from math import pi

from datetime import date

from tabulate import tabulate

# Ex 1

class Cerc:

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

self.raza = raza

self.culoare = culoare

def descrie\_cerc(self):

print(f'Raza este {self.raza} cm iar culoarea {self.culoare}.')

def aria(self):

return pi \* self.raza \*\* 2

def diametru(self):

return self.raza \* 2

def circumferinta(self):

return 2 \* pi \* self.raza

cerc1 = Cerc(8, 'verde')

cerc1.descrie\_cerc()

print(cerc1.aria())

print(cerc1.diametru())

print(cerc1.circumferinta())

cerc2 = Cerc(12, 'galben')

cerc2.descrie\_cerc()

print(cerc2.aria())

print(cerc2.diametru())

print(cerc2.circumferinta())

# Ex 2

class Dreptunghi:

def \_\_init\_\_(self, lungime, latime, culoare):

self.lungime = lungime

self.latime = latime

self.culoare = culoare

def descrie(self):

print(f'Acest dreptunghi are lungimea {self.lungime} cm, latimea {self.latimea} cm si culoarea {self.culoare}.')

def aria(self):

return self.lungime \* self.latime

def perimetru(self):

return 2 \* self.lungime + 2 \* self.latime

def schimba\_culoarea(self, noua\_culoare):

self.culoare = noua\_culoare

dreptunghi1 = Dreptunghi(10, 5, 'albastru')

dreptunghi1.descrie()

print(dreptunghi1.aria())

print(dreptunghi1.perimetru())

dreptunghi1.schimba\_culoarea('verde')

dreptunghi1 = Dreptunghi(18,11 , 'rosu')

dreptunghi1.descrie()

print(dreptunghi1.aria())

print(dreptunghi1.perimetru())

dreptunghi1.schimba\_culoarea('mov')

# Ex 3

class Angajat:

def \_\_init\_\_(self, nume, prenume, salariu):

self.nume = nume

self.prenume = prenume

self.salariu = salariu

def descrie(self):

print(f'Angajatul {self.nume} {self.prenume} are salariul {self.salariu}.')

def nume\_complet(self):

print(f'Numele complet al angajatului este {self.nume} {self.prenume}.')

def salariu\_lunar(self):

print(f'Salariul lunar este de {self.salariu}.')

def salariu\_anual(self):

return self.salariu \* 12

def marire\_salariu(self, procent):

return self.salariu \* procent / 100

angajat1 = Angajat('Pop', 'Andrei', 4000)

angajat1.descrie()

angajat1.nume\_complet()

angajat1.salariu\_lunar()

print(angajat1.salariu\_anual())

print(angajat1.marire\_salariu(10))

angajat2 = Angajat('Toman', 'Ionela', 7000)

angajat2.descrie()

angajat2.nume\_complet()

angajat2.salariu\_lunar()

print(angajat2.salariu\_anual())

print(angajat2.marire\_salariu(25))

# Ex 4

class Cont:

def \_\_init\_\_(self, iban, titular\_cont, sold):

self.iban = iban

self.titular\_cont = titular\_cont

self.sold = sold

def afisare\_sold(self):

print(f'Titularul {self.titular\_cont} are in contul {self.iban} suma de {self.sold} lei.')

def debitare\_cont(self, suma):

return self.sold - suma

def creditare\_cont(self, suma):

return self.sold + suma

cont1 = Cont('P16BTRL9182RI', 'Rus Ioan', 10000)

cont1.afisare\_sold()

print(cont1.debitare\_cont(2500))

print(cont1.creditare\_cont(7000))

cont2 = Cont('P25BTRL177084SM', 'Socol Mircea', 6000)

cont2.afisare\_sold()

print(cont2.debitare\_cont(1200))

print(cont2.creditare\_cont(3000))

