Java - Assessment

Duration: 2 hours

Time: 4:00 PM till 6:00 pm

Case study Overview

This Case Study involves generating a train ticket and storing the ticket to the Collection.

The application should take train number and passenger details as input.

The details of the train should be stored in a Collection. The train details to be stored in the Collection are Train number, Train Name, Source Station, Destination Station and Ticket Price. The train details should be fetched from the Collection using the train number provided by the user.

The passenger name, age and gender should be accepted from the user.

The application should generate a train ticket using the train details and passenger details. The generated ticket should contain the following details

- PNR Number, Travel date, Train Number, Train Name, Source Station, Destination Station
- Name, age, gender and ticket fare for each passenger
- Total Ticket price

The generated ticket should be stored & retrieved as per requirement below.

Your client facing team has collected the requirements for you and the design team has created an initial design. You need to develop the Application based on the below requirements and design.

Note:

- Classes should be encapsulated
- Java naming conventions should be followed
- Appropriate commenting should be done

Business Requirements

· The Trains should be fetched from Collection using train number
· Travel date should be after current Date
· The PNR number should be auto-generated by adding first character of source station, first character of destination station, travel date(yyyymmdd) and a running counter starting from 100.
    - Example: PNR for train from Bangalore to Mumbai with travel date 21/01/2017 wouldbe BM_20170121_100

· The ticket Fare for each passenger is calculated using the following rules
    - For age < = 12, fare is 50% of ticket price regardless of gender
    - For age > = 60, fare is 60% of ticket price regardless of gender
    - For Females, 25% discount on the ticket price

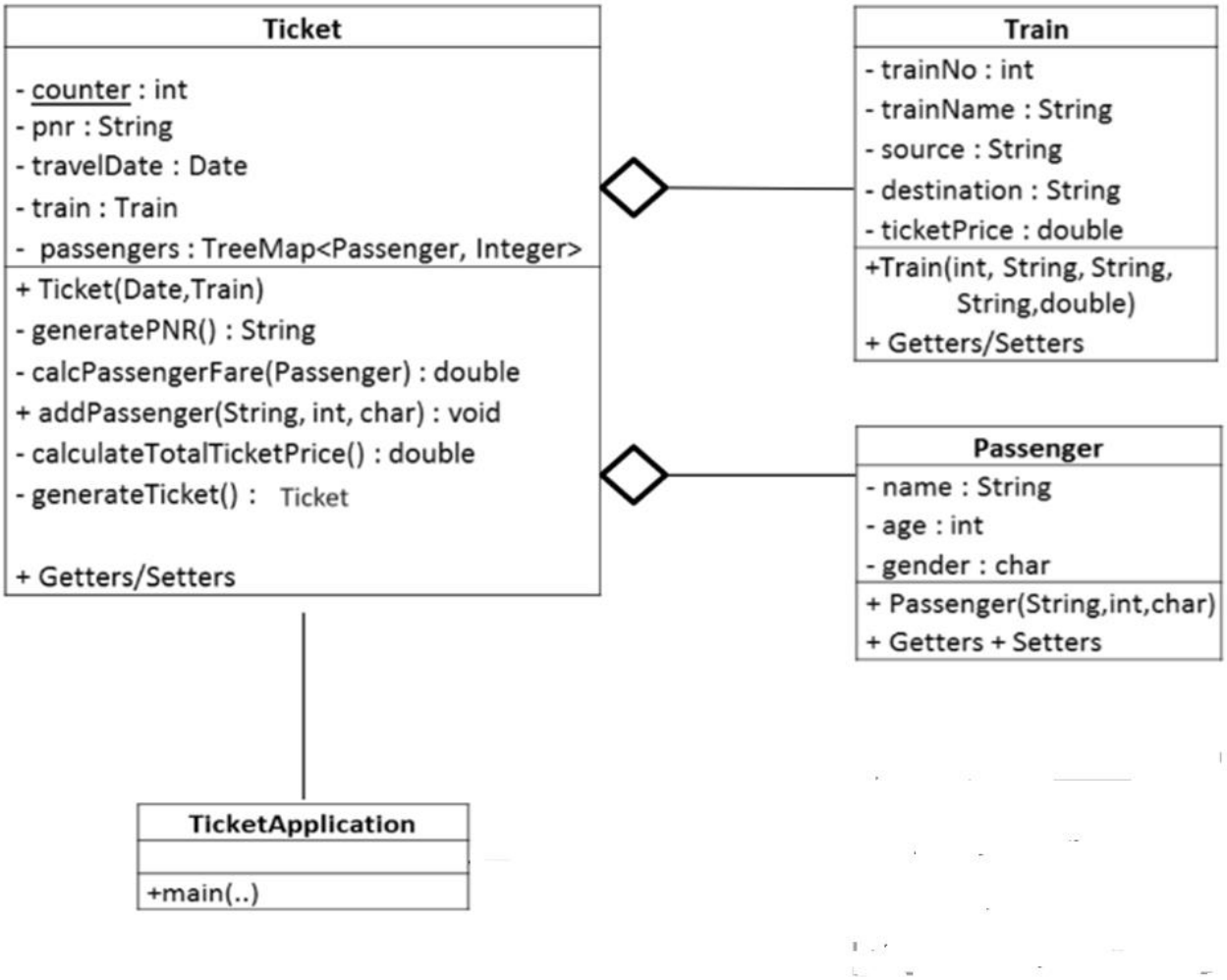
· The Ticket details should be displayed as per below format. The passenger's should be in ascending order based on name.

```
PNR : ----------
Train no : -----
Train Name : ----------
From : ----------
To : ----------
Travel Date : dd/mm/yyyy

Passengers :
Name Age Gender Fare
---------- ----- x,xxx.xx
---------- ----- x,xxx.xx
---------- ----- x,xxx.xx
Total Price : xx,xxx.xx
```

