 10.1-10.3, в кожному по п’ять тест-кейсів. На відміну від минулих лабораторних робіт в структуру тестового набору було додано колонку Preliminary Steps (Попередні кроки). В цій колонці описується назва вхідного/вихідного файлу та зміст файлу.

Під час реалізації задачі 10.1-10.3 мовою програмування C++ було опрацьовано роботу з файловим потоком (ofstream, ifstream), запис/дозапис/читання та відкриття/закриття файлу.

Для підрахунку знаків оклику та знаків питання в задачі 10.1 було обрано варіант з ітерацією кожного символу змісту файла. Під час кожної ітерації була перевірка на символ та перевірка чи є символ голосною літерою. Якщо символ не є голосною літерою, то додаємо його до змінної типу string, де зберігається фінальна версія тексту без голосних літер.

Під час роботи над задачею 10.2 потрібно було ознайомитись з постановою КМУ №55 «Про впорядкування транслітерації українського алфавіту латиницею». В цій постанові надається український алфавіт та аналог латиницею. В деяких випадках були такі винятки, що велика літера та маленька по-різному записуються латиницею. Для реалізації цієї задачі я обрав масив розмірністю 256 комірок типу string. З таблиці кодів кожного символу кодування Windows-1251 я відшукав літери українського алфавіту та використав коди в якості індексу. Після розробки такого рішення з’явилась можливість по індексу (тобто коду літери) отримати аналог латиницею. Додатково потрібно було звернути увагу на примітки з буквосполученням «зг» та м’яким знаком/апострофом. Було додано перевірку на м’який знак та апостроф. Для примітки з буквосполученням перевіряється теперішний символ та попередній. У випадку, якщо теперішній символ є літерою «г», а попередній - «з», то додаємо між ними “g”. Кінцем задачі 10.2 було додавання дати й часу дозапису інформації. Для цього було обрано функції time та ctime з заголовкого файлу <ctime>.

Задача 10.3 потребувала тих знань, що були здобуті під час виконання минулих лабораторних робіт. Потрібно було використати функцію s\_calculation з власного модуля ModulesHrebeniuk та дописати число b у двійковому коді. Для переводу числа b у двійковий код я використав клас bitset, що надається заголовковим файлом <bitset>.

Для тестування тестових наборів було реалізовано тестовий драйвер під назвою проекту TestDriver. Було реалізовано протоколювання процесу тестування, а саме: виведення вхідних даних, отриманий від функції результат та статус кожного тест-кейса (passed/failed). Перед виведенням результату тест-кейса у особи, що тестує ПЗ запитується про коректність виконання тест-кейсу. Це було реалізовано через те, що я обрав напівавтоматизоване тестування, де тестувальник вручну виконує певні кроки. Під час виконання роботи тестового драйверу усі тестові кейси з трьох задач були завершені зі статусом passed.

Завдяки цій лабораторній роботі я покращив свої знання та практичні навички у проектуванні та розробці модульних проектів, розробці тестових драйверів, роботі з файловими потоками (запис/читання тощо) та у роботі з різноманітними типами даних. Мені було цікаво виконувати цю лабораторну роботу, дізнаватись нову інформацію та проектувати задачі такого рівня.

**Контрольні запитання і завдання**

**1.** *Поясніть сутність понять області видимості об’єктів та простору імен у мові програмування С++.*

Область видимості об'єкта в C++ визначає, в якій частині програми цей об'єкт доступний і може бути використаний. Простір імен дозволяє групувати пов'язані між собою імена (змінних, функцій, класів тощо) під одним іменем, щоб уникнути конфліктів імен з іншими частинами програми.

**2.** *Яким є час життя локального об’єкта (змінної)? Наведіть приклад оголошення локальної та глобальної змінних одного зі складових типів С/С++.*

Час життя локального об'єкта (змінної) обмежений блоком коду, в якому він був оголошений. Локальна змінна створюється при вході у блок і знищується при виході з нього.

**3.** *Якою є область видимості змінної, оголошеної у циклі С++? Відповідь обґрунтуйте та доведіть на прикладі.*

Змінна, оголошена в циклі, має область видимості, обмежену цим циклом. Вона недоступна за межами циклу.

