

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.301.173.320.05 ЛР

Виконав студент гр. _____ 320 _____

_____ Перцев Кирило
(підпис, дата) (П.І.Б.)

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_5`, 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.

15.	Створити список з трьох точок, порахувати відстань між першою і третьою, пересунути другу на 45 вгору.
------------	--------------------------------------------------------------------------------------------------------

Рисунок 1 – Завдання 2

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

ВИКОНАННЯ РОБОТИ

Завдання 1. Створення класу Point_5.

Всі поля класу Point_5. Всього 3 поля:

__x – координата x
 __y – координата y
 __point_count – кількість точок

Всі методи класу Point_5:

Конструктор – деструктор:

__init__ – конструктор
 __del__ – деструктор

2 сеттера:

set_x – записує координату x
 set_y – записує координату y

2 геттера:

get_x – повертає координату x
 get_y – повертає координату y

Інші методи:

shift – робить зсув точки
 get_count – повертає номер точки

Лістинг коду класу Point_5 наведено в дод. А (стор. 5). Екрани роботи програми показані в дод. Б.

Алгоритм вирішення показано в дод. В.

Завдання 2. Вирішення задачі task15.

Вхідні дані: list_of3_points – список з трьох точок, дійсний тип.

Вихідні дані: сповіщення в консолі про довжину між першою і третьою точкою.

Лістинг коду вирішення задачі task15 наведено в дод. А (стор. 5). Екрани роботи програми показані в дод. Б.

Алгоритм вирішення показано в дод. В.

Завдання 3. Побудова точок (функція show_points).

Вхідні дані: list_of_points – список з трьох точок, дійсний тип.

Вихідні дані: графічне вікно з відображеними точками.

Лістинг коду вирішення задачі наведено в дод. А (стор. 5). Екрани роботи програми показані в дод. Б.

Алгоритм вирішення показано в дод. В.

Завдання 4. Збереження точок у файл (функція `save_points`).

Вхідні дані: `list_of_points` – список з трьох точок, дійсний тип.

Вихідні дані: текстовий файл зі збереженими точками.

Лістинг коду вирішення задачі наведено в дод. А (стор. 5). Екрани роботи програми показані в дод. Б.

Алгоритм вирішення показано в дод. В.

ВИСНОВКИ

Під час виконання лабораторної роботи було вивчено теоретичний матеріал щодо основ програмування на мові Python з використанням об'єктів і класів. Було з'ясовано, що на відміну від традиційних поглядів, коли програму розглядали як набір підпрограм, або як перелік інструкцій комп'ютеру, ООП-програми можна вважати сукупністю об'єктів. Відповідно до парадигми об'єктно-орієнтованого програмування, кожен об'єкт здатний отримувати повідомлення, обробляти дані, та надсилати повідомлення іншим об'єктам. Кожен об'єкт — своєрідний незалежний автомат з окремим призначенням та відповідальністю.

Matplotlib використовують для візуалізації даних будь-якої складності. Бібліотека дозволяє будувати різні варіанти графіків: лінійні, тривимірні, діаграми розсіювання та інші, а також комбінувати їх. На прикладі завдання 3 було розглянуто використання цієї бібліотеки.

ДОДАТОК А

Лістинг коду програми до задач 1, 2, 3, 4

Файл main.py

```

import matplotlib.pyplot as plt
from Point_5 import Point_5
import math

# Основна функція
def main():
    # пустий масив точок
    points = []
    # Вводимо координати точок
    # та створюємо екземпляри класу Point_5
    print("Enter X,Y values for 3 points:")
    for i in range(3):
        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 = Point_5(tmp_x, tmp_y)
            print(Point_5.get_count())
            points.append(tmp_point)

    # Відображення точок до змін
    show_points(points)
    # Обробка точок за варіантом
    task15(points)
    # Відображення точок після змін
    show_points(points)
    # Збереження координат точок у файлі
    save_points(points)

# Функція для обробки точок за варіантом
def task15(list_of3_points):
    """Створити список з трьох точок, порахувати відстань між першою і
    третьою, пересунути другу на 45 вгору."""
    point_1 = list_of3_points[0]
    point_2 = list_of3_points[1]
    point_3 = list_of3_points[2]
    length = math.sqrt(math.pow(point_2.get_x() - point_1.get_x(), 2) +
    math.pow(point_2.get_y() - point_1.get_y(), 2))
    point_2.shift(0.0, 45.0)
    print("Length = {}".format(length))

# відображення графічних об'єктів
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()

# збереження координат у файлу

