Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №7

з дисципліни «Спеціалізовані мови програмування»

на тему

«Робота з API та веб-сервісами»

Виконав:

Костюк В. О.

Перевірив:

Щербак С. С.

Львів 2024

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

**План 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.

**Реалізація:**

**Файл api\_handler.py:**  
import requests

import logging

# Налаштування URL API

API\_URL = "https://jsonplaceholder.typicode.com/posts"  # JSON Placeholder API

class APIHandler:

    """Клас для роботи з API"""

    def fetch\_data(self):

        """Метод для отримання даних з API"""

        try:

            response = requests.get(API\_URL)

            response.raise\_for\_status()

            data = response.json()

            logging.info("Дані успішно отримані.")

            return data

        except requests.exceptions.RequestException as e:

            logging.error(f"Помилка API: {e}")

            print(f"Помилка API: {e}")

            return None

**Файл data\_saver.py:**

import json

import csv

class DataSaver:

    """Клас для збереження даних у різні формати"""

    def save\_data(self, data, file\_format="json"):

        """Збереження даних у обраному форматі"""

        filename = f"data.{file\_format}"

        try:

            if file\_format == "json":

                with open(filename, "w") as f:

                    json.dump(data, f, indent=4)

            elif file\_format == "csv":

                with open(filename, "w", newline='') as f:

                    writer = csv.writer(f)

                    writer.writerow(data[0].keys())

                    for row in data:

                        writer.writerow(row.values())

            elif file\_format == "txt":

                with open(filename, "w") as f:

                    f.write(str(data))

            print(f"Дані збережено у файл {filename}.")

        except Exception as e:

            print(f"Помилка при збереженні даних: {e}")

**Файл data\_visualizer.py:**

from tabulate import tabulate

import json

from colorama import Fore, Style

class DataVisualizer:

    """Клас для візуалізації даних"""

    def display\_data(self, data, display\_type="table"):

        """Відображення даних у табличному або списковому форматі"""

        if display\_type == "table":

            print(Fore.BLUE + Style.BRIGHT + tabulate(data, headers="keys", tablefmt="grid") + Style.RESET\_ALL)

        elif display\_type == "list":

            for item in data:

                print(Fore.GREEN + json.dumps(item, indent=4) + Style.RESET\_ALL)

**Файл user\_input\_handler.py:**

class UserInputHandler:

    """Клас для обробки введення користувача"""

    def get\_user\_choice(self):

        print("1. Показати всі дані (таблиця)")

        print("2. Показати дані у форматі списку")

        print("3. Зберегти дані")

        return input("Виберіть опцію: ")

    def validate\_choice(self, choice):

        return choice in ["1", "2", "3"]

**Файл main.py:**

from lab7.api\_handler import APIHandler

from lab7.user\_input\_handler import UserInputHandler

from lab7.data\_visualizer import DataVisualizer

from lab7.data\_saver import DataSaver

def main():

    api\_handler = APIHandler()

    user\_input\_handler = UserInputHandler()

    data\_visualizer = DataVisualizer()

    data\_saver = DataSaver()

    # Отримуємо дані з API

    data = api\_handler.fetch\_data()

    if data is None:

        return

    # Головний цикл програми

    while True:

        choice = user\_input\_handler.get\_user\_choice()

        if user\_input\_handler.validate\_choice(choice):

            if choice == "1":

                data\_visualizer.display\_data(data, display\_type="table")

            elif choice == "2":

                data\_visualizer.display\_data(data, display\_type="list")

            elif choice == "3":

                file\_format = input("Оберіть формат (json/csv/txt): ").strip().lower()

                data\_saver.save\_data(data, file\_format=file\_format)

        else:

            print("Неправильний вибір. Спробуйте ще раз.")

        next\_action = input("Бажаєте продовжити? (y/n): ").strip().lower()

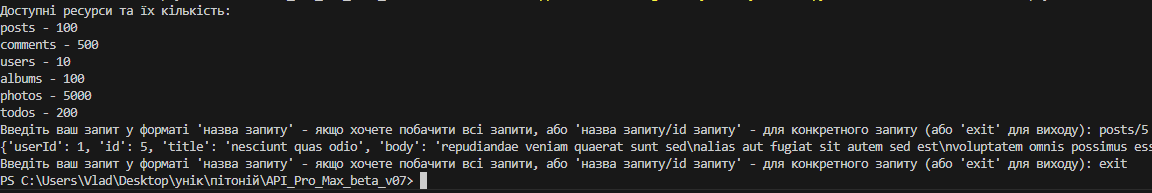
        if next\_action != "y":

            break

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

    main()

**Результат виконання:**

****

**Висновок:** на цій лабораторній роботі я створив API-проект використовуючи сторонній ресурс з даними