Simple E-commerce Cart System

Code Walkthrough

1. Product Hierarchy:

- The code starts with a hierarchy of products, including a base class Product, and two subclasses Laptop and Headphones.
- This demonstrates inheritance, where Laptop , Headphones and mobile inherit attributes and methods from the base Product class.

2. Discount Strategies:

- The DiscountStrategy interface defines the contract for discount strategies, and two concrete implementations (PercentageOffDiscount and BuyOneGetOneFree) provide different ways to apply discounts.
- The PercentageOffDiscount applies a percentage off the original price, considering the quantity.
- The BuyOneGetOneFree strategy implements a simple "Buy One Get One Free" discount.

3. Shopping Cart and Cart Items:

- The CartItem class represents an item in the shopping cart, containing a product and its quantity.
- The ShoppingCart class manages a list of CartItem instances and has a reference to a DiscountStrategy. It uses the strategy to calculate the total bill.
- The shopping cart supports adding, updating, and removing items, and it displays the current items in the cart.

4. Main Class (ECommerceCartSystem):

- The main class sets up products (laptop ,headphones, mobile) and displays them with their attributes.
- It uses a ShoppingCart instance with a selected discount strategy (PercentageOffDiscount with 0% off initially).
- A while loop allows the user to interact with the system by selecting various options (add item, update quantity, remove item, view cart, calculate total, exit).

5. **User Input and Switch Case:**
    - The code takes user input using a Scanner to select options.
    - The switch case handles different user options and interacts with the ShoppingCart accordingly.
6. **Dynamic Product Management:**
    - Products are managed dynamically using collections (ArrayList and LinkedHashMap), making it easy to add, update, or remove products from the system.
7. **Code Organization:**
    - Code is organized into classes and methods, promoting encapsulation and maintainability.
8. **Error Handling:**
    - Code includes some basic error handling, such as checking if a product exists before adding, updating, or removing it from the cart.
9. **Loop and Exit:**
    - Used a while loop to allow continuous interaction until the user chooses to exit.

****

- YADDALA PAVAN KUMAR REDDY