```

```
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
            f.write(f"{num+1}: {point.get_x()}; {point.get_y()}\n")

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

Файл Point_5.py

```
class Point_5:
    """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)
        Point_5.__point_count += 1

    def __del__(self):
        print("Point has been deleted")
        Point_5.__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 Point_5.__point_count
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

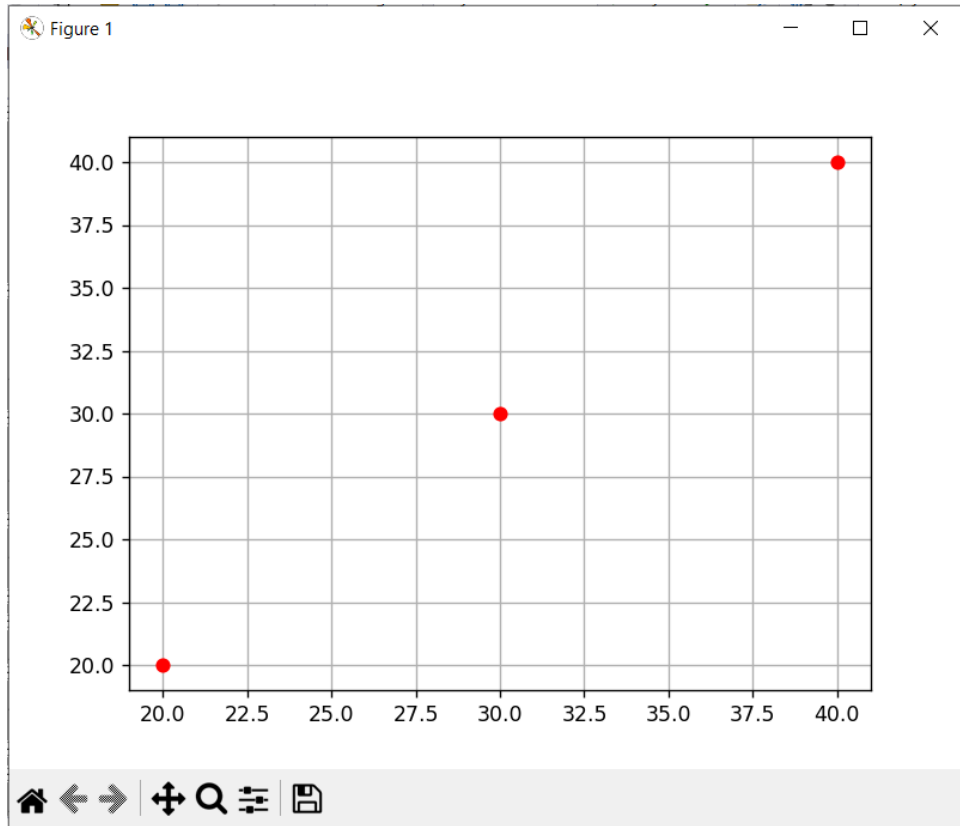


Рисунок Б.1 – Графік введених точок

