

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 5

На тему: “Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.”

З дисципліни: «Основи програмування»

до:

Практичних Робіт до блоку № 5

Виконав:

Студент групи ШІ-11

Голейчук Іван Миколайович

Львів 2024

Тема роботи:

"Основи роботи з файлами та рядковими змінними у програмуванні на C++"

Мета роботи:

Розробити системне розуміння принципів роботи з файлами та рядковими змінними у мові C++. Дослідити методи взаємодії з текстовими та бінарними файлами, а також функціональність стандартної бібліотеки для обробки файлів. Ознайомитися зі створенням власних бібліотек, організацією коду, та вивчити практичні прийоми парсингу, серіалізації даних і обробки помилок.

Теоретичні відомості:

Вступ до Роботи з Файлами:

Основні операції з файлами: відкриття, читання, запис, закриття

Робота з файловими дескрипторами

C-style читання з файлу та запис до файлу

Перевірка стану файлу: перевірка помилок, кінець файлу

Базові приклади читання та запису в файл

Символи і Рядкові Змінні:

Робота з char та string: основні операції і методи

Стрічкові літерали та екранування символів

Конкатенація, порівняння та пошук у рядках

Текстові Файли:

Особливості читання та запису текстових файлів

Обробка рядків з файлу: getline, ignore, peek

Форматування тексту при записі: setw, setfill, setprecision

Парсинг текстових файлів: розділення на слова, аналіз структури

Обробка помилок при роботі з файлами

Бінарні Файли:

Вступ до бінарних файлів: відмінності від текстових, приклади (великі дані, ігрові ресурси, зображення)

Читання та запис бінарних даних

Робота з позиціонуванням у файлі: seekg, seekp

Серіалізація об'єктів у бінарний формат

Стандартна бібліотека та робота з файлами:

Огляд стандартної бібліотеки для роботи з файлами

Потоки вводу/виводу: ifstream, ofstream, fstream

Обробка помилок при роботі з файлами

Створення й використання бібліотек:

Вступ до створення власних бібліотек у C++

Правила розбиття коду на header-и(.h) та source(.cpp) файли

Статичні проти динамічних бібліотек: переваги та використання

Інтерфейси бібліотек: створення, документування, версіонування

Використання сторонніх бібліотек у проектах

Індивідуальний план опрацювання теорії:

Вступ до Роботи з Файлами:

Опрацьовано:

Розглянуто основні операції роботи з файлами: відкриття, читання, запис, закриття.

Вивчено принципи роботи з файловими дескрипторами та виконання C-style операцій з файлами. Проведено аналіз перевірки стану файлів (помилки, кінець файлу) з демонстрацією базових прикладів.

Джерела інформації:

- Лекції Олександра Пшеничного
- Практичні заняття
- Використання штучного інтелекту (чат GPT)
- YouTube

Символи і Рядкові Змінні:

Опрацьовано:

Вивчено основні операції з типами даних `char` і `string`, включаючи роботу зі стрічковими літералами, екрануванням символів, конкатенацією, порівнянням та пошуком у рядках.

Розглянуто приклади ефективної роботи з рядковими змінними.

Джерела інформації:

- Лекції Олександра Пшеничного
- Практичні заняття
- Використання штучного інтелекту (чат GPT)
- YouTube

Текстові Файли:

Опрацьовано:

Досліджено специфіку роботи з текстовими файлами: методи читання (`getline`, `ignore`, `peek`) і запису даних, використання функцій форматування тексту (`setw`, `setfill`, `setprecision`). Вивчено техніки парсингу файлів для аналізу їхньої структури та обробки рядків. Здійснено практику перевірки помилок при роботі з файлами.

Джерела інформації:

- Лекції Олександра Пшеничного
- Практичні заняття
- Використання штучного інтелекту (чат GPT)
- YouTube

Бінарні Файли:

Опрацьовано:

Вивчено основи роботи з бінарними файлами, їх відмінності від текстових. Ознайомлено з методами читання, запису, серіалізації даних і позиціонування у файлі (`seekg`, `seekp`).

Наведено приклади використання бінарних файлів для збереження великих даних.

