UNIVERSIDADE DA CORUÑA

Software Design

Design Assignment (2022-2023)

INSTRUCTIONS: **Deadline: December 16, 2022 (until 23:59)**.

- **The exercises must be solved applying design principles and patterns. A solution that does not use them will be considered incorrect.**
- **Report**: For each exercise you must produce a report that includes the following:
  - Explanation of the **design principles** you have used (particularly the **SOLID** principles) and exactly where they have been used (name the specific classes from your code).
  - Explanation of the **design pattern(s)** you have used. For each pattern you will include the following:
    - **Brief explanation of the chosen pattern** and the rationale for its use.
    - **Class diagram** where the classes involved in the pattern are shown. It is important to indicate the role played by each class in the diagram itself, with UML annotations.
    - **Dynamic diagrams** (sequence, communication or state-machine) showing the dynamic behavior of fundamental aspects of the code. You must decide what type of diagram is more suitable for each exercise.
- **Code and submission**:
  - You must upload a single IntelliJ IDEA project to your repository in a folder whose name is your group followed by the `-DA` suffix. For example, `DS-11-01-DA`.
  - A package will be created for each exercise with the names: `e1` and `e2`..
  - You must include unit tests and check the code coverage.
  - The documentation with the design and the principles and patterns will be submitted as a `PDF` file in a `doc` folder inside the IntelliJ project.
- **Evaluation**:
  - This assignment represents 1/3 of the final practicum grade. The documentation and the code will be evaluated based on the following criteria:
  - **Quality of documentation**: suitableness of the chosen patterns and principles, clarity of the explanations, quantity and clarity of the submitted diagrams, match between design and code, etc.
  - **Quality of code**: correct implementation of design patterns and principles, correct application of the object-oriented philosophy, match between design and code, validity of the tests, etc.

1. **Online shopping system**

   We propose an online shopping system with the following steps:

   - *ShoppingCart*: You may add items to the shopping cart as long as there is enough stock available for the chosen product. You can also remove added items from the cart. The moment you want to finalize the purchase you can do a *check-out* to finish the shopping
   - *CheckOut*: At this point the client can go back to the previous step and keep managing the cart (adding more items). Alternatively, and in order to avoid confusion, the system lets you remove the items from the previous step or change the quantities (assuming the stock allows it).
   - *Payment*: After choosing the products and their quantities, the client pays for the order, which is now considered confirmed/paid.
   - *Cancelled*: After the payment, the system offers a window of 24 hours to cancel it.
   - *Completed*: After the payment, if 24 hours have passed and no cancellation has been made, the order begins to be prepared and is considered completed.

   The class `Product` must include, among others, a "stock" attribute to know the number of items available. All the operations that affect the stock of a product must update it accordingly.

   The class `Order` will have a `screenInfo()` method that offers the basic information for the application. Next, we indicate the information that must be displayed for each phase the order goes through, and the chosen format. Additionally, some concrete examples are provided.

```
        * When starting a new order
        Example: Order Number: 1111
        Phase: Shopping -- Welcome to online shop

        * ShoppingCart, CheckOut: number of products in the cart
        Example 1: Order Number: 1111
        Phase: Shopping -- 3 products
        Example 2: Order Number: 1111
        Phase: Chek Out: 4 products

        * Payment order: number of products in the cart, time of the order.
        Example: Order Number: 1111
        Phase: Paid Order: 4 products -- date 2022-10-31 19:06:37

        * Completed order: number of products in the cart
        Example: Order Number: 1111
        Phase: Completed Order: 4 products

        * Cancelled o Completed order
        Example: Order Number: 1111
        Phase: Cancelled Order
```

   In addition, each event must be recorded (a change in the cart, a change of phase in the order, etc.). This way, a *log* is obtained with all the information about the order. Considering the specified format for recording the events, an example of said *log* would be as follows:

```
    Order 1111: Shopping Phase
    - Add: Item: 111 - Quantity: 10 -> Shopping Cart -- Products : 1
    - Add: Item: 222 - Quantity: 2 -> Shopping Cart -- Products : 2
    - Add: Item: 333 - Quantity: 3 -> Shopping Cart -- Products : 3
    - Add: Item: 444 - Quantity: 4 -> Shopping Cart -- Products : 4
    - Remove: Item: 444 -> Shopping Cart -- Products : 3
    - Add: Item: 444 - Quantity: 5 -> Shopping Cart -- Products : 4
    Order 1111: Check Out Phase
    - Modify: Item: 111 - Quantity: 1 -> CheckOut Order -- Products: 4
    Order 1111: Payment Phase
```

In the future, the problem might be expanded to include the process of delivery. That is, knowing whether the order has been delivered to the client or to an intermediary (postal system) who is waiting for the client to pick it up; or whether it has been rejected, lost in transit, etc.

Develop a solution based on design principles and patterns for suitably representing each order and its phases and the changes between them, as well as facilitating the addition of new phases in the future to our basic online shopping system.

2. **Management of alerts on the parameters of the tanks of an aquarium**

   You must develop a new system for managing alerts on the parameters of the tanks of the Aquarium Finisterrae of A Coruña.

   Each tank contains several sensors. Each sensor measures a single numeric parameter periodically. The sensors measure properties like the level of oxygen, the PH or the temperature. Changes in these parameters will be simulated via setters in the corresponding classes.

   For each sensor you can configure a series of alerts, which can be of two types: orange and red. An alert is assigned to a single sensor, even though a sensor may have multiple alerts assigned to it. Each alert specifies two numeric ranges. The first range indicates the normal values for the parameter measured by the sensor. If the value remains within this range, the alert is not fired. When the value goes outside the accepted range, the orange alert is activated. The second range, which is wider than the first, specifies the minimum and maximum levels for the orange alert. If the value goes outside that range, the red alert is activated.

   The Aquarium's tanks have control devices that react to these situations. They are associated with a series of alarms that, whenever an orange or red alarm is activated, can take action immediately.

   Besides, there are a number of people whose task is to take care of the tanks. Each person can subscribe to a specific set of alerts. When an orange or red alert is fired, a report is added for the subscribed personnel. The reports are stored in a priority queue (depending on the type of alert) so they can be dealt with during the maintenance tasks. Each report contains the following information: type of alert (red or orange), name and location of the tank, name of the alert, name of the parameter, level of the parameter and date and time of the alert. An alert is only communicated if there is a change of state, except when it is disabled, in which case it is not communicated. Below we show a possible example of a report:

```
Alerts of Maintenance Seals
RED Alerts:
* RED Alert:
Seals Swimming Pool, Outside
Oxygen control seals: parameter Oxygen, level -1,000
17:08:09 11/11/2022

ORANGE Alerts:
* ORANGE Alert:
Seals Swimming Pool, Outside
Oxygen control seals: parameter Oxygen, level 4,000
17:08:09 11/11/2022

* ORANGE Alert:
Seals Swimming Pool, Outside
Oxygen control seals: parameter Oxygen, level 3,000
17:08:09 11/11/2022
```

   The system must be designed to easily include new sensors that fire alerts (e.g. water level) and new elements that receive said alerts and react to them.

   Develop a solution, based on design principles and patterns, that allows to solve efficiently the problem.