Object-Oriented Programming Lab#6, Spring 2022

Today's Topics

- Class/Object, Constructor,
- package
- Array (Reference Type)
- ArrayList
- JOptionPane

ArrayLIst:

Action	**Code**
Creating an ArrayList	ArrayList<T> list = new ArrayList<T>();
Adding element to arraylist	list.add(T);
Accessing an element	List.get(int index)
Size of arraylist	list.size();

Online Store- Problem Description

Develop an application for an online store which will help **a store owner** to keep the record of its items/items and run the business. The store contains different types of items e.g., **Food items, clothing, electronics** and many more. For simplicity, you can work with just 3 types of items as mentioned in previous line. Each **item** has some common characteristics e.g., ***name, id, category, price*** and ***quantity***. Like all other online stores, you need to implement the following functionalities in your application.

1. User can browse through the items – View all items.
2. User can view the list of items of specific category e.g., all food items, all clothing etc.
3. User can view the details of a specific item.
4. Add item to store
5. Sell an item

Implementation

Note: The following design is just one possible option. You can modify it according to your need. You are free to add additional attributes, methods.

Possible classes:

1. **Item (**under store package**)**

   - Attributes: **name, id, category, price, quantity**
   - Constructor
   - Method:
     - public void updatePrice(double ***newPrice***)
       - update the ***price*** to ***newPrice***
     - display() or toString()

2. **Shop** (under store package)

   - Attributes: name, ArrayList<Item> **items** or Item array
   - Constructor
   - Method:
     - public void addItem(String **name,** String **id,** String **category,** double **price,** int **count**)
       - Search for the item in the **items** array using the **id**. If the item is found increase the **quantity** attribute of the product by the **count** amount. If the item is not available in the list, create an **Item** object using the parameters and add the object to the array/arraylist **items**.
     - public void viewItem(String id)
       - Search for the item in the **items** array using the **id**. If the item is found, call the display or print.
     - public void viewItems()
       - call the **display**() method for each item in the **items** array/arraylist.

- public void viewItems(String category)
    - Search for **all the item**s in the **items** array using the **category**. Call display method for all those items.
- public void sellAnItem(String id, int quantity)
    - search for the item in "***items***" array using the **id**. If the item is found in the list, **quantity** will decrease. If the item is not found, a message should display

3. ShopApp (under store.app package**)**

- Add Main method. Inside the main, create an object of **Shop** class and then provide the following **menu** and call appropriate method using the **Shop** object. Take input from users as needed.
    (a) 1 to view all
    (b) 2 to view specific category
    (c) 3 to view a specific item
    (d) 4 to add item
    (e) 5 to sell item
    (f) 0 to exit