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 КІ-23

Яровець Т. 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

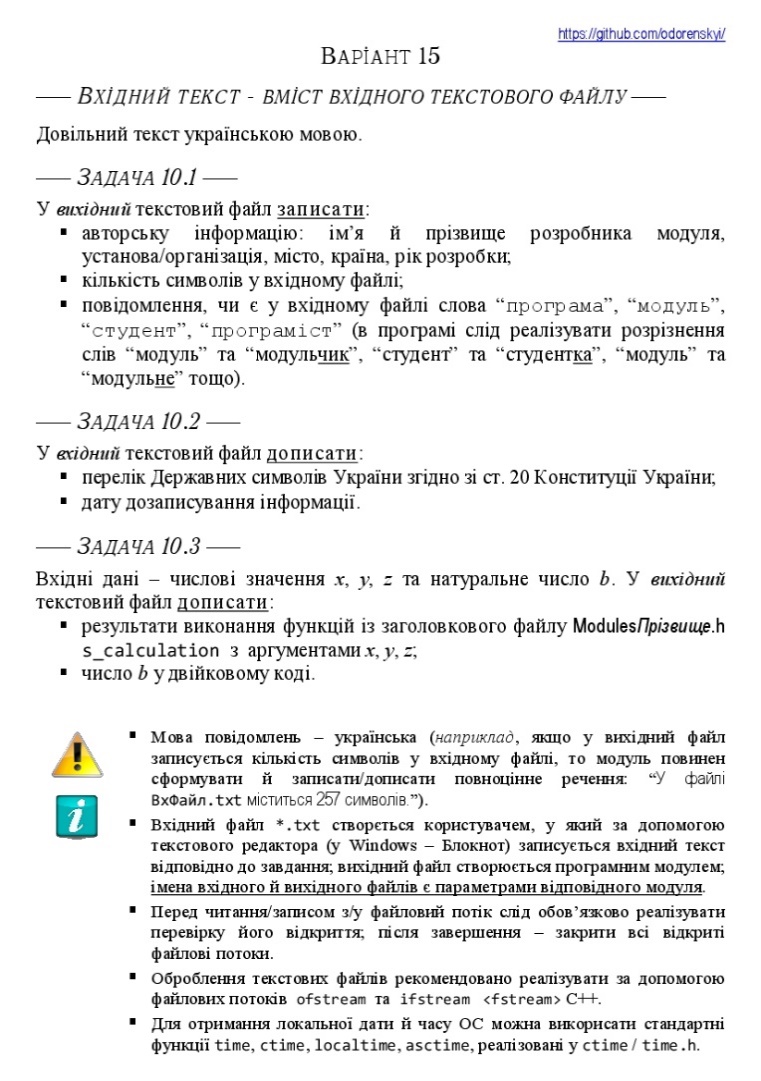
**Мета:** полягає у набутті ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування С++ програмних модулів створення й оброблення даних типів масив, структура, об’єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символьної інформації.

**Завдання:**

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 №15**



**Завдання по варіанту:**

**Аналіз та постановка задачі 10.1**

Вх: file\_name - типу const char\*

Вих: Оновлений файл file\_name.txt, оновлення мають авторську інформацію та кількість символів у вхідному файлі, повідомлення щодо наявності заданих у ТЗ слів.

**Аналіз та постановка задачі 10.2**

Вх: file\_name - типу const char\*

Вих: дозапис у file\_name.txt, а саме перелік символів України згідно зі ст. 20 Конституції України, дату дозапису інформції.

**Аналіз та постановка задачі 10.3**

Вх: x, y, z, b – значення типу int, та file\_name – типу const char\*, ім’я вихідного файлу.

Вих: Створений файл file\_name.txt з результатом виконання функції із заголовкового файлу ModulesYarovets.h s\_calculation з аргументами x, y , z (x - виключено ще з 8 лабораторної роботи, бо не приймає ніякої участі у роботі ПЗ);

Число b у двійковому коді.

**Лістинг задачі 10.1:**

void task10\_1(const char\* file\_name)

{

fstream fileread (file\_name);

string buffer;

string information = "У вхідному файлі було знайдено такі слова:";

string allInfo;

int countChar;

if(!fileread.is\_open())

{

ofstream file(file\_name);

file.close();

fstream fileread(file\_name);

}

if (fileread.is\_open())

{

while(getline(fileread, buffer))

{

allInfo += buffer;

}

countChar = allInfo.length();

for(int i = 0; i < countChar; i++)

{

if(isascii(buffer[i]))

{

buffer[i] = '/';