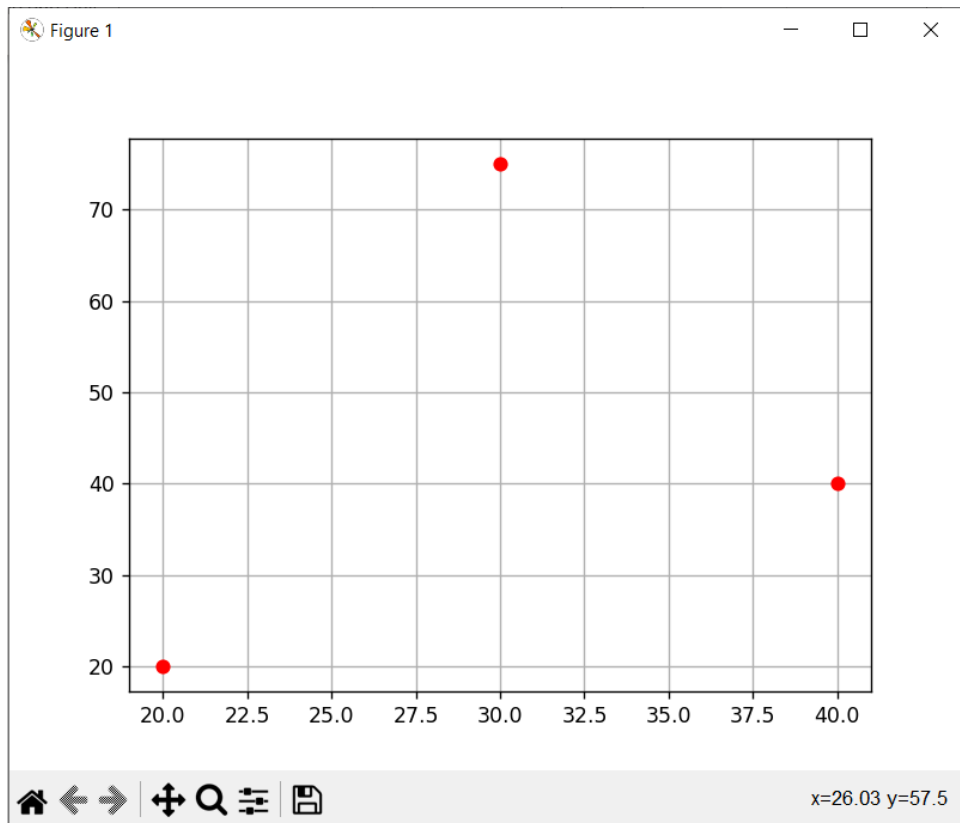
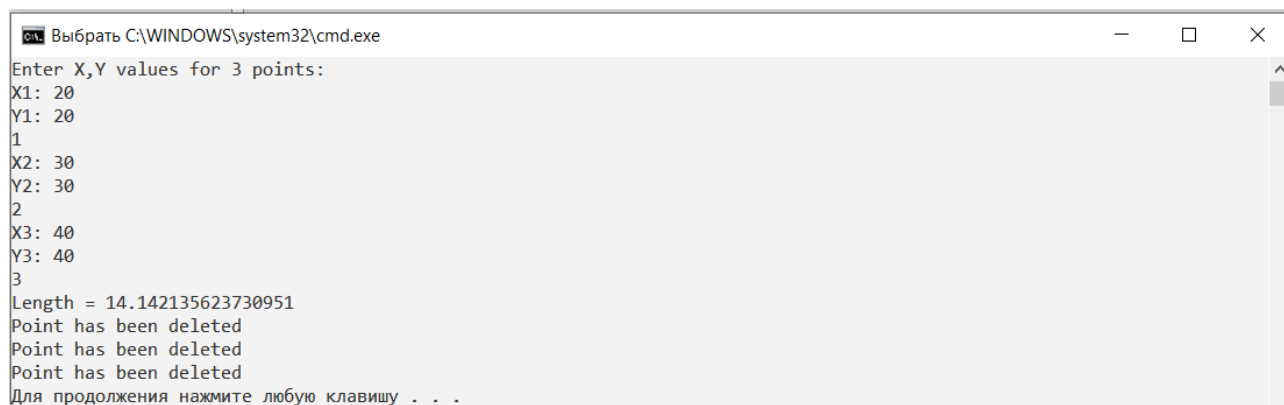
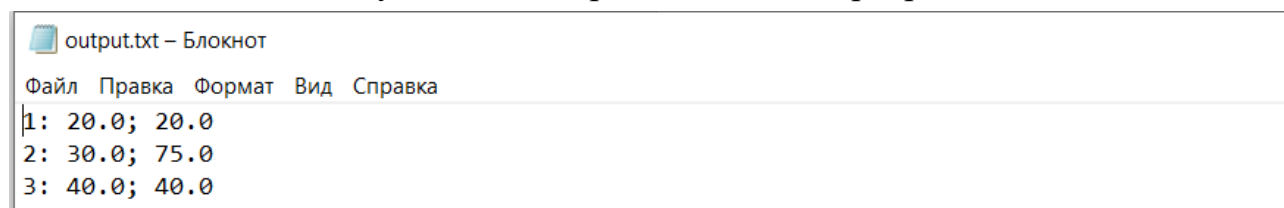


Рисунок Б.2 – Графік зі зміненою точкою



```
Выбрать C:\WINDOWS\system32\cmd.exe
Enter X,Y values for 3 points:
X1: 20
Y1: 20
1
X2: 30
Y2: 30
2
X3: 40
Y3: 40
3
Length = 14.142135623730951
Point has been deleted
Point has been deleted
Point has been deleted
Для продолжения нажмите любую клавишу . . .
```

Рисунок Б.3 – Экран виконання програми



```
output.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
1: 20.0; 20.0
2: 30.0; 75.0
3: 40.0; 40.0
```

