Звіт до лабораторної роботи №2

Завдання 1

Мета роботи:

Розробити програму, яка:

1. Відкриває текстовий файл **data.txt** для читання.
2. Підраховує кількість рядків у файлі.
3. Виводить кожен рядок разом із кількістю слів у ньому.
4. Записує результати у новий файл **results.txt**.

Опис програми:

Програма *reader.py* реалізує функцію **processfile**, яка:

- Приймає назви вхідного та вихідного файлів як аргументи.
- Виконує обробку файлу згідно з вимогами завдання.

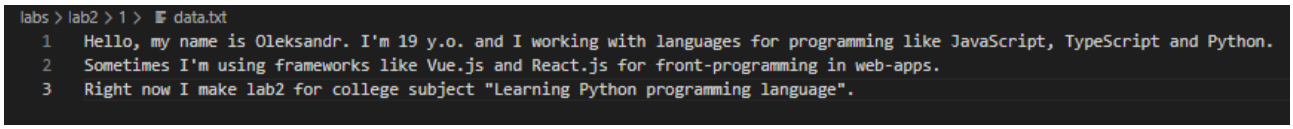
Кроки виконання

1. **Відкриття файлу для читання**:
    - За допомогою **with open(input_filename, 'r', encoding='utf-8') as infile** програма відкриває файл для читання. Використання **encoding='utf-8'** забезпечує коректне відображення тексту в разі наявності нестандартних символів.
    - Усі рядки зчитуються за допомогою **infile.readlines()** і зберігаються у списку **lines**.
2. **Підрахунок кількості рядків**:
    - Програма визначає кількість рядків за допомогою **len(lines)** і виводить це значення в термінал.
3. **Обробка кожного рядка та підрахунок слів**:
    - За допомогою циклу **for i, line in enumerate(lines, start=1)** програма проходить кожен рядок, де **i** – номер рядка.
    - Кількість слів у рядку визначається за допомогою **len(line.split())**, де **line.split()** розділяє рядок на слова за пробілами.
    - Результат додається до списку *results* у форматі **"Рядок {i}: {line.strip()} (Кількість слів: {word_count})\n"**.
4. **Запис результатів у файл results.txt**:
    - Програма відкриває новий файл **results.txt** для запису.
    - Спочатку записується загальна кількість рядків, а потім у кожному рядку виводиться текст разом із кількістю слів.
    - Програма виводить повідомлення про успішний запис результатів.

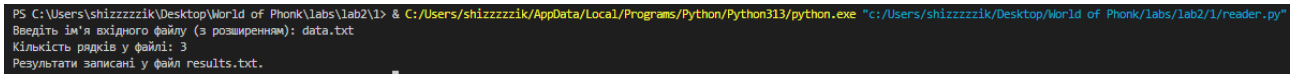
5. **Обробка помилок**:
    - У випадку, якщо вхідний файл не знайдено, програма виводить повідомлення про помилку *FileNotFoundError*.
    - Для інших можливих помилок додається загальне виключення **Exception**, яке виводить текст помилки.

Тестування програми

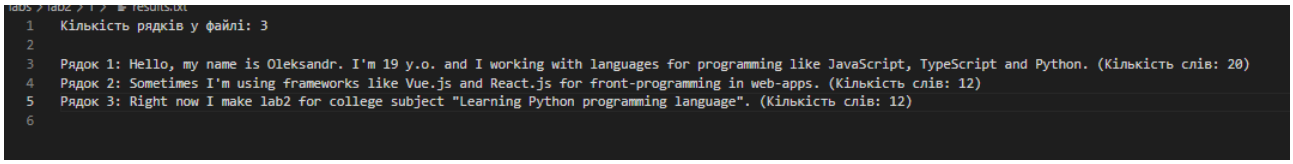
data.txt



Запуск програми reader.py



results.txt



Висновок

У процесі виконання завдання я розробив програму, яка обробляє текстовий файл: зчитує дані, аналізує їх, і записує результати в новий файл. Програма коректно обробляє випадки відсутності файлу, а також форматує вихідні дані, що дає змогу отримати зручний для читання файл результатів.