Джерела інформації:

- Лекції Олександра Пшеничного
- Практичні заняття
- Використання штучного інтелекту (чат GPT)
- YouTube

Стандартна бібліотека та робота з файлами:

Опрацьовано:

Досліджено можливості стандартної бібліотеки для роботи з файлами: використання потоків (`ifstream`, `ofstream`, `fstream`), обробка помилок, а також інтеграція сторонніх бібліотек у проєкти.

Джерела інформації:

- Лекції Олександра Пшеничного
- Практичні заняття
- Використання штучного інтелекту (чат GPT)
- YouTube

Створення й використання бібліотек:

Опрацьовано:

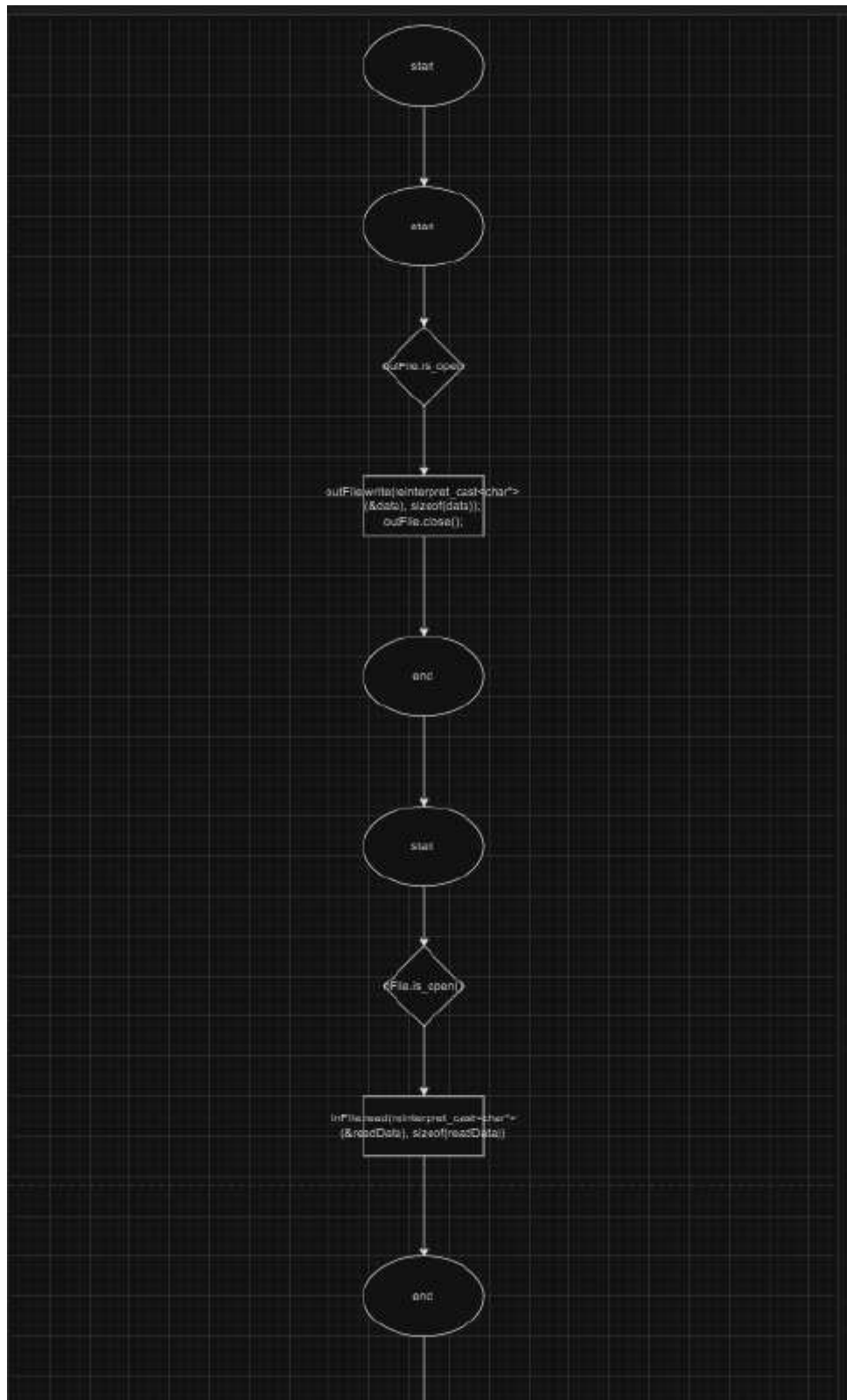
Розглянуто основи створення власних бібліотек у C++, правила розподілу коду між файлами (`header` і `source`). Вивчено переваги статичних і динамічних бібліотек, принципи документування та версіонування інтерфейсів, інтеграцію бібліотек у проєкти.