Рисунок Б.4 – Текстовий файл для Завдання 4

ДОДАТОК В

Діаграми активності зо завдань 1 та всієї програми

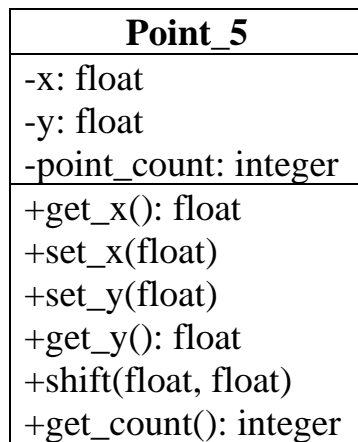


Рисунок В.1 – Діаграма класу Point_5

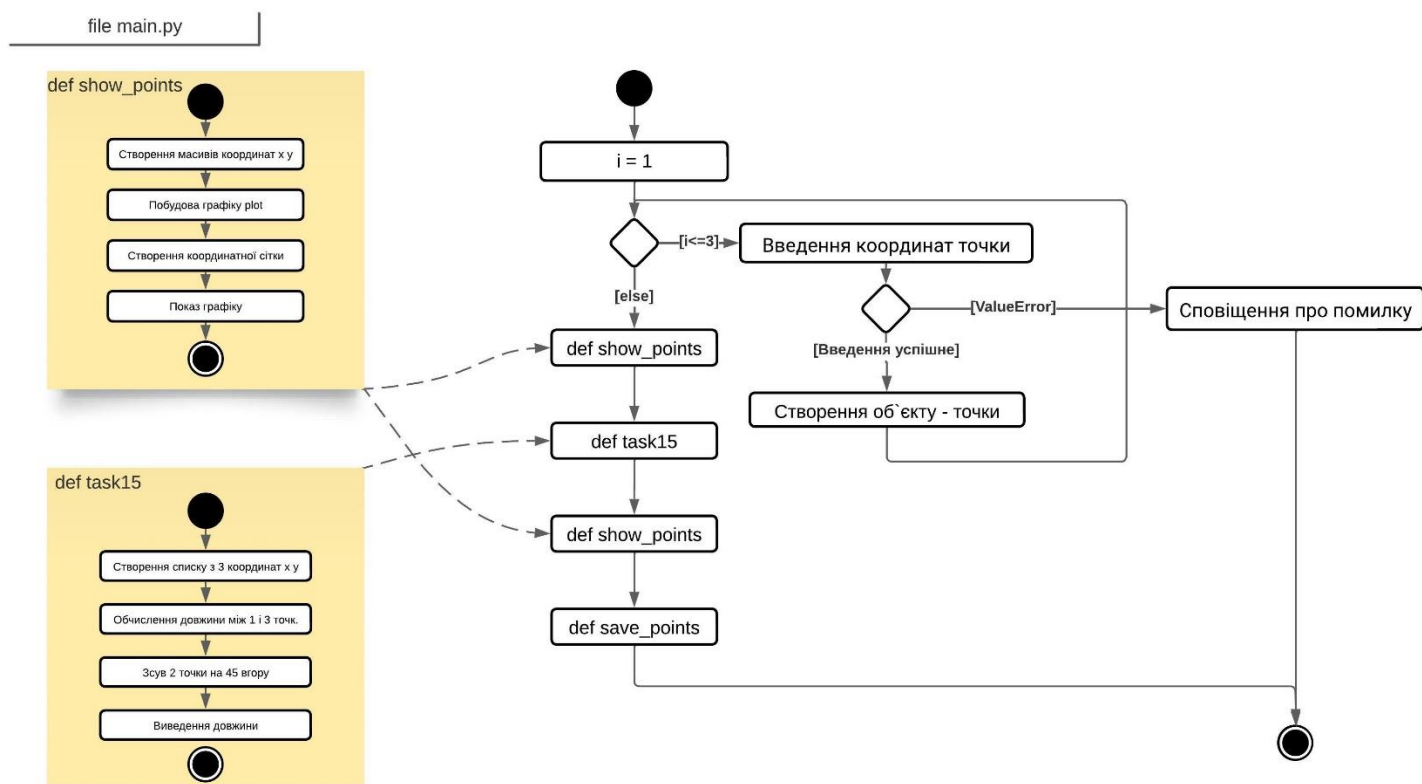


Рисунок В.2 – Діаграма активності для основного сценарію роботи з об'єктами.