**4.** *Якими є призначення й синтаксис запису директиви extern? Наведіть приклад її застосування під час реалізації міжмодульних змінних або функцій.*

Директива extern вказує, що змінна або функція визначена десь в іншому файлі. Це дозволяє використовувати цю змінну/функцію в поточному файлі.

**5.** *У чому полягає відмінність між масивом і рядком С/С++ з погляду реалізації їх оброблення?*

Відмінність між масивом і рядком полягає в тому, що рядок в є масивом символів, який завершується нуль-символом ('\0'), що служить позначенням кінця рядка. Це означає, що рядки обробляються спеціальним чином, особливо при роботі зі стандартними функціями для роботи з рядками, такими як strlen, strcpy тощо. В той час як масив може містити будь-які дані (не тільки символи) і не має вбудованого механізму для позначення кінця даних, окрім його розміру.

**6.** *Яким є синтаксис запису та оголошення об’єднання у С/С++?*

Об'єднання дозволяє зберігати різні типи даних в одній області пам'яті. В кожен момент часу в об'єднанні може зберігатися лише одне значення.

Синтаксис:

union MyUnion {

int intValue;

float floatValue;

char charValue;

};

**7.** *Дайте визначення винятку (exception). Як у С++ реалізовується його оброблення?*

Виняток - це спеціальний об'єкт, який сигналізує про виникнення особливої ситуації під час виконання програми. Оброблення винятків реалізується за допомогою конструкцій try, catch і throw.

Приклад:

try { // блок коду, виконання якого може призвести до помилки

// перевірка на помилку

throw N; // збудження винятку з генерацією її коду для catch

} catch (формальний\_параметр\_N) {

// обробник винятку

}

**8.** *Яким є синтаксис оголошення одно- й двовимірних масивів даних у С/С++ та за допомогою якої операції і за якими індексами здійснюється доступ до їх елементів?*

Синтаксис:

- Одновимірний масив: тип назва\_масиву[розмір];

- Двовимірний масив: тип назва\_масиву[розмір1][розмір2];

Доступ до елементів масиву здійснюється за допомогою індексації: назва\_масиву[індекс] для одновимірного масиву та назва\_масиву[індекс1][індекс2] для двовимірного масиву. Індекси починаються з нуля.

**9.** *Яким є призначення наступних рядкових функцій з заголовкового файлу cstring (string.h): strstr, strlen, strcpy, strncpy, strcat, strncat, strcmp, strncmp, stricmp, strnicmp, strchr, strcspn, strspn, strprbk, atof, atoi, atol?*

- strstr: Повертає покажчик на перше входження підрядка в рядку;

- strlen: Повертає довжину рядка без урахування завершального нуль-символу;

- strcpy: Копіює рядок, включаючи завершальний нуль-символ, в інший рядок;

- strncpy: Копіює перші N символів одного рядка в інший рядок;

- strcat: Додає один рядок до іншого;

- strncat: Додає перші N символів одного рядка до іншого;

- strcmp: Порівнює два рядки і повертає результат порівняння;

- strncmp: Порівнює перші N символів двох рядків;

- stricmp: Порівнює два рядки без урахування регістру;

- strnicmp: Порівнює перші N символів двох рядків без урахування регістру;

- strchr: Повертає покажчик на перше входження символу в рядку;

- strcspn: Повертає кількість символів у початковій частині рядка, які не містяться в іншому рядку;

- strspn: Повертає кількість символів у початковій частині рядка, які містяться в іншому рядку;

- strpbrk: Повертає покажчик на перший символ у рядку, який входить в заданий набір символів;

- atof: Перетворює рядок на число з плаваючою крапкою;

- atoi: Перетворює рядок на ціле число;

- atol: Перетворює рядок на довге ціле число.

**10.** *Яке призначення, синтаксис опису і оголошення структур (struct) С/С++? Чим вони відрізняються від масивів?*

Структура - це користувацький тип даних, який дозволяє об'єднати декілька змінних різних типів під одним ім'ям. Структури використовуються для представлення складних типів даних. На відміну від масивів, які містять елементи одного типу, структури можуть включати елементи різних типів. Доступ до полів структури здійснюється за допомогою крапки.

