Tema 7 Cei 4 piloni ai OOP

Ex:

ABSTRACTION

Clasa abstractă FormaGeometrica

Conține un field PI=3.14

Conține o metodă abstractă aria (opțional)

Conține o metodă a clasei descrie() - aceasta printează pe ecran ‘Cel mai

probabil am colturi’

INHERITANCE

Clasa Pătrat - moștenește FormaGeometrica

constructor pentru latură

ENCAPSULATION

latura este proprietate privată

Implementează getter, setter, deleter pentru latură

Implementează metoda cerută de interfață (opțional, doar dacă ai ales să

implementezi metoda abstractă aria)

Clasa Cerc - moștenește FormaGeometrica

constructor pentru rază

raza este proprietate privată

Implementează getter, setter, deleter pentru rază

Implementează metoda cerută de interfață - în calcul folosește field PI

mostenit din clasa părinte (opțional, doar dacă ai ales să implementezi metoda

abstractă aria)

POLYMORPHISM

Definește o nouă metodă descrie - printează ‘Eu nu am colturi’

Creează un obiect de tip Patrat și joacă-te cu metodele lui

from abc import ABC, abstractmethod

class FormaGeometrica(ABC):

PI = 3.14

@abstractmethod

def aria(self):

pass

def descrie(self):

print('Cel mai probabil am colturi.')

class Patrat(FormaGeometrica):

def \_\_init\_\_(self, latura):

self.\_\_latura = latura

def get\_latura(self):

return self.\_\_latura

def set\_latura(self, latura\_noua):

self.\_\_latura = latura\_noua

return latura\_noua

def del\_latura(self):

print('Latura a fost stearsa.')

del self.\_\_latura

def aria(self):

return self.\_\_latura \*\* 2

class Cerc(FormaGeometrica):

def \_\_init\_\_(self, raza):

self.\_\_raza = raza

def get\_raza(self):

return self.\_\_raza

def set\_raza(self, raza\_noua):

self.\_\_raza = raza\_noua

return raza\_noua

def del\_raza(self):

print('Raza a fost stearsa.')

del self.\_\_raza

def aria(self):

return FormaGeometrica.PI \* self.\_\_raza \*\* 2

def descrie\_cerc(self):

print('Eu nu am colturi.')

patrat = Patrat(3)

print(patrat.get\_latura())

print(patrat.del\_latura())

print(patrat.set\_latura(4))

print(patrat.aria())

patrat.descrie()

cerc = Cerc(2)

print(cerc.get\_raza())

print(cerc.del\_raza())

print(cerc.set\_raza(5))

print(cerc.aria())

cerc.descrie\_cerc()