МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра ІСМ



ЗВІТ

Про виконання лабораторної роботи №8

«Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python»

З дисципліни

«Спеціалізовані мови програмування»

Студентки групи РІ – 31

Нащецької Яни Вікторівни

Прийняв викладач

Щербак С.С.

**Лабораторна робота № 8. Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python**

**Мета**: 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)

**Основний код програми:**

from ...interfaces.VisualizationInterface import VisualizationInterface

import matplotlib.pyplot as plt

class BarChartStrategy(VisualizationInterface):

    def visualize(self, data\_x, data\_y):

        if not data\_x or not data\_y or len(data\_x) != len(data\_y):

            raise ValueError("Помилка: Дані для графіка некоректні.")

        plt.bar(data\_x, data\_y)

        plt.xlabel("Категорії")

        plt.ylabel("Значення")

        plt.title("Стовпчастий графік")

        plt.show()

import matplotlib.pyplot as plt

from ...interfaces.VisualizationInterface import VisualizationInterface

class LineChartStrategy(VisualizationInterface):

    def visualize(self, data\_x, data\_y):

        if not data\_x or not data\_y or len(data\_x) != len(data\_y):

            raise ValueError("Помилка: Дані для графіка некоректні.")

        plt.plot(data\_x, data\_y, label="Лінійний графік")

        plt.xlabel("X (вісь)")

        plt.ylabel("Y (вісь)")

        plt.title("Лінійний графік даних")

        plt.legend()

        plt.show()

from abc import ABC, abstractmethod

class VisualizationStrategy(ABC):

    @abstractmethod

    def visualize(self):

        pass

import os

import pandas as pd

from .factories.VisualizationFactory import VisualizationFactory

file\_path = r"d:\NU\_LP\3\_year\1\_semestr\SPL\LW\lab\_8\data\people.csv"

print(f"Trying to load file from: {file\_path}")

class CSVVisualization:

    def \_\_init\_\_(self):

        self.visualization\_factory = VisualizationFactory()

    def load\_data(self, file\_path):

        try:

            if not os.path.isfile(file\_path):

                print(f"Файл {file\_path} не знайдено.")

                return None

            data = pd.read\_csv(file\_path)

            print("Дані успішно завантажено.")

            return data

        except Exception as e:

            print(f"Помилка при завантаженні CSV: {e}")

            return None

    def visualize\_data(self, data, visualization\_type, x="Index", y="Salary"):

        try:

            strategy = self.visualization\_factory.create\_strategy(visualization\_type)

            if x == "Index":

                data\_x = pd.Series(data.index)

            else:

                if x not in data.columns:

                    print(f"Колонка {x} не знайдена в даних.")

                    return

                data\_x = pd.to\_numeric(data[x], errors="coerce")

            if y not in data.columns:

                print(f"Колонка {y} не знайдена в даних.")

                return

            data\_y = pd.to\_numeric(data[y], errors="coerce")

            if data\_x.isnull().any() or data\_y.isnull().any():

                print("Одне з значень у колонках має недопустимі або пропущені значення.")

                return

            strategy.visualize(data\_x, data\_y)

        except ValueError as e:

            print(f"ValueError: {e}")

        except Exception as e:

            print(f"Помилка: {e}")

visualization = CSVVisualization()

data = visualization.load\_data(file\_path)

if data is not None:

    visualization.visualize\_data(data, "line", x="Index", y="Salary")

**Висновок:** Виконавши ці завдання, я створила багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib. Цей проект покращив мої навички візуалізації даних, дозволивши досліджувати результати з різноманітними наборами даних