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»

Виконав студент гр. 312ст

Ковальов Олег

(підпис, дата) (П.І.Б.)

Перевірив

(підпис, дата)

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 (записуює 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 х,

−

зсув по у.

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\_х; координата\_у – для непарних варіантів*

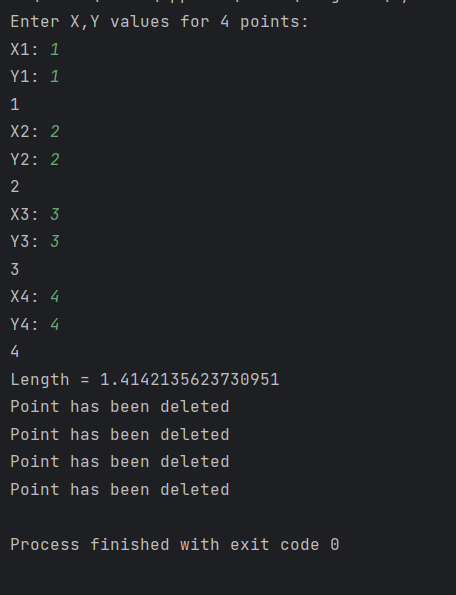
*(номер) координата\_х:координата\_у – для парних варіантів*

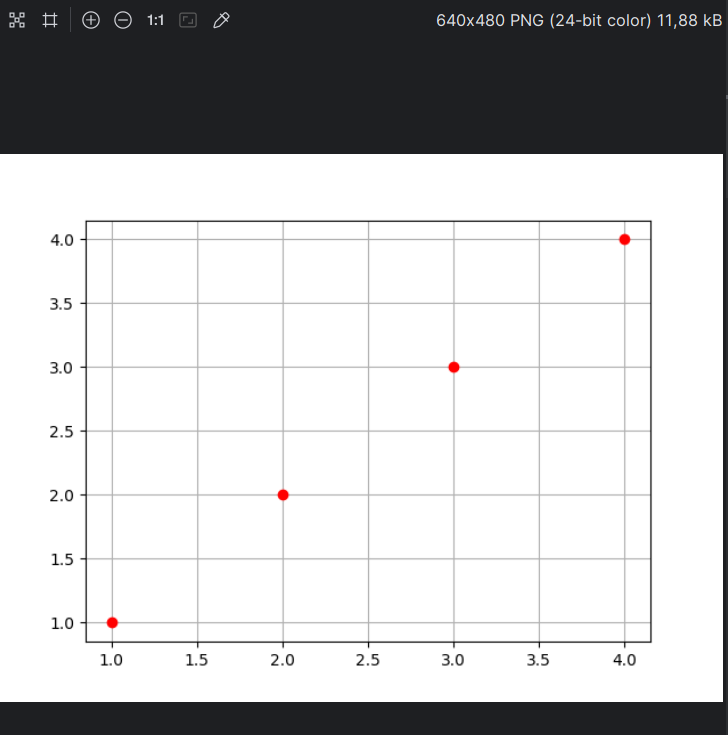


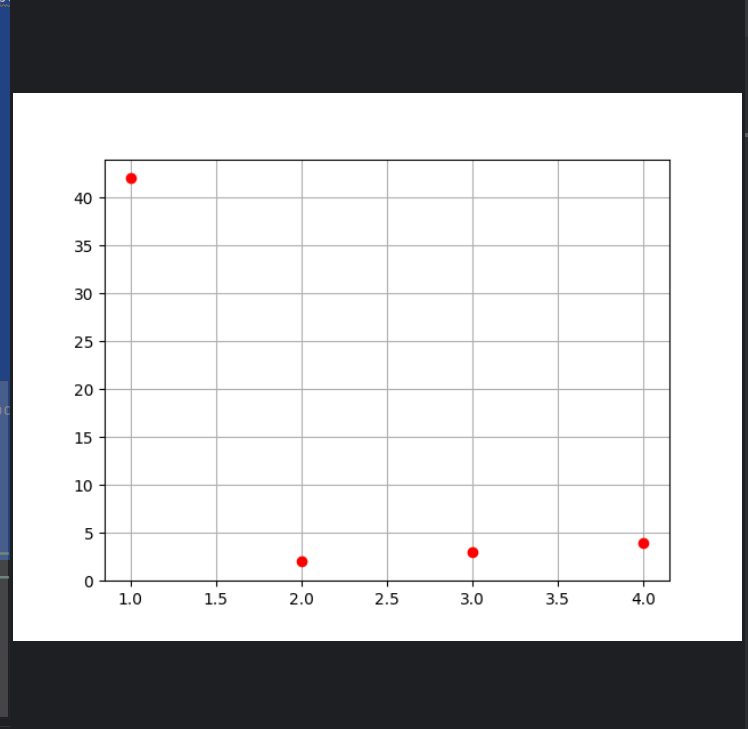
|  |
| --- |
| Point |
| - \_\_x: float = 0.0  - \_\_y: float = 0.0  - \_\_point\_count: int = 0 |
| + \_\_init\_\_(x: float, y: float)  + \_\_del\_\_()  + get\_x(): float  + set\_x(value: float): void  + set\_y(value: float): void  + get\_y(): float  + shift(x\_shift: float, y\_shift: float): void  + get\_count(): int |

Лістинг коду  
Клас  
class Point:  
   
  
 \_\_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.\_\_point\_count += 1  
  
 def \_\_del\_\_(self):  
 print("Point has been deleted")  
 Point.\_\_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.\_\_point\_count

Main:  
import matplotlib.pyplot as plt  
from Point import Point  
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 = Point(tmp\_x, tmp\_y)  
 print(Point.get\_count())  
 points.append(tmp\_point)  
 show\_points(points)  
 task1(points)  
 show\_points(points)  
 save\_points(points)  
  
  
def task1(list\_of4\_points):  
  
 point\_1 = list\_of4\_points[0]  
 point\_2 = list\_of4\_points[1]  
 point\_3 = list\_of4\_points[2]  
 point\_4 = list\_of4\_points[3]  
  
  
 length = math.sqrt(math.pow(point\_4.get\_x() - point\_3.get\_x(), 2) + math.pow(point\_4.get\_y() - point\_3.get\_y(), 2))  
 point\_1.shift(0.0, 41.0)  
 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()







Висновок

У процессі виконання роботи я отримав теоретичні знання з основ програмування на мові Python звикористанням об'єктів і класів, навички використання бібліотеки для візуалізації масивів даних, і навчитися розробляти скрипти для роботи з об'єктами призначених для користувача класів.