from math import sin, cos

class Punto:

"Classe per gestire un punto geometrico."

def \_\_init\_\_(self,x,y,z):

self.x=x

self.y=y

self.z=z

def Distanza(self):

"Determina la distanza del punto dall'origine."

return ( self.x\*\*2 + self.y\*\*2 + self.z\*\*2 )\*\*0.5

def Trasla(self,Dx,Dy,Dz):

self.x += Dx

self.y += Dy

self.z += Dz

def Ruota(self,x0,y0,z0,ax,ay,az):

self.x, self.y, self.z = self.x-x0, self.y-y0, self.z-z0

self.x, self.y, self.z = self.x\*cos(az)-self.y\*sin(az), \

self.x\*sin(az)+self.y\*cos(az), self.z

self.x, self.y, self.z = self.x, self.y\*cos(ax)-self.z\*sin(ax), \

self.y\*sin(ax)+self.z\*cos(ax)

self.x, self.y, self.z = self.x\*cos(-ay)-self.z\*sin(-ay), self.y, \

self.x\*sin(-ay)+self.z\*cos(-ay)

self.x, self.y, self.z = self.x+x0, self.y+y0, self.z+z0

class Punto2D(Punto):

"Classe per gestire un punto geometrico nel piano x, y."

def \_\_init\_\_(self,x,y):

Punto.\_\_init\_\_(self,x,y,0)

def Trasla(self,Dx,Dy):

Punto.Trasla(self,Dx,Dy,0)

def Ruota(self,x0,y0,az):

Punto.Ruota(self,x0,y0,0.,0.,0.,az)

def Anomalia(self):

return arctan2(self.y, self.x)

def \_\_str\_\_(self):

return "( %f, %f, %f )" % (self.x,self.y,self.z)

def \_\_repr\_\_(self):

return "( %f, %f, %f )" % (self.x,self.y,self.z)

def \_\_lt\_\_(self,altro):

return self.Distanza() < altro.Distanza()

def \_\_le\_\_(self,altro):

return self.Distanza() <= altro.Distanza()

def \_\_eq\_\_(self,altro):

return self.Distanza() == altro.Distanza()

def \_\_ge\_\_(self,altro):

return self.Distanza() >= altro.Distanza()

def \_\_gt\_\_(self,altro):

return self.Distanza() > altro.Distanza()

def \_\_iter\_\_(self):

return self

def \_\_next\_\_(self):

if not hasattr(self,'index'):

self.index = 0

if self.index == 3:

self.index = 0

raise StopIteration

elif self.index == 0:

self.index += 1

return self.x

elif self.index == 1:

self.index += 1

return self.y

elif self.index == 2:

self.index += 1

return self.z