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

Заріцкий В. А.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А. С.

Кропивницький – 2025

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. **Варіант №12**

**Завдання до лабораторної роботи**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesZaritskyi.а (проект ModulesZaritskyi лабораторних робіт №8–9).
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3.

# 1. Аналіз задачі 10.1

## 1.1. Вимоги та постановка задачі

* **Мета**: Створити вихідний текстовий файл із записом авторської інформації, інформації про мову вхідного речення та обробленого речення.
* **Вхідні дані**: Текстовий файл, який містить довільне речення (як українською, так і англійською).
* **Обробка речення**:
  + **Визначення мови**: Проаналізувати вміст файла для визначення, чи переважають латинські символи (англійська) чи кириличні (українська).
  + **Обробка англійського речення**: Перетворити всі літери у верхній регістр.
  + **Обробка українського речення**: Видалити слова «лінощі», «сесія», «академзаборгованість» (з урахуванням можливих варіацій регістру).
* **Запис вихідних даних**:
  + Запис авторської інформації:

"Інформація про автора: Віктор Заріцкий, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025."

* + Запис інформації про мову вхідного речення.
  + Запис обробленого речення відповідно до визначеної логіки.
* **Особливості реалізації**:
  + Файл для запису створюється, якщо не існує, або очищається (перезаписується) перед записом.
  + Перевірка відкриття файлового потоку як для читання, так і для запису.

## 1.2. Архітектурне проектування (10.1)

* **Модуль читання файлу**:
  + Читає вхідне речення з файлу.
  + Перевірка успішного відкриття файлу.
* **Модуль аналізу тексту**:
  + Функція для аналізу символів: якщо більшість символів – латиниця, визначається англійська мова, інакше – українська.
  + Якщо англійський текст – застосовується функція конвертації до верхнього регістру.
  + Якщо український текст – застосовується функція пошуку та виключення слів "лінощі", "сесія", "академзаборгованість".
* **Модуль запису файлу**:
  + Створює або очищує вихідний файл.
  + Записує послідовно: авторські дані, інформацію про мову, оброблене речення.
  + Закриває файловий потік після запису.

## 1.3. Детальний псевдокод для 10.1

function processTask10\_1(inputFileName, outputFileName):

// Відкриття вхідного файлу для читання

open inputFileName as inFile for reading

if inFile не відкрився:

вивести повідомлення про помилку та завершити роботу

read sentence from inFile

close inFile

// Визначення мови речення

if sentence містить переважно латинські символи:

language = "англійська"

processedSentence = convertToUppercase(sentence)

else:

language = "українська"

processedSentence = removeForbiddenWords(sentence, ["лінощі", "сесія", "академзаборгованість"])

// Відкриття вихідного файлу для запису (створення/очищення)

open outputFileName as outFile for writing

if outFile не відкрився:

вивести повідомлення про помилку та завершити роботу

// Запис даних до вихідного файлу

write to outFile: "Розробник: Віктор Заріцкий, Центральноукраїнський Національний Технічний УніверсИТЕТ, Кропивницький, Україна, 2025."

write to outFile: "Мова речення: " + language

write to outFile: "Оброблене речення: " + processedSentence

close outFile

# 2. Аналіз задачі 10.2

## 2.1. Вимоги та постановка задачі

* **Мета**: Доповнити (дозаписати) вхідний текстовий файл додатковою інформацією.
* **Додаткова інформація**:
  + Текст статті 62 Закону України "Про вищу освіту". Цей текст може бути збережений як константа або завантажуватись із зовнішнього джерела.
  + Дата та час моменту дозапису (отримання локальної дати та часу через стандартні функції з <ctime>/<time.h>).
* **Режим роботи з файлом**:
  + Вхідний файл відкривається в режимі допису (append).
  + Потрібна перевірка відкриття потоку для дозапису.
  + Після операцій потік слід закрити.

## 2.2. Архітектурне проектування (10.2)

* **Модуль дозапису інформації**:
  + Функція для відкриття файлу в режимі append.
  + Функція отримання поточного часу у вигляді рядка.
  + Запис тексту статті 62 та рядка з датою й часом.
* **Обробка файлу**:
  + Відкриття файлу, перевірка успішного відкриття, дозапис та закриття потоку.

## 2.3. Детальний псевдокод для 10.2

function processTask10\_2(inputFileName):

// Відкриття файлу у режимі дозапису

open inputFileName as file in append mode

if file не відкрився:

вивести повідомлення про помилку та завершити роботу

// Завантаження тексту статті 62 Закону "Про вищу освіту"

textArticle = "Текст статті 62 Закону України 'Про вищу освіту'..."

// Отримання поточного часу

currentTime = getLocalTime()

// Запис інформації

write to file: textArticle

write to file: "Дата та час дозапису: " + currentTime

close file

# 3. Аналіз задачі 10.3

## 3.1. Вимоги та постановка задачі

* **Мета**: Доповнити вихідний текстовий файл результатами числових обчислень.
* **Вхідні дані**:
  + Три числові значення: **x**, **y**, **z**.
  + Натуральне число **b**.
* **Обчислення**:
  + Виклик функції s\_calculation(x, y, z) з заголовкового файлу ModulesZaritskyi для отримання певного результату.
  + Перетворення числа **y** до двійкового формату (через алгоритм послідовного ділення на 2 з накопиченням остач).
* **Вивід**:
  + Дописування до вихідного файлу результату виконання функції s\_calculation.
  + Дописування числа **y** у вигляді рядка, що є двійковим представленням.
* **Режим роботи з файлом**:
  + Вихідний файл відкривається в режимі дозапису (append) із відповідною перевіркою відкриття.

## 3.2. Архітектурне проектування (10.3)

* **Модуль числових обчислень**:
  + Функція для виклику s\_calculation з параметрами **x**, **y**, **z**.
  + Функція, яка приймає значення **y** і повертає рядок у двійковій формі.
* **Модуль запису числових результатів**:
  + Відкриття вихідного файлу (режим дозапису).
  + Запис результатів обчислень у форматовані повідомлення (наприклад, "Результат виконання s\_calculation(x, y, z) = ...", "Число y у двійковій системі числення: ...").

## 3.3. Детальний псевдокод для 10.3

function processTask10\_3(outputFileName, x, y, z, b):

// Виклик функції s\_calculation з ModulesZaritsky

resultCalculation = s\_calculation(x, y, z)

// Перетворення числа y до двійкового формату

binaryY = convertToBinary(y)

// Відкриття вихідного файлу для дозапису

open outputFileName as outFile in append mode

if outFile не відкрився:

вивести повідомлення про помилку та завершити роботу

// Запис результатів числових обчислень

write to outFile: "Результат виконання s\_calculation(" + x + ", " + y + ", " + z + ") = " + resultCalculation

write to outFile: "Число " + y + " у двійковій системі числення: " + binaryY

close outFile

## Алгоритм перетворення десяткового числа в двійкове:

function convertToBinary(num):

if num == 0:

return "0"

binaryRepresentation = ""

while num > 0:

remainder = num % 2

binaryRepresentation = remainder + binaryRepresentation // накопичення символів зліва направо

num = num // 2

return binaryRepresentation