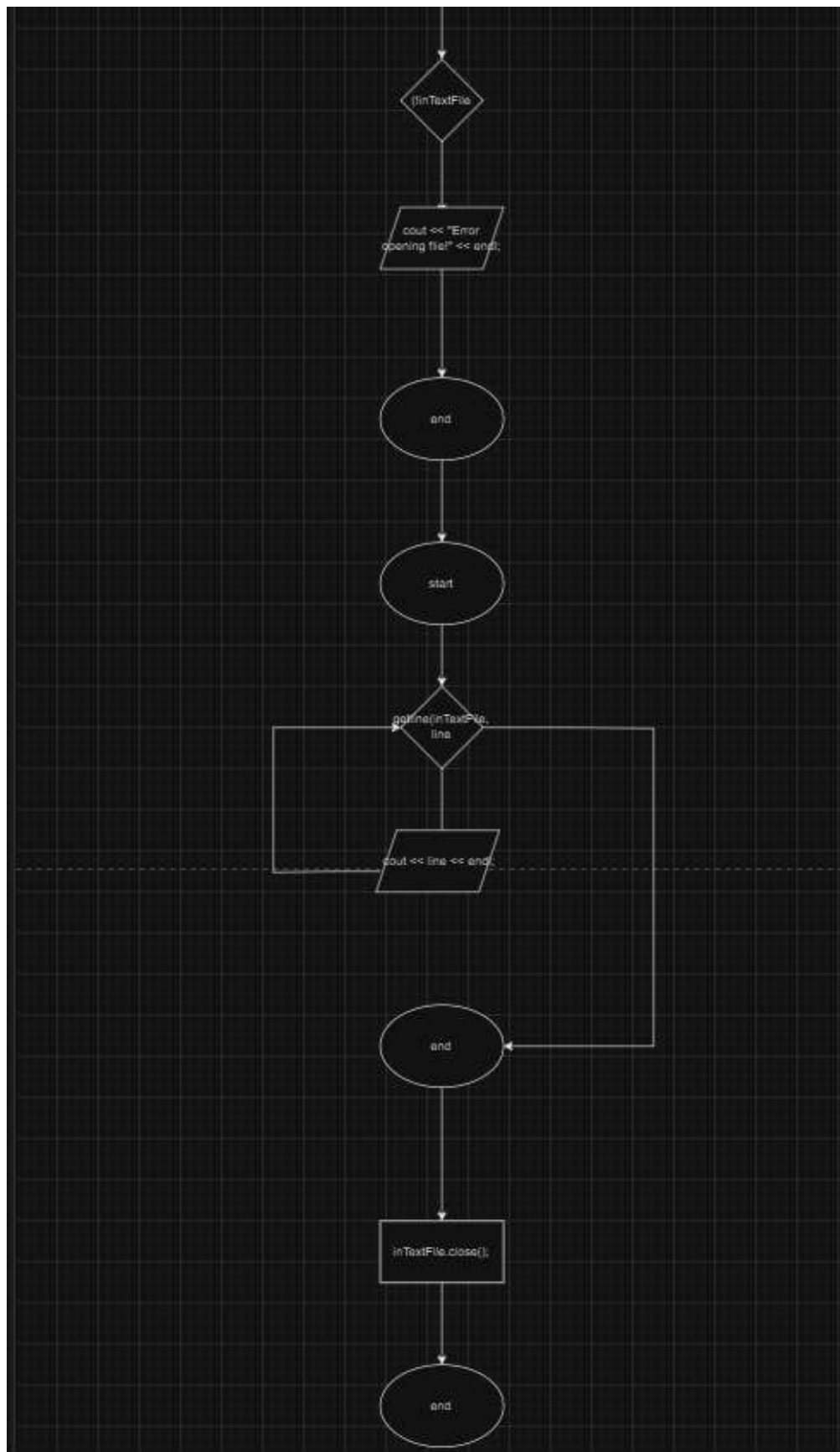
Джерела інформації:

- Лекції Олександра Пшеничного
- Практичні заняття
- Використання штучного інтелекту (чат GPT)
- YouTube

Виконання роботи:

Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-9)





Epic 5 Task 3 - Lab# programming: VNS Lab 6



vns_lab_6_task_2_variant_23_ivan_holeichuk (1).cpp

23. Структура "Стадіон":

- назва;
- рік будівлі;
- кількість площадок;
- види спорту.

Знищити всі елементи, у яких рік будівлі менше заданого, додати 2 елементи перед елементом із зазначеним номером.

Epic 5 Task 4 - Lab# programming: VNS Lab 8



vns_lab_8_task_2_variant_23_ivan_holeichuk (1).cpp

23. Для рядка знайти довжину найкоротшого слова.

John Black - Epic 5 Task 5 - Lab# programming: VNS Lab 9



vns_lab_9_task_2_variant_23_ivan_holeichuk (1).cpp

23.

- 1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких міститься два однакових слова.
- 2) Визначити номер слова, у якому найбільше букв «А».

Epic 5 Task 6 - Lab# programming: Algotester Lab 4

Lab 4v1

Limits: 1 sec., 256 MiB

Вам дано 2 цілих чисел масиви, розміром N та M .

Ваше завдання вивести:

1. Різницю $N-M$
2. Різницю $M-N$
3. Їх перетин
4. Їх об'єднання
5. Їх симетричну різницю

Input

У першому рядку ціле число N - розмір масиву 1

У другому рядку N цілих чисел - елементи масиву 1

У третьому рядку ціле число M - розмір масиву 2

У четвертому рядку M цілих чисел - елементи масиву 2

Output

Вивести результат виконання 5 вищезазначених операцій у форматі:

У першому рядку ціле число N - розмір множини

У наступному рядку N цілих чисел - посортована у порядку зростання множина



algotester_lab_4.1_variant_1_ivan_holeichuk (1).cpp



algotester_lab_4.2_variant_1_ivan_holeichuk (1).cpp

Epic 5 Task 7 - Lab# programming: Algotester Lab 6



algotester_lab_6_variant_1_ivan_holeichuk (1).cpp

Lab 6v1

Limits: 2 sec., 256 MiB

Вам дано N слів та число K .

Ваше завдання перерахувати букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останньої до першої у алфавіті. Букву потрібно виводити лише один раз.

У випадку якщо таких букв немає - вивести "Empty!".

Input

Цілі числа N та K - загальна кількість слів та мінімальна кількість слів щоб враховувати букви цього слова в результаті.

