Lab07

IT618 Enterprise Computing

PM Jat, DA-IICT, Winter'2017

In this lab we learn to create Data Access Objects.

1. A skeleton DAO for **`InventoryItem`** class of Lab04 is given below. Implement the same. Suggested Relation Design is also given.

   Complete the implementation of **`InventoryItemDAO`** class.

```
package pjava.jdbc.dao;

import pjava.inventory.InSufficientStock;
import pjava.inventory.InventoryItem;
import pjava.inventory.ItemNotFound;
import pjava.inventory.ItemExists;

public class InventoryDAO {

    InventoryDAO() {
    }
    public void InsertItem(InventoryItem items)
                              throws ItemExists, DAOException {
    }
    public void InsertItems(InventoryItem[] items)
                              throws ItemExists, DAOException {
        //should commit only if all items are successfully inserted otherwise should rollback
    }
    public void updateItem(InventoryItem items)
                             throws ItemNotFound, DAOException {
    }
    public void upDateItems(InventoryItem[] items)
                             throws ItemNotFound, DAOException {
        //should commit only if all items are updated otherwise rollback
    }
    public InventoryItem getItem(long item_code)
                             throws ItemNotFound, DAOException {
    }
```

continued …

```
    public InventoryItem[] getItems() throws DAOException {
    }
    public void setPageLength(int length) {
    }
    public InventoryItem[] getPaginatedItems(int page_no)
                              throws DAOException {
    }
    public InventoryItem[] getItemsCategory(int category)
                              throws DAOException {
    }
    public void addStock(int item_code, int qty)
                             throws ItemNotFound, DAOException {
    }
    public void withdrawStock(int item_code, int qty)
            throws ItemNotFound, InSufficientStock, DAOException {
    }
    public InventoryItem[] itemsUnderStock() throws DAOException{
    }
    public double totalInventoryCost() throws DAOException {
    }
    public void deleteItems(int[] items_codes)
                           throws ItemNotFound, DAOException {
        //should commit only if all items are delleted otherwise rollback
    }
    public void deleteItem(int item_code)
                           throws ItemNotFound, DAOException {
    }
    private int page_length = 20;
}
```