}

}

int countStudent = buffer.find("/студент/");

int countModule = buffer.find("/модуль/");

int countDev = buffer.find("/програміст/");

int countApp = buffer.find("/програма/");

if (countStudent != -1) information += " студент";

if (countModule != -1)

{

if (information.length() > 43)

{

information += ",";

}

information += " модуль";

}

if (countDev != -1)

{

if (information.length() > 43)

{

information += ",";

}

information += " програміст";

}

if (countApp != -1)

{

if (information.length() > 43)

{

information += ",";

}

information += " програма";

}

if (countStudent == -1 && countModule == -1 && countDev == -1 && countApp == -1)

{

information = "У файлі не було знайдено слів: студент, модуль, програміст, програма";

}

information += ".";

fileread.close();

ofstream filewrite(file\_name);

filewrite << "Кількість символів у даному тексті: " << countChar << "\n";

filewrite << information << "\n";

filewrite << "Модуль було розроблено студентом Центральноукраїнського національного технічого університету - Яровцем Тарасом; м. Кропивницький; Україна; 2024 рік.";

filewrite.close();

}

}

**Алгоритм задачі 10.1:**

ПОЧАТОК

1. Відкрити файл.
2. Перевірити чи відкрито файл, якщо ні – створити та відкрити.
3. Порахувати кількість символів у тексті файлу.
4. Перетворити усі символи, що не є буквами у ‘/’.
5. Знайти задані слова.
6. Записати, які слова було знайдено.
7. Закрити файл.
8. Відкрити файл для запису.
9. Записати інформацію відповідно до вимог.
10. Закрити файл.

КІНЕЦЬ

**Лістинг задачі 10.2:**

void task10\_2(const char\* file\_name)

{

system("chcp 65001 > nul");

fstream filewrite(file\_name, ios\_base::app);

filewrite << "\n" << "Згідно зі ст. 20 Конституції України державними символами є: Державний Прапор України, Державний Герб України і Державний Гімн України." <<"\n";

time\_t now = time(NULL);

tm \*localeTime = localtime(&now);

filewrite << 1900 + localeTime->tm\_year;

filewrite.close();

}

**Алгоритм задачі 10.2:**

ПОЧАТОК

1. Відкрити файл для запису.
2. Записати до файлу щодо Конституції України.
3. Отримати час та дату у даний момент часу.
4. Записати інформацію відповідно до вимог.
5. Закрити файл.

КІНЕЦЬ

**Лістинг задачі 10.3:**

void task10\_3(int y, int z, int b, const char\* file\_name)

{

float res\_calc = s\_calculation(y, z);

int bin = 0, k = 1;

while (b)

{

bin += (b%2) \* k;

k \*= 10;

b/=2;

}

string b\_in\_bin = to\_string(bin);

fstream filewrite(file\_name, ios\_base::app);

if (res\_calc == -1)

{

filewrite << "\n" << "Неможливо визначити, тому що корінь не може бути від'ємним!" << "\n";

}

else

{

filewrite << "\n" << "S: " << setprecision(1) << fixed << res\_calc <<"\n";

}

filewrite << "b у двійковій: " << b\_in\_bin <<"\n";

filewrite.close();

}

**Алгоритм задачі 10.3:**

ПОЧАТОК

1. Викликати функцію s\_calculation() з відповідними параметрами.
2. Записати результат функції s\_calculation() у змінну res\_calc.
3. Перевести задане число у змінній B у двійкову форму.
4. Відкрити файл для запису.
5. Записати інформацію відповідно до вимог.
6. Закрити файл.

КІНЕЦЬ

**Лістинг TestDriver:**

#include <iostream>

#include "ModulesYarovets.h"

#include <fstream>

using namespace std;

int main()

{

system("chcp 65001 & cls");

cout << "TASK 10.1 TEST:" << endl;

ofstream test1 ("name.txt");

test1 << "Центральноукраїнський національний технічний університет - вищий навчальний заклад, який готує студентів за спеціальностями, які актуальні для галузей економіки країни.";

test1.close();

task10\_1("name.txt");

ofstream test2 ("sun.txt");

test2 << "Йде до університету студент який гарно вчиться.";

test2.close();

task10\_1("sun.txt");

ofstream test3 ("cat.txt");

test3 << "Кіт м'яко муркоче.";

test3.close();

task10\_1("cat.txt");

task10\_1("information.txt");

ofstream test5 ("bird.txt");

test5 << "Пташка летить високо.";

test5.close();

task10\_1("bird.txt");

cout << "END OF TEST 10.1" << endl;

cout << "TASK 10.2 TEST:" << endl;

ofstream test6 ("ball.txt");

test6 << "Діти граються м'ячем.";

test6.close();

task10\_2("ball.txt");

ofstream test7 ("water.txt");

