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 ТИІВ 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 КБ-24

Бовкун Д. О.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А. С.

Кропивницький – 2025

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

Мета роботи полягає у набутті грунтовних 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 libModules Прізвище. а (проект Modules Прізвище лабораторних робіт №8-9).
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв'язування задач 10.1-10.3.

Варіант № 17

Аналіз та постановка задач

Задача 10.1

**Мета:** Обробити вхідний текстовий файл та записати у вихідний файл певну інформацію.  
**Вхідні дані:** Текстовий файл з довільним текстом українською мовою.  
**Вихідні дані:** Текстовий файл, який містить:

1. Авторську інформацію (ім'я, прізвище, установа, місто, країна, рік розробки).
2. Кількість абзаців у вхідному тексті.
3. Перевірку наявності слів "Україна", "університет", "блокнот" (з урахуванням різних форм слів).

**Кроки виконання:**

1. Відкрити вхідний файл для читання та вихідний файл для запису (якщо він існує, перезаписати).
2. Записати авторську інформацію у вихідний файл.
3. Підрахувати кількість абзаців у вхідному файлі (абзаци можуть бути розділені порожніми рядками або іншими символами).
4. Перевірити наявність слів "Україна", "університет", "блокнот" у тексті, ігноруючи форми слів (наприклад, "університету").
5. Записати результати у вихідний файл.
6. Закрити файли.

**Особливості:**

Використання файлових потоків ifstream та ofstream.

Перевірка на коректність відкриття файлів.

Обробка текстових даних з урахуванням української мови.

Задача 10.2

**Мета:** Дописати у вхідний текстовий файл додаткову інформацію.  
**Вхідні дані:** Текстовий файл з довільним текстом.  
**Вихідні дані:** Той самий файл, доповнений:

1. Кількістю крапок у тексті.
2. Датою та часом дозапису.

**Кроки виконання:**

1. Відкрити вхідний файл для читання та запису (у режимі додавання).
2. Підрахувати кількість крапок у тексті.
3. Отримати поточну дату та час.
4. Дописати у файл кількість крапок, дату та час.
5. Закрити файл.

**Особливості:**

Використання функцій для роботи з датою та часом (time, ctime).

Режим відкриття файлу для додавання (ios::app).

Задача 10.3

**Мета:** Обробити числові дані та дописати результати у вихідний файл.  
**Вхідні дані:** Числові значення x, y, z та натуральне число b.  
**Вихідні дані:** Текстовий файл, доповнений:

1. Результатом функції s\_calculation(x, y, z) з модуля ModulesПрізвище.h.
2. Числом b у двійковому коді.

**Кроки виконання:**

1. Відкрити вихідний файл для запису (у режимі додавання).
2. Викликати функцію s\_calculation та записати результат у файл.
3. Перевести число b у двійковий код та записати результат у файл.
4. Закрити файл.

**Особливості:**

Використання заголовкового файла ModulesПрізвище.h.

Переведення числа у двійкову систему.

Вимоги до програмного забезпечення

1. **Інтерфейс:**

Програма повинна приймати імена вхідного та вихідного файлів як параметри.

Повідомлення мають бути українською мовою.

1. **Обробка файлів:**

Перевірка на коректність відкриття файлів.

Закриття всіх файлових потоків після виконання.

Якщо вихідний файл існує, його вміст має бути перезаписано.

1. **Технічні вимоги:**

Використання бібліотеки <fstream> для роботи з файлами.

Використання бібліотеки <ctime> для роботи з датою та часом.

Архітектура програми

Програма складатиметься з таких модулів:

1. **Головний модуль:**

Керує виконанням завдань 10.1, 10.2, 10.3.

Обробляє параметри командного рядка (імена файлів).

1. **Модуль для задачі 10.1:**

Читає вхідний файл, аналізує текст, записує результати у вихідний файл.

1. **Модуль для задачі 10.2:**

Дописує інформацію у вхідний файл.

1. **Модуль для задачі 10.3:**

Виконує обчислення та дописує результати у вихідний файл.

1. **Допоміжні функції:**

Перевірка наявності слів у тексті.

Переведення числа у двійковий код.

Детальне проектування

1. **Функція для задачі 10.1:**

void processTask10\_1(const string& inputFile, const string& outputFile) {

ifstream in(inputFile);

ofstream out(outputFile, ios::trunc);

// Перевірка файлів, обробка тексту, запис результатів

}

1. **Функція для задачі 10.2:**

void processTask10\_2(const string& inputFile) {

fstream file(inputFile, ios::in | ios::out | ios::app);

// Підрахунок крапок, отримання дати/часу, запис

}

1. **Функція для задачі 10.3:**

void processTask10\_3(const string& outputFile, double x, double y, double z, int b) {

ofstream out(outputFile, ios::app);

// Виклик s\_calculation, переведення b у двійковий код, запис

}

1. **Допоміжні функції:**

bool containsWord(const string& text, const string& word) — перевірка наявності слова у тексті.

string toBinary(int n) — переведення числа у двійковий код.