

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 4

з дисципліни «Реалізація класу і робота з об'єктами»

Тема: " Структурування програм з використанням  
функцій"

ХАІ.301 . Інженерія мобільних додатків. 312ст.1 ЛР

Виконав студент гр. \_\_\_\_\_312ст\_\_\_\_\_

\_\_\_\_\_Васильєв Б.А.\_\_\_\_\_  
(підпис, дата) (П.І.Б.)

Перевірів

\_\_\_\_\_к.т.н., доц. О. В. Гавриленко  
\_\_\_\_\_ас. В. О. Білозерський  
(підпис, дата) (П.І.Б.)

## МЕТА РОБОТИ

Застосувати теоретичні знання з основ програмування на мові Python з використанням об'єктів і класів, навички використання бібліотеки для візуалізації масивів даних, і навчитися розробляти скрипти для роботи з об'єктами призначених для користувача класів.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Створити список з чотирьох точок, порахувати відстань між четвертою та третьою, пересунути першу на 29 вправо. Варіант 10 представлено на сторінці 3

## ВИКОНАННЯ РОБОТИ

Завдання 1: Вирішення задачі Point\_10

Вхідні дані:

points: список точок на площині, кожна точка є об'єктом класу Point\_10 (тип: list[Point\_10], обмеження: координати точок в межах від -100 до 100, кількість точок: 4).

Вихідні дані:

distance: відстань між третьою та четвертою точкою (тип: float).

modified\_points: список точок після зміщення першої точки (тип: list[Point\_10]).

file\_output: вміст текстового файлу "points\_data.txt", що містить координати всіх точок після зміщення (тип: str).

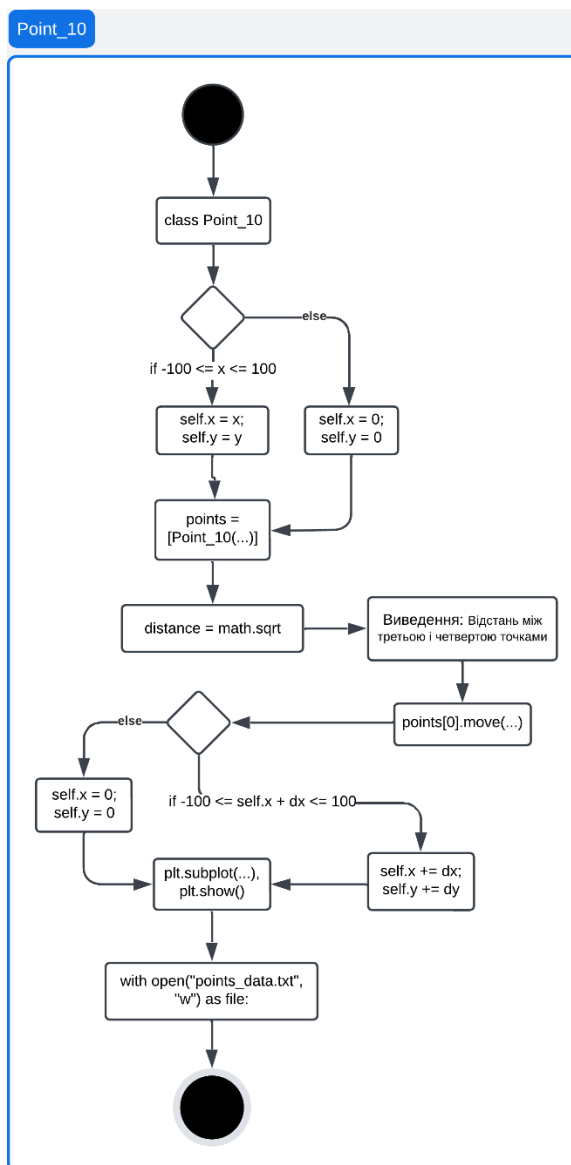


Рисунок 1 діаграма Завдання 1: Вирішення задачі Point\_10

Лістинг коду вирішення задачі наведено в дод. А (стор. 4). Екран роботи програми показаний на рис. Б.2.

## ВИСНОВКИ

У цьому завданні я реалізував функціонал для роботи з точками на площині: обчислення відстані між точками, зміщення точки вправо та візуалізація результатів на графіку. Це дозволило потренуватися в обробці об'єктів, перевірці умов, а також у збереженні та виведенні результатів у файл.