test7 << "Вода прозора і чиста.";

test7.close();

task10\_2("water.txt");

ofstream test8 ("autumn.txt");

test8 << "Листя опадає восени.";

test8.close();

task10\_2("autumn.txt");

ofstream test9 ("book.txt");

test9 << "Мама читає книгу.";

test9.close();

task10\_2("book.txt");

task10\_2("all.txt");

cout << "END OF TEST 10.2" << endl;

cout << "TASK 10.3 TEST:" << endl;

ofstream test11 ("dog.txt");

test11 << "Собака бігає швидко.";

test11.close();

task10\_3(7, 8, 3, "dog.txt");

ofstream test12 ("wind.txt");

test12 << "Вітер дме сильно.";

test12.close();

task10\_3(6, 15, 113, "wind.txt");

ofstream test13 ("flowers.txt");

test13 << "Квіти пахнуть приємно.";

test13.close();

task10\_3(2, 4, 87, "flowers.txt");

ofstream test14 ("radical.txt");

test14.close();

task10\_3(5, 5, 0, "radical.txt");

task10\_3(0, 0, 43, "see.txt");

cout << "END OF TEST 10.3" << endl;

system("pause");

return 0;

}

**Висновок:** Для виконання завдань мною було опрацьовано зміст бібліотеки fstream, клас ofstream та його функції, клас ifstream та його функції.

У завданні 10.1. мною було проаналізовано вимоги до програмного забезпечення за завданням 10.1. й створено функцію (task10\_1) та додано до статичної бібліотеки libModulesYarovets.a, що містить у собі функції s\_calculation, bofort, socks, bit також було створено оновлено заголовковий файл ModulesYarovets.h, що вже містить прототип функції s\_calculation, bofort, socks, bit.

У завданні 10.2. мною було проаналізовано вимоги до програмного забезпечення за завданням 10.2. й створено функцію (task10\_2) та додано до статичної бібліотеки libModulesYarovets.a, що містить у собі функції s\_calculation, bofort, socks, bit, task10\_1, також було створено оновлено заголовковий файл ModulesYarovets.h, що вже містить прототип функції s\_calculation, bofort, socks, bit, task10\_1.

У завданні 10.3. мною було проаналізовано вимоги до програмного забезпечення за завданням 10.3. й створено функцію (task10\_3) та додано до статичної бібліотеки libModulesYarovets.a, що містить у собі функції s\_calculation, bofort, socks, bit, task10\_1, task10\_2, також було створено оновлено заголовковий файл ModulesYarovets.h, що вже містить прототип функції s\_calculation, bofort, socks, bit, task10\_1, task10\_2.

Було створено 3 Тест-сьюти з 5 тестовими випадками за якими пізніше були проведені тестування статичної бібліотеки libModulesYarovets.a за допомогою консольного застосунку TestDriver.

**Контрольні запитання:**

1. Яке призначення та синтаксис запису блоку-контроля try - throw - catch у мові програмування С++?

У мові програмування C++ блоки try, throw і catch використовуються для обробки виключень.

try {

// Код, який може викликати виключення

if (/\* якась умова \*/) {

throw /\* вираз (об'єкт або значення) \*/;

}

} catch (/\* тип виключення \*/ e) {

// Код обробки виключення

} catch (/\* інший тип виключення \*/ e) {

// Код обробки іншого виключення

}

1. Наведіть приклад опису й використання міжмодульної змінної.

Міжмодульна змінна — це змінна, доступна з кількох модулів (файлів) програми. Її оголошення здійснюється за допомогою ключового слова `extern`. Приклад запису:

Перший модуль:

#include <iostream>

extern int intermodule;

void view\_intermodule() {

cout << "Значення міжмодульної змінної: " << intermodule << endl;

}

Другий модуль:

int intermodule = 10;

1. Яку область видимості матимуть об’єкти (змінні, типи, константи тощо), описані в тілі функції main С++?

Змінні, оголошені всередині функції main(), можуть використовуватися лише в межах цієї функції і знищуються після завершення її виконання.

1. Здійсніть порівняльний аналіз змінної типу enum та масиву.

Порівняння

* Читабельність і безпека: Enum значно покращує читабельність і безпеку коду за рахунок використання іменованих констант, тоді як масиви можуть містити будь-які значення одного типу.
* Гнучкість: Масиви є більш гнучкими в зберіганні даних різних типів і розмірів, у той час як enum обмежений визначеним набором значень.
* Область застосування: Enum краще підходить для фіксованих наборів станів або опцій (наприклад, дні тижня, кольори), тоді як масиви використовуються для зберігання великих наборів однорідних даних (наприклад, список чисел, колекція об'єктів).