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 «Об'єктно-орієнтоване програмування СУ»

Тема:«Реалізація класу і робота з об'єктами»

ХАІ.301.272.321.4.ЛР

Виконав студент гр.
_____321__Новіков Олексій Сергійович

Перевірив
к.т.н., доц. О. В. Гавриленко
ас. В. О. Білозерський
(підпис, дата)
(П.І.Б.)

2023

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. Визначити клас Point_*n* (*n – номер варіанту*), який реалізує

абстракцію з атрибутами:

1)

дві дійсні координати точки на площині (властивості, приховані

змінні екземпляра),

–

для кожної метод-геттер (повертає відповідну координату),

–

для кожної метод-сеттер (записує відповідну координату, якщо

вона у межах [-100, 100], інакше – дорівнює 0))

2)

кількість створених екземплярів точки (змінна класу),

3)

метод класу (повертає кількість створених примірників),

4)

конструктор з двома параметрами (за замовчуванням),

5)

деструктор, що виводить відповідне повідомлення,

6)

метод, що змінює координати точки з двома вхідними дійсними

параметрами:

–

зсув по x,

–

зсув по y.

Завдання 2. Виконати операції з об'єктами даного класу відповідно до

варіанту (*див. таб.1*).

Завдання 3. Використовуючи пакет matplotlib, відобразити створені об'єкти

в графічному вікні до і після змін.

Завдання 4. Зберегти координати точок у текстовому файлі у форматі:

номер: координата_x; координата_y – для непарних варіантів

(номер) координата_x:координата_y – для парних варіантів

	третьою, пересунути першу на 33 вправо.
12.	Створити список з чотирьох точок, порахувати відстань між другою і

	четвертою, пересунути третю на 36 вправо.

Таб. 1

Point
- __x: float = 0.0 - __y: float = 0.0 - __point_count: int = 0
+ __init__(x: float, y: float) + __del__() + get_x(): float + set_x(value: float): void + set_y(value: float): void + get_y(): float + shift(x_shift: float, y_shift: float): void + get_count(): int

Лістинг коду

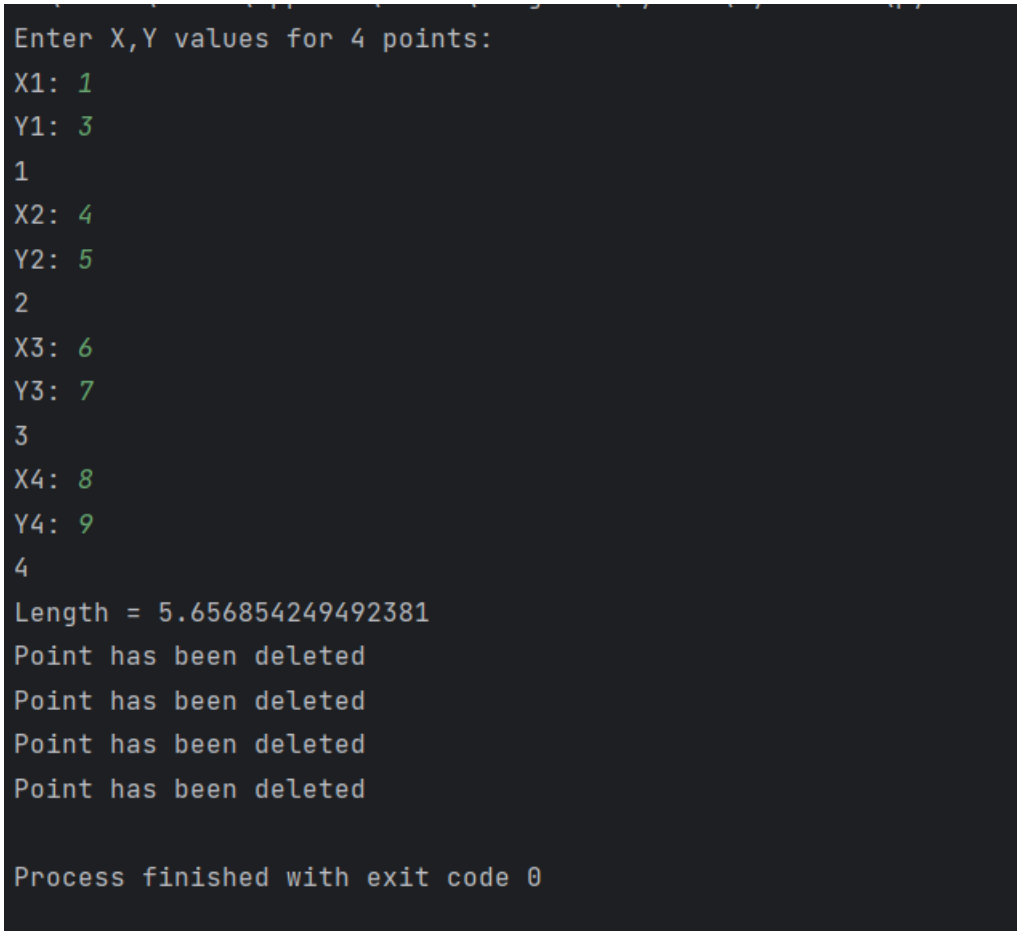
Клас

```
class point1:
    """
    A class to represent a point in 2D space"""

    __x: float = 0.0
    __y: float = 0.0
    __point_count: int = 0

    def __init__(self, x: float, y: float):
        self.set_x(x)
        self.set_y(y)
        point1.__point_count += 1

    def __del__(self):
