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

Виконала:

ст. гр. ІТ-32

Ольга ЧИГИРИК

Прийняв

доцент каф. ІСМ:

Сергій ЩЕРБАК

|  |  |
| --- | --- |
| **Балів** | **Дата** |
|  |  |

Львів-2023

**Мета**: 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)

**Main.py**

import console8  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 console8.main()

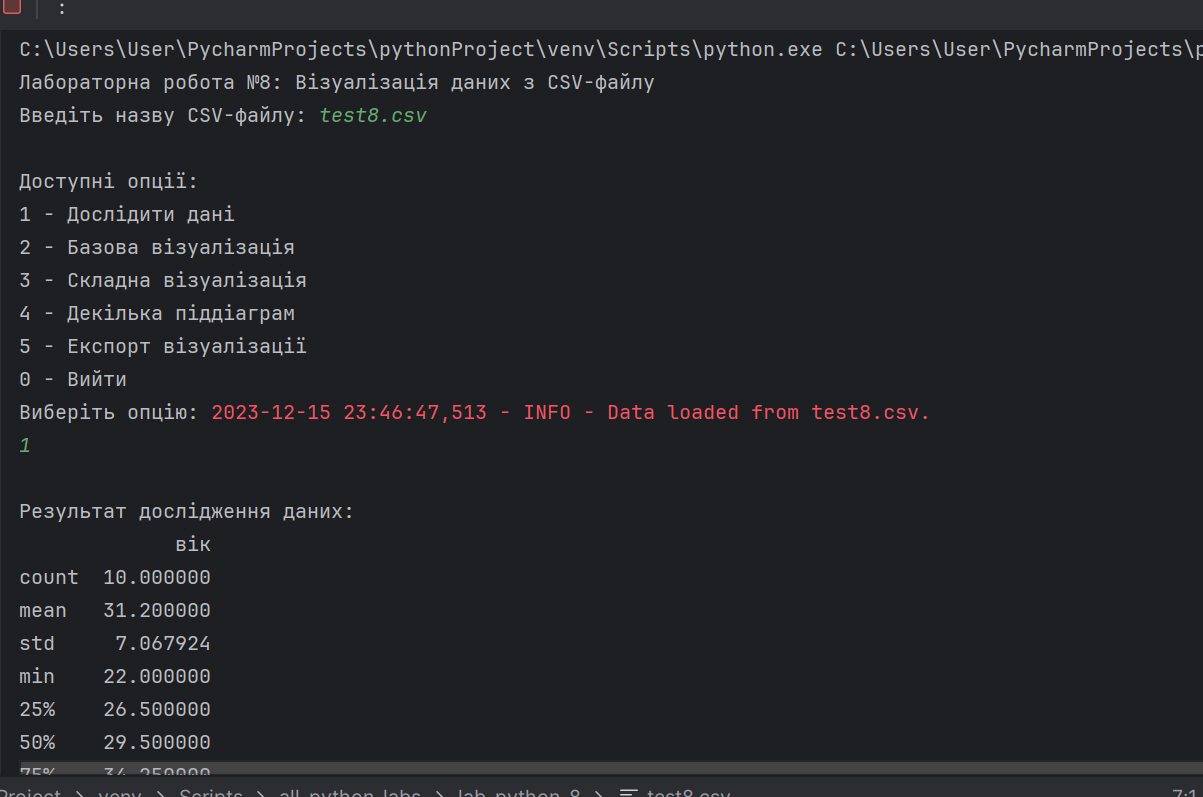
**Console.py**

from data\_visualizer8 import DataVisualizer  
#from Scripts.all\_python\_labs.shared\_lib.data\_manager import DataManager  
  
def main():  
 print("Лабораторна робота №8: Візуалізація даних з CSV-файлу")  
 while True:  
 csv\_file = input("Введіть назву CSV-файлу: ")  
 visualizer = DataManager(csv\_file)  
  
 if visualizer.data is not None:  
 break  
 else:  
 print("Файл не знайдено або виникла помилка. Будь ласка, спробуйте ще раз.")  
  
 while True:  
 print("\nДоступні опції:")  
 print("1 - Дослідити дані")  
 print("2 - Базова візуалізація")  
 print("3 - Складна візуалізація")  
 print("4 - Декілька піддіаграм")  
 print("5 - Експорт візуалізації")  
 print("0 - Вийти")  
  
 choice = input("Виберіть опцію: ")  
  
 if choice == '1':  
 print("\nРезультат дослідження даних:")  
 print(visualizer.explore\_data())  
  
 elif choice == '2':  
 col = input("Введіть назву колонки для базової візуалізації: ")  
 visualizer.basic\_visualization(col)  
  
 elif choice == '3':  
 col\_x = input("Введіть назву першої колонки: ")  
 col\_y = input("Введіть назву другої колонки: ")  
 visualizer.advanced\_visualization(col\_x, col\_y)  
  
 elif choice == '4':  
 cols = input("Введіть назви колонок через кому для множинних піддіаграм: ").split(',')  
 visualizer.multiple\_subplots(cols)  
  
 elif choice == '5':  
 filename = input("Введіть назву файлу для експорту: ")  
 visualizer.export\_visualization(filename)  
  
 elif choice == '0':  
 break  
  
 else:  
 print("Невідома опція. Спробуйте знову.")

