

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

ЗВІТ
про виконання лабораторної роботи № 7
з дисципліни
«Спеціалізовані мови програмування»

Виконав:
студент гр. ІТ-32
Федик В.А

Прийняв:
доц.каф. ІСМ
Щербак С.С.

Львів-2023

Мета. Створення консольного об'єктно - орієнтованого додатка з використанням API.

Хід виконання:

Завдання 1. Вибір провайдера API Виберіть надійний API, який надає через HTTP необхідні дані для віддаленого зберігання, вивантаження або реалізуйте свій. Для прикладу це може бути jsonplaceholder.org.

Завдання 2. Інтеграція API Виберіть бібліотеку для роботи з API та обробки HTTP запитів (для прикладу це може бути бібліотека Requests). Інтегруйте обраний API в ваш консольний додаток на Python. Ознайомтеся з документацією API та налаштуйте необхідний API-ключ чи облікові дані.

Завдання 3. Введення користувача Розробіть користувацький інтерфейс, який дозволяє користувачам візуалізувати всі доступні дані в табличному вигляді та у вигляді списку. Реалізуйте механізм для збору та перевірки введеного даних користувачем.

Завдання 4. Розбір введення користувача Створіть розбірник для видобування та інтерпретації виразів користувача на основі регулярних виразів, наприклад, для візуалізації дат, телефонів, тощо. Переконайтеся, що розбірник обробляє різні формати введення та надає зворотний зв'язок про помилки.

Завдання 5. Відображення результатів Реалізуйте логіку для візуалізації даних через API в консолі. Обробляйте відповіді API для отримання даних у вигляді таблиць, списків. Заголовки таблиць, списків мають виділятися кольором та шрифтом, які задається користувачем.

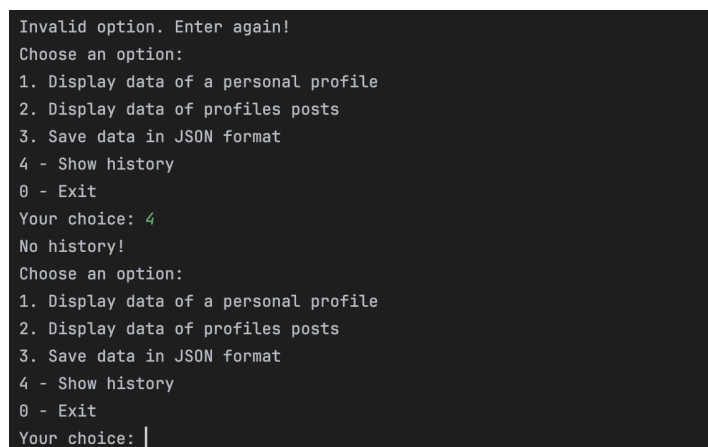
Завдання 6. Збереження даних Реалізуйте можливості збереження даних у чіткому та читабельному форматі JSON, CSV та TXT.

Завдання 7. Обробка помилок Розробіть надійний механізм обробки помилок для керування помилками API, некоректним введенням користувача та іншими можливими проблемами. Надавайте інформативні повідомлення про помилки.

Завдання 8. Ведення історії обчислень Включіть функцію, яка реєструє запити користувача, включаючи введені запити та відповідні результати. Дозвольте користувачам переглядати та рецензувати історію своїх запитів.

Завдання 9. Юніт-тести Напишіть юніт-тести для перевірки функціональності вашого додатку. Тестуйте різні операції, граничні випадки та сценарії помилок.

На рис. 1 зображено знімок екрану із середовища розробки.



```
Invalid option. Enter again!
Choose an option:
1. Display data of a personal profile
2. Display data of profiles posts
3. Save data in JSON format
4 - Show history
0 - Exit
Your choice: 4
No history!
Choose an option:
1. Display data of a personal profile
2. Display data of profiles posts
3. Save data in JSON format
4 - Show history
0 - Exit
Your choice: |
```

Рис. 1 Виконання програми

Посилання на GitHub-репозиторій із кодом: <https://github.com/fedykvitalik2004/spl>

Висновки: Я створив проєкт, який надав мені цінний досвід роботи з API, дизайну користувацького інтерфейсу, валідації введення, обробки помилок та тестування.