# Ex 1 optional

class Factura:

seria = 'F1623'

def \_\_init\_\_(self, numar, nume\_produs, cantitate, pret\_buc):

self.numar = numar

self.nume\_produs = nume\_produs

self.cantitate = cantitate

self.pret\_buc = pret\_buc

def schimba\_cantitatea(self, cantitate):

self.cantitate = cantitate

def schimba\_pretul(self, pret):

self.pret\_buc = pret

def schimba\_nume\_produs(self, nume):

self.nume\_produs = nume

def genereaza\_factura(self):

tabel = [[f'Factura seria {Factura.seria} numar {self.numar}'],

[f'Data: {date.today()}'],

['Produs', 'cantitate', 'pret bucata', 'Total'],

[self.nume\_produs, self.cantitate, self.pret\_buc, self.cantitate\*self.pret\_buc]]

print(tabulate(tabel, stralign="center"))

factura1 = Factura(1, 'Telefon', 8, 1500)

factura1.schimba\_cantitatea(7)

factura1.schimba\_pretul(2000)

factura1.schimba\_nume\_produs('Tableta')

factura1.genereaza\_factura()

factura2 = Factura(2, 'scaun', 6, 499)

factura2.schimba\_cantitatea(3)

factura2.schimba\_pretul(2599)

factura2.schimba\_nume\_produs('pat')

factura2.genereaza\_factura()

# Ex 2 optional

class Masina:

marca = 'BMW'

viteza\_actuala = 0

culoare = 'gri'

culori\_disponibile = ('alb', 'negru', 'albastru', 'mov', 'rosu')

inmatriculata = False

def \_\_init\_\_(self, model, viteza\_maxima):

self.model = model

self.viteza\_maxima = viteza\_maxima

def descrie(self):

print('marca:', self.marca)

print('viteza actuala:', self.viteza\_actuala)

print('culoare:', self.culoare)

print('model:', self.model)

print('viteza maxima:', self.viteza\_maxima)

def inmatriculare(self):

self.inmatriculata = True

return self.inmatriculata

def vopseste(self, culoare):

if culoare in self.culori\_disponibile:

self.culoare = culoare

return f'Noua culoare a masinii este {self.culoare}.'

else:

return f'Eroare. Culoarea {culoare} nu este disponibila. Alegeti alta culoare.'

def accelereaza(self, viteza):

if viteza < 0:

return 'Eroare. Viteza trebuie sa fie mai mare decat 0.'

elif viteza > self.viteza\_maxima:

viteza = self.viteza\_maxima

return f'Masina accelereaza pana la {viteza} km/h.'

else:

return f'Masina accelereaza pana la {viteza} km/h.'

def franeaza(self):

self.viteza\_actuala = 0

return self.viteza\_actuala

masina1 = Masina('X5', 320)

masina1.descrie()

print(masina1.inmatriculare())

print(masina1.vopseste('galben'))

print(masina1.accelereaza(330))

print(masina1.franeaza())

masina1 = Masina('X7', 340)

masina1.descrie()

print(masina1.inmatriculare())

print(masina1.vopseste('albastru'))

print(masina1.accelereaza(280))

print(masina1.franeaza())

# Ex 3 optional

class TodoList:

todo = {}

def adauga\_task(self, nume, descriere):

self.todo[nume] = descriere

return self.todo

def finalizeaza\_task(self, nume):

self.todo.pop(nume)

return self.todo

def afiseaza\_todo\_list(self):

return self.todo.keys()

def afiseaza\_detalii\_suplimentare(self, nume\_task):

if nume\_task not in self.todo:

intreb\_user = input('Doriti sa adaugati taskul in todo list ? Raspundeti cu "da" sau "nu": ')

if intreb\_user == 'nu':

return 'La revedere'

else:

self.todo[nume\_task] = input('Introduceti detalii pentru noul task: ')

return self.todo

mylist = TodoList()

print(mylist.adauga\_task('cumparaturi', 'du-te la piata si ia fructe si legume'))

print(mylist.adauga\_task('curatenie', 'aspira si da cu mopu in toata casa'))

print(mylist.afiseaza\_todo\_list())

print(mylist.afiseaza\_detalii\_suplimentare('du masina in service'))

print(mylist.finalizeaza\_task('cumparaturi'))

myotherlist = TodoList()

print(myotherlist.adauga\_task('du gunoiul', 'diseara la ora 22'))

print(myotherlist.adauga\_task('spala masina', 'maine la 10 la Jetpoint'))

print(myotherlist.afiseaza\_todo\_list())

print(myotherlist.afiseaza\_detalii\_suplimentare('vizita medic familie'))

print(myotherlist.finalizeaza\_task('du gunoiul'))