МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж



Лабораторна робота №8

з дисципліни Спеціалізовані мови програмування

на тему

Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python

Виконав:

студент групи РІ-21сп

Дмитрій Сас

Львів – 2024

**Мета виконання лабораторної роботи:** Розробка додатка для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП (наслідування, інкапсуляція, поліморфізм)

**План роботи**

**Завдання 1: Вибір CSV-набору даних**

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

**Завдання 2: Завантаження даних з CSV**

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

**Завдання 3: Дослідження даних**

Визначте екстремальні значення по стовцям

**Завдання 4: Вибір типів візуалізацій**

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: Підготовка даних**

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

**Завдання 6: Базова візуалізація**

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: Розширені візуалізації**

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

**Завдання 8: Декілька піддіаграм**

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

**Завдання 9: Експорт і обмін**

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

**Текст програмної реалізації:**

**data\_visualizer.py:**import pandas as pd

import matplotlib.pyplot as plt

import os

class CSVVisualizer:

    def \_\_init\_\_(self, filename):

        self.filename = filename

        self.data = None

        # Перевірка наявності файлу, створення, якщо не існує

        if not os.path.exists(self.filename):

            self.create\_sample\_csv()

        self.load\_data()

    def create\_sample\_csv(self):

        # Приклад базового CSV-файлу

        sample\_data = {

            "Date": ["2024-01-01", "2024-02-01", "2024-03-01", "2024-04-01"],

            "Value1": [10, 15, 20, 25],

            "Value2": [20, 25, 30, 35]

        }

        df = pd.DataFrame(sample\_data)

        df.to\_csv(self.filename, index=False)

        print(f"Створено новий CSV-файл із базовими даними: {self.filename}")

    def load\_data(self):

        try:

            self.data = pd.read\_csv(self.filename)

            print("Дані завантажено успішно.")

        except Exception as e:

            print(f"Помилка завантаження даних: {e}")

    def display\_data\_summary(self):

        if self.data is not None:

            print("Огляд даних:\n", self.data.describe())

        else:

            print("Дані ще не завантажені.")

    def plot\_data(self):

        if self.data is not None:

            plt.plot(self.data["Date"], self.data["Value1"], label="Value1")

            plt.plot(self.data["Date"], self.data["Value2"], label="Value2")

            plt.xlabel("Date")

            plt.ylabel("Values")

            plt.legend()

            plt.title("Basic Plot of Value1 and Value2 Over Time")

            plt.show()

        else:

            print("Дані ще не завантажені.")

    def plot\_multiple\_subplots(self):

        if self.data is not None and len(self.data.columns) > 2:

            fig, axs = plt.subplots(1, 2, figsize=(10, 5))

            axs[0].plot(self.data["Date"], self.data["Value1"], color='b')

            axs[0].set\_title("Value1 Over Time")

            axs[1].plot(self.data["Date"], self.data["Value2"], color='r')

            axs[1].set\_title("Value2 Over Time")

            plt.show()

        else:

            print("Необхідно більше стовпців для багатопанельної діаграми.")

# Використання класу

visualizer = CSVVisualizer("ваш\_файл.csv")

visualizer.display\_data\_summary()

visualizer.plot\_data()

visualizer.plot\_multiple\_subplots()

