

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

Рак В.П.

Прийняв:

Щербак 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**

Визначити екстремальні значення по стовпцям

### **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 обробити набір даних за необхідністю для візуалізації. Це може включати виправлення даних, фільтрацію, агрегацію або трансформацію.

### **Завдання 6: Базова візуалізація**

Створіть базу 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. Поекспериментувати з різними функціями Matplotlib та налаштуваннями.

### **Завдання 8: Декілька піддіаграм**

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

### **Завдання 9: Експорт і обмін**

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

## Виконання роботи

### Текст програмної реалізації:

#### **base\_visualization.py:**

```
import os

import matplotlib.pyplot as plt

from abc import ABC, abstractmethod
```

```
class BaseVisualization(ABC):
```

```
    """
```

```
    Абстрактний клас для візуалізації даних.
```

```
    Атрибути:
```

```
    - `data`: Дані, які будуть використовуватись для побудови графіку.
```

```
    Методи:
```

```
    - `plot(self, *args, **kwargs)`: Абстрактний метод для побудови графіку.
```

```
    - `export_chart(self, filename, format='png')`: Метод для експорту графіку у файл.
```

```
    """
```

```
    def __init__(self, data):
```

```
        """
```

```
        Ініціалізує об'єкт візуалізації з даними.
```

```
        Параметри:
```

```
        - `data`: Дані для побудови графіку.
```

```
        """
```

```
        self.data = data
```

```
    @abstractmethod
```

```
    def plot(self, *args, **kwargs):
```

```
        """
```

```
        Абстрактний метод для побудови графіку. Повинен бути реалізований у підкласах.
```

Параметри:

- ``*args`, `**kwargs``: Додаткові параметри для побудови графіку.

"""

pass

```
def export_chart(self, filename, format='png'):
```

"""

Експортує побудований графік у файл.

Зберігає графік у файл в директорії 'lab8' з заданим ім'ям та форматом.

Параметри:

- ``filename``: Ім'я файлу для збереження графіку.

- ``format``: Формат збереження файлу (за замовчуванням 'png').

Повертає:

- Нічого не повертає, але виводить шлях до збереженого файлу.

"""

```
# Створюємо шлях до директорії lab8 і зберігаємо графік
```

```
save_dir = 'd:/lpnu-specialized-programming-languages-2024/lab8'
```

```
save_path = os.path.join(save_dir, f'{filename}.{format}')
```

```
plt.savefig(save_path)
```

```
print(f'Графік експортовано як {save_path}')
```

## **visualizations.py:**

```
import matplotlib.pyplot as plt
```

```
from .base_visualization import BaseVisualization
```

```
class LineChart(BaseVisualization):
```

```
    def plot(self, x_column, y_column, save_as=None):
```

```
plt.plot(self.data[x_column], self.data[y_column])
plt.xlabel(x_column)
plt.ylabel(y_column)
plt.title('Лінійний графік')
if save_as:
    self.export_chart(save_as)
plt.show()
```

```
class BarChart(BaseVisualization):
    def plot(self, x_column, y_column, save_as=None):
        plt.bar(self.data[x_column], self.data[y_column])
        plt.xlabel(x_column)
        plt.ylabel(y_column)
        plt.title('Стовпчиковий графік')
        if save_as:
            self.export_chart(save_as)
        plt.show()
```

```
class ScatterPlot(BaseVisualization):
    def plot(self, x_column, y_column, save_as=None):
        plt.scatter(self.data[x_column], self.data[y_column])
        plt.xlabel(x_column)
        plt.ylabel(y_column)
        plt.title('Діаграма розсіювання')
        if save_as:
            self.export_chart(save_as)
        plt.show()
```

```
class PieChart(BaseVisualization):
    def plot(self, column, save_as=None):
        plt.pie(self.data[column].value_counts(), labels=self.data[column].unique(),
        autopct='%1.1f%%')
        plt.title('Секторний графік')
```

```
if save_as:
    self.export_chart(save_as)
plt.show()
```

### **data\_handler.py:**

```
import pandas as pd
```

```
class DataHandler:
```

```
    def __init__(self, filepath):
        self.data = pd.read_csv(filepath)

    def get_column_extremes(self):
        return self.data.describe().loc[['min', 'max']]

    def filter_data(self, condition):
        return self.data.query(condition)
```

### **display.py:**

```
def display_data_overview(data_handler):
    """Відображає огляд даних."""
    print("Огляд даних:\n", data_handler.get_column_extremes())
```

### **main.py:**

```
from lab8.dal.data_handler import DataHandler
from lab8.bll.visualizer import Visualizer
from lab8.ui.display import display_data_overview
```

```
"""
```

Основна функція для завантаження даних, їх візуалізації та порівняння графіків.

Ця функція виконує кілька кроків:

1. Завантажує дані з файлу CSV за допомогою класу `DataHandler`.
2. Виводить первинний огляд даних (мінімальні та максимальні значення для кожного стовпця).
3. Ініціалізує візуалізатор (`Visualizer`) з даними та створює різні візуалізації:
  - Лінійний графік для порівняння продажів.
  - Стопчиковий графік для категорій.
  - Діаграму розсіювання для порівняння зросту та ваги.
  - Секторний графік для порівняння регіонів.
4. Порівнює два графіки розсіювання (Зріст vs Вага) та два лінійних графіки (Продажі vs Кількість).

Параметри:

- Немає параметрів, функція працює з файлами та класами, визначеними у відповідних модулях.