N стрічок s

Output

У першому рядку ціле число M - кількість унікальних букв

У другому рядку унікальні букви через пробіли

Constraints

$$1 \leq K \leq N \leq 10^5$$

$$1 \leq |s_i| \leq 10$$

$$s_i \in a..z$$

Epic 5 Task 8 - Practice# programming: Class Practice Task



practice_work_task_1_ivanholeichuk (2).cpp



practice_work_task_2_ivanholeichuk (2).cpp

Задача №1 – Запис текстової стрічки у файл із заданим ім'ям

Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };  
FileOpResult write_to_file(char *name, char *content);
```

Умова задачі:

- створити файл із заданим ім'ям; якщо файл існує – перезаписати його вміст
- написати код стійкий до різних варіантів вхідних параметрів
- `name` – ім'я; може не включати шлях
- записати у файл вміст стрічки `content`; прочитати `content` із стандартного вводу
- повернути статус операції: `Success` – все пройшло успішно, `Failure` – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файлу.

Мета задачі

Розуміння методів роботи з файлами: Робота з файлами є одним з базових навиків програмування. Реалізація функції створення та запису в файл допоможе освоїти практичні навички роботи з файлами з використанням стандартної бібліотеки C++. Для виконання завдання студент має навчитись використовувати методи відкриття файлу, запису масиву даних у файл, закриття файлу та обробки помилок чи станів операції на кожному з етапів.

Розвиток алгоритмічного мислення: Запис у файл включає набір операцій, які найкраще вкладаються в концепцію алгоритма, як список детальних кроків. Імплементація цієї функції наочно демонструє створення алгоритма у програмуванні.

Освоїти навички роботи з текстовими стрічками: завдання допоможе освоїти роботу з C стрічка, які є масивами з нульовим символом в кінці. Типові концепції при роботі з C стрічками це арифметика вказників, ітерація по стрічці, копіювання частини стрічки, розбиття на токени по заданому символу.

Розвинути навички розв'язувати задачі: Запис у файл може супроводжуватись набором станів (немає доступу на створення, недостатньо місця, ін.), які необхідно передбачити у алгоритмі. Аналіз цих станів дозволяє розвинути навик розв'язання інженерних задач у програмуванні.

Задача №2 – Копіювання вмісту файлу у інший файл

Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };  
FileOpResult copy_file(char *file_from, char *file_to);
```

Умова задачі:

- копіювати вміст файлу з ім'ям `file_from` у файл з ім'ям `file_to`; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів
- `file_from`, `file_to` – можуть бути повним або відносним шляхом
- повернути статус операції: `Success` – все пройшло успішно, `Failure` – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файлу.

Мета задачі

Розуміння методів роботи з файлами: Робота з файлами є одним з базових навиків програмування. Реалізація функції копіювання вмісту файлу допоможе освоїти практичні навички роботи з файлами з використанням стандартної бібліотеки C++. Для виконання завдання студент має навчитись використовувати методи відкриття файлу, читання вмісту файлу, запису масиву даних у файл, закриття файлу та обробки помилок чи станів операції на кожному з етапів.

Розвиток алгоритмічного мислення: Читання та запис у файл включає набір операцій, які найкраще вкладаються в концепцію алгоритма, як список детальних кроків. Імплементація цієї функції наочно демонструє створення алгоритма у програмуванні.

Освоїти навички роботи з потоком даних: завдання допоможе освоїти роботу з потоками даних (концепція реалізована в STL як набір класів "stream" - `fstream`, `stringstream`, `ifstream` та ін.). Концепція потоку даних дозволяє абстрагувати роботу з джерелами та приймачами даних та писати з її допомогою високорівневий код.

Розвинути навички розв'язувати задачі: Операції читання з файлу та запис у файл можуть супроводжуватись набором різних станів (немає доступу на читання чи створення, недостатньо місця, ін.), які необхідно передбачити у алгоритмі. Аналіз цих станів дозволяє розвинути навик розв'язання інженерних задач у програмуванні.

Epic 5 Task 9 - Practice# programming: Self Practice Task



self_practice_work_ivan_holeichuk (1).cpp

Робота у команді:



Ми зібрались одинраз та обговорили всі деталі епіку. Домовились що якщо в когось будуть питання то зберемось щераз.

Висновок: У результаті роботи вивчено основи роботи з файлами та рядковими змінними у C++. Розглянуто базові операції з файлами (відкриття,

читання, запис, закриття), роботу з файловими дескрипторами та перевірку стану файлів. Досліджено обробку текстових і бінарних файлів, включаючи форматування, парсинг і серіалізацію даних.

Проаналізовано методи роботи з рядковими змінними (`char`, `string`), їх об'єднання, порівняння та пошук. Вивчено використання потоків вводу/виводу (`ifstream`, `ofstream`, `fstream`) і основи створення власних бібліотек у C++. Розглянуто обробку помилок для забезпечення стабільності програм.

Ці знання створюють базу для вирішення складних програмних завдань.