from enum import Enum

from datetime import datetime

import pickle

class Type(Enum):

hatchback = "hatchback"

sedan = "sedan"

suv = "suv"

class Car:

def \_\_init\_\_(self, make, model, carID, price, carType):

self.\_make = make

self.\_model = model

self.\_carID = carID

self.\_price = price

self.\_carType = carType

# getters and setters

def getMake(self):

return self.\_make

def setMake(self, make):

self.\_make = make

def getModel(self):

return self.\_model

def setModel(self, model):

self.\_model = model

def getCarID(self):

return self.\_carID

def setCarID(self, carID):

self.\_carID = carID

def getPrice(self):

return self.\_price

def setPrice(self, price):

self.\_price = price

def getCarType(self):

return self.\_carType

def setCarType(self, carType):

self.\_carType = carType

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

return "Car ID: {}\nMake: {}\nModel: {}\nPrice: {}\nType: {}".format(

self.\_carID, self.\_make, self.\_model, self.\_price, self.\_carType.value)

def display\_car\_details(self, car\_id):

if car\_id == self.\_carID:

print(str(self))

else:

print("Car with ID {} not found".format(car\_id))

class Sale:

def \_\_init\_\_(self, employeeID, carID, salePrice):

self.\_employeeID = employeeID

self.\_carID = carID

self.\_salePrice = salePrice

# getters and setters

def getEmployeeID(self):

return self.\_employeeID

def setEmployeeID(self, employeeID):

self.\_employeeID = employeeID

def getCarID(self):

return self.\_carID

def setCarID(self, carID):

self.\_carID = carID

def getSalePrice(self):

return self.\_salePrice

def setSalePrice(self, salePrice):

self.\_salePrice = salePrice

class Person:

def \_\_init\_\_(self, firstName, lastName, gender, dateOfBirth):

self.\_firstName = firstName

self.\_lastName = lastName

self.\_gender = gender

self.\_dateOfBirth = dateOfBirth

# getters and setters

def getFirstName(self):

return self.\_firstName

def setFirstName(self, firstName):

self.\_firstName = firstName

def getLastName(self):

return self.\_lastName

def setLastName(self, lastName):

self.\_lastName = lastName

def getGender(self):

return self.\_gender

def setGender(self, gender):

self.\_gender = gender

def getDateOfBirth(self):

return self.\_dateOfBirth

def setDateOfBirth(self, dateOfBirth):

self.\_dateOfBirth = dateOfBirth

class Customer(Person):

def \_\_init\_\_(self, firstName, lastName, gender, dateOfBirth):

super().\_\_init\_\_(firstName, lastName, gender, dateOfBirth)

class JobTitle(Enum):

manager = "Manager"

salesperson = "Salesperson"

class Employee(Person):

def \_\_init\_\_(self, firstName, lastName, gender, dateOfBirth, employeeID, basicSalary, department, jobTitle, passportNumber):

super().\_\_init\_\_(firstName, lastName, gender, dateOfBirth)

self.\_employeeID = employeeID

self.\_basicSalary = basicSalary

self.\_department = department

self.\_jobTitle = jobTitle

self.\_passportNumber = passportNumber

self.\_sales = []

# getters and setters

def getEmployeeID(self):

return self.\_employeeID

def setEmployeeID(self, employeeID):cod

self.\_employeeID = employeeID

def getBasicSalary(self):

return self.\_basicSalary

def setBasicSalary(self, basicSalary):

self.\_basicSalary = basicSalary

def getDepartment(self):

return self.\_department

def setDepartment(self, department):

self.\_department = department

def getJobTitle(self):

return self.\_jobTitle

def setJobTitle(self, jobTitle):

self.\_jobTitle = jobTitle

def getPassportNumber(self):

return self.\_passportNumber

def setPassportNumber(self, passportNumber):

self.\_passportNumber = passportNumber

def getSales(self):

return self.\_sales

def addSale(self, sale):

self.\_sales.append(sale)

def calculate\_salary(self):

total\_sales = sum(sale.\_salePrice for sale in self.\_sales)

salesperson\_commission = total\_sales \* 0.065

manager\_commission = total\_sales \* 0.035

company\_profit = total\_sales - salesperson\_commission - manager\_commission

salary = self.\_basicSalary

if self.\_jobTitle == "Manager":

salary += manager\_commission

elif self.\_jobTitle == "Salesperson":

salary += salesperson\_commission

return salary

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

return "Employee ID: {}\nName: {} {}\nGender: {}\nDate of Birth: {}\nBasic Salary: {}\nDepartment: {}\nJob Title: {}\nPassport Number: {}".format(

self.\_employeeID, self.\_firstName, self.\_lastName, self.\_gender, self.\_dateOfBirth, self.\_basicSalary, self.\_department, self.\_jobTitle, self.\_passportNumber)

def display\_employee\_details(self, employee\_id):

if employee\_id == self.\_employeeID:

print(str(self))

else:

print("Employee with ID {} not found".format(employee\_id))

def display\_sales\_details(self, employee\_id):

if employee\_id == self.\_employeeID:

if len(self.\_sales) == 0:

print("This employee has no sales.")

else:

print("Sales details for employee ID {}:".format(self.\_employeeID))

for sale in self.\_sales:

print("Car ID: {}, Sale Price: {}".format(sale.\_carID, sale.\_salePrice))

else:

print("Employee with ID {} not found".format(employee\_id))