Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Фізико-технічний інститут

Кафедра фізико-технічних засобів захисту інформації

**Протокол лабораторної роботи № 9**

з дисципліни Програмування 4

за темою

Принципи програмування на мові Python

**Виконав:**

Студентка 2 курсу

Групи ФЕ-71

Христюк І.В.

**Прийняв:**

Доц. Прогонов Д.О.

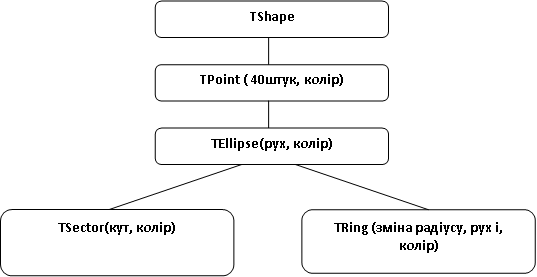
**Київ 2019**

1.Тема: Побудова об’єктної ієрархії геометричних фігур

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

## Порядок виконання роботи

1. Ознайомтеся з теоретичною частиною
2. Реалізуйте програмно приклади приведені в тексті



**Реалізація алгоритму**

import numpy as np  
import matplotlib.pyplot as plt  
import matplotlib.patches as pts  
import menu  
from math import isnan, isinf  
  
colors = ['r', 'g', 'b', 'y', 'red', 'green', 'blue', 'yellow']  
class TShape:  
 pass  
  
class TPoint(TShape):  
  
 #initializator  
 def \_\_init\_\_(self, coordX, coordY, color):  
 self.coordX, self.coordY = float(coordX), float(coordY)  
 if color in colors:  
 self.color = color  
 else:  
 print("Unable to set this color.")  
 exit(-1)  
 def buildPoint(self):  
 plt.scatter(self.coordX, self.coordY, color=self.color)  
  
 def changeCollor(self, new\_color):  
 if new\_color in colors:  
 self.color = new\_color  
 else:  
 print("Unable to change color.")  
 exit(-1)  
  
class TEllipse(TPoint):  
 def \_\_init\_\_(self, a, b, x0, y0, color):  
 self.a = a  
 self.b = b  
 self.center = [x0,y0]  
 self.angle = 0  
 self.color = color  
  
 def showFigure(self):  
 # Compute ellipse parameters  
 a = self.a/2 # Semimajor axis  
 x0 = self.center[0] # Center x-value  
 y0 = self.center[1] # Center y-value  
 b = self.b/2 # Semiminor axis  
 phi = self.angle # Angle betw major axis and x-axis  
  
 # Parametric plot in t  
 points = []  
 t = 0  
 while t < 2\*np.pi:  
 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(TPoint(x, y, self.color))  
 t += 0.16  
 t = 0  
 x, y = [], []  
 for i in range(len(points)):  
 points[i].buildPoint()  
 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+=0.16  
 plt.plot(x, y, self.color)  
 plt.show()  
  
 def spin(self):  
 self.angle += np.pi/2  
  
 def move(self, x, y):  
 self.center[0] += x  
 self.center[1] += y  
  
 def changeCollor(self):  
 while True:  
 color = input("Введите цвет фигуры: ")  
 if not color in colors:  
 print("Такой цвет недоступен.")  
 continue  
 else:  
 break  
 self.color = color  
  
class TSector(TEllipse):  
 def \_\_init\_\_(self, a, b, x0, y0, color, angle):  
 self.a = a  
 self.b = b  
 self.center = [x0, y0]  
 self.angle = angle  
 self.color = color  
 self.shAngle = angle  
  
 def showFigure(self):  
 # Compute ellipse parameters  
 a = self.a/2 # Semimajor axis  
 x0 = self.center[0] # Center x-value  
 y0 = self.center[1] # Center y-value  
 b = self.b/2 # Semiminor axis  
 phi = self.angle # Angle betw major axis and x-axis  
  
 # Parametric plot in t  
 points = []  
 t = 0  
 while t <= self.shAngle:  
 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(TPoint(x, y, self.color))  
 t += 0.16  
 t = 0  
 x, y = [], []  
 for i in range(len(points)):  
 points[i].buildPoint()  
 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+=0.16  
 plt.plot(x, y, self.color)  
 plt.plot([x0, x[len(points)-1]], [y0, y[len(points)-1]], self.color)  
 plt.plot([x0, x[0]], [y0, y[0]], self.color)  
 plt.show()  
  
 def changeAngle(self):  
 while True:  
 try:  
 angle = float(input("Введите угол, который занимает сектор(в градусах): "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if angle < 0:  
 print("Введите угол больше, или равный 0.")  
 continue  
 elif isnan(angle):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(angle):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else:  
 angle = (angle\*np.pi)/180  
 break  
 self.shAngle = angle  
  