        print("Point has been deleted")
```

```
        point1.__point_count -= 1

    def get_x(self):
        return self.__x

    def set_x(self, value):
        if value <= 100 and value >= -100:
            self.__x = value
        else:
            self.__x = 0.0

    def set_y(self, value):
        if value <= 100 and value >= -100:
            self.__y = value
        else:
            self.__y = 0.0

    def get_y(self):
        return self.__y

    def shift(self, x_shift: float, y_shift: float):
        self.set_x(self.get_x() + x_shift)
        self.set_y(self.get_y() + y_shift)

    @staticmethod
    def get_count():
        return point1.__point_count

        Main:
import matplotlib.pyplot as plt
from point1 import point1
import math
```

```
# Основна функція
def main():
    points = []
    print("Enter X,Y values for 4 points:")
    for i in range(4):
        try:
            tmp_x = float(input("X{}: ".format(i+1)))
            tmp_y = float(input("Y{}: ".format(i + 1)))
        except ValueError:
            print("Wrong values for points!")
            exit()
        else:
            tmp_point = point1(tmp_x, tmp_y)
            print(point1.get_count())
            points.append(tmp_point)
    show_points(points)
    task1(points)
    show_points(points)
    save_points(points)


def task1(list_of3_points):

    point_1 = list_of3_points[0]
    point_2 = list_of3_points[1]
    point_3 = list_of3_points[2]
    point_4 = list_of3_points[3]

    length = math.sqrt(math.pow(point_4.get_x() - point_2.get_x(), 2) +
math.pow(point_4.get_y() - point_2.get_y(), 2))
    point_3.shift(36.0, 0.0)
    print("Length = {}".format(length))

def show_points(list_of_points):
```

```
    # work with plot
    x = [point.get_x() for point in list_of_points]
    y = [point.get_y() for point in list_of_points]
    plt.plot(x, y, 'ro')
    plt.grid()
    plt.show()
def save_points(list_of_points):
    with open("output.txt", "w") as f:
        for num, point in enumerate(list_of_points):  # 0: point1, 1: point2,
2: point3, 3: point4
            #f.write(f"{num+1}: {point.get_x()}; {point.get_y()}\n")
            f.write(f"({num+1}) {point.get_x()}:{point.get_y()}\n")


if __name__ == '__main__':
    main()
```



