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 КБ-24

Науменко О. В.

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.

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 10.1–10.3.9.4 на основі функцій статичної бібліотеки libModulesПрізвище.а.

**Варіант №15:**

**ЗАДАЧА 10.1**

**Формулювання задачі:**

**У вихідний текстовий файл записати:**

-авторську інформацію: ім’я й прізвище розробника модуля, установа/організація, місто, країна, рік розробки

-кількість символів у вхідному файлі

-повідомлення, чи є у вхідному файлі слова “програма”, “модуль”, “студент”, “програміст” (у програмі слід реалізувати розрізнення слів “модуль” та “модульчик”, “студент” та “студентка”, “модуль” та “модульне” тощо)

**Вимоги до програмного забезпечення:**

-підтримка аналізу тексту українською мовою

-розпізнавання слів із коренями або частинами базових форм (стемінг або просте порівняння з використанням find/regex)

-перевірка наявності ключових слів

-підтримка формування текстового звіту з результатами аналізу

**Архітектура рішення:**

-input.txt — вхідний файл з довільним українським текстом

-output.txt — вихідний файл із результатами

-analyze\_text() — функція для обчислення кількості символів та пошуку ключових слів

-write\_output() — функція для створення текстового звіту

-contains\_word\_variants(word, text) — допоміжна функція, яка перевіряє наявність слова з можливими варіаціями

**Проєктування програмних модулів:**

mod\_text\_analysis.cpp

-int count\_characters(const std::string &text) - повертає кількість символів

-bool contains\_word(const std::string &text, const std::vector<std::string> &variants) - перевіряє наявність слова та його форм

-void write\_report(const std::string &filename, ...) - записує звіт

mod\_main.cpp

-зчитування вхідного тексту

-виклик модулів аналізу

-запис результату

**ЗАДАЧА 10.2**

**Формулювання задачі:**

**У вихідний текстовий файл дописати:**

-перелік Державних символів України згідно зі ст. 20 Конституції

України

-дату дописування інформації

**Вимоги до програмного забезпечення:**

-можливість додавання інформації у вже наявний файл

-автоматичне зчитування поточної дати

-можливість локалізованого виводу тексту

**Архітектура рішення:**

-output.txt — файл, який уже існує після Задачі 10.1

-append\_symbols() — функція для відкриття файлу в режимі

дописування та запису тексту

-get\_current\_date() — функція для отримання поточної дати в заданому

форматі

**Проєктування програмних модулів:**

mod\_constitution.cpp

-std::string get\_ukrainian\_symbols() - повертає перелік символів

-std::string get\_current\_date() - формує дату у форматі ДД.ММ.РРРР

-void append\_info(const std::string &filename) - виконує дописування у

файл

**ЗАДАЧА 10.3**

**Формулювання задачі:**

Вхідні дані — числові значення x, y, z та натуральне число b  
**У вихідний текстовий файл дописати:**

-результат виконання функції з заголовкового файлу

Modules/Прізвище.h — s\_calculation(x, y, z)

-число b у двійковому коді

**Вимоги до програмного забезпечення:**

-підключення зовнішнього заголовкового файлу з реалізацією функції

-підтримка математичних обчислень

-конвертація десяткового числа у двійкову систему

**Архітектура рішення:**

-Modules/Прізвище.h — файл із функцією s\_calculation(x, y, z)

-output.txt — файл, у який додається результат функції та двійкове

представлення b

-binary\_conversion() — функція для переведення числа у двійковий

вигляд

**Проєктування програмних модулів:**

mod\_math\_ops.cpp

-int s\_calculation(int x, int y, int z) - реалізація згідно з індивідуальним

варіантом

-std::string to\_binary(int number) - повертає рядок з двійковим

поданням

-void append\_math\_results(const std::string &filename, int x, int y, int z, int

b) - виконує обчислення та запис результату

Додатки:

**Test Suite — ЗАДАЧА 10.1**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Preliminary Steps | Action (test steps) | Expected Result |
| 1 | input1.txt: "Це моя програма для студентів" | analyze\_text("input1.txt") | К-сть символів: 34 - "програма": так - "студент": так |
| 2 | input2.txt: "Тут нічого схожого немає" | analyze\_text("input2.txt") | К-сть символів: 28 - ключові слова: ні |
| 3 | input3.txt: "Модульчик допомагає студенткам" | analyze\_text("input3.txt") | К-сть символів: 36 - модифіковані форми слів знайдені |
| 4 | input4.txt: "Програміст створив модульне середовище" | analyze\_text("input4.txt") | К-сть символів: 45 - всі чотири ключові слова в похідній формі |
| 5 | input5.txt: "" (порожній файл) | analyze\_text("input5.txt") | К-сть символів: 0 - ключові слова: ні |
| 6 | input6.txt: "МоДуЛь СтУдЕнТ пРоГрАмА" | analyze\_text("input6.txt") | К-сть символів: 27 - слова в різному регістрі розпізнані |
| 7 | input7.txt: "Цей текст довгий..." (1000+ символів) | analyze\_text("input7.txt") | К-сть символів: >1000 - ключові слова: ні |
| 8 | input8.txt: "Програмісти і студенти працюють з модулями" | analyze\_text("input8.txt") | К-сть символів: 55 - форми всіх ключових слів виявлені |

**Test Suite — ЗАДАЧА 10.2**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Preliminary Steps | Action (test steps) | Expected Result |
| 1 | output1.txt існує | append\_symbols("output1.txt") | Додано символи + поточна дата |
| 2 | output2.txt з фразою "Результат аналізу" | append\_symbols("output2.txt") | Текст залишився, символи додано |
| 3 | output3.txt — порожній | append\_symbols("output3.txt") | Додано тільки символи і дату |
| 4 | output4.txt — містить вже символи | append\_symbols("output4.txt") | Дублювання — дозволено (символи додаються знову) |
| 5 | output5.txt — містить лише 1 рядок | append\_symbols("output5.txt") | Додано блок з символами та датою |
| 6 | output6.txt — файл у DOS-форматі з \r\n | append\_symbols("output6.txt") | Символи додано коректно у новому форматі |
| 7 | output7.txt — файл з табуляціями | append\_symbols("output7.txt") | Форматування не порушено, додано коректно |
| 8 | output8.txt — файл недоступний (імітація помилки прав) | append\_symbols("output8.txt") | Очікується виняток/повідомлення про неможливість запису |

**Test Suite — ЗАДАЧА 10.3**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Preliminary Steps | Action (test steps) | Expected Result |
| 1 | x=2, y=3, z=4, b=5 | append\_math\_results("res1.txt", 2,3,4,5) | `s\_calculation(2,3,4)=14` + `b=101` |
| 2 | x=1, y=1, z=1, b=10 | append\_math\_results("res2.txt", 1,1,1,10) | `=3` + `b=1010` |
| 3 | x=5, y=0, z=2, b=15 | append\_math\_results("res3.txt", 5,0,2,15) | `=7` + `b=1111` |
| 4 | x=0, y=0, z=0, b=0 | append\_math\_results("res4.txt", 0,0,0,0) | `=0` + `b=0` |
| 5 | x=100, y=200, z=300, b=255 | append\_math\_results("res5.txt", 100,200,300,255) | `s=...` + `b=11111111` |
| 6 | x=3, y=5, z=7, b=1 | append\_math\_results("res6.txt", 3,5,7,1) | `s=...` + `b=1` |
| 7 | x=123, y=321, z=456, b=16 | append\_math\_results("res7.txt", 123,321,456,16) | `s=...` + `b=10000` |
| 8 | x=7, y=8, z=9, b=1024 | append\_math\_results("res8.txt", 7,8,9,1024) | `s=...` + `b=10000000000` |