

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №7
з дисципліни
СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ
на тему
Робота з API та веб-сервісами

Виконав:
ст. гр. РІ-21сп
Рак В.П.
Прийняв:
Щербак С.С.

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

Завдання

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 паттерна проектування для реалізації імплементації цієї лабораторної роботи. Для прикладу, це може бути паттерн 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). Інтегрувати 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. Обробляти 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.

Виконання роботи

Текст програмної реалізації:

main.py:

"""

Основний модуль програми, який керує взаємодією з користувачем, отримує дані через API, відображає їх та надає можливість збереження у файли.

Функція:

- `main()` : основна функція програми для керування процесом отримання, відображення даних та їх збереження.

"""

```
from dal import APIClient, DataSaver
from bll import Repository, UnitOfWork
from ui import UserInterface
```

```
def main():
```

```
    """
```

Основна функція програми. Запускає цикл взаємодії з користувачем, отримує дані через API,

відображає їх у таблиці та надає можливість збереження даних у форматах JSON, CSV, або TXT.

Програма працює в нескінченному циклі, поки користувач не вибере 'quit'.

Користувач може вибрати між переглядом даних користувачів або постів.

Після перегляду даних, користувач може зберегти їх у вибраному форматі.

```
    """
```

```
    api_url = "https://jsonplaceholder.typicode.com"
```

```
    api_client = APIClient(api_url)
```

```
    repository = Repository(api_client)
```

```
    uow = UnitOfWork(repository)
```

```
    while True:
```

```
        # Запитуємо користувача про ввід.
```

```
user_input = UserInterface.prompt_user_input()

# Перевіряємо правильність введеного значення.
if not UserInterface.validate_input(user_input):
    continue

# Вихід з програми, якщо введено 'quit'.
if user_input == 'quit':
    break

# Отримуємо дані через UnitOfWork.
data = uow.get_all_data()

# Відображаємо дані залежно від вибору користувача.
if user_input == 'users':
    UserInterface.display_data(data['users'], format_type='table', color=True)
elif user_input == 'posts':
    UserInterface.display_data(data['posts'], format_type='table', color=True)

# Запитуємо користувача, чи хоче він зберегти дані.
save_option = input("Do you want to save the data? (y/n): ").strip().lower()

# Якщо користувач хоче зберегти, запитуємо формат.
if save_option == 'y':
    file_format = input("Choose file format (json/csv/txt): ").strip().lower()

# Зберігаємо у вибраному форматі.
if file_format == 'json':
    DataSaver.save_to_json(data[user_input], f"{user_input}.json")
elif file_format == 'csv':
    DataSaver.save_to_csv(data[user_input], f"{user_input}.csv")
elif file_format == 'txt':
    DataSaver.save_to_txt(data[user_input], f"{user_input}.txt")
```

else:

```
print("Invalid format.")
```

Перевірка, чи є цей файл основним для виконання.

```
if __name__ == "__main__":
```

```
    main()
```

Результат роботи програми:

```
Enter 'users' to view users, 'posts' to view posts, 'quit' to exit:
Your choice: users

+-----+-----+-----+-----+-----+-----+
| id | name | username | email | company | address |
+-----+-----+-----+-----+-----+-----+
| 1 | Leanne Graham | Bret | Sincere@april.biz | {} | {'street': 'Kulas Light', 'suite': 'Apt. 556', 'city': 'Gwenborough', 'zipcode': '92998-3874', 'geo': {'lat': '-37.3159', 'lng': '81.1496'}} |
| 2 | Ervin Howell | Antonette | Shanna@melissa.tv | {} | {'street': 'Victor Plains', 'suite': 'Suite 879', 'city': 'Wisokyburgh', 'zipcode': '90566-7771', 'geo': {'lat': '-43.9509', 'lng': '-34.4618'}} |
| 3 | Clementine Bauch | Samantha | Nathan@yesenia.net | {} | {'street': 'Douglas Extension', 'suite': 'Suite 847', 'city': 'McKenziehaven', 'zipcode': '59590-4157', 'geo': {'lat': '-68.6182', 'lng': '-47.0653'}} |
| 4 | Patricia Lebsack | Karianne | Julianne.Oconner@kory.org | {} | {'street': 'Hoeger Mall', 'suite': 'Apt. 692', 'city': 'South Elvis', 'zipcode': '53919-4257', 'geo': {'lat': '29.4572', 'lng': '-164.2998'}} |
| 5 | Chelsey Dietrich | Kamren | Lucio.Hettinger@annie.ca | {} | {'street': 'Skiles Walks', 'suite': 'Suite 351', 'city': 'Roscoeview', 'zipcode': '33263', 'geo': {'lat': '-31.8129', 'lng': '62.5342'}} |
| 6 | Mrs. Dennis Schulist | Leopoldo Corkery | Karley.Dach@jasper.info | {} | {'street': 'Norberto Crossing', 'suite': 'Apt. 950', 'city': 'South Christy', 'zipcode': '23585-1337', 'geo': {'lat': '-71.4197', 'lng': '71.7478'}} |
| 7 | Kurtis Weissnat | Elwyn Skiles | Telly.Hoeger@billy.biz | {} | {'street': 'Rex Trail', 'suite': 'Suite 289', 'city': 'Howemouth', 'zipcode': '58804-1099', 'geo': {'lat': '24.8918', 'lng': '21.8984'}} |
| 8 | Nicholas Runolfsson | Maxine Mienow | Sherwood@rosamond.me | {} | {'street': 'Ellsworth Summit', 'suite': 'Suite 729', 'city': 'Aliyaview', 'zipcode': '45169', 'geo': {'lat': '-14.3998', 'lng': '-120.7677'}} |
| 9 | Glenna Reichert | Delphine | Chain.McDermott@dana.io | {} | {'street': 'Dayna Park', 'suite': 'Suite 449', 'city': 'Bartholomebury', 'zipcode': '76495-3109', 'geo': {'lat': '24.6463', 'lng': '-168.8889'}} |
| 10 | Clementina DuBuque | Moriah Stanton | Rey.Padberg@karina.biz | {} | {'street': 'Kattie Turnpike', 'suite': 'Suite 198', 'city': 'Lebsackbury', 'zipcode': '31428-2261', 'geo': {'lat': '-38.2386', 'lng': '57.2232'}} |
+-----+-----+-----+-----+-----+-----+

Do you want to save the data? (y/n):
```

Рис. 1 - Результат виведення users

Висновок: У ході виконання лабораторної роботи я створив проект, який надає цінний досвід роботи з API, дизайну користувацького інтерфейсу, валідації введення, обробки помилок та тестування. Цей проект виконує різні дії з використанням API, а також тестує додаток та виводить Coverage Report.