 def changeCollor(self):  
 while True:  
 color = input("Введите цвет фигуры: ")  
 if not color in colors:  
 print("Такой цвет недоступен.")  
 continue  
 else:  
 break  
 self.color = color  
  
class TRing(TEllipse):  
 def \_\_init\_\_(self, r, xCenter, yCenter, color):  
 self.radius = r  
 self.center = [xCenter, yCenter]  
  
 self.color = color  
  
 def showFigure(self):  
 x0 = self.center[0]  
 y0 = self.center[1]  
 rad = self.radius  
 # Parametric plot in t  
 points = []  
 t = 0  
 phi = 0  
 #build points  
 while t < 2\*np.pi:  
 x = x0 + rad \* np.cos(t) \* np.cos(phi) - rad \* np.sin(t) \* np.sin(phi)  
 y = y0 + rad \* np.cos(t) \* np.sin(phi) + rad \* np.sin(t) \* np.cos(phi)  
 points.append(TPoint(x, y, self.color))  
 t += 0.16  
 t = 0  
 x, y = [], []  
 #build connections  
 for i in range(len(points)):  
 points[i].buildPoint()  
 x.append(x0 + rad \* np.cos(t) \* np.cos(phi) - rad \* np.sin(t) \* np.sin(phi))  
 y.append(y0 + rad \* np.cos(t) \* np.sin(phi) + rad \* np.sin(t) \* np.cos(phi))  
 t+=0.16  
 plt.plot(x, y, self.color)  
 plt.show()  
  
 def changeRadius(self, new\_radius):  
 self.radius = new\_radius  
  
 def changeCollor(self):  
 while True:  
 color = input("Введите цвет фигуры: ")  
 if not color in colors:  
 print("Такой цвет недоступен.")  
 continue  
 else:  
 break  
 self.color = color  
  
 def move(self, x, y):  
 self.center[0] += x  
 self.center[1] += y  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 while True:  
 fig, ax = plt.subplots()  
 plt.grid()  
 userAnswer = menu.showMenu()  
 if userAnswer == '1':  
 data = menu.buildEllipse()  
 a, b = data[0], data[1]  
 coordsCenter = [data[2], data[3]]  
 color = data[4]  
 newEllipse = TEllipse(a, b, coordsCenter[0], coordsCenter[1], color)  
 newEllipse.showFigure()  
 while True:  
 ans = input("Действия над фигурой: \n 1- Сдвинуть \n 2- Изменить цвет \n 3 - Повернуть на 90 градусов. \n 0 - Перейти в предыдущее меню. \n Ваш выбор -> ")  
 if ans == '1':  
 while True:  
 try:  
 move\_X = float(input("Введите смещение по X: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 if not menu.checkNum(move\_X):  
 continue  
 try:  
 move\_Y = float(input("Введите смещение по Y: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 if not menu.checkNum(move\_Y):  
 continue  
 break  
 newEllipse.move(move\_X, move\_Y)  
 plt.grid()  
 newEllipse.showFigure()  
 elif ans == '2':  
 newEllipse.changeCollor()  
 plt.grid()  
 newEllipse.showFigure()  
 elif ans == '3':  
 newEllipse.spin()  
 plt.grid()  
 newEllipse.showFigure()  
 elif ans == '0':  
 break  
 else:  
 print("Такого пункта меню не существует.")  
  
 elif userAnswer == '2':  
 data = menu.buildSector()  
 a, b = data[0], data[1]  
 coordsCenter = [data[2], data[3]]  
 color = data[4]  
 angle = data[5]  
 newSector = TSector(a, b, coordsCenter[0], coordsCenter[1], color, angle)  
 plt.grid()  
 newSector.showFigure()  
 while True:  
 ans = input("Действия над фигурой: \n 1- Изменить угол \n 2- Изменить цвет \n 0 - Перейти в предыдущее меню. \n Ваш выбор -> ")  
 if ans == '1':  
 newSector.changeAngle()  
 plt.grid()  
 newSector.showFigure()  
 elif ans == '2':  
 newSector.changeCollor()  
 plt.grid()  
 newSector.showFigure()  
 elif ans == '0':  
 break  
 else:  
 print("Такого пункта меню не существует.")  
  
