МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №85

з дисципліни Спеціалізовані мови програмування

на тему

Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python

~~Робота з API та веб-сервісами~~

Виконав:
студент групи PI-21сп
Владислав ДМИТРЕНКО

Formatted: Ukrainian
Formatted: Ukrainian
Formatted: Ukrainian
Formatted: Ukrainian
Formatted: Centered
Formatted: Ukrainian
Formatted: Ukrainian

Львів – 2024

Мета: Розробка додатка для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП (наслідування, інкапсуляція, поліморфізм) ~~Створення консольного об'єктно - орієнтованого додатка з використанням API та патернів проектування~~

План роботи

Завдання 1: Оберіть CSV-набір даних, який ви хочете візуалізувати. Переконайтеся, що він містить відповідні дані для створення змістовних візуалізацій.

Завдання 2: Напишіть код для завантаження даних з CSV-файлу в ваш додаток Python. Використовуйте бібліотеки, такі як Pandas, для спрощення обробки даних.

Завдання 3: Визначте екстремальні значення по стовцям

Завдання 4: Визначте, які типи візуалізацій підходять для представлення вибраних наборів даних. Зазвичай це може бути лінійні графіки, стовпчикові діаграми, діаграми розсіювання, гістограми та секторні діаграми.

Завдання 5: Попередньо обробіть набір даних за необхідністю для візуалізації. Це може включати виправлення даних, фільтрацію, агрегацію або трансформацію.

Завдання 6: Створіть базову візуалізацію набору даних, щоб переконатися, що ви можете відображати дані правильно за допомогою Matplotlib. Розпочніть з простої діаграми для візуалізації однієї змінної.

Завдання 7: Реалізуйте більш складні візуалізації, виходячи з характеристик набору. Поекспериментуйте з різними функціями Matplotlib та налаштуваннями.

Завдання 8: Навчіться створювати кілька піддіаграм в межах одного малюнка для відображення декількох візуалізацій поруч для кращого порівняння.

Formatted: Centered

Formatted: Font: Bold, Ukrainian

Formatted: Ukrainian

Formatted: Font: Bold, Ukrainian

Formatted: Ukrainian

Formatted: Font: Bold, Ukrainian

Formatted: Ukrainian

Formatted: Font: Bold, Ukrainian

Formatted: Ukrainian

Formatted: Font: Bold, Ukrainian

Formatted: Ukrainian

Formatted: Font: Bold, Ukrainian

Formatted: Ukrainian

Formatted: Font: Bold, Ukrainian

Formatted: Ukrainian

Завдання 9: Реалізуйте функціональність для експорту візуалізацій як зображень (наприклад, PNG, SVG) або інтерактивних веб-додатків (наприклад, HTML)

~~**Завдання 1:** 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 реалізації імплементації цієї лабораторної роботи. Для прикладу, це може бути патерн Unit of Work та Repository~~

~~**Завдання 2:** 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:** 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:** 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:** 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:** Реалізуйте можливості збереження даних у чіткому та читабельному форматі JSON, CSV та TXT~~

Formatted: Font: Bold, Ukrainian
Formatted: Ukrainian
Formatted: Font: Not Bold, Ukrainian
Formatted: Justified
Formatted: Ukrainian

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

~~**Завдання 8:** Включіть функцію, яка реєструє запити користувача, включаючи введені запити та відповідні результати. Дозвольте користувачам переглядати та рецензувати історію своїх запитів.~~

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

Результати тестування:

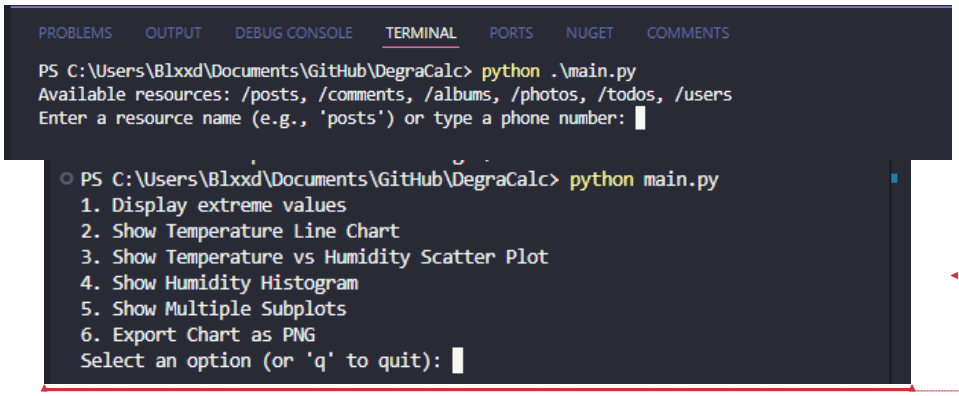

Рис. 1. Користувацький інтерфейс~~Вибір ресурсу з даними.~~

```
Available resources: /posts, /comments, /albums, /photos, /todos, /users
Enter a resource name (e.g., 'posts') or type a phone number: todos
                                    Data
```

