Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра інформаційних систем та мереж

******

**ЗВІТ**

**ДО ЛАБОРАТОРНОЇ РОБОТИ №8  
з дисципліни  
«Спеціалізовані мови програмування»  
на тему:**  
«Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python»

***Виконала:*** *студентка групи РІ-31  
 Діана ВІЙЧУК*

***Прийняв:****к. т. н., доцент* *Сергій ЩЕРБАК*

## МЕТА РОБОТИ

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

## ЗАВДАННЯ

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)

## ПРОГРАМНА РЕАЛІЗАЦІЯ

Файл */runner.py* .

import sys

import os

lab8\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..", ".."))

sys.path.append(lab8\_root)

from src.main import main

if \_\_name\_\_ == "\_\_main\_\_":

main()

Файл */src/bll/classes/csv/data\_repository.py* .

import pandas as pd

class DataRepository:

def \_\_init\_\_(self, file\_path):

self.file\_path = file\_path

self.data = self.load\_data()

def load\_data(self):

try:

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

data['date'] = pd.to\_datetime(data['date'])

return data

except FileNotFoundError:

print("Файл не знайдено!")

return pd.DataFrame()

def get\_data(self):

return self.data

Файл */src/bll/classes/csv/data\_strategy.py* .

class DataProcessingStrategy:

def process(self, data):

raise NotImplementedError("Метод повинен бути реалізований у підкласах")

class ExtremeValuesStrategy(DataProcessingStrategy):

def process(self, data):

extreme\_values = data.agg(['min', 'max'])

return extreme\_values

Файл */src/bll/classes/csv/visualization.py* .

import matplotlib.pyplot as plt

class Visualization():

def \_\_init\_\_(self, data):

self.data = data

self.figure = None

def plot(self, x\_column, y\_column):

raise NotImplementedError("Метод повинен бути реалізований у підкласах")

def save(self, filename):

if self.figure:

self.figure.savefig(filename, bbox\_inches='tight', pad\_inches=0.1)

print(f"Графік збережено як {filename}")

else:

print("Спочатку потрібно створити графік перед збереженням.")

class LineChart(Visualization):

def plot(self, x\_column, y\_column):

self.figure, ax = plt.subplots()

ax.plot(self.data[x\_column], self.data[y\_column])

ax.set\_xlabel(x\_column)

ax.set\_ylabel(y\_column)

ax.set\_title("Лінійний графік")

ax.tick\_params(axis='x', rotation=45)

class BarChart(Visualization):

def plot(self, x\_column, y\_column):

self.figure, ax = plt.subplots()

ax.bar(self.data[x\_column], self.data[y\_column])

ax.set\_xlabel(x\_column)

ax.set\_ylabel(y\_column)

ax.set\_title("Стовпчикова діаграма")

ax.tick\_params(axis='x', rotation=45)

class ScatterPlot(Visualization):

def plot(self, x\_column, y\_column):

self.figure, ax = plt.subplots()

ax.scatter(self.data[x\_column], self.data[y\_column])

ax.set\_xlabel(x\_column)

ax.set\_ylabel(y\_column)

ax.set\_title("Діаграма розсіювання")

class CombinedCharts(Visualization):

def plot(self, x\_column, y\_column1, y\_column2):

self.figure, axs = plt.subplots(1, 2, figsize=(15, 5))

axs[0].plot(self.data[x\_column], self.data[y\_column1])

axs[0].set\_title(f"{y\_column1} vs {x\_column}")

axs[0].set\_xlabel(x\_column)

axs[0].set\_ylabel(y\_column1)

axs[1].bar(self.data[x\_column], self.data[y\_column2])

axs[1].set\_title(f"{y\_column2} vs {x\_column}")

axs[1].set\_xlabel(x\_column)

axs[1].set\_ylabel(y\_column2)

plt.tight\_layout()

class VisualizationFactory:

def create\_visualization(visualization\_type, data):

if visualization\_type == 'line':

return LineChart(data)

elif visualization\_type == 'bar':

return BarChart(data)

elif visualization\_type == 'scatter':

return ScatterPlot(data)

elif visualization\_type == 'combined':

return CombinedCharts(data)

else:

raise ValueError(f"Невідомий тип візуалізації: {visualization\_type}")

def export\_visualization(visualization, filename):

visualization.save(filename)

Файл */src/main.py*

import sys

import os

lab8\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..", ".."))

sys.path.append(lab8\_root)

from src.bll.classes.csv.data\_repository import DataRepository

from src.ui.menu import menu

def main():

repo = DataRepository('sources/seattle-weather.csv')

data = repo.get\_data()

menu(data)

if \_\_name\_\_ == "\_\_main\_\_":

main()

## ВИСНОВКИ

## На цій лабораторній роботі я розробила додаток для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП (наслідування, інкапсуляція, поліморфізм).

Посилання на GitHub: <https://github.com/Dinasi4ka/Calculator.git>