Міністерство освіти та науки України

Національний технічний університет України

«Київський Політехнічний Інститут ім. Ігоря Сікорського»

ФТІ

Кафедра ФТЗЗІ

**Лабораторна робота №9**

з дисципліни: «Програмування 4»

# на тему:

# «Побудова об’єктної ієрархії геометричних фігур»

Виконав:

Студент групи ФЕ-81

Кривчук Назарій

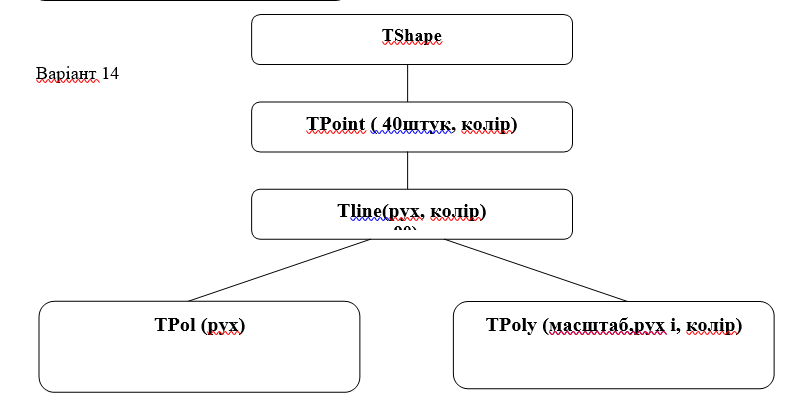
Перевірив:

Доцент Прогонов Д.О.

Київ 2020

**Мета роботи:** засвоїти базові поняття ООП на прикладі побудови ієрархії геометричних фігур засобами мови програмування Python.

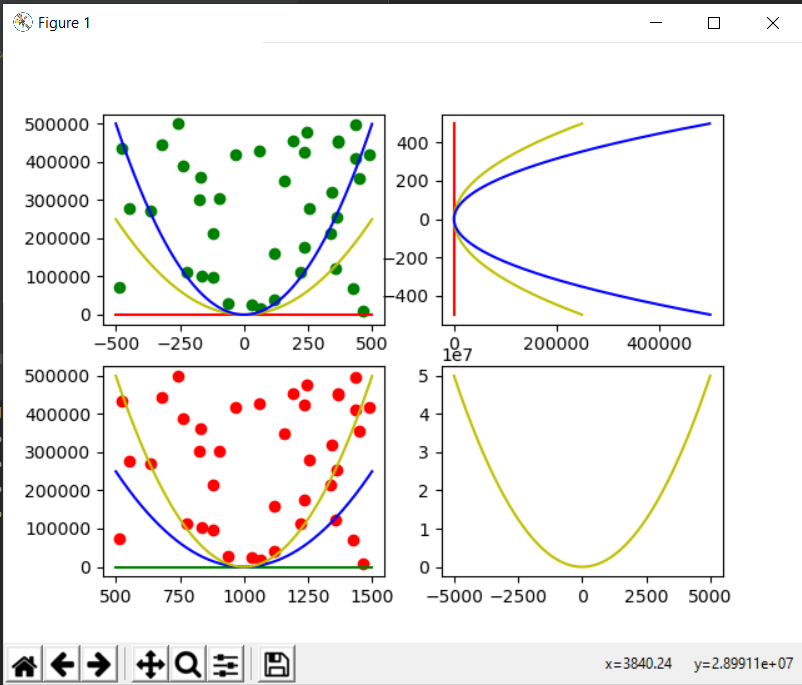
**Завдання:**

****

**Код реалізації:**

import matplotlib.pyplot as plt  
import numpy as np  
import random  
  
  
class TShape:  
 def \_\_init\_\_(self, name):  
 self.name = name  
  
 def draw(self):  
 print("Draw " + self.name)  
  
 def move(self, x, y):  
 print("Move")  
  
 def rotate(self):  
 print("Rotate")  
  
 def change\_color(self, color):  
 print("Changed color")  
  