 elif userAnswer == '3':  
 data = menu.buildRing()  
 radius = data[0]  
 coordsCenter = [data[1], data[2]]  
 color = data[3]  
 newRing = TRing(radius, coordsCenter[0], coordsCenter[1], color)  
 plt.grid()  
 newRing.showFigure()  
 while True:  
 ans = input("Действия над фигурой: 1- Сдвинуть \n 2- Изменить радиус \n 3- Изменить цвет \n 0 - Перейти в предыдущее меню. \n Ваш выбор -> ")  
 if ans == '1':  
 while True:  
 try:  
 move\_X = float(input("Введите смещение по X: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 if not menu.checkNum(move\_X):  
 continue  
 try:  
 move\_Y = float(input("Введите смещение по Y: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 if not menu.checkNum(move\_Y):  
 continue  
 break  
 newRing.move(move\_X, move\_Y)  
 plt.grid()  
 newRing.showFigure()  
 elif ans == '2':  
 radius = 0  
 while True:  
 try:  
 radius = float(input("Введите радиус окружности: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if radius <= 0:  
 print("Размер должен быть больше нуля.")  
 continue  
 elif isnan(radius):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(radius):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else:  
 break  
 newRing.changeRadius(radius)  
 plt.grid()  
 newRing.showFigure()  
 elif ans == '3':  
 newRing.changeCollor()  
 plt.grid()  
 newRing.showFigure()  
 elif ans == '0':  
 break  
 else:  
 print("Такого пункта меню не существует.")  
  
 elif userAnswer == '4':  
 exit()

from math import isnan, isinf  
from numpy import pi  
colors = ['r', 'g', 'b', 'y', 'red', 'green', 'blue', 'yellow']  
def showMenu():  
 availableAnswers = ['1', '2', '3', '4']  
 while True:  
 print("Выберите, какую фигуру построить:")  
 print("1 - Эллипс; \n 2 - Сегмент; \n 3 - Обод; \n 4 - выход из программы.")  
 answ = input("Ваш выбор -> ")  
 if not answ in availableAnswers:  
 print("Неверный выбор. Такого пункта не существует.")  
 continue  
 else:  
 break  
 return answ  
  
#================Ellipse===========================  
def buildEllipse():  
 while True:  
 try:  
 a = float(input("Введите размер первой полуоси: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if a <= 0:  
 print("Размер должен быть больше нуля.")  
 continue  
 elif isnan(a):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(a):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 try:  
 b = float(input("Введите размер второй полуоси: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if b <= 0:  
 print("Размер должен быть больше нуля.")  
 continue  
 elif isnan(b):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(b):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 try:  
 coordX = float(input("Введите x-координату центра: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if isnan(coordX):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(coordX):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 try:  
 coordY = float(input("Введите y-координату центра: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if isnan(coordY):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(coordY):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 color = input("Введите цвет фигуры: ")  
 if not color in colors:  
 print("Такой цвет недоступен.")  
 continue  
 else: break  
  
 return [a, b, coordX, coordY, color]  
  
#================Sector===========================  
def buildSector():  
 while True:  
 try:  
 a = float(input("Введите размер первой полуоси: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if a <= 0:  
 print("Размер должен быть больше нуля.")  
 continue  
 elif isnan(a):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(a):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 try:  
 b = float(input("Введите размер второй полуоси: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if b <= 0:  
 print("Размер должен быть больше нуля.")  
 continue  
 elif isnan(b):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(b):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 try:  
 coordX = float(input("Введите x-координату центра: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if isnan(coordX):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(coordX):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
 while True:  
 try:  
 coordY = float(input("Введите y-координату центра: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if isnan(coordY):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(coordY):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 color = input("Введите цвет фигуры: ")  
 if not color in colors:  
 print("Такой цвет недоступен.")  
 continue  
 else: break  
 while True:  
 try:  
 angle = float(input("Введите угол, который занимает сектор(в градусах): "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if angle < 0:  
 print("Введите угол больше, или равный 0.")  
 continue  
 elif isnan(angle):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(angle):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else:  
 angle = (angle\*pi)/180  
 break  
 return [a, b, coordX, coordY, color, angle]  
  
  
#================Ring===========================  
def buildRing():  
  
 while True:  
 try:  
 radius = float(input("Введите радиус окружности: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if radius <= 0:  
 print("Размер должен быть больше нуля.")  
 continue  
 elif isnan(radius):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(radius):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 try:  
 coordX = float(input("Введите x-координату центра: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if isnan(coordX):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(coordX):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 try:  
 coordY = float(input("Введите y-координату центра: "))  
 except ValueError:  
 print("Вы ввели не число.")  
 continue  
 if isnan(coordY):  
 print("Невозможно выполнить действие с не числом.")  
 continue  
 elif isinf(coordY):  
 print("Невозможно выполнить действие с бесконечностью.")  
 continue  
 else: break  
  
 while True:  
 color = input("Введите цвет фигуры: ")  
 if not color in colors:  
 print("Такой цвет недоступен.")  
 continue  
 else: break  
  
 return [radius, coordX, coordY, color]  
  
  
def checkNum(number):  
 if isnan(number):  
 print("Невозможно выполнить действие с не числом.")  
 return False  
 elif isinf(number):  
 print("Невозможно выполнить действие с бесконечностью.")  
 return False  
 else:  
 return True