Hands-On for Java SE using JDBC

Vehicle Service Station

Aim

- To develop a console-based and GUI application for a vehicle service station /garage.
- To provide various services to customers, prepare bills and track business by service station.

Technologies used

- Java Programming:
  - Class & Objects, Inheritance & Polymorphism, Composition / Association / Aggregation
  - Java Collections, Serialization, File Io, swing

Duration

- Console Based Implementation: 30 Hours
- GUI & Integration with classes: 8 Hours

Description

- Vehicle service station provide following services to customer:
  - Maintenance/Repairing:
    - Customer can ask to repair one or more areas e.g. dash board, brake system, etc.
    - Each repairing may need to change few parts e.g. dash board repairing may need to change speedo-meter cable, dash board dial, few indicators on dash board, etc.
    - For each repairing some labor charges might be applicable other than pricing of replaced spare parts.
    - Maintenance can also contain just fitting/tightening few screws, oiling of some parts for better performance; but not replacing any spare part. In this case only labor charges are applicable.
  - Engine/Gear Oil change:
    - Occasionally customer may ask or service engineer may advice to change engine and/or gear oil for better performance. Example, Servo4T, Castrol, etc. The engine oil is packaged in quantity of 900ml/1000ml (or any other quantity) for a fixed price.
    - Other than engine oil, few fuel additives can be added for smoother performance of the vehicle. These additives are added into fuel tank. Like engine oil they are also sold in fixed quantity and fixed price.
    - As service station get reasonable margin in selling these oil packs, no separate charges are taken from customer for changing oil.
- For each servicing request from customer for a vehicle may including repairing of multiple parts and/or changing engine oil and/or adding additives.

- A bill is prepared for each service request separately.
  - The detailed bill should print following details:
    - Service station Name.
    - Service Date.
    - Customer Name and Vehicle Details for which services were provided.
    - Each spare part costing
    - Maintenance Charges i.e. labor charges.
    - Oil/Additive price
    - Bill Amount + 12.6 % = Final Bill

- The minimum information for any vehicle is its manufacturer, model and vehicle number (on number plate). Note that customer may have more than one vehicle.
- Also, customer information should be maintained like his/her name, address, mobile number and vehicles. For sake of simplicity customers are identified by their mobile no.
- Service station should maintain daily billing details, so that at any time current cash collection can be calculated. Also, as it stores all old bills, business for any given date can be calculated.

- As console application starts a menu appears for service station manager:
  - Customer
    - Add Customer (Add new customer)
    - Display All Customers (Display all customers from database)
    - Display Specific Customer Details (Input mobile no and display the respective customer)
    - Edit Customer (Edit the mobile no and address of customer)
    - Delete Customer (Delete the customer from database)

  - Vehicle
    - Add Vehicle (Add new vehicle)
    - Display All Vehicles (Display all the vehicles from database)
    - Display Customer Vehicles (Display all the vehicles of specified customer)
    - Edit Vehicle (Edit the vehicle number)
    - Delete Vehicle (Delete the vehicle from database)

  - Service Request
    - Select Customer Vehicle (To do in this menu)
      - Choose customer from existing list of customers.
      - Display his/her vehicle details.
      - Choose vehicle from the list or input details of new vehicle and store it for future reference.

- Process Request (It consists of submenu)
  - New Service
    Service center can create a new service for the selected customer vehicle
  - Existing Service
    If service is already created select the existing service
  - Maintenance
    As per the requirements mentioned in above description
  - Repairing
    As per the requirements mentioned in above description
  - Oil/Additive Change/Add
    As per the requirements mentioned in above description

- Prepare and Display Bill
  Display the bill as per given in the above description
- Get Payment Form Customer
  Get the payment amount and add in payments table

o Parts
- Add Part (Add new part)
- Display All Parts (Display all parts from database)
- Edit Part Price (Edit the price of specific part based on part id)
- Delete Part (Delete a specific part based on part id.)

o Today's Business (Compute total cash from list of all bills (paid amount) of today and display it.)

o Given Date's Business (Compute total cash from list of all bills (paid amount) of user given date and display it.)

Note:

o Ensure that none of the classes (except main class) contain code related to console input/output. This
will make code less portable to user/web interface.
o Service station should maintain list of customers as set (java.util.HashSet<>).
o Service station should maintain list of bills as list (java.util.LinkedList<>).
o Customer should contain list of vehicles as Map (java.util.HashMap<String,Vehicle> □ Key is vehicle
number and Value is vehicle object).
o Each repairing/maintenance will store list of replaced spare parts is stored as List (java.util.LinkedList<>).

- Details of multiple services provided (during single service request) should be maintained as List (java.util.LinkedList<>).
- Use serialization and deserialization to store and load customer details.
- Implement equals(), hashCode() in designed classes, so that they will be useful for searching.
- Design, implement and utilize appropriate exception classes.
- Design GUI application which will allow handling multiple service requests concurrently. For more details refer screen shots or application exe.