

Order management task

Background :

- we have a solution that manages order status and print order receipt, we have 5 statuses: pending, accepted, processing, delivering, receiving, and rejected.
- We have a product class that manages product details and calculates the price based on the attributes.
- we have a shipping class that manages the shipping method and calculate the tax based on the countries and the method type.
- we have user and address classes that interact with our solution.

Your assessment :

- re-factoring this code (restructure this design, change names, separate the responsibility, add new classes if you need that).
- we need to extend the statuses, shipping methods and countries so we need a mechanism to do that in our solution.
- we need to print the receipt with another format like XML, JSON, and so on to interact with another system.
- we need to handle the exceptions.
- use design patterns and any data structure you want but avoid over-engineering or making a lot of layers of abstraction.

Our expectation :

- you know PHPUnit if you don't know it, Google it.
- you are using PHP latest version .
- write a unit test to cover this code.
- when you finish this task upload it in Google drive and send the link for it and not upload it in github .
- write readme file to explain your effort.

Our Evaluation :

- the testOrderHaveReceipt test must be the same without any change.
- We make sure you did not change existing plain text output in any way.
- We evaluate the cleanliness of your code, in terms of the size and tidiness of classes and methods, the quality of the names of your variables, classes, methods, the accuracy and usefulness of the comments if they are found, etc...

- We evaluate your design approach, how you used OOP to your advantage, how you use patterns and abstractions and whether they add value or are just there to show off your knowledge.
- We evaluate your JSON representation for proper understanding of JSON nature and how it represents data elements.

Bonus :

- use docker.
- add cancelled and returned status.
- add a new shipping method called (fetcher).
- add XML formatting to represent the receipt.