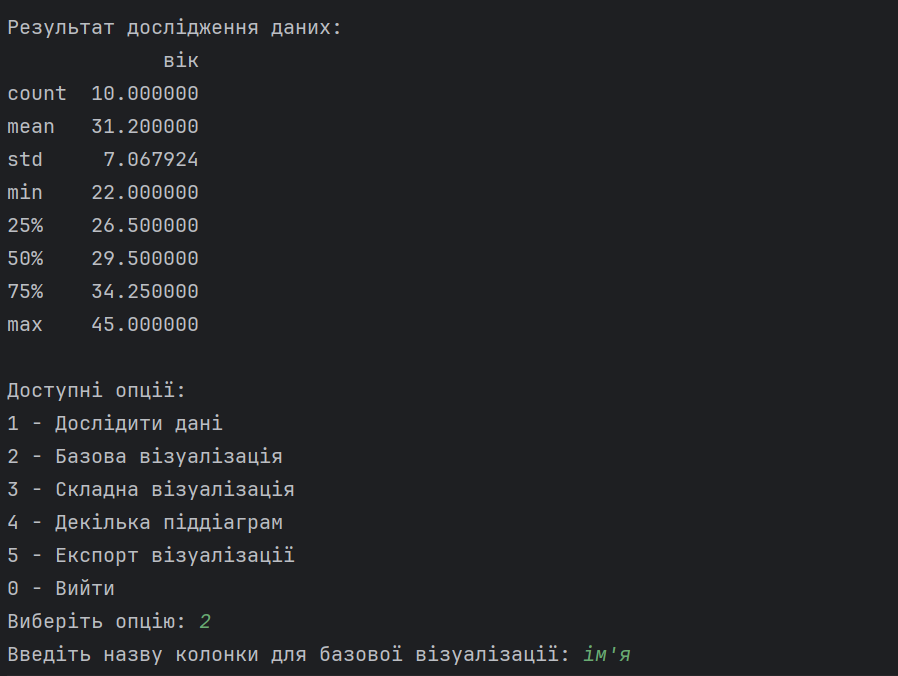
from data\_visualizer8 import DataVisualizer  
#from Scripts.all\_python\_labs.shared\_lib.data\_manager import DataManager  
  
def main():  
 print("Лабораторна робота №8: Візуалізація даних з CSV-файлу")  
 while True:  
 csv\_file = input("Введіть назву CSV-файлу: ")  
 visualizer = DataManager(csv\_file)  
  
 if visualizer.data is not None:  
 break  
 else:  
 print("Файл не знайдено або виникла помилка. Будь ласка, спробуйте ще раз.")  
  
 while True:  
 print("\nДоступні опції:")  
 print("1 - Дослідити дані")  
 print("2 - Базова візуалізація")  
 print("3 - Складна візуалізація")  
 print("4 - Декілька піддіаграм")  
 print("5 - Експорт візуалізації")  
 print("0 - Вийти")  
  
 choice = input("Виберіть опцію: ")  
  
 if choice == '1':  
 print("\nРезультат дослідження даних:")  
 print(visualizer.explore\_data())  
  
 elif choice == '2':  
 col = input("Введіть назву колонки для базової візуалізації: ")  
 visualizer.basic\_visualization(col)  
  
 elif choice == '3':  
 col\_x = input("Введіть назву першої колонки: ")  
 col\_y = input("Введіть назву другої колонки: ")  
 visualizer.advanced\_visualization(col\_x, col\_y)  
  
 elif choice == '4':  
 cols = input("Введіть назви колонок через кому для множинних піддіаграм: ").split(',')  
 visualizer.multiple\_subplots(cols)  
  
 elif choice == '5':  
 filename = input("Введіть назву файлу для експорту: ")  
 visualizer.export\_visualization(filename)  
  
 elif choice == '0':  
 break  
  
 else:  
 print("Невідома опція. Спробуйте знову.")