Class Diagram

The application consists of the following classes



Core Java Case Study – Train Ticket Generation

Class Ticket

- Create 3-4 Train Object and Store it in the Collection
- findTrain method takes trainNo as input and returns a Train object, if train is found, else throw TrainNotFoundException
- generatePNR method builds the PNR as per the business rules and returns the generated PNR
- calcPassengerFare method takes a passenger object as parameter, calculates the fare for the passenger and returns the ticket fare for that passenger
- addPassenger method take passenger details as parameters, creates passenger object and adds the passenger as a key and the calculated passenger fare as the value in TreeMap 'passengers'
- calculateTotalTicketPrice() method returns the total ticket price by adding up all the passenger fares stored in the TreeMap passengers
- generateTicket method generates the ticket as per the given format and returns the Ticket.

Class TicketApplication

- Contains a main method and is the starting point of Application
- Uses a Scanner to get the trainNo, travelDate, number of Passengers, and details of each passenger
- Use findTrain to find the train. If train is not found, it displays the message "Train with given number does not exist"
- Adds passengers to the Ticket
- Display the ticket to user

Sample Input and Output

Sample1

Enter the Train Number
1003
Enter Travel Date
01/01/2017
Enter Number of Passengers
2
Enter Passenger Name
Kumar
Enter Age
30
Enter Gender(M/F)
M
Enter Passenger Name
Vinay
Enter Age
32
Enter Gender(M/F)
M

Output
Ticket Booked with PNR : BM_20170101_100

Ticket should be displayed

Sample2
Enter the Train Number
9999
Output
Train with given train number does not exist

Sample3
Enter the Train Number
1003
Enter Travel Date
01/01/2000

Output
Travel Date is before current date

Instructions:

- **Follow coding standards, naming conventions, formatting**
- **Include Comments wherever possible**
- **Do not copy code from any source (internet/batch mates)**
- **Once done with the coding, push it to GitLab and provide access to the Repo.**
- **Follow Good Practices**
- **No Unnecessary Printing/Whitespaces**