  
class TPoint(TShape):  
 def \_\_init\_\_(self, x=None, y=None, color='g', count = 40):  
 if x is None:  
 x = [random.randint(-500, 500) for i in range(0, count)]  
 if y is None:  
 y = [random.randint(0, 500000) for i in range(0, count)]  
 self.x = x  
 self.y = y  
 TShape.\_\_init\_\_(self, "Point")  
 self.color = color  
 self.graph = None  
  
 def draw(self, ax):  
 self.graph = ax.plot(self.x, self.y, 'o', color=self.color)  
 # ax.setp(self.graph, color=self.color)  
  
 def move(self, x, y, ax=None):  
 for i in range(0, len(self.x)):  
 self.x[i] += x  
 for i in range(0, len(self.y)):  
 self.y[i] += y  
 if not ax == None:  
 self.draw(ax)  
  
 def change\_color(self, color):  
 print("Changed color")  
 self.color = color  
  
  
class TLine(TPoint):  
 def \_\_init\_\_(self, color='r'):  
 TShape.\_\_init\_\_(self, "Line")  
 TPoint.\_\_init\_\_(self, np.arange(-500, 500, 1), [50 for i in range(0, 1000)], color)  
  
 def draw(self, ax):  
 self.graph = ax.plot(self.x, self.y, color=self.color)  
  
 def rotate(self, ax=None):  
 TPoint.\_\_init\_\_(self, self.y, self.x, self.color)  
 if not ax == None:  
 self.draw(ax)  
  
  
class TPol(TLine):  
 def \_\_init\_\_(self, points\_num=1000, a=1.0, b=1.0, c=1.0, color='y', ):  
 TShape.\_\_init\_\_(self, "TPol")  
 x = np.arange(((-1) \* points\_num / 2), (points\_num / 2), 1)  
 TPoint.\_\_init\_\_(self, x, self.get\_y(x, a, b, c), color)  
  
 def get\_y(self, x, a, b, c):  
 y = []  
 for i in x:  
 y.append((a \* (i \*\* 2)) + (b \* i) + c)  
 return y  
  
  
class TPoly(TLine):  
 def \_\_init\_\_(self, koefs=None, points\_num=1000, color='b'):  
 if koefs == None:  
 koefs = [2.0, 1.0, 1.0]  
 TShape.\_\_init\_\_(self, "TPoly")  
 x = np.arange(((-1) \* points\_num / 2), (points\_num / 2), 1)  
 TPoint.\_\_init\_\_(self, x, self.get\_y(x, koefs), color)  
 self.koefs = koefs  
  
 def get\_y(self, x, koefs):  
 y = []  
 for i in x:  
 num = 0  
 stepen = len(koefs) - 1  
 for k in koefs:  
 num += k \* (i \*\* stepen)  
 stepen -= 1  
 y.append(num)  
 return y  
  
 def mashtab(self, points\_num, ax=None):  
 x = np.arange(((-1) \* points\_num / 2), (points\_num / 2), 1)  
 TPoint.\_\_init\_\_(self, x, self.get\_y(x, self.koefs), self.color)  
 if not ax == None:  
 self.draw(ax)  
  
  
point = TPoint()  
line = TLine()  
pol = TPol()  
poly = TPoly()  
  
figs, (ax1, ax2) = plt.subplots(nrows=2, ncols=2)  
  
point.draw(ax1[0])  
line.draw(ax1[0])  
pol.draw(ax1[0])  
poly.draw(ax1[0])  
  
line.rotate(ax1[1])  
pol.rotate(ax1[1])  
poly.rotate(ax1[1])  
  
line.rotate()  
pol.rotate()  
poly.rotate()  
  
point.change\_color('r')  
line.change\_color('g')  
pol.change\_color('b')  
poly.change\_color('y')  
  
point.move(1000, 0, ax2[0])  
line.move(1000, 0, ax2[0])  
pol.move(1000, 0, ax2[0])  
poly.move(1000, 0, ax2[0])  
  
poly.mashtab(10000, ax2[1])  
  
plt.show()

**Приклад виконання програми:**

****