Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра ІСМ



**Звіт**

до лабораторної роботи № 7

з дисципліни

​*Спеціалізовані мови програмування*

на тему:

“**Робота з API та веб-сервісами**”

Виконав студент РІ-31

**Шапошніков Дмитро**

Прийняв: Щербак С.С.

Львів – 2024

**Мета роботи:** Створення консольного об’єктно - орієнтованого додатка з використанням API та патернів проектування

**Завдання лабораторної роботи**

Завдання 1: Вибір провайдера API та патернів проектування

Виберіть надійний API, який надає через HTTP необхідні дані для віддаленого зберігання, вивантаження або реалізуйте свій. Для прикладу це може бути jsonplaceholder.org. Крім того, оберіть 2-3 патерна проектування для реалізаціі імплементаціі цієї лабораторноі роботи. Для прикладу, це може бути патерн Unit of Work та Repository

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 користувальницький 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 виділяться 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.

Файл запуску:

    def run(self):

        print("\nWelcome to the News API Interface!")

        while True:

            try:

                print("\nMenu:")

                print("1. Search news")

                print("2. Get top headlines")

                print("3. View search history")

                print("4. Save last results")

                print("5. Change title color")

                print("6. Exit")

                choice = input("\nEnter your choice (1-6): ")

                if choice == '1':

                    self.search\_news()

                elif choice == '2':

                    self.get\_top\_headlines()

                elif choice == '3':

                    self.history.display()

                elif choice == '4':

                    all\_results = self.uow.repository.get\_all()

                    if not all\_results:

                        print("No results to save. Please perform a search first.")

                    else:

                        filetype = input("Enter file type to save (json/csv/txt): ").lower()

                        if filetype in ['json', 'csv', 'txt']:

                            self.save\_to\_file(all\_results[-1], f"news\_results.{filetype}", filetype)

                        else:

                            print("Invalid file type. Please choose json, csv, or txt.")

                elif choice == '5':

                    self.change\_title\_color()

                elif choice == '6':

                    print("Thank you for using the News API Interface. Goodbye!")

                    break

                else:

                    print("Invalid choice. Please enter a number between 1 and 6.")

            except Exception as e:

                print(f"An error occurred: {e}")

                print("Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

    API\_KEY = "3b51eb9e98f24ee79b98ffcc873ebfc9"

    news\_api = NewsAPI(API\_KEY)

    ui = UserInterface(news\_api)

    ui.run()

**Висновок**

Виконавши ці завдання, я створив проект, який надасть мені цінний досвід роботи з API, дизайну користувацького інтерфейсу, валідації введення, обробки помилок та тестування.

GitHub: https://github.com/RomchykPrytula/SMP