

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”

Факультет прикладної математики

Лабораторна робота №2

Тема: «Програмування і алгоритми»

Виконала: студентка групи КМ-31
Буряк Софія

КИЇВ — 2024

Мета

Розробити програмний засіб, який відображує за заданим датасетом точки на координатній площині і зберігає зображення в одному з графічних форматів.

Обрана бібліотека

Візуалізація даних є важливим інструментом для аналізу та представлення інформації. Python пропонує низку бібліотек для візуалізації, серед яких однією з найпопулярніших є Matplotlib. Ця бібліотека дозволяє створювати графіки різного типу, такі як лінійні графіки, гістограми, розсіювання (scatter plot) тощо.

Основні компоненти:

1. Зчитування даних: декодування текстових файлів із координатами у формат, придатний для обробки.
2. Налаштування полотна: визначення розмірів графіка та розміщення точок відповідно до заданих координат.
3. Збереження результатів: генерація файлу з графіком у зручному графічному форматі.

Усі методи, які використовувалися з бібліотеки:

- plt.figure — створення полотна.
- plt.scatter — нанесення точок на графік.
- plt.xlim, plt.ylim — налаштування меж графіка.
- plt.title, plt.xlabel, plt.ylabel — додавання підписів.
- plt.grid — активація сітки.
- plt.savefig — збереження результату у файл.
- plt.show — виведення графіка на екран.

Хід виконання

1. Для читання координат була розроблена функція `read_coordinates`. Функція відкриває файл, читає кожен рядок і обробляє його, перетворюючи рядкові значення у числові.

Ключові етапи:

- Відкриття файлу у режимі читання.
- Використання методу `split()` для розділення координат.
- Перетворення текстових даних у числовий формат за допомогою функції `map(int, ...)`

2. Для візуалізації точок була створена функція `plot_points`.

Ключові етапи:

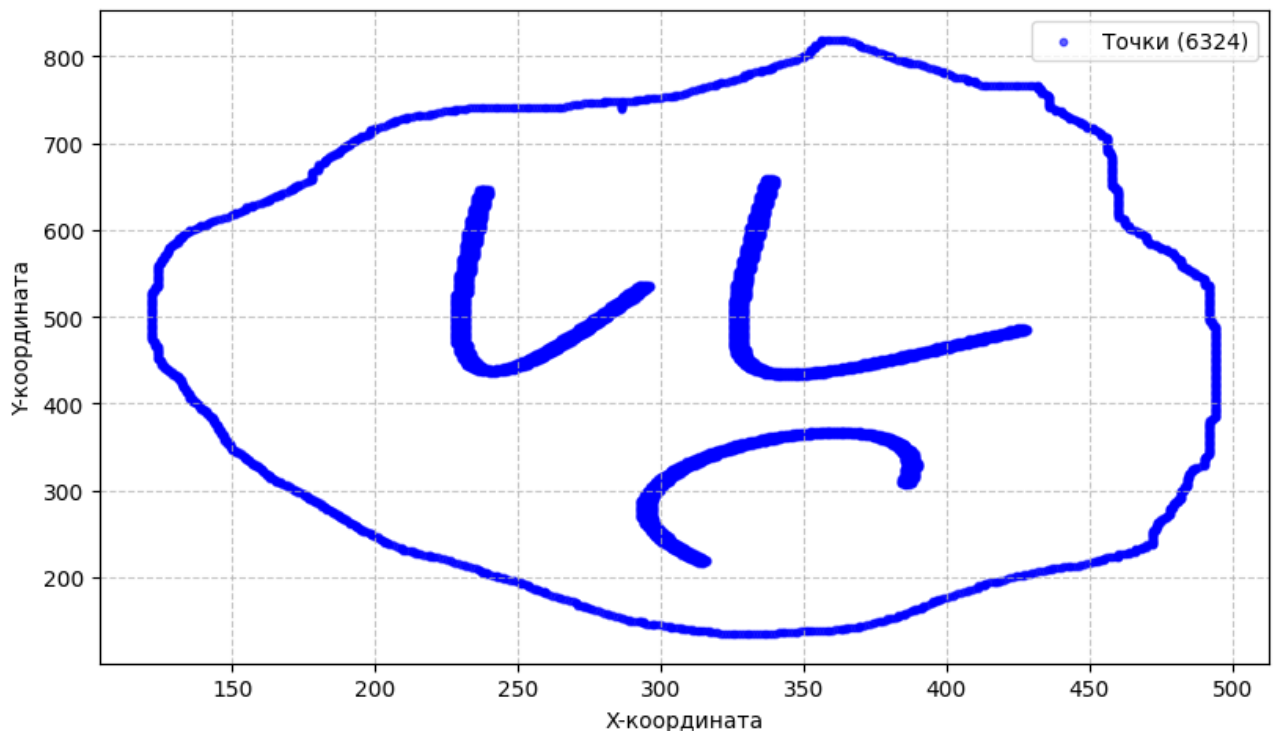
- Налаштовує розмір полотна через метод `plt.figure`.
- Будує точки за допомогою методу `plt.scatter`, що дозволяє задавати їхній розмір, прозорість та колір.
- Відображає сітку, підписи осей і заголовок графіка.

