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 _____»

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

ХАІ.301.174.322.11 ЛР

Виконав студент гр. _____322_____

_____Шаєнко Віталій Андрійович___
(підпис, дата) (П.І.Б.)

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

2024

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_21, 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.

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:

Task21. Створити список з трьох точок, порахувати відстань між першою і третьою, пересунути другу на 15 вниз і на 25 вліво.

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі Task21.

Вхідні дані (ім’я, опис, тип, обмеження):

A,B,C — точки з координатами (x;y),

float

Вихідні дані (ім’я, опис, тип):

distance - float, відстань між точками A та C
Виведення графіку з точками

Алгоритм вирішення показано нижче чи показано на рис. 1
Діаграму класу показано на рис. 2

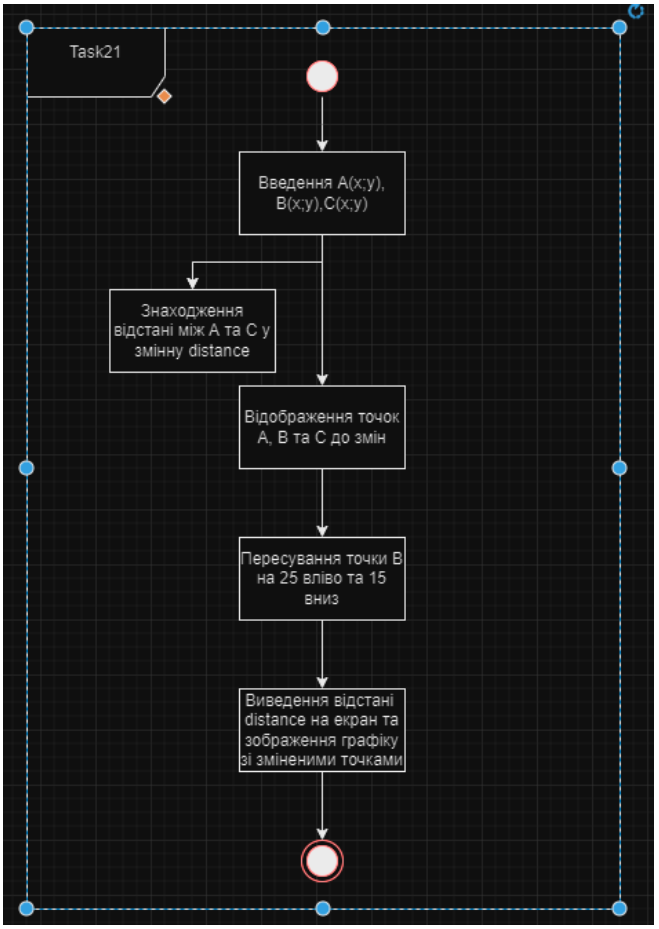

Рисунок 1 – Алгоритм вирішення Task21

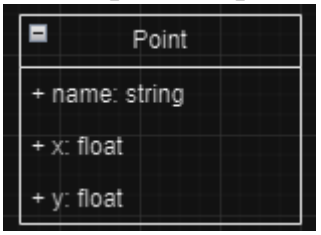

Рисунок 2 – Діагарама класу Point

Лістинг коду вирішення задачі наведено в дод. А (стор. 6).
Екран роботи програми показаний на рис. Б.1.

ВИСНОВКИ

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

ДОДАТОК А.1
Лістинг коду основної програми запуску

```
import lab4

task_list = {
    "1": lab4.task,
}

if __name__ == '__main__':
    choice = input("Оберіть завдання:\n1.Завдання№21 || [0-Вихід]: \n")

    while choice != "0":
        if choice in task_list.keys():
          task_list[choice]()
        else:
          print("Такої відповіді немає!")

        choice = input("Оберіть завдання знову [0-Вихід]: ")
```

