from orders import Order, Cart

from products import Cloth, Gadget

from users import User

u = User("Abduvali Rajabov", "abdushka@sfak.com",

"99 999 88 78", 1984, "M",

"Qo'qon shahar, Marifat 28")

u1 = User("Abdushukur Holmatov", "hheehee@sfak.com",

"91 988 09 78", 1981, "M",

"Andijon shahar, Shaytanat 128")

u2 = User("Toshtemir Toshov", "stone@sfak.com",

"91 008 09 07", 2001, "M",

"Toshkent, Toshko'mirchi 101")

p1 = Cloth("T-shirt 99", 23, "09.09.2023", u2, "Tommy Hilfiger", "L", "white")

p2 = Cloth("Jacket 101", 320, "09.09.2023", u2, "Tommy Hilfiger", "XL", "black", )

p3 = Cloth("Trousers ff4", 132, "09.09.2023", u2, "Fashion", "M", "khaki")

p4 = Cloth("T-shirt ", 48, "09.09.2023", u2, "Levi's", "XL", "cyan")

p5 = Gadget("iPhone 13 pro Max", 1200, "09.09.2022", u2, "Apple", "smartphone", "new")

p6 = Gadget("Galaxy S23 pro Max", 1100, "08.09.2022", u2, "Samsung", "smartphone", "new")

c1 = Cart(1, u, "10.10.2023", "pending")

c2 = Cart(2, u1, "10.11.2023", "pending")

o1 = Order(p1, 4, c1)

o2 = Order(p4, 2, c1)

o3 = Order(p6, 1, c1)

o4 = Order(p5, 2, c2)

o5 = Order(p2, 5, c2)

c1.get\_narx()

c2.get\_narx()

print("Foydalanuvchi daromadi: ", u2.get\_daromad())

class Cart:

def \_\_init\_\_(self, id, user, sana, holat):

self.\_\_id = id

self.\_\_user = user

self.\_\_sana = sana

self.holat = holat

self.\_\_narx = 0

self.\_\_bonus = 0

def get\_narx(self):

print(f"Narx:{self.\_\_narx}\nBonus:{self.\_\_bonus}")

return self.\_\_narx

def set\_narx(self, a):

self.\_\_narx += a-a/20

self.\_\_bonus += a/20

class Order:

def \_\_init\_\_(self, product, amount, cart):

self.product = product

self.amount = amount

self.cart = cart

self.narx = product.\_narx \* amount

cart.set\_narx(self.narx)

product.\_user.set\_daromad(self.narx)

class Product:

def \_\_init\_\_(self, nom, narx, sana, user):

self.\_nom = nom

self.\_narx = narx

self.\_sana = sana

self.\_user = user

class Cloth(Product):

def \_\_init\_\_(self, nom, narx, sana, user, brend, size, rang):

super().\_\_init\_\_(nom, narx, sana, user)

self.\_brend = brend

self.\_size = size

self.\_rang = rang

class Gadget(Product):

def \_\_init\_\_(self, model, narx, sana, user, brend, tur, holat):

super().\_\_init\_\_(model, narx, sana, user)

self.brend = brend

self.tur = tur

self.holat = holat

class User:

def \_\_init\_\_(self, fish, email, telefon, yil, jins, manzil):

self.\_\_manzil = manzil

self.\_\_jins = jins

self.\_\_yil = yil

self.\_\_telefon = telefon

self.\_\_email = email

self.\_\_fish = fish

self.\_\_daromad = 0

def set\_daromad(self, a):

self.\_\_daromad += a

def get\_daromad(self):

return self.\_\_daromad

def get\_manzil(self):

return self.\_\_manzil

def get\_jins(self):

return self.\_\_jins

def get\_email(self):

return self.\_\_email

def get\_telefon(self):

return self.\_\_telefon

def get\_fish(self):

return self.\_\_fish

def get\_yil(self):

return self.\_\_yil

# class University:

# def \_\_init\_\_(self, ism):

# self.ism = ism

# self.departments = []

# def add\_department(self, department):

# self.departments.append(department)

# class Department:

# def \_\_init\_\_(self, ism):

# self.name = ism

# self.professors = []

# self.students = []

# def add\_professor(self, professor):

# self.professors.append(professor)

# def add\_student(self, student):

# self.students.append(student)

# class Professor:

# def \_\_init\_\_(self, ism):

# self.name = ism

# self.department = None

# self.courses\_taught = []

# def assign\_department(self, department):

# self.department = department

# def add\_course(self, course):

# self.courses\_taught.append(course)

# def assign\_grade(self, student, course, grade):

# student.add\_grade(course, grade)

# class Student:

# def \_\_init\_\_(self, ism):

# self.ism = ism

# self.department = None

# self.courses\_taken = []

# self.grades = {}

# def assign\_department(self, department):

# self.department = department

# def add\_course(self, course):

# self.courses\_taken.append(course)

# def add\_grade(self, course, grade):

# self.grades[course] = grade

# class Course:

# def \_\_init\_\_(self, ism):

# self.ism = ism

# university = University("ABC University")

# department1 = Department("Computer Science")

# department2 = Department("Mathematics")

# university.add\_department(department1)

# university.add\_department(department2)

# professor1 = Professor("John Doe")

# professor1.assign\_department(department1)

# department1.add\_professor(professor1)

# professor1.add\_course(Course("Dasturlash faniga kirish"))

# student1 = Student("Alice Smith")

# student1.assign\_department(department1)

# department1.add\_student(student1)

# student1.add\_course(Course("Dasturlash faniga kirish"))

# professor1.assign\_grade(student1, "Dasturlash faniga kirish", "A")

# print("Student:", student1.ism)

# print("Grade in Introduction to Programming:", student1.grades["Introduction to Programming"])

# class Character:

# def \_\_init\_\_(self, name, health):

# self.name = name

# self.health = health

# def take\_damage(self, damage):

# self.health -= damage

# if self.health <= 0:

# print(f"{self.name} has been defeated!")

# def attack(self, target):

# print(f"{self.name} attacks {target.name}!")

# def interact(self):

# print(f"{self.name} interacts with the game environment.")

# class PlayerCharacter(Character):

# def \_\_init\_\_(self, name, health):

# super().\_\_init\_\_(name, health)

# self.inventory = []

# def pick\_up\_item(self, item):

# self.inventory.append(item)

# print(f"{self.name} picks up {item}!")

# def use\_item(self, item):

# if item in self.inventory:

# self.inventory.remove(item)

# print(f"{self.name} uses {item}!")

# else:

# print(f"{self.name} doesn't have {item} in their inventory!")

# class NonPlayerCharacter(Character):

# def \_\_init\_\_(self, name, health, quest):

# super().\_\_init\_\_(name, health)

# self.quest = quest

# def give\_quest(self):

# print(f"{self.name} gives a quest: {self.quest}!")

# class Item:

# def \_\_init\_\_(self, name):

# self.name = name

# class Battle:

# def \_\_init\_\_(self, player, enemy):

# self.player = player

# self.enemy = enemy

# def start\_battle(self):

# print(f"A battle begins between {self.player.name} and {self.enemy.name}!")

# while self.player.health > 0 and self.enemy.health > 0:

# self.player.attack(self.enemy)

# self.enemy.attack(self.player)

# print("The battle ends!")

# player = PlayerCharacter("Player 1", 100)

# enemy = NonPlayerCharacter("Enemy 1", 50, "Defeat 10 monsters")

# player.interact()

# enemy.give\_quest()

# battle = Battle(player, enemy)

# battle.start\_battle()

# sword = Item("Sword")

# player.pick\_up\_item(sword)

# player.use\_item(sword)