**Data\_visualizer.py**

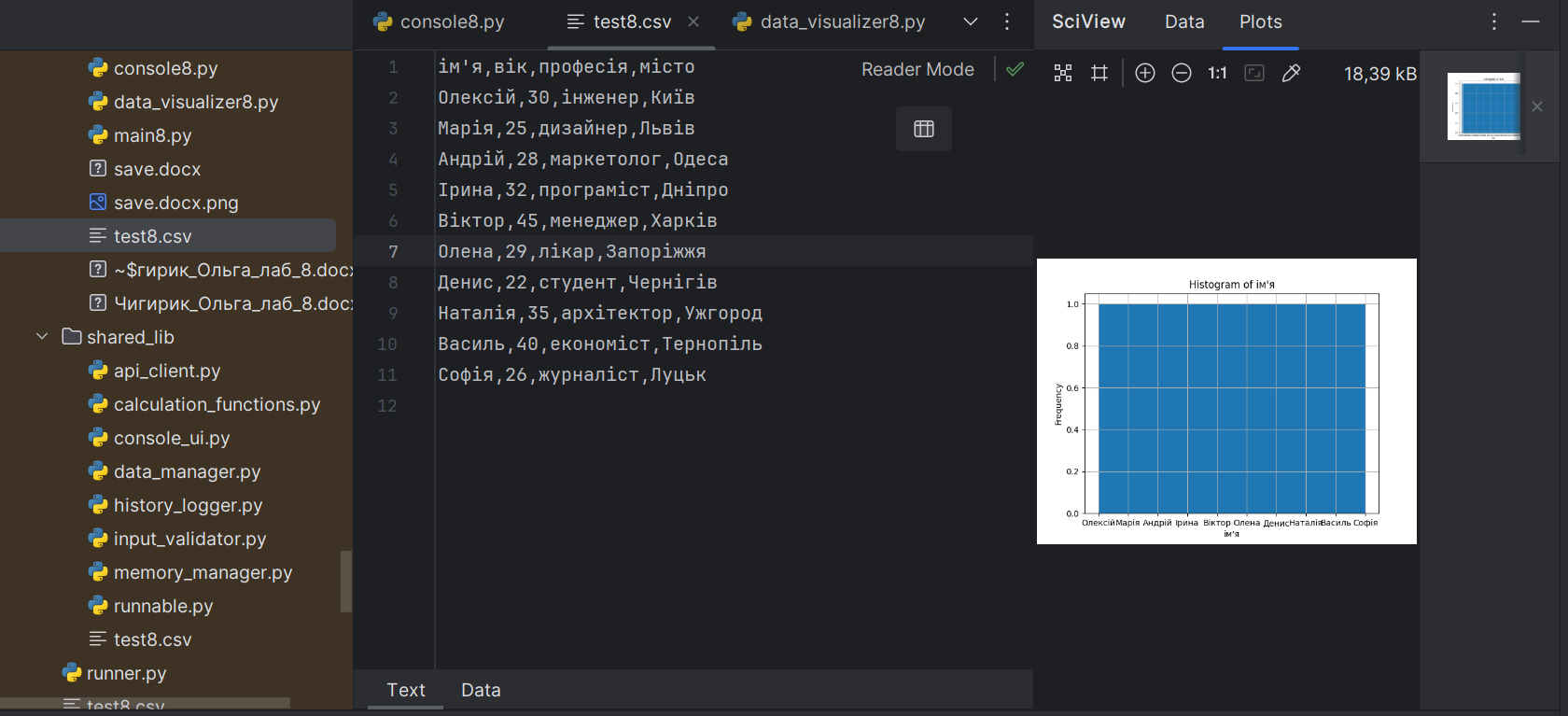
import pandas as pd  
import matplotlib.pyplot as plt  
  
  
class DataVisualizer:  
 def \_\_init\_\_(self, csv\_file):  
 self.data = self.load\_data(csv\_file)  
  
 @staticmethod  
 def load\_data(csv\_file):  
 # Завантаження даних  
 try:  
 return pd.read\_csv(csv\_file)  
 except FileNotFoundError:  
 print(f"Файл не знайдено: {csv\_file}")  
 return None  
  
 def explore\_data(self):  
 # Дослідження даних  
 if self.data is not None:  
 return self.data.describe()  
 else:  
 return "Дані не завантажено."  
  
 def basic\_visualization(self, column\_name):  
 # Базова візуалізація (наприклад, гістограма)  
 if self.data is not None and column\_name in self.data.columns:  
 self.data[column\_name].hist()  
 plt.xlabel(column\_name)  
 plt.ylabel('Частота')  
 plt.title(f'Гістограма для {column\_name}')  
 plt.show()  
 else:  
 print(f"Колонка {column\_name} не знайдена.")  
  
 def advanced\_visualization(self, column\_x, column\_y):  
 # Складна візуалізація (наприклад, діаграма розсіювання)  
 if self.data is not None:  
 plt.scatter(self.data[column\_x], self.data[column\_y])  
 plt.xlabel(column\_x)  
 plt.ylabel(column\_y)  
 plt.title(f'Діаграма розсіювання: {column\_x} vs {column\_y}')  
 plt.show()  
 else:  
 print("Дані не завантажено або колонки не знайдено.")  
  
 def multiple\_subplots(self, columns):  
 # Створення декількох піддіаграм  
 if self.data is not None:  
 fig, axs = plt.subplots(len(columns), figsize=(10, 5))  
 for i, col in enumerate(columns):  
 if col in self.data.columns:  
 axs[i].plot(self.data[col])  
 axs[i].set\_title(col)  
 else:  
 print(f"Колонка {col} не знайдена.")  
 plt.tight\_layout()  
 plt.show()  
 else:  
 print("Дані не завантажено.")  
  
 def export\_visualization(self, filename, file\_format='png'):  
 # Експорт останньої візуалізації  
 plt.savefig(f'{filename}.{file\_format}')



*Рис. 1 Результат виконання завдання*



*Рис. 2 Результат виконання завдання*



*Рис. 3 Результат виконання завдання*

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