Головною особливістю є автоматичне визначення меж графіка з урахуванням діапазону координат точок. Межі коригуються додатковим відступом (5%), щоб точки не "торкались" країв полотна.

3. Графік зберігається у файл формату PNG. Для цього використовується метод `plt.savefig`, який дозволяє налаштувати параметри якості (DPI), кольору фону тощо. У разі успіху виводиться повідомлення, а у разі помилки — обробляється виняток.

Результат

Візуалізація координат точок



Код

```
import matplotlib.pyplot as plt

# Функція для зчитування координат із файлу
def read_coordinates(file_name):
    coordinates = []
    try:
```

```

        with open(file_name, "r") as file:
            for line in file:
                x, y = map(int, line.split()) # Розділяємо і перетворюємо в числа
                coordinates.append((x, y))
    except FileNotFoundError:
        print(f"Файл {file_name} не знайдено!")
        return []
    return coordinates

# Основна функція для побудови графіка
def plot_points(points, canvas_size=(960, 540), output_file="output_image.png"):
    if not points:
        print("Список точок порожній, нічого виводити.")
        return

    # Створюємо полотно з правильними розмірами
    width, height = canvas_size
    plt.figure(figsize=(width/100, height/100), dpi=100)

    # Розпаковуємо координати
    x_coords, y_coords = zip(*points)

    # Створюємо графік
    plt.scatter(x_coords, y_coords,
                c='blue',
                s=10,
                alpha=0.6,
                label=f"Точки ({len(points)})")

    plt.title("Візуалізація координат точок", pad=15)
    plt.xlabel("X-координата")
    plt.ylabel("Y-координата")

    # Додаємо сітку
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.legend()

    # Додаємо відступи пропорційно до діапазону даних
    x_range = max(x_coords) - min(x_coords)
    y_range = max(y_coords) - min(y_coords)
    padding_x = x_range * 0.05 # 5% відступ
    padding_y = y_range * 0.05

    plt.xlim(min(x_coords) - padding_x, max(x_coords) + padding_x)
    plt.ylim(min(y_coords) - padding_y, max(y_coords) + padding_y)

    # Зберігаємо результат
    try:
        plt.savefig(output_file,
                    bbox_inches='tight',
                    dpi=100,
                    facecolor='white',
                    edgecolor='none')
    
```

```
        print(f"Графік успішно збережено у файл: {output_file}")
    except Exception as e:
        print(f"Помилка при збереженні файлу: {e}")

plt.show()

plt.close()

file_name = "DS1.txt"
points = read_coordinates(file_name)
plot_points(points)
```