SE311 SPRING 2023-2024 / 15-03-2024
Lab Work 4 – Composite Pattern
Goal: Practice on Composite Pattern

HepsiTrendy11 is an e-commerce platform where users can look for items from different categories and add them to their basket to be bought.

The menu of the shop contains different categories that may have also categories inside. For instance; Electronics category has TV's and PC's but Fashion category has two sub-categories: men and women.

In this lab, you will represent this hierarchical menu by using composite pattern. Below you can find explanations on the classes of this hierarchy. However, "which class is which participant?" question is yours to discover.

HepsiTrendy11: This is the interface that defines the common operations in the menu.

Product: This is a concrete implementation of the HepsiTrendy11 interface that represents a single product they sell. Each product has a name, price, and description. The class also implements operations such as display() to print its details.

Category: This is an implementation of the HepsiTrendy11 interface that represents a category of products on the menu. It contains a list of child components and implements operations such as "add" and "remove". As it is mentioned previously a category can contain other categories and products, forming a hierarchical structure.

You may start from scratch or you may use the Composite Pattern sample code from Blackboard as a starting point. Composite Pattern sample code is very similar to this structure and it may be comfortable for some of you to do this lab by modifying the sample code.

1) After you implement the system, you should be able to create a menu and print everything in the menu. To display, you can utilize the display() method from Composite Pattern sample code. You may use the data from second page to create a menu.

2) Next, please add a method called "*boolean find(String)*". This method will take a name as parameter and will traverse the hierarchy to find the product or category with that name and returns "true" OR "false" if it does not exist.

3) Next, add another method called "*int totalPrice(String)*". It will take a category name and returns the total price of products under it. First, it must check whether the category exists in the hierarchy or not.

4) Furthermore, we want to make sure that nobody could add anything to a specific product. To achieve this, use "Safety" implementation discussed in class.

Lastly, in your main function test your code. Create the menu with given data below and display it. Then, test your *find* function by calling it once with "Entrees" parameter and once more with "Soft Drinks" parameter.

EXAMPLE SHOP MENU (“+” Category, “-“ Single Product)

+Electronics
    +TV
        -OLED TV
        -QLED TV
    +PC
        -RAM
        -SSD
+Fashion
    +Men
        -Suit
    +Women
        -Shirt
        -Skirt
+Outdoor
    -Tent

+Cosmetics
    +Skin Care
        -Face Cream
        -Sun Protector
    -Shampoo
    -Parfum