Синтаксис:

struct MyStruct {

typeOne varOne;

typeTwo varTwo;

};

**11.** *Що таке член (поле) структури struct С/С++ та який оператор реалізовує доступ до нього?*

Член (поле) структури - це змінна, оголошена всередині структури. Доступ до члена структури здійснюється за допомогою оператора крапки.

**12.** *Що у С/С++ називають дескриптором структури (struct)? За якими правилами він утворюється і задля чого використовується програмістом?*

Дескриптором структури називають ім'я структури, яке використовується для ідентифікації типу структури. Дескриптор утворюється при описі структури після ключового слова struct. Він використовується для оголошення змінних цього типу структури та доступу до її членів.

**13.** *Що слід виконати (записати у вихідному коді програми) для оголошення змінної типу структура, якщо означений тип описано у заголовковому файлі?*

Для оголошення змінної типу структура, описаної у заголовковому файлі, потрібно включити цей заголовковий файл та оголосити змінну, використовуючи дескриптор структури.

**14.** *Мовою програмування С/С++ наведіть приклад оголошення структури (struct), один із членів якої ― структура.*

struct Address {

char street[50];

char city[20];

};

struct Person {

char name[20];

int age;

struct Address address;

};

**15.** *Наведіть синтаксис оголошення змінної множинного типу та перелічіть допустимі операції над нею. Чим з погляду використання відрізняються змінні типу масив і типу множина?*

Синтаксис: set<тип\_елементів> ім'я\_множини;

Допустимі операції над множиною:

- Додавання елемента: insert();

- Видалення елемента: erase();

- Перевірка наявності елемента: count() або find();

- Отримання розміру множини: size();

- Очищення множини: clear().

Відмінності між масивом і множиною:

- Масив зберігає елементи в заданому порядку, а множина - в відсортованому порядку;

- Масив допускає дублікати елементів, а множина - ні;

- Масив має фіксований розмір, а множина може динамічно змінювати свій розмір;

- Масив забезпечує прямий доступ до елементів за індексом, а множина - ні.

**16.** *Що у С/С++ розуміють під явним і неявним перетворення типів та яким чином реалізовується кожен з них?*

Явне перетворення типів здійснюється за допомогою операторів приведення типу, наприклад:

int x = 10;

double y = (double) x; // явне перетворення int в double

Неявне перетворення типів відбувається автоматично, коли значення одного типу присвоюється змінній іншого сумісного типу, наприклад:

int x = 10;

double y = x; // неявне перетворення int в double

**17.** *Наведіть приклад оголошення потокового об’єкта ofstream з заголовкового файла fstream С++. Яким чином за його допомогою здійснюється запис даних у файловий потік (файл)?*

Приклад:

ofstream outputFile("output.txt");

outputFile << "Певна інформація, яку я хочу записати у файл :)" << endl;

outputFile.close();

**18.** *У якому режимі відкриються файлові потоки, об’єкти яких оголошені типом fstream, ofstream, ifstream відповідно?*

- fstream: відкриває файл для читання і запису;

- ofstream: відкриває файл для запису;

- ifstream: відкриває файл для читання.

**19.** *Яке призначення наступних функцій-членів потокового об’єкта ifstream С++: open(), eof(), close()?*

- open(): відкриває файл для читання;

- eof(): перевіряє, чи досягнуто кінець файлу;

- close(): закриває файл.

**20.** *Поясніть призначення кожної з перелічених констант режимів відкриття файлових потоків у С++: ios\_base::app, ios\_base::ate, ios\_base::in, ios\_base::out, ios\_base::trunc (простір імен – std).*

- ios\_base::app: дописує дані в кінець файлу;

- ios\_base::ate: встановлює позицію запису/читання в кінець файлу;

- ios\_base::in: відкриває файл для читання;

- ios\_base::out: відкриває файл для запису;

- ios\_base::trunc: очищає вміст файлу при відкритті.