Lab04 - "Save an Order" using JDBC

[IT618, Enterprise Computing, Autumn'23]

Instructor: PM Jat (pm_jat@daiict.ac.in)

In this lab, let you implement a functionality of `"Save an Order"` using JDBC. Use the same `DBCon.getConnection()` from https://github.com/pmjat/j2ee/blob/master/j2ee/jdbc-demo/src/main/java/pmj/jee/jdbc/DBCon.java for establishing DB connection (you may, however, require to change `DBConfig` as per your database)

Suppose you are given data of an Order in a JSON file `order.json`. The same has been pasted below for your input.

File: `order.json`

```
{
  "order": {
    "orderNo":123,
    "customer": {
      "id": 1,
      "name": "Mayank",
      "email": "mayank@gmail.com",
      "billingAddress": {
         "address": "A-23, Sector 2",
         "city": "Ahmedabad",
         "pin": "382007"
        }
    },
    "orderItems":[
      {
        "productId":27,
        "price": 325,
        "qty":1
      },
      {
        "productId":19,
        "price": 550,
        "qty":2
      },
      {
        "productId":21,
        "price": 999,
        "qty":1
      }
    ],
    "shippingAddress": {
         "address": "F703, Sky Height, Gayathri Nagar",
         "city": "Bangalore",
         "pin": "560021"
    },
    "payment_txnId": "abelif879rft"
  }
}
```

Let you do the following here:

1. Create class `Customer` with required fields of the customer (as per in json file)
2. Create class `Address` with required fields of address (as per in JSON file)
3. Create class `OrderItem` with required fields of Order Item (as per in JSON file)
4. Create a class `Order` as an aggregation of Customer, Address, List of Order Items, and other fields as shown in JSON file.

Let us have the following tables: `Customer`, `Order`, `OrderItem`, and `Product`
Product table stores attributes: `ID`, `Description`, `Rate`, `QtyInStock`. You can assume that `ProductID` in the order item is an existing product id in the `Product` table.

Your job now is to implement a DAO for order. Let the DAO have the following interface having only two functions `save` and `read`.

```
class OrderDAO {
    void saveOrder(Order order) { ... }
        //this function saves the parameter order.
    Order readOrder(int order_no) { ... }
        //this function reads all order data from respective
        //tables and build the corresponding `Order` object and returns.
        //If supplied order_no is no existent then it returns `null`.
}
```

Now you also create a tester class that does following:

1. Read in the order from the given JSON file.
2. Save the order into the database by calling `saveOrder` function
3. Read the same order from the database using `readOrder`
4. Print the Order.

PUT everything in package `jee.lab04`

Deliverables of this exercise: Source Code of required classes and tester class.