```python
from enum import Enum
#####################################################################################################
#                                              Account class                                        #
#####################################################################################################
class Account:
    def __init__(self, username, password, email, phone_number):
        self.__username = username
        self.__password = password #must be hash before store
        self.__email = email
        self.__phone_number = phone_number

class Admin(Account):
    __product_catalog = []

    def __init__(self, username, password, email, phone_number, product_catalog, coupon, promotion, order):
            Account.__init__(self, username, password, email, phone_number)
            self.__product_catalog = product_catalog # ProductCataog object
            self.__coupon = coupon # Coupon object
            self.__promotion = promotion # Promotion object
            self.__order = order # Order object (specific user)
    #some method for managing our web


class User(Account):
    def __init__(self, username, password, email, phone_number, person_data, address, cart, order,
 order_history, coupon):
        Account.__init__(self, username, password, email, phone_number)
        self.__person_data = person_data
        self.__address = [] # List of Shipping_Address Object
        self.__cart = cart # Cart object
        self.__order_history = order_history # OrderHistory object
        self.__coupons = [] # List for store Coupon object

class Shipping_Address:
    def __init__(self, name_surname, phone_number, address, sub_district, district, province, postal_code):
        self.__name_surname = name_surname
        self.__phone_number = phone_number
        self.__address = address
        self.__sub_district = sub_district
        self.__district = district
        self.__province = province
        self.__postal_code = postal_code
#####################################################################################################
#                                              Product class                                        #
#####################################################################################################
class Product:
    def __init__(self, type, product_id, option, brand, image, price, quantity, detail, promotion):
            self.__type = type
            self.__product_id = product_id
            self.__option = option
            self.__brand = brand
            self.__image = image # Path to the image source
            self.__price = price
            self.__quantity = quantity
            self.__detail = detail
            self.__promotion = promotion # List of Promotion Object

class Item(Product):
    def __init__(self, type, product_id, option, brand, image, price, quantity, detail, amount):
         Product.__init__(self, type, product_id, option, brand, image, price, quantity, detail)
         self.__amount = amount

class ProductCatalog:
     def __init__(self, product):
            self.__products = [] # List of Product object

#####################################################################################################
#                                    MarketPlace Element class                                      #
#####################################################################################################
class Promotion:
     def __init__(self, duration, description):
        self.__duration = duration
        self.__description = description

class FlatDiscount(Promotion):
    def __init__(self, duration, description, discount):
        Promotion.__init__(self, duration, description)
        self.__discount = discount

class PercentageDiscount(Promotion):
    def __init__(self, duration, description, discount_percent):
        Promotion.__init__(self, duration, description)
        self.__discount_percent = discount_percent


class Coupon:
     def __init__(self, quantity, duration, code_id, discount, description):
        self.__duration = duration
        self.__discount = discount
        self.__description = description
        self.__quantity = quantity
        self.__code_id = code_id

class Cart:
     def __init__(self, items, total_price, coupon):
        self.__items = [] # List for store Item object
        self.__total_price = 0
        self.__coupon = coupon # Coupon object (use a coupon for discount)

class Order:
     def __init__(self, order_date, delivery_expect_date, payment_method, tracking_number, total_price,
order_id, status):
        self.__order_date = order_date
        self.__delivery_expect_date = delivery_expect_date
        self.__payment_method = payment_method
        self.__tracking_number = tracking_number
        self.__total_price = total_price
        self.__order_id = order_id
        self.__status = status

class OrderHistory:
     def __init__(self, orders):
          self.__orders = [] # list for store Orders

#####################################################################################################
#                                              Payment class                                        #
#####################################################################################################
class Payment:
     def __init__(self, transaction_id, status, create_on):
          self.__transaction_id = transaction_id
          self.__status = status
          self.__create_on = create_on

class CreditCardTransaction(Payment):
     def __init__(self, transaction_id, status, create_on, name_on_card, card_id, CVC, due_date):
          Payment.__init__(self, transaction_id, status, create_on)
          self.__name_on_card = name_on_card
          self.__card_id = card_id
          self.__CVC = CVC
          self.__due_date = due_date

class CashOnDeliveryTransaction(Payment):
     def __init__(self, transaction_id, status, create_on):
          Payment.__init__(self, transaction_id, status, create_on)

class Cashier:
     def __init__(self, net_price, payment_method, payment):
          self.__net_price = net_price
          self.__payment_method = payment_method # PaymentMethod object
          self.__payment = payment # Payment object

class PaymentMethod(Enum):
     CreditCardTransaction, CardOnDelivery = 1, 2
```