ДОДАТОК А.2
Лістинг коду вирішення задач Task21

```
import matplotlib.pyplot as plt
import math

class Point:
    def __init__(self, name, x, y):
        self.name = name
        self.x = float(x)
        self.y = float(y)

    def find_length(self, other):
        return math.sqrt((other.x - self.x) ** 2 + (other.y - self.y) ** 2)

    def move(self, x_skip, y_skip):
        self.x += x_skip
        self.x = self.check_x()
        self.y += y_skip
        self.y = self.check_y()

    def show(self):
        print(self.x, self.y)

    def check_x(self):
        if self.x >= 100 or self.x <= -100:
            self.x = 0
        else:
            return self.x

    def check_y(self):
        if self.y >= 100 or self.y <= -100:
            self.y = 0
        else:
            return self.y
```

```
def task():
     point_list = [
        Point("A", float(input("Enter x for A: ")), float(input("Enter y for A:
")),
        Point("B", float(input("Enter x for B: ")), float(input("Enter y for B:
")),
        Point("C", float(input("Enter x for C: ")), float(input("Enter y for C:
")))
        ]

     for point in point_list:
          print(point.name,": (", point.x,";" ,point.y, ") ")

     distance = point_list[0].find_length(point_list[2])
     print(f"Відстань між точками {point_list[0].name} та {point_list[2].name}:
{distance}")

     print("Перед изменением:"), point_list[1].show()
     point_list[1].move(-25, -15)
     print("После изменения(Точка перемещена на 25 единиц влево и на 15 вниз):"),
point_list[1].show()

     graph(point_list)


def graph(point_list):
    x = [point.x for point in point_list]
    y = [point.y for point in point_list]

    for i, point in enumerate(point_list):
        plt.text(point.x + 1, point.y + 1, point.name, fontsize=10, color='black')

    plt.plot(x,y, 'ro')
    plt.grid()
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

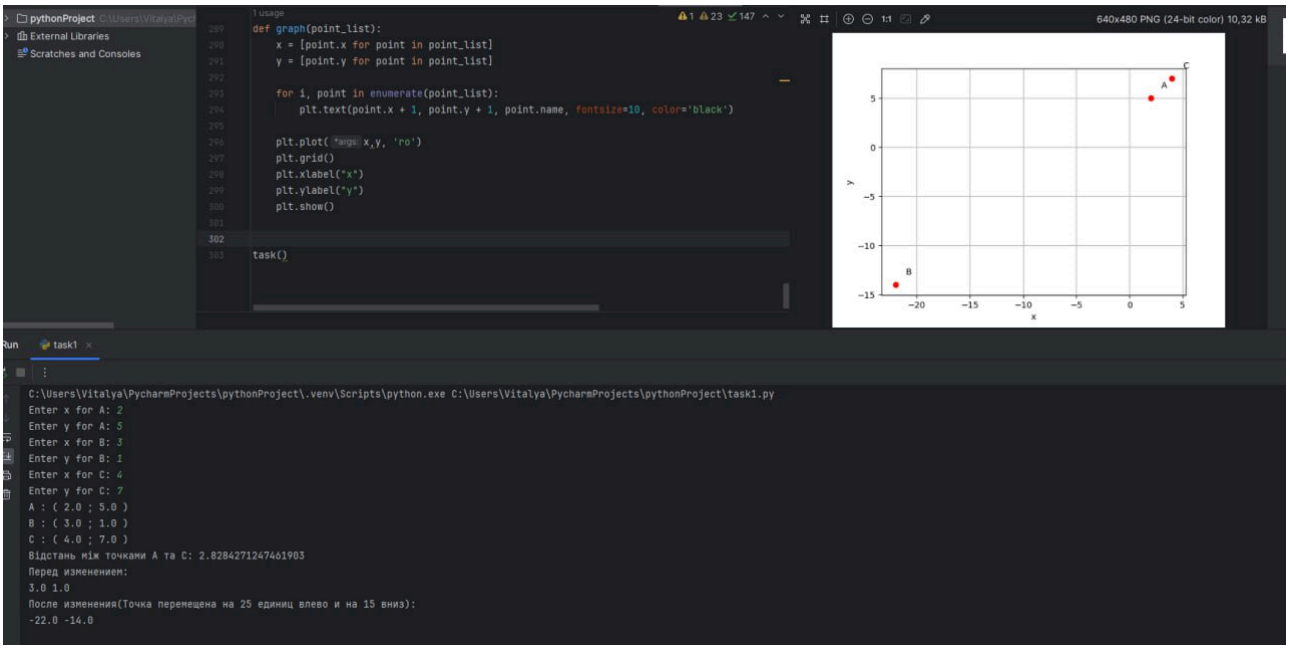

Рисунок Б.1 – Екран виконання програми для вирішення завдання Task21