

**Мета роботи** 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 №14:**

1. **Аналіз задачі 10.1:**

**Мета**: аналіз вхідного текстового файлу та створення узагальненої інформації на основі його вмісту.

Необхідно реалізувати:

* Запис авторської інформації: ім’я, прізвище розробника, організація, місто, країна, рік розробки.
* Витягти та записати всі символи верхнього регістру з вхідного файлу.
* Проаналізувати, чи всі рядки завершуються розділовими знаками (як у вірші), й записати відповідне повідомлення.
* Весь вірш переписати в один стовпчик по 4 рядки.

**Архітектурне проектування 10.1:**

#### Вхідні дані:

input.txt — вхідний файл з віршем.

#### Вихідні дані:

output.txt — файл з результатами аналізу тексту.

#### Функції:

getAuthorInfo() — повертає авторську інформацію.

extractUppercase(fileName) — отримує всі символи верхнього регістру.

checkPunctuation(fileName) — перевіряє, чи кожен рядок закінчується розділовим знаком.

writePoemVertically(fileName) — записує вірш в один стовпчик по 4 рядки.

**Псевдокод до задачі 10.1:**

Відкрити input.txt для читання

Відкрити output.txt для запису (очистити, якщо існує)

1. Авторська інформація:

автор ← getAuthorInfo()

записати автор у output.txt

2. Великі літери:

upperLetters ← extractUppercase("input.txt")

записати upperLetters у output.txt

3. Перевірка розділових знаків:

punctuationResults ← checkPunctuation("input.txt")

записати punctuationResults у output.txt

4. Запис вірша стовпчиком:

writePoemVertically("input.txt", "output.txt")

Закрити обидва файли

1. **Аналіз задачі 10.2:**

**Мета:** доповнення вихідного файлу мета-інформацією.

Необхідно дописати:

* Розмір файлу у кілобайтах (КБ).
* Дату і час модифікації файлу — момент виконання програми.

### **Архітектурне проєктування 10.2:**

#### Вхідні дані:

output.txt — файл після виконання задачі 10.1.

#### Вихідні дані:

output.txt з дописаними метаданими.

#### Функції:

getFileSizeKB(fileName) — обчислює розмір файлу в КБ.

getCurrentDateTime() — отримує дату і час останньої модифікації.

**Псевдокод до задачі 10.2:**

Відкрити output.txt у режимі додавання

1. Отримати розмір файлу:

size ← getFileSizeKB("output.txt")

записати "Розмір файлу: size КБ" у output.txt

2. Отримати дату і час:

dateTime ← getCurrentDateTime()

записати "Дата і час: dateTime" у output.txt

Закрити output.txt

1. **Аналіз задачі 10.3:**

**Мета**: обробка вхідних числових даних та їх аналітичне представлення у вихідному файлі.

Необхідно реалізувати:

* Вхідні дані: цілі значення x, y, z та натуральне b.
* Виклик функції s\_calculation(x, y, z) із заголовкового файлу та запис результату.
* Представлення числа y у двійковій формі (переведення з десяткової системи).

### **Архітектурне проєктування 10.3:**

#### Вхідні дані:

Числові значення: x, y, z, b.

#### Вихідні дані:

output.txt з результатами обчислення і двійковим поданням y.

#### Функції:

s\_calculation(x, y, z) — обчислення з зовнішнього заголовкового файлу Modules/Kondratenko.h.

toBinary(y) — переведення числа y у двійкову систему числення.

**Псевдокод до 10.3:**

Ввід x, y, z, b (з клавіатури або у коді)

Відкрити output.txt у режимі додавання

1. Обчислення:

результат ← s\_calculation(x, y, z)

записати "Результат обчислення: результат" у output.txt

2. Подання y у двійковій системі:

yBinary ← toBinary(y)

записати "Число y в двійковій формі: yBinary" у output.txt

Закрити output.txt