Userid	Id	Title	Completed
1	1	delectus aut autem	False
1	2	quis ut nam facilis et officia qui	False
1	3	fugiat veniam minus	False
1	4	et porro tempora	True
1	5	laboriosam mollitia et enim quasi adipisci quia provident illum	False
1	6	qui ullam ratione quibusdam voluptatem quia omnis	False
1	7	illo expedita consequatur quia in	False
1	8	quo adipisci enim quam ut ab	True
1	9	molestiae perspiciatis ipsa	False
1	10	illo est ratione doloremque quia maiores aut	True
1	11	vero rerum temporibus dolor	True
1	12	ipsa repellendus fugit nisi	True
1	13	et doloremque nulla	False
1	14	repellendus sunt dolores architecto voluptatum	True
1	15	ab voluptatum amet voluptas	True
1	16	accusamus eos facilis sint et aut voluptatem	True
1	17	quo laboriosam deleniti aut qui	True
1	18	dolorum est consequatur ea mollitia in culpa	False
1	19	molestiae ipsa aut voluptatibus pariatur dolor nihil	True
1	20	ullam nobis libero sapiente ad optio sint	True
2	21	suscipit repellat esse quibusdam voluptatem incidunt	False
2	22	distinctio vitae autem nihil ut molestias quo	True
2	23	et itaque necessitatibus maxime molestiae qui quas velit	False

```
Select an option (or 'q' to quit): 1
Temperature: Min=-8.751103306122198, Max=34.334323546035684
Humidity: Min=20.10315523092394, Max=99.34085119341344
WindSpeed: Min=0.0205347525669341, Max=19.902143508601725
Rainfall: Min=0.0, Max=2.5
1. Display extreme values
2. Show Temperature Line Chart
3. Show Temperature vs Humidity Scatter Plot
4. Show Humidity Histogram
5. Show Multiple Subplots
6. Export Chart as PNG
Select an option (or 'q' to quit):
```

Рис. 2. Вивід даних

10	198	quis eius est sint explicabo	True
10	199	numquam repellendus a magnam	True
10	200	ipsam aperiam voluptates qui	False

```
Choose a format to save data (json, csv, txt): csv
Data saved to data_20241128_220956.csv
PS C:\Users\Blxxd\Documents\GitHub\DegraCalc>
```

