Python OOP - Second Exam - Moed A

Customer’s Order Management Software System

Your mission is to create a software system that implements a customer’s orders management system.
Your system should handle the orders of the website customers.

Your system should support two different types of orders:

- Regular order
- VIP order

Each Order has the following properties:

- Id
- name
- delivery address
- List of items in the order
- order customer
- order total price
- payment type (CREDIT CARD / CASH / CHECK / OTHER)
- order date

The order id will be unique for each order, we can’t have two orders with the same order id.

At the constructor each order will call a function to calculate the order total price. The order total price will be calculated by the sum of the prices of all the items inside the items list.

In addition, if the order is VIP order then the total order price should get a discount according to the discount specified in the customer discount property.
If the order is a regular order the total price will be without any discount.
Before calculating a VIP order total price you should check that the customer that made the order is indeed a VIP customer.
If it’s not a vip customer but the order is a VIP order your system should throw an error.

Each Order Item has the following properties:

- Id
- item name
- item price

The item id will be unique for each order, we can’t have two items with the same item id.

Each Customer has the following properties:

- Id
- first name
- last name
- email
- delivery address
- customer type (REGULAR / VIP)
- customer discount (can be null)
- List of favorite items
- customer gift

The customer id will be unique for each order, we can’t have two customers with the same customer id.

Whenever a customer creates a new order your system should automatically add all the items in that order to the customer favorite items list. If the item already exists in this list you should not add it again (items with the same name but with different id are the same items).

In addition, each customer can update his favorite items list meaning he can add more items to the list or remove items he doesn't want.

Your system should support the ability to give a customer gift to customers.
Customer gift can be of any type but must have the following function:
open_gift() → when calling this function your system should print:
“Congratulations! you got a new gift! Enjoy!”

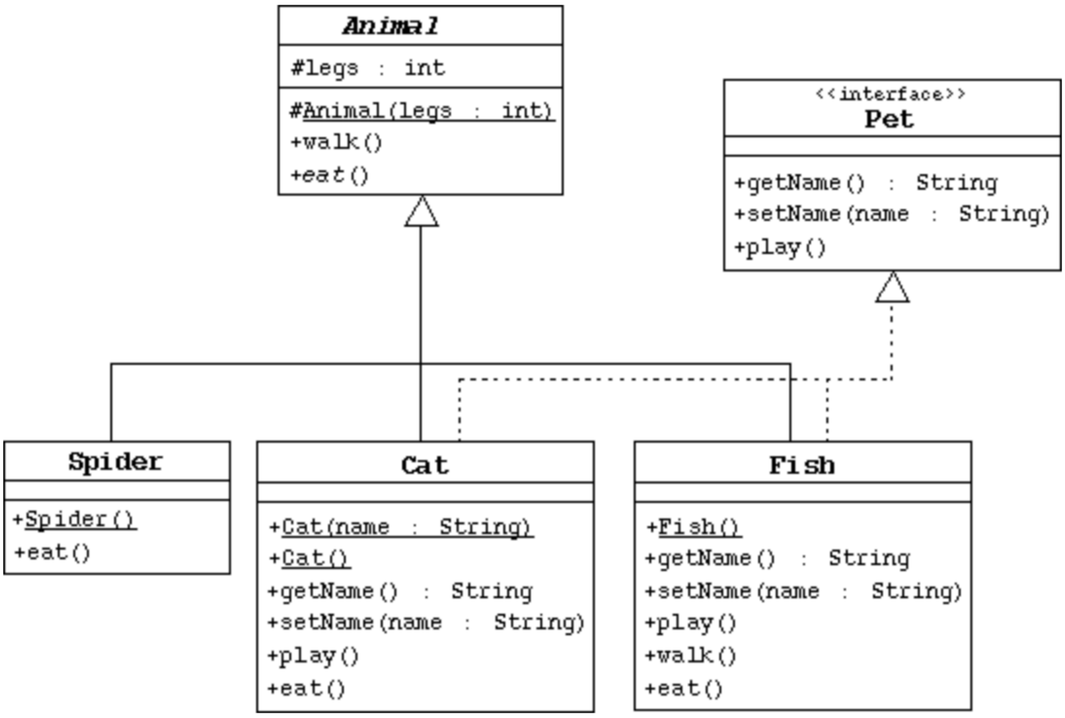
Your system should support the ability to give gifts to customers by calling the function take_gift(gift)

In addition the customer should have the ability to open the gift if he has one by calling the gift open_gift() function.

Bonus (10 points)

Implement the following software system according to the UML diagram below.

Use inheritance / composition / interfaces and abstract classes wherever you need to.



Good Luck :)