## ДОДАТОК А

### Лістинг коду програми до завдання Proc24

```
import math
import matplotlib.pyplot as plt

class Point_10:
    """Клас для представлення точки на площині."""
    instance_count = 0

    def __init__(self, x=0, y=0):
        self.x = x if -100 <= x <= 100 else 0
        self.y = y if -100 <= y <= 100 else 0
        Point_10.instance_count += 1

    def __del__(self):
        Point_10.instance_count -= 1

    def move(self, dx, dy):
        self.x = self.x + dx if -100 <= self.x + dx <= 100 else 0
        self.y = self.y + dy if -100 <= self.y + dy <= 100 else 0

# точки
points = [Point_10(10, 20), Point_10(30, 40), Point_10(50, 60), Point_10(70,
80)]

# Відстань між третьою і четвертою точками
distance = math.sqrt((points[2].x - points[3].x) ** 2 + (points[2].y -
points[3].y) ** 2)
print(f"Відстань між третьою і четвертою точками: {distance}")

# Зміщення першої точки
points[0].move(29, 0)

# Графік точок
def plot_points(points, title, color):
    x, y = [p.x for p in points], [p.y for p in points]
    plt.scatter(x, y, color=color)
    for i, (xi, yi) in enumerate(zip(x, y), 1):
        plt.text(xi, yi, f"{i}")
    plt.title(title)
    plt.grid()
```

```
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plot_points([Point_10(10, 20), Point_10(30, 40), Point_10(50, 60),
Point_10(70, 80)], "До змін", "blue")
plt.subplot(1, 2, 2)
plot_points(points, "Після змін", "red")
plt.show()

# Збереження координат у файл
with open("points_data.txt", "w") as file:
    file.writelines([f"({i}) {p.x}:{p.y}\n" for i, p in enumerate(points,
1)])

print("Координати точок збережено у файл 'points_data.txt'.")
```

## ДОДАТОК Б

## Скрін-шоти вікна виконання програми

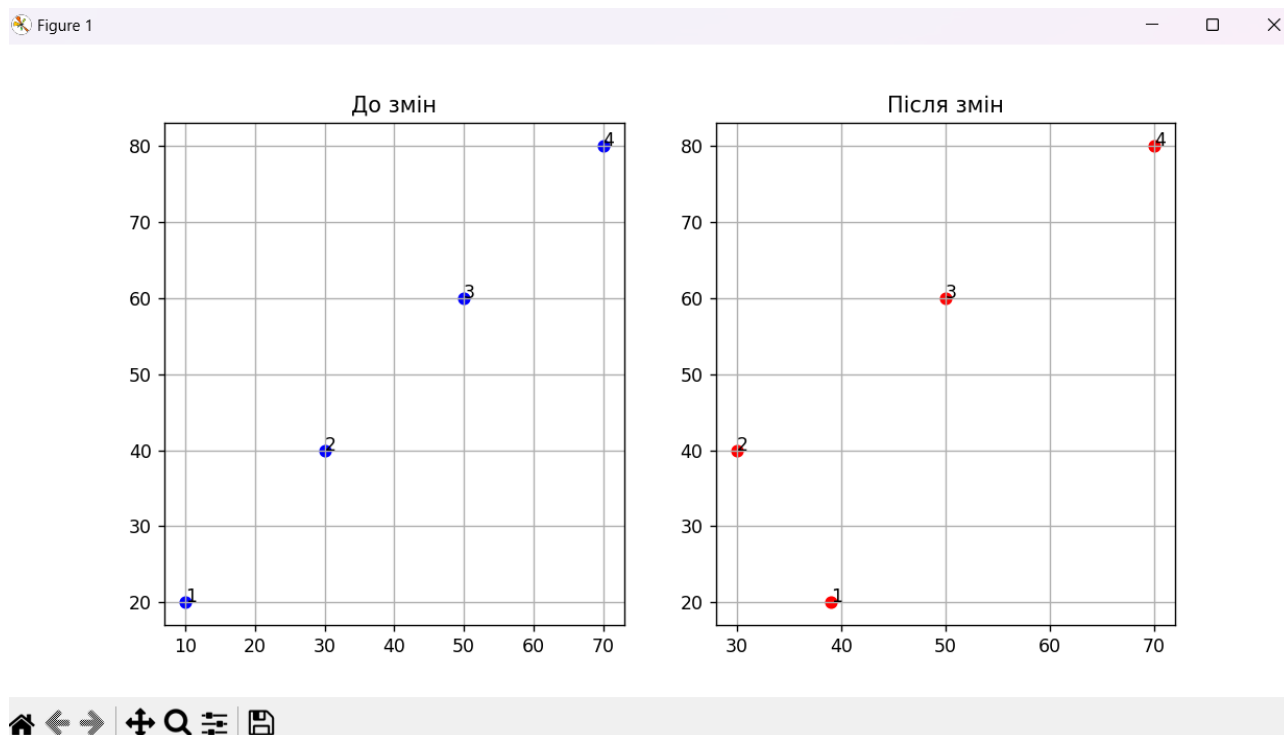


Рисунок Б.2 – Екран виконання програми для вирішення завдання Point\_10

```
oop 5.1.py  points_data.txt x
1 (1) 39:20
2 (2) 30:40
3 (3) 50:60
4 (4) 70:80
```

Рисунок Б.3 – Збереження даних в текстовому файлі завдання Point\_10