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

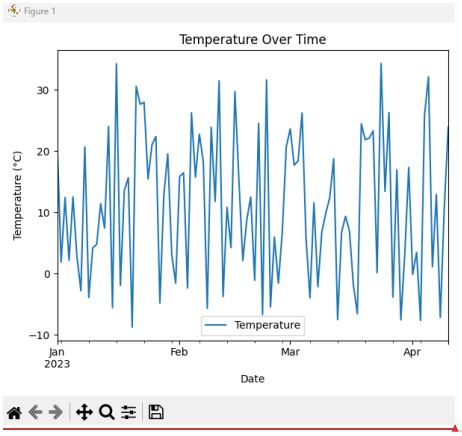

Рис 3. Вигляд лінійної діаграми~~Збереження даних у файлі~~

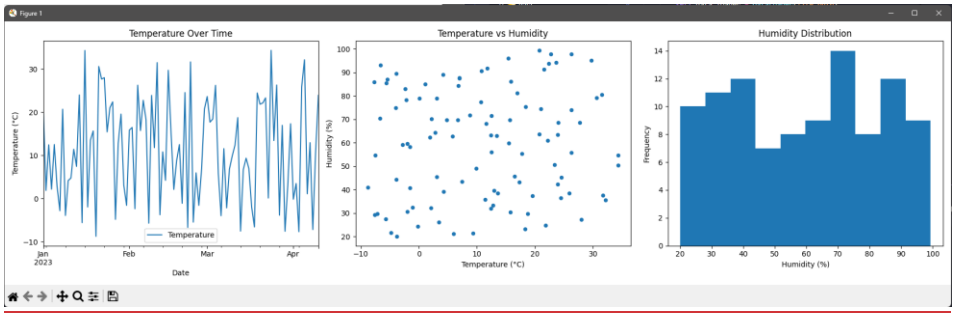

Рис. 4. Вигляд всіх діаграм разом

Formatted: Ukrainian

data_20241128_220956.csv U

data_20241128_220956.csv > data

```
userId,id,title,completed
1,1,delectus aut autem,False
1,2,quis ut nam facilis et officia qui,False
1,3,fugiat veniam minus,False
1,4,et porro tempora,True
1,5,laboriosam mollitia et enim quasi adipisci quia provident illum,False
1,6,qui ullam ratione quibusdam voluptatem quia omnis,False
1,7,illo expedita consequatur quia in,False
1,8,quo adipisci enim quam ut ab,True
1,9,molestiae perspiciatis ipsa,False
1,10,illo est ratione doloremque quia maiores aut,True
1,11,vero rerum temporibus dolor,True
1,12,ipsa repellendus fugit nisi,True
1,13,et doloremque nulla,False
1,14,repellendus sunt dolores architecto voluptatum,True
1,15,ab voluptatum amet voluptas,True
1,16,accusamus eos facilis sint et aut voluptatem,True
1,17,quo laboriosam deleniti aut qui,True
1,18,dolorum est consequatur ea mollitia in culpa,False
```

~~Рис 4. Результат збереження файлу~~

Текст контроллера даних ~~текс функції транзакції~~

Formatted: Font: (Default) Times New Roman, 14 pt

Formatted: Ukrainian

Formatted: Normal, Space After: 10 pt, Line spacing: Multiple 1,15 li

```
class DataController:
    def __init__(self, file_path):
        self.data_loader = DataLoader(file_path)
        self.data = self.data_loader.load_data()
        if self.data.empty:
            print("Warning: The data is empty or could not be loaded
correctly.")
        self.analysis = DataAnalysis(self.data)

    def display_extremes(self):
        extremes = self.analysis.get_extreme_values()
        for key, value in extremes.items():
            print(f'{key}: Min={value[0]}, Max={value[1]}')

    def get_plot_instance(self, plot_type):
        if plot_type == "line":
            return LineChart()
        elif plot_type == "scatter":
            return ScatterPlot()
        elif plot_type == "histogram":
            return Histogram()
        else:
            return None

    def plot_chart(self, plot_type):
        plot_instance = self.get_plot_instance(plot_type)
```

```
        if plot_instance:
            plot_instance.plot(self.data)
            plot_instance.show()
# core/transaction.py
from datetime import datetime

class Transactions:
    def __init__(self):
        self.history = []

    def register_request(self, query, result):
        self.history.append({
            "query": query,
            "result": result,
            "timestamp": datetime.now().isoformat()
        })

    def log_save_action(self, phone, format, filename):
        """Log the saving action to a file with details."""
        entry = {
            "action": "save",
            "phone": phone,
            "format": format,
            "filename": filename,
            "timestamp": datetime.now().isoformat()
        }
        self.history.append(entry)
        self._write_log(entry)

    def _write_log(self, entry, filename="logs/history.log"):
        with open(filename, "a") as file:
            if "action" in entry:
                file.write(f"{entry['timestamp']} - Action:
{entry['action']}, Phone: {entry['phone']}, Format: {entry['format']},
Filename: {entry['filename']}\n")
            else:
                file.write(f"{entry['timestamp']} - Query: {entry['query']},
Result Count: {len(entry['result'])}\n")

    def save_history(self, filename="logs/history.log"):
        with open(filename, "a") as file:
            for entry in self.history:
                self._write_log(entry, filename)
        self.history.clear()
```

Текст функції експорту діаграми в PNG ~~з'єднання з API~~

```
class ExportController:
    def export_plot(self, plot_instance, filename, format="png"):
        plot_instance.save(filename, format)

    def export_as_html(self, plot_instance, filename):
        plot_instance.save_as_html(filename)

    def prompt_and_export(self, plot_instance, filename_base, format="png"):
```

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

```
        save_choice = input("Do you want to save this chart? (y/n): ")
        if save_choice.lower() == 'y':
            filename = f"{filename_base}.{format}"
            self.export_plot(plot_instance, filename, format)
            print(f"Chart saved as {filename}.")
```

```
# api/api_client.py
import requests
from core.error_handler import APIError, ErrorHandler

class APIClient:
    def __init__(self, base_url):
        self.base_url = base_url

    def get_data(self, endpoint):
        try:
            response = requests.get(f"{self.base_url}/{endpoint}")
            response.raise_for_status()
            return response.json()
        except requests.RequestException as e:
            ErrorHandler.handle_error(APIError(f"API request failed: {e}"))
            return None
```

Висновки: В ході виконання лабораторної роботи було створено багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib. ~~проєкт, який надав цінний досвід роботи з API, дизайну користувацького інтерфейсу, обробки помилок та тестування~~

Formatted: Font: (Default) Times New Roman, 14 pt, Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian