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

Владислав ДМИТРЕНКО

Львів – 2024

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

**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.

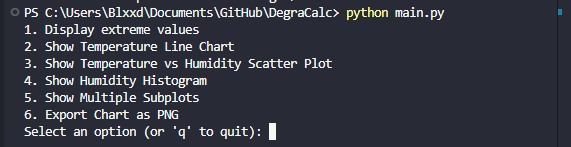
**Завдання 6:** Створіть базову візуалізацію набору даних, щоб переконатися, що ви можете відображати дані правильно за допомогою Matplotlib. Розпочніть з простої діаграми для візуалізації однієї змінної.

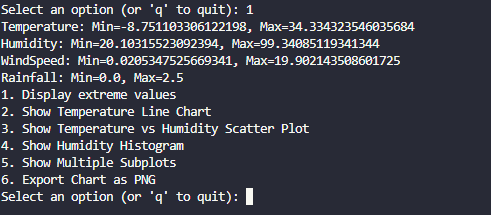
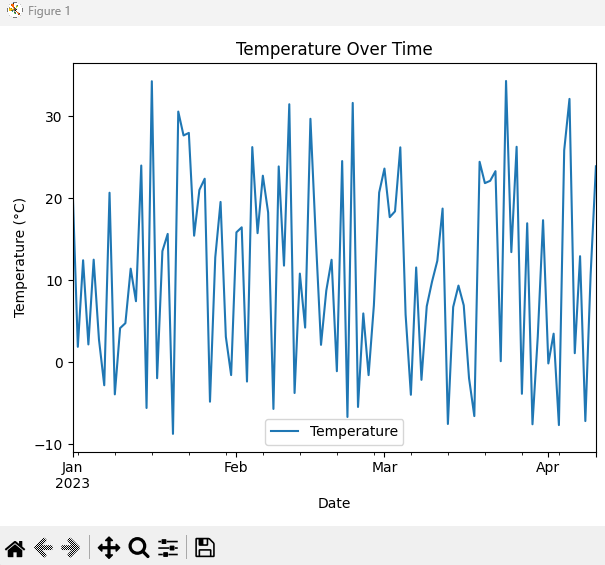
**Завдання 7:** Реалізуйте більш складні візуалізації, виходячи з характеристик набору. Поекспериментуйте з різними функціями Matplotlib та налаштуваннями.

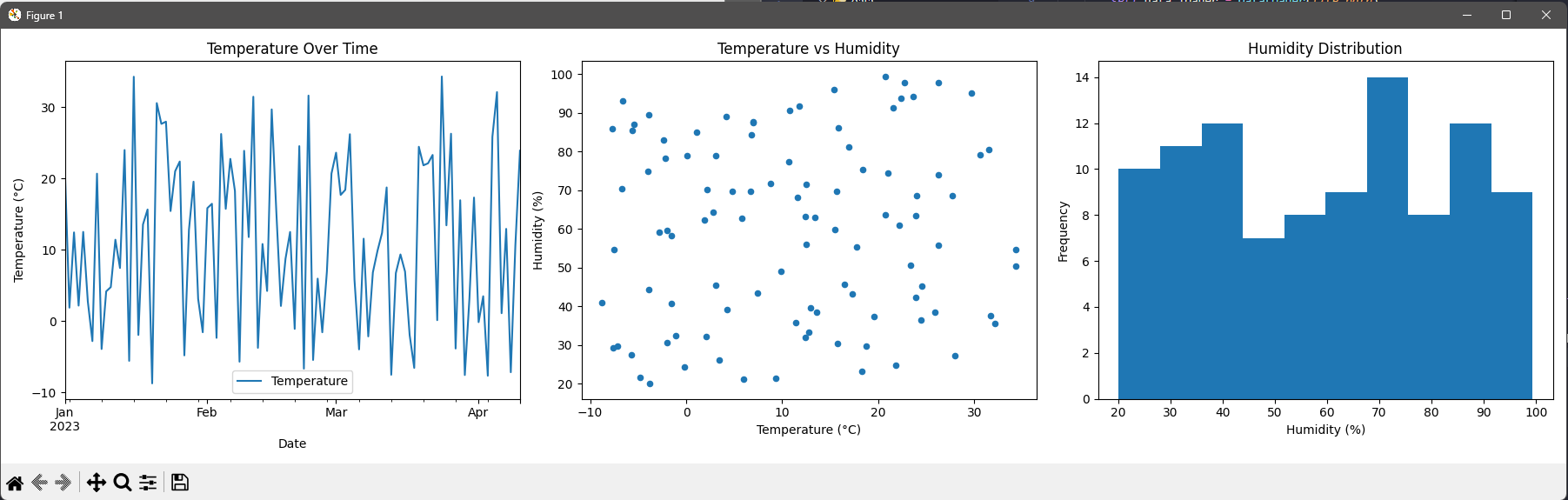
**Завдання 8:** Навчіться створювати кілька піддіаграм в межах одного малюнка для відображення декількох візуалізацій поруч для кращого порівняння.

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

**Результати тестування:**

****  
Рис. 1. Користувацький інтерфейс

  
Рис. 2. Вивід даних  
  
Рис 3. Вигляд лінійної діаграми

  
Рис. 4. Вигляд всіх діаграм разом

Текст контроллера даних

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()

Текст функції експорту діаграми в PNG

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"):

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}.")

**Висновки:** В ході виконання лабораторної роботи було створено багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib.