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:

ст. гр. РІ-21сп

Шептицький Р.М

Прийняв:

Шербак С.С.

Львів-2024

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

**Хід 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)

**Виконання роботи**

**Папка Classes:**

**Вміст файлу bar\_chart\_visualizer.py:**

from matplotlib import pyplot as plt  
from Classes.data\_visualizer import DataVisualizer  
  
class BarChartVisualizer(DataVisualizer):  
 def process\_data(self):  
 self.\_data = self.\_data.groupby('Humidity')['Temperature'].sum()  
 print("Дані оброблено для стовпчикової діаграми.")  
  
 def plot(self):  
 self.\_data.plot(kind='bar', color='skyblue')  
 plt.title("Стовпчикова діаграма")  
 plt.xlabel("Вологість повітря")  
 plt.ylabel("Температура")  
 print("Стовпчикову діаграму побудовано.")

**Вміст файлу combined\_chart\_visualizer.py:**

from Classes.data\_visualizer import DataVisualizer  
import matplotlib.pyplot as plt  
  
class CombinedChartVisualizer(DataVisualizer):  
 def process\_data(self):  
 self.bar\_data = self.\_data.groupby('Humidity')['Temperature'].sum()  
 self.line\_data = self.\_data.sort\_values('Date')  
 self.scatter\_data = self.\_data[['X', 'Y']]  
  
 def plot(self):  
 fig, axes = plt.subplots(1, 3, figsize=(15, 5))  
  
 # Стовпчикова діаграма  
 self.bar\_data.plot(kind='bar', color='skyblue', ax=axes[0])  
 axes[0].set\_title("Стовпчикова діаграма")  
 axes[0].set\_xlabel("Вологість повітря")  
 axes[0].set\_ylabel("Температура")  
  
 # Лінійний графік  
 axes[1].plot(self.line\_data['Date'], self.line\_data['Temperature'], marker='o', linestyle='-', color='green')  
 axes[1].set\_title("Лінійний графік")  
 axes[1].set\_xlabel("Дата")  
 axes[1].set\_ylabel("Температура")  
 axes[1].tick\_params(axis='x', rotation=45)  
  
 # Діаграма розсіювання  
 axes[2].scatter(self.scatter\_data['X'], self.scatter\_data['Y'], color='red', alpha=0.5)  
 axes[2].set\_title("Діаграма розсіювання")  
 axes[2].set\_xlabel("X")  
 axes[2].set\_ylabel("Y")  
  
 plt.tight\_layout()

**Вміст файлу data\_visualizer.py:**

from abc import ABC, abstractmethod  
import pandas as pd  
import matplotlib.pyplot as plt  
import os  
  
  
class DataVisualizer(ABC):  
 def \_\_init\_\_(self, file\_path):  
 self.file\_path = file\_path  
 self.\_data = None  
  
 def visualize(self):  
 try:  
 self.load\_data()  
 self.process\_data()  
 self.plot()  
 self.export()  
 except KeyError as e:  
 print(f"Помилка: Відсутній необхідний стовпець у файлі - {e}")  
 except FileNotFoundError:  
 print("Помилка: Файл не знайдено.")  
 except pd.errors.EmptyDataError:  
 print("Помилка: Файл порожній або має некоректний формат.")  
  
 def load\_data(self):  
 self.\_data = pd.read\_csv(self.file\_path)  
 print("Дані завантажено успішно.")  
  
 @abstractmethod  
 def process\_data(self):  
 pass  
  
 @abstractmethod  
 def plot(self):  
 pass  
  
 def export(self):  
 save\_option = input("Бажаєте зберегти діаграму? (так/ні): ").strip().lower()  
 if save\_option == 'так':  
 file\_format = input("Оберіть формат файлу для збереження (png/svg): ").strip().lower()  
 if file\_format not in ['png', 'svg']:  
 print("Невірний формат. Діаграму буде збережено у форматі PNG за замовчуванням.")  
 file\_format = 'png'  
  
 file\_name = input("Введіть назву файлу (без розширення): ").strip()  
 if not file\_name:  
 file\_name = "chart"  
  
 folder\_path = 'data'  
 if not os.path.exists(folder\_path):  
 os.makedirs(folder\_path)  
  
 file\_name\_with\_extension = os.path.join(folder\_path, f"{file\_name}.{file\_format}")  
 try:  
 plt.tight\_layout()  
 plt.savefig(file\_name\_with\_extension)  
 print(f"Діаграму збережено як '{file\_name\_with\_extension}'.")  
 except Exception as e:  
 print(f"Помилка під час збереження файлу: {e}")  
 else:  
 print("Діаграму не збережено.")  
 plt.show()

**Вміст файлу line\_chart\_visualizer.py:**

from Classes.data\_visualizer import DataVisualizer  
import matplotlib.pyplot as plt  
  
class LineChartVisualizer(DataVisualizer):  
 def process\_data(self):  
 self.\_data.sort\_values('Date', inplace=True)  
 print("Дані оброблено для лінійного графіка.")  
  
 def plot(self):  
 plt.plot(self.\_data['Date'], self.\_data['Temperature'], marker='o', linestyle='-', color='green')  
 plt.title("Лінійний графік")  
 plt.xlabel("Дата")  
 plt.ylabel("Температура")  
 plt.xticks(rotation=45)  
 print("Лінійний графік побудовано.")

**Вміст файлу min\_max\_visualizer.py:**

import pandas as pd  
  
from Classes.data\_visualizer import DataVisualizer  
  
class MinMaxVisualizer(DataVisualizer):  
 def visualize(self):  
 try:  
 self.load\_data()  
 self.process\_data()  
 except FileNotFoundError:  
 print("Помилка: Файл не знайдено.")  
 except pd.errors.EmptyDataError:  
 print("Помилка: Файл порожній або має некоректний формат.")  
  
 def process\_data(self):  
 print("\nМінімальні та максимальні значення для кожної колонки:")  
  
 for column in self.\_data.columns:  
 column\_min = self.\_data[column].min()  
 column\_max = self.\_data[column].max()  
 print(f"{column}: Minimum = {column\_min}, Maximum = {column\_max}")  
  
 def plot(self):  
 pass

**Вміст файлу scatter\_plot\_visualizer.py:**

from Classes.data\_visualizer import DataVisualizer  
import matplotlib.pyplot as plt  
  
class ScatterPlotVisualizer(DataVisualizer):  
 def process\_data(self):  
 print("Дані готові для діаграми розсіювання.")  
  
 def plot(self):  
 plt.scatter(self.\_data['X'], self.\_data['Y'], color='red', alpha=0.5)  
 plt.title("Діаграма розсіювання")  
 plt.xlabel("X")  
 plt.ylabel("Y")  
 print("Діаграму розсіювання побудовано.")

**Папка Config:**

**Вміст файлу data.csv:**

Humidity,Temperature,Date,X,Y  
52%,+12,2024-11-01,1,2  
47%,+3,2024-11-02,2,3  
50%,+22,2024-11-03,3,5  
35%,+11,2024-11-04,4,7  
57%,+13,2024-11-05,6,10  
55%,+20,2024-11-06,7,13  
40%,+15,2024-11-07,9,15  
42%,+12,2024-11-08,10,15  
45%,+4,2024-11-09,12,13  
64%,+6,2024-11-10,13,15  
23%,+15,2024-11-11,14,17  
43%,+12,2024-11-12,16,20  
41%,+18,2024-11-13,17,21  
49%,+26,2024-11-14,19,23

**Папка UI:**

**Вміст файлу user\_interface.py:**

import os  
import sys  
  
lab8\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), ".."))  
sys.path.append(lab8\_root)  
  
from Classes.bar\_chart\_visualizer import BarChartVisualizer  
from Classes.line\_chart\_visualizer import LineChartVisualizer  
from Classes.scatter\_plot\_visualizer import ScatterPlotVisualizer  
from Classes.min\_max\_visualizer import MinMaxVisualizer  
from Classes.combined\_chart\_visualizer import CombinedChartVisualizer  
  
def main():  
 while True:  
 print("\nОберіть опцію:")  
 print("1. Стовпчикова діаграма")  
 print("2. Лінійний графік")  
 print("3. Діаграма розсіювання")  
 print("4. Показати мінімальні та максимальні значення")  
 print("5. Відобразити всі діаграми одночасно")  
 print("6. Вийти")  
  
 choice = input("Введіть номер опції: ")  
  
 if choice == '6':  
 print("Програма завершена.")  
 break  
  
 file\_path = input("Введіть шлях до CSV-файлу: ")  
  
 if choice == '1':  
 visualizer = BarChartVisualizer(file\_path)  
 elif choice == '2':  
 visualizer = LineChartVisualizer(file\_path)  
 elif choice == '3':  
 visualizer = ScatterPlotVisualizer(file\_path)  
 elif choice == '4':  
 visualizer = MinMaxVisualizer(file\_path)  
 elif choice == '5':  
 visualizer = CombinedChartVisualizer(file\_path)  
 else:  
 print("Невірний вибір. Спробуйте ще раз.")  
 continue  
  
 visualizer.visualize()

**В кориневій папці лабораторної файл runner.py:**

from UI.user\_interface import main  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Результат виконання: а саме виведення усіх діаграма на рисунку 1 та виведення мінімальних та максимальних значень рисунок 2.

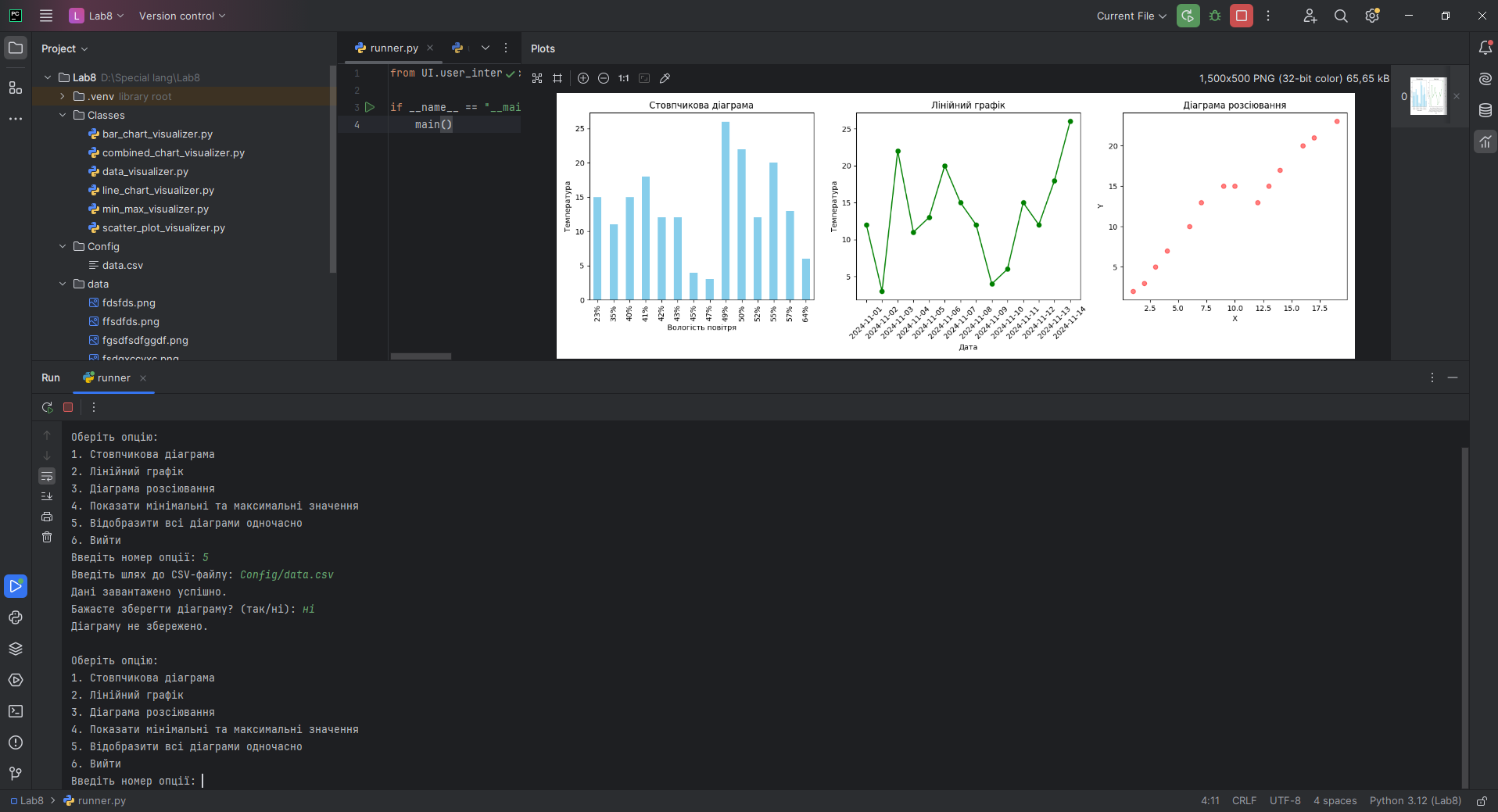


Рисунок 1 – Вивід усіх діаграм

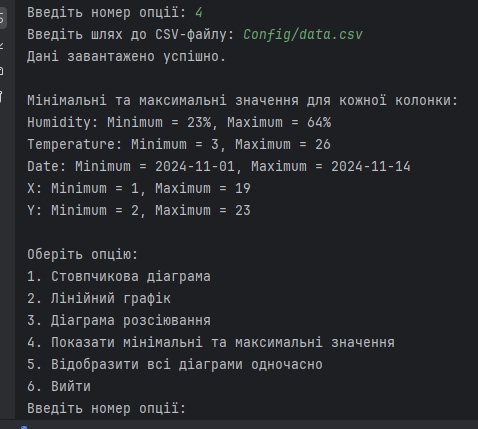


Рисунок 2 – Вивід мінімальних та максимальних значень

**Висновок:** У ході виконання лабораторної роботи я створив багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib. Дана програма генерує стовпчикову діаграму, лінійний графік та діаграму розсіювання. Також дає можливість відобразити усі діаграми разом та показати мінімальні та максимальні значення.