Міністерство освіти та науки України

Національний технічний університет України

«Київський Політехнічний Інститут ім. Ігоря Сікорського»

Фізико-Технічний Інститут

Кафедра ФТЗЗІ

**Лабораторна робота №9**

з дисципліни: «Програмування 4»

# на тему:

# «Побудова об’єктної ієрархії геометричних фігур на мові Python»

# 

Виконав:

Студент групи ФЕ-81

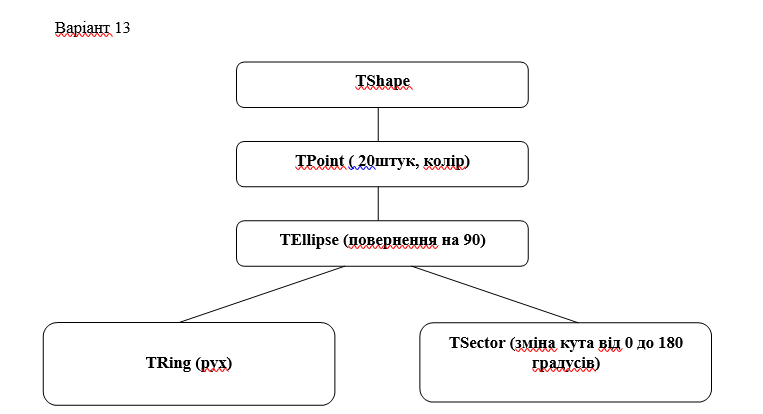
Кочерга Валерій

Перевірив:

Доцент Прогонов Д.О.

Київ 2020

Завдання:



Код програми:

import matplotlib  
import numpy as np  
import matplotlib.pyplot as plt  
import numpy.random as rnd  
  
  
class Tshape:  
 def \_\_init\_\_(self, center, colour, scale):  
 self.center = center  
 self.colour = colour  
 self.scale = scale  
  
  
class TPoint(Tshape):  
 def \_\_init\_\_(self, center, colour, width, scale):  
 Tshape.\_\_init\_\_(self, center, colour, scale)  
 self.width = width  
 self.colour = colour  
  
 def set\_colour(self, colour):  
 self.colour = str(colour)  
  
  
class Ellipse(TPoint):  
 def \_\_init\_\_(self, w, h, a, flagRand=False, xy=0.0, center=2.0, color="r", scale=1.0):  
 Tshape.\_\_init\_\_(self, center, color, scale)  
 if not xy:  
 self.xy = rnd.rand(2)  
 else:  
 self.xy = xy  
 self.flagR = flagRand  
 self.width = w  
 self.height = h  
 self.angle = a \* 360  
  
 def change\_angle(self, angle\_1):  
 self.angle = angle\_1  
  
 def draw(self):  
 e = matplotlib.patches.Ellipse(xy=self.xy, width=self.width, height=self.height, angle=self.angle)  
 fig = plt.figure(0)  
 ax = fig.add\_subplot(111, aspect='equal')  
 ax.add\_artist(e)  
 e.set\_clip\_box(ax.bbox)  
 e.set\_alpha(rnd.rand())  
 if self.flagR:  
 e.set\_facecolor(rnd.rand(3))  
 else:  
 e.set\_facecolor(self.color)  
 ax.set\_xlim(-2, 10)  
 ax.set\_ylim(-2, 10)  
  
 plt.show()  
  
  
class Ring(Ellipse):  
 def \_\_init\_\_(self, r, xCenter, yCenter, colour="g"):  
 self.radius = r  
 self.center = [xCenter, yCenter]  
 self.colour = colour  
  
 def move(self, x, y):  
 self.center[0] += x  
 self.center[1] += y  
  
 def draw(self):  
 x0 = self.center[0]  
 y0 = self.center[1]  
 # Parametric plot in t  
 points = []  
 t = 0  
 phi = 0  
 # build points  
 while t < 2 \* np.pi:  
 x = x0 + self.radius \* np.cos(t) \* np.cos(phi) - self.radius \* np.sin(t) \* np.sin(phi)  
 y = y0 + self.radius \* np.cos(t) \* np.sin(phi) + self.radius \* np.sin(t) \* np.cos(phi)  
 points.append((x, y, self.colour))  
 t += 0.31  
 t = 0  
 x, y = [], []  
 # build connections  
 for i in range(len(points)):  
 x.append(x0 + self.radius \* np.cos(t) \* np.cos(phi) - self.radius \* np.sin(t) \* np.sin(phi))  
 y.append(y0 + self.radius \* np.cos(t) \* np.sin(phi) + self.radius \* np.sin(t) \* np.cos(phi))  
 t += 0.31  
 plt.plot(x, y, self.colour)  
 plt.show()  
  
  
class Sector(Ellipse):  
 def \_\_init\_\_(self, a, b, x0, y0, radius, flagRand=True, angle=0, colour="b"):  
 self.angle = angle  
 self.a = a  
 self.b = b  
 self.center = [x0, y0]  
 self.radius = radius  
 self.colour = colour  
 self.flagR = flagRand  
  
 def draw(self):  
 # Compute ellipse parameters  
 a = self.a / 2  
 x0 = self.center[0]  
 y0 = self.center[1]  
 b = self.b / 2  
 phi = self.angle  
  
 # Parametric plot in t  
 points = []  
 t = 0  
 while t <= self.angle:  
 x = x0 + a \* np.cos(t) \* np.cos(phi) - b \* np.sin(t) \* np.sin(phi)  
 y = y0 + a \* np.cos(t) \* np.sin(phi) + b \* np.sin(t) \* np.cos(phi)  
 points.append((x, y, self.colour))  
 t += 0.31  
 t = 0  
 x, y = [], []  
 for i in range(len(points)):  
 x.append(x0 + a \* np.cos(t) \* np.cos(phi) - b \* np.sin(t) \* np.sin(phi))  
 y.append(y0 + a \* np.cos(t) \* np.sin(phi) + b \* np.sin(t) \* np.cos(phi))  
 t += 0.31  
 plt.plot(x, y, self.colour)  
 plt.plot([x0, x[len(points) - 1]], [y0, y[len(points) - 1]], self.colour)  
 plt.plot([x0, x[0]], [y0, y[0]], self.colour)  
 plt.show()  
  
  
t = TPoint(5.0, 'none', 1.0, 1.0)  
t.set\_colour('red')  
  
e = Ellipse(3, 5, 232, flagRand=True)  
e.draw()  
e.change\_angle(90)  
e.draw()  
  
r = Ring(2, 3, 3)  
r.draw()  
r.move(2, 2)  
r.draw()  
  
e = Sector(3, 3, 5, 5, 2)  
e.draw()  
e.change\_angle(np.pi)  
e.draw()

Приклад виконання програми:

