Homework #2 - Praxis 2022 - 1

Made by: Team 6: Camilo Funez y Andres Felipe Gaviria Lora

Good implementations:

In general: The project follows the conventions of “camelCase” which is recommended for Java, the use of functions with specific and unique task.

Class Item: In this class, we identify: High cohesion, because every method and parameter that it has, feels proper to the entity "item" in the context of the project. Low coupling, because the functions of the code, though related, can work and can be modified without involving the others.

Changes made:

1. Using polymorphism, we inherit from the class “Item”, having in mind the “Open Closed Principle” and create a new class for every type of item (“ItemNormal”, “ItemAged”, “ItemTicket” and “ItemLegendary”). In specific, every class overrides the method “updateQuality” using it in his very own way in order to work correctly for the specific cases of every type.

2. In each implementation of “updateQuality” we tried to write conditions more legible and avoid nested ones, using Guard Clauses, at the cost of making some lines of code a little longer.

3. We wanted to accomplish the principle of Dependency Inversion” by making the class “ItemService” a class that represents a schema for the other item classes.

4. We did an effort to follow the Liskov Substitution Principle, modifying the class item in a way that making changes in its children classes, does not affect the intent of the class.

5. Also, we acknowledge that the place of “updateQuality” in the definition of the class is an improvement for cohesion, given the relationship between its purpose and the class.

6. Functions and class names are intended to be self-explanatory.