Завдання 2

Мета роботи

Розробити програму, яка:

1. Виконує GET-запит до API сервісу JSONPlaceholder для отримання списку постів.
2. Зберігає отримані дані у форматі JSON у файл posts.json.

Опис програми

Програма f**etch_and_save_posts.py** використовує бібліотеки **requests** для виконання HTTP-запитів та **json** для збереження отриманих даних у форматі **JSON**. Головна функція програми – **fetch_and_save_posts**, що приймає *URL* для запиту та ім'я файлу для збереження даних.

Кроки виконання

1. **Виконання GET-запиту:**
    - Програма ініціює запит **GET** до *API* сервісу за допомогою **requests.get(url)**.
    - Після виконання запиту перевіряється статус-код відповіді. Якщо запит успішний (*код 200*), дані передаються далі для обробки.
2. **Обробка отриманих даних:**

- Якщо запит завершився успішно, програма використовує метод response.json() для отримання даних у форматі JSON. Це список об'єктів, де кожен об’єкт представляє окремий пост з полями, такими як userId, id, title та body.

3. **Збереження даних у файл:**
    - Отримані дані зберігаються у файл **posts** у форматі *JSON*, використовуючи **json.dump**.
    - Аргумент **ensure_ascii=False** дозволяє зберігати текст у вигляді *Unicode*, а **indent=4** забезпечує читабельне форматування *JSON* із відступами.
4. **Обробка помилок:**
    - Якщо статус-код відповіді відрізняється від 200, програма виводить повідомлення про невдалий запит разом із кодом статусу.
    - Загальне виключення **Exception** обробляє будь-які інші помилки, що можуть виникнути під час виконання програми.

Тестування програми

Запуск програми fetch_and_save_posts.py

```
PS C:\Users\shizzzzzik\Desktop\World of Phonk\labs\lab2\2> & C:/Users/shizzzzzik/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shizzzzzik/Desktop/World of Phonk/labs/lab2/2/fetch_and_save_posts.py"
Дані успішно збережено у файл posts.json
```

Збережені дані отримані з GET запиту у posts.json


```
labs > lab2 > 2 > {} posts.json > ...
[
    {
        "userId": 10,
        "id": 96,
        "title": "quaerat velit veniam amet cupiditate aut numquam ut sequi",
        "body": "in non odio excepturi sint eum\nlabore voluptates vitae quia qui et\ninventore itaque rerum\nveniam non exercitationem delectus aut"
    },
    {
        "userId": 10,
        "id": 97,
        "title": "quas fugiat ut perspiciatis vero provident",
        "body": "eum non blanditiis soluta porro quibusdam voluptas\nvel voluptatem qui placeat dolores qui velit aut\nvel inventore aut cumque culpa explicabo aliquid at\r
    },
    {
        "userId": 10,
        "id": 98,
        "title": "laboriosam dolor voluptates",
        "body": "doloremque ex facilis sit sint culpa\nsoluta assumenda eligendi non ut eius\nsequi ducimus vel quasi\nveritatis est dolores"
    },
    {
        "userId": 10,
        "id": 99,
        "title": "temporibus sit alias delectus eligendi possimus magni",
        "body": "quo deleniti praesentium dicta non quod\naut est molestias\nmolestias et officia quis nihil\nitaque dolorem quia"
    },
    {
        "userId": 10,
        "id": 100,
        "title": "at nam consequatur ea labore ea harum",
        "body": "cupiditate quo est a modi nesciunt soluta\nipsa voluptas error itaque dicta in\nautem qui minus magnam et distinctio eum\naccusamus ratione error aut"
    }
]
```


Висновок

У процесі виконання цього завдання я навчився:

1. Виконувати HTTP-запити за допомогою бібліотеки requests.
2. Перетворювати отриману відповідь у формат JSON та зберігати її у файл.
3. Обробляти можливі помилки при виконанні запитів та збереженні файлів.

Програма коректно реалізує отримання даних з API та їх збереження, що робить її зручною для отримання та архівування інформації.