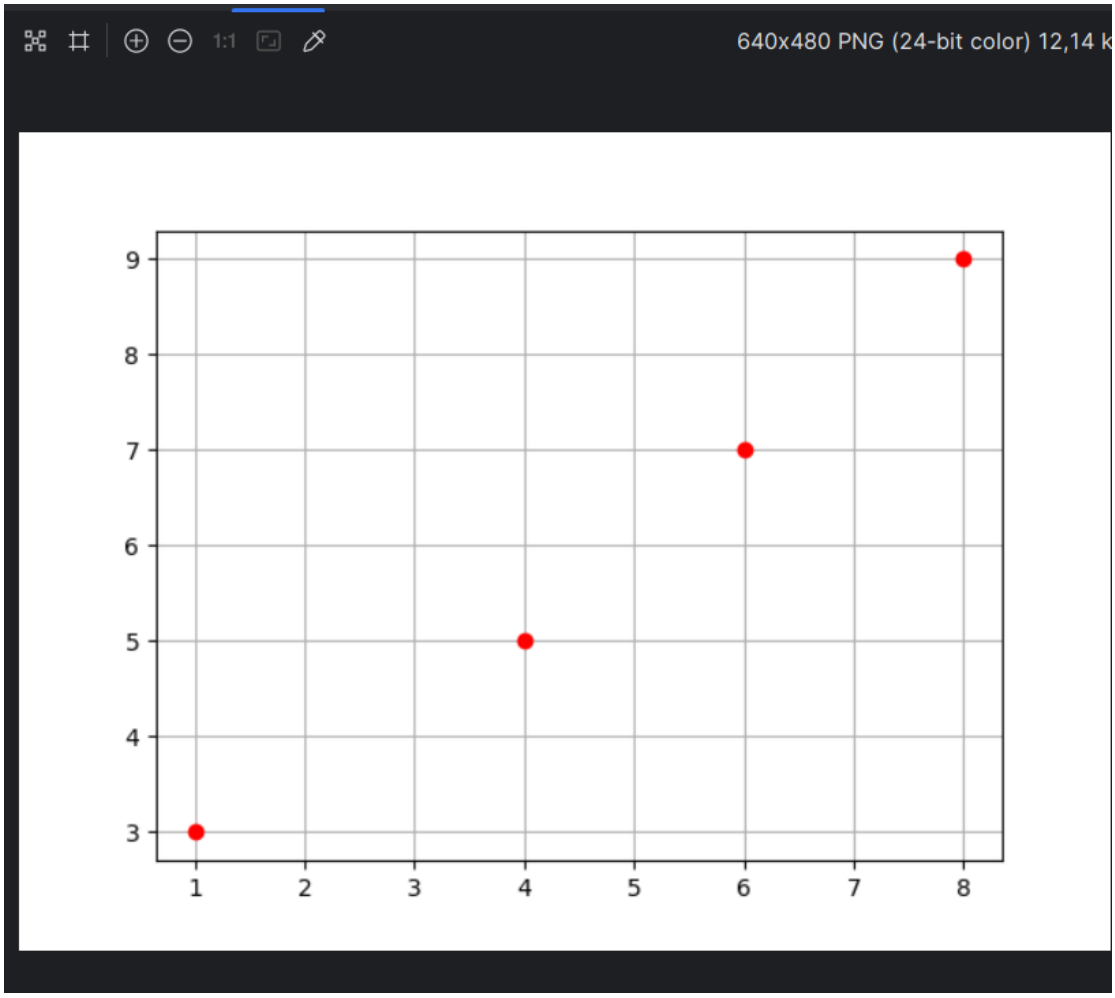


Рисунок 2 «Діаграма»

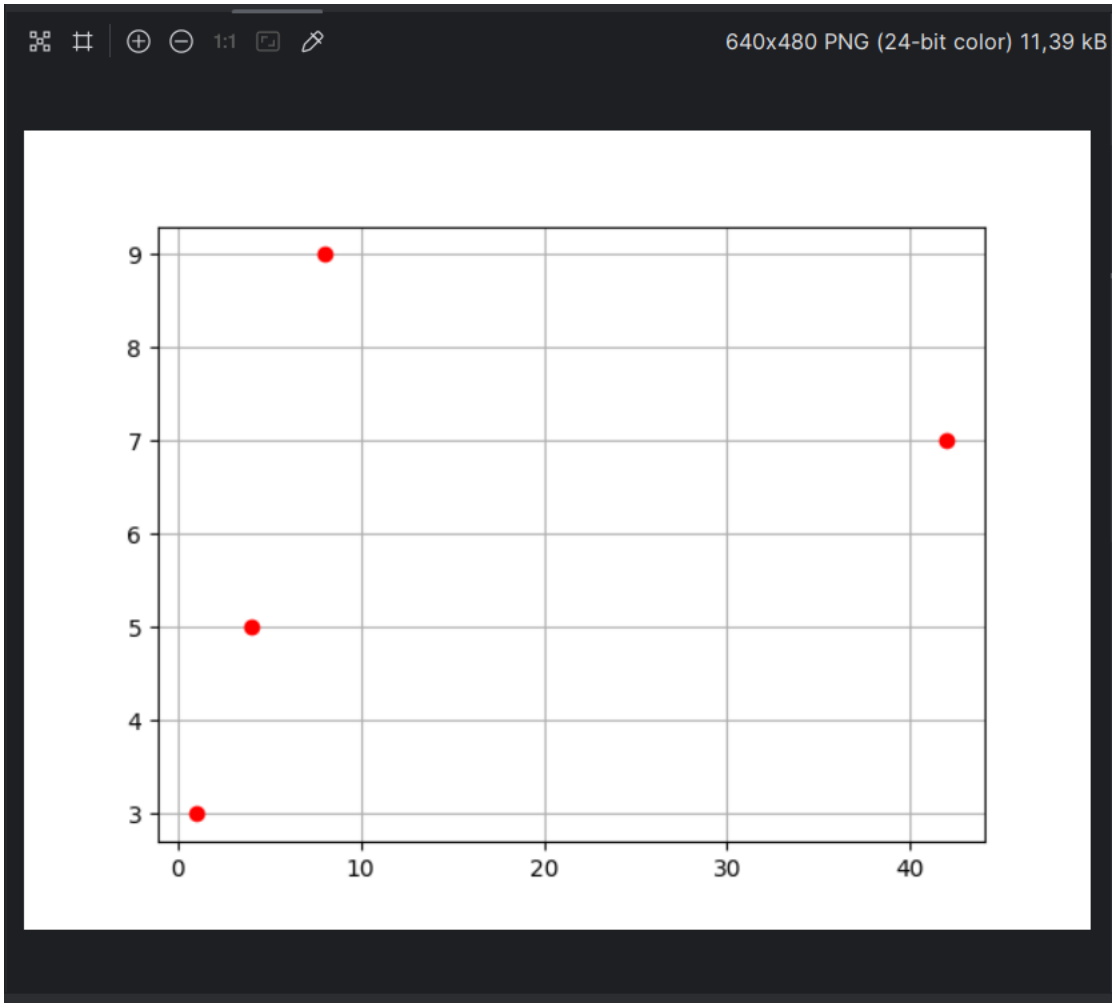


Рисунок 3 «Діаграма»

Висновок

У процесі виконання роботи я отримав 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.