**Результати тестування:**

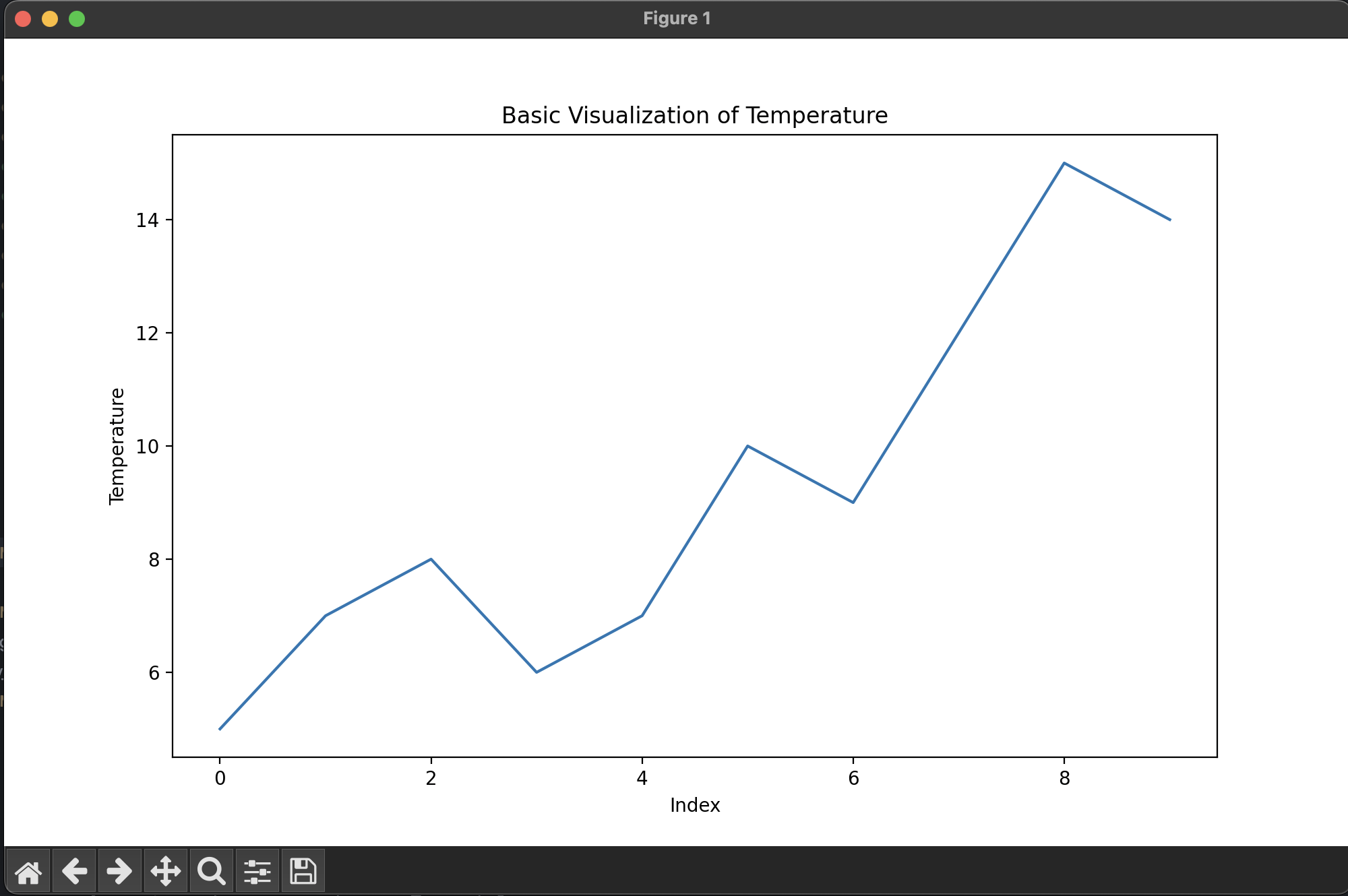


Рис. 1. Результат виводу графіка температури

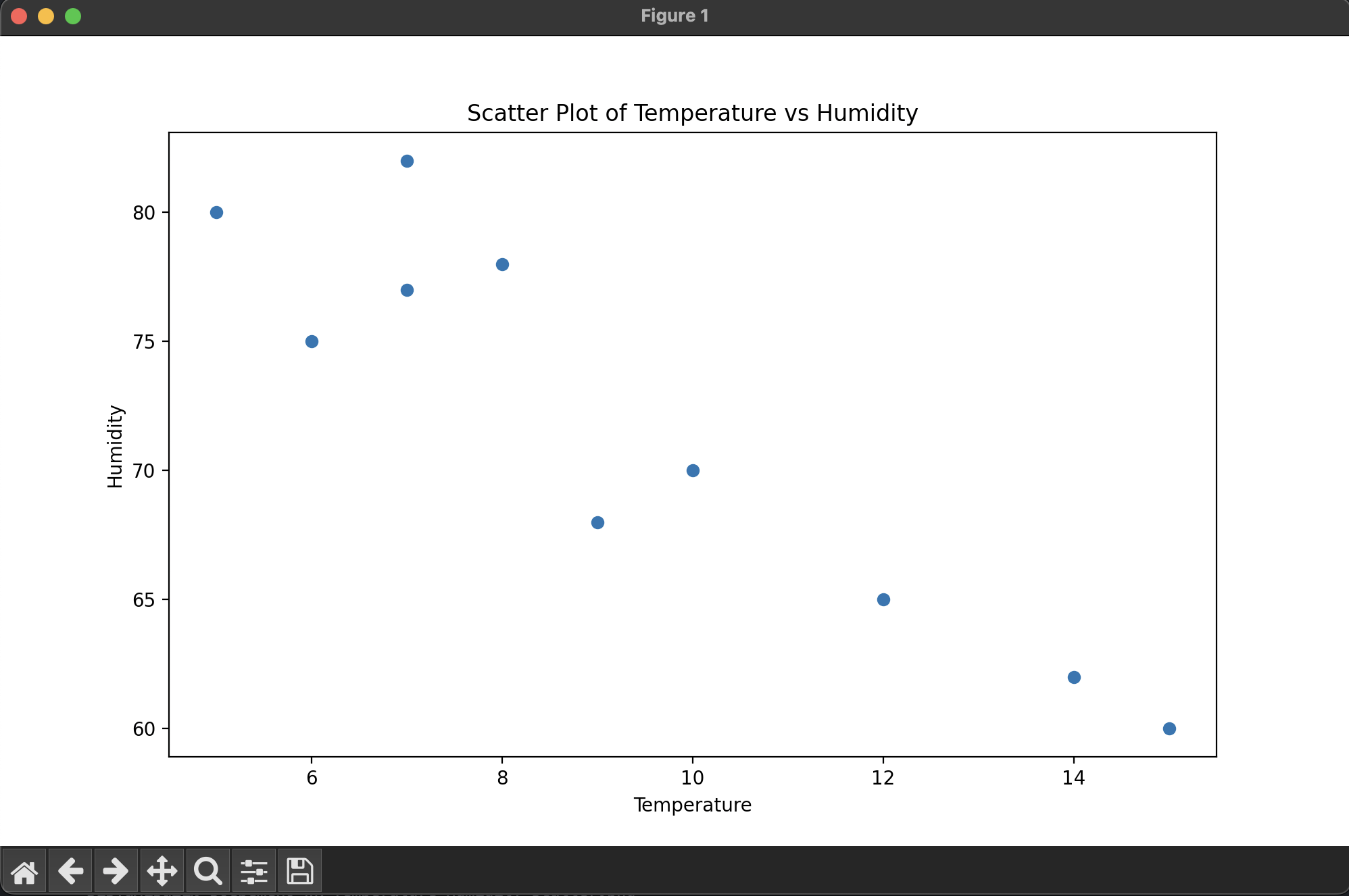


Рис. 2. Результат виводу графіка температури та вологості

**Висновки:** на цій лабораторній роботі було розроблено додаток для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП (наслідування, інкапсуляція, поліморфізм).