Повертає:

- Нічого не повертає. Всі операції виконуються за допомогою виведення на екран та збереження файлів з графіками.

"""

def main():

    filepath = 'lab8/data.csv'

    # Завантажуємо дані і показуємо первинний огляд

    handler = DataHandler(filepath)

    display\_data\_overview(handler)

    # Ініціалізуємо візуалізатор з даними

    visualizer = Visualizer(handler.data)

    # Відображаємо та зберігаємо різні візуалізації

    visualizer.display\_line\_chart('Date', 'Sales', save\_as='line\_chart\_sales')

    visualizer.display\_bar\_chart('Category', 'Count', save\_as='bar\_chart\_category')

    visualizer.display\_scatter\_plot('Height', 'Weight', save\_as='scatter\_plot\_height\_weight')

```
visualizer.display_pie_chart('Region', save_as='pie_chart_region')
```

```
# Порівнюємо два графіки розсіювання: Height vs Weight
```

```
visualizer.compare_charts('scatter', 'Height', 'Weight', 'Height')
```

```
# Порівнюємо два лінійних графіки: Sales vs Count
```

```
visualizer.compare_charts('line', 'Date', 'Sales', 'Count')
```

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

```
    main()
```

## Результат роботи програми:

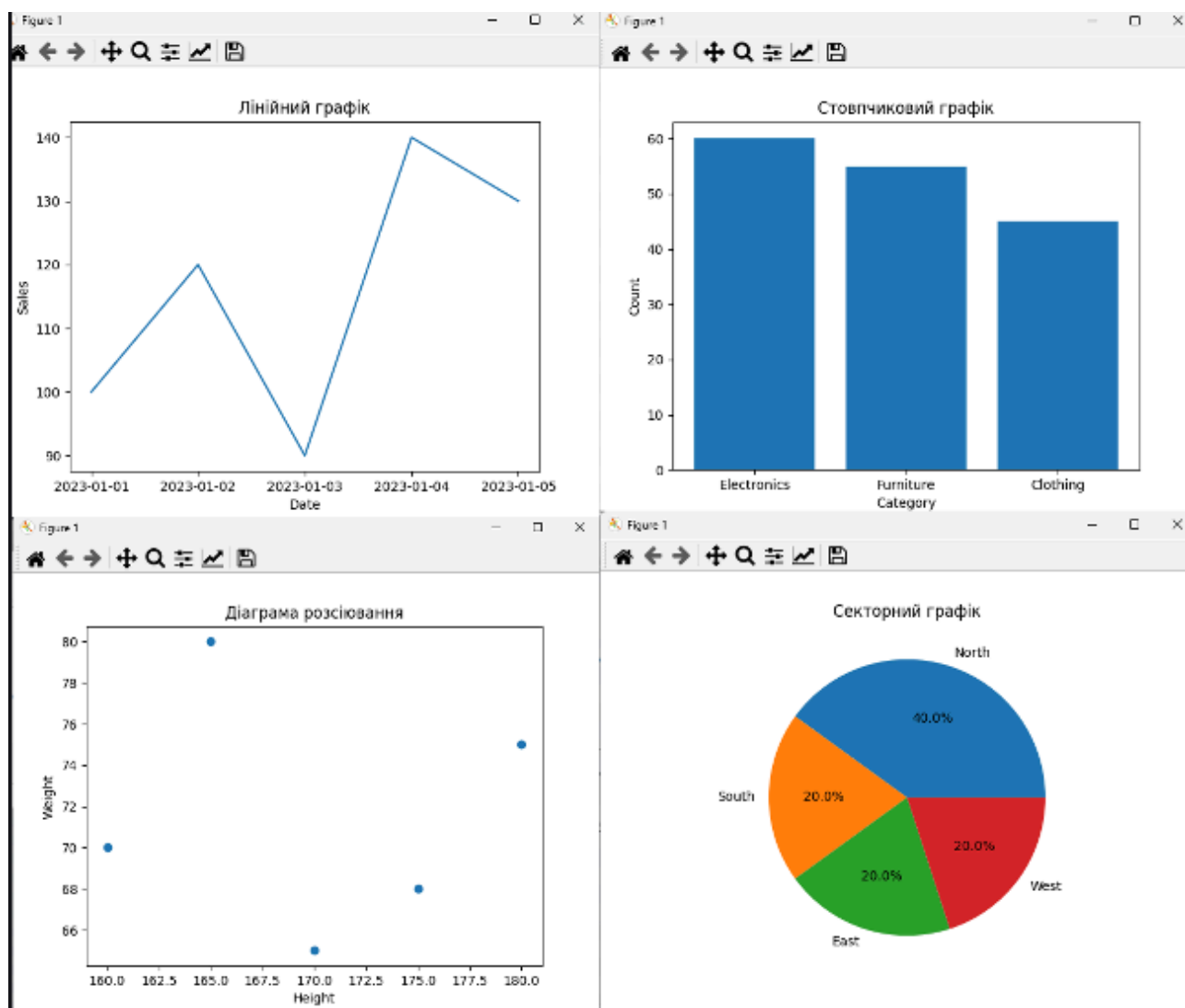


Рис. 1 – Вивід діаграм



```
Огляд даних:
      Sales  Count  Height  Weight  Population
min    90.0   30.0   160.0   65.0      800.0
max   140.0   60.0   180.0   80.0     1200.0
```

Рис. 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.