CASESTUDY

Core Java – Train Ticket

Contents

Case study Overview

This Case Study involves generating a train ticket and printing the ticket to a File.

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

The details of the train should be stored in a database. The train details to be stored in database are Train number, Train Name, Source Station, Destination Station and Ticket Price. The train details should be fetched from the database 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 written to a File.

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 database 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 would be 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 file name for storing the generated ticket should be the PNR Number.

- The Ticket details should be stored in a file as per below format. The passenger's should written in ascending order based on name.

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

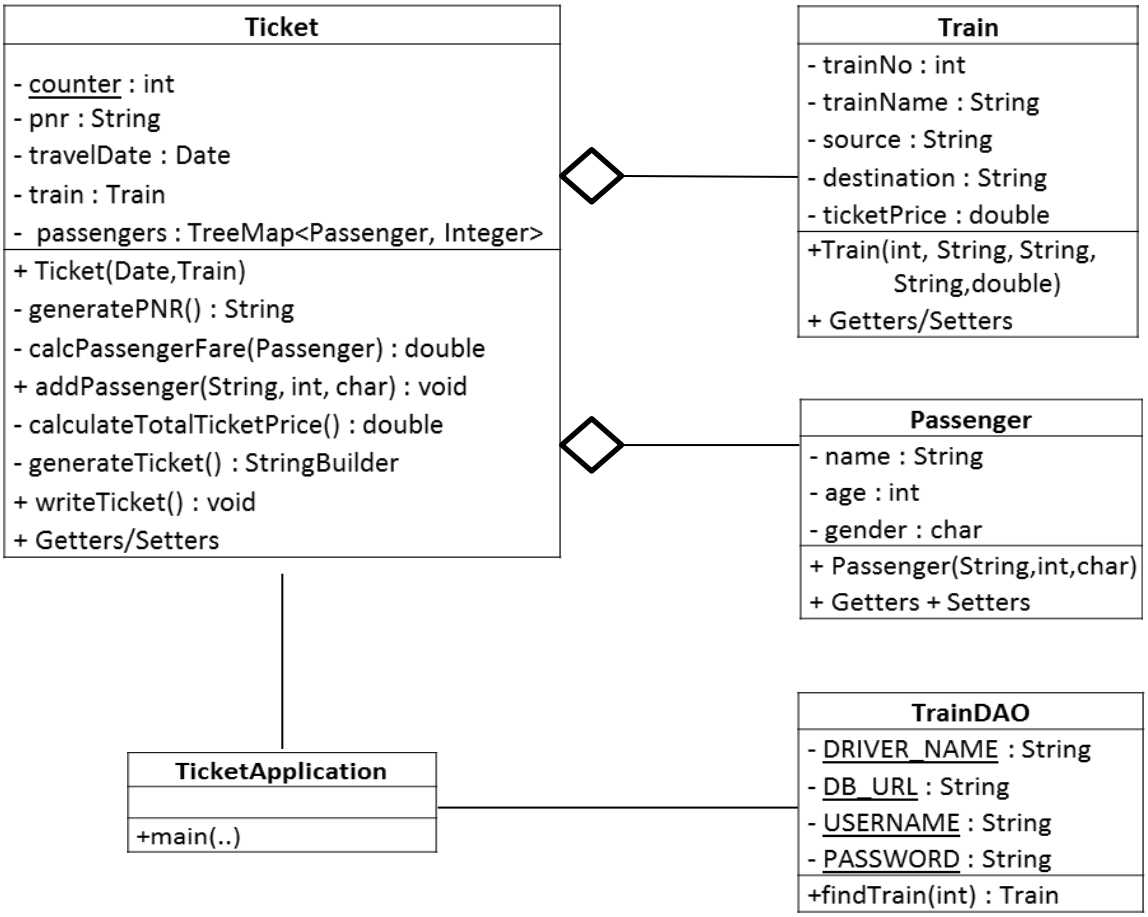
Passengers :

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



Class Ticket

- 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 return the ticket in a StringBuilder
- writeTicket method writes the generated ticket to a File

Class TrainDAO

- findTrain method takes trainNo as input and returns a Train object, if train is found, else returns null

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
- Uses TrainDAO 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
- Writes the ticket to a file

Sample Input and Output

Sample1

Enter the Train Number
1003
Enter Travel Date
01/01/2017
Enter Number of Passengers
2
Enter Passenger Name
Ghouse
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 written to the file with filename BM_20170101_100.txt

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

Table Design and Script

TABLE – TRAINS	
Column Name	**Column Type**
TRAIN_NO	NUMBER(5,0)
TRAIN_NAME	VARCHAR2(40)
SOURCE	VARCHAR2(20)
DESTINATION	VARCHAR2(20)
TICKET_PRICE	NUMBER(8,2)

Execute the below DDL to create the above table and insert Sample data

```
--------------------------------------------------------
-- DDL for Table TRAINS
--------------------------------------------------------
 DROP TABLE TRAINS;

 CREATE TABLE "TRAINS"
  (	"TRAIN_NO" NUMBER(5,0) PRIMARY KEY,
	"TRAIN_NAME" VARCHAR2(40),
	"SOURCE" VARCHAR2(20),
	"DESTINATION" VARCHAR2(20),
	"TICKET_PRICE" NUMBER(8,2)
  ) ;

Insert into TRAINS (TRAIN_NO,TRAIN_NAME,SOURCE,DESTINATION,TICKET_PRICE) values (1001,'Shatabdi
Express','Bangalore','Delhi',2500);
Insert into TRAINS (TRAIN_NO,TRAIN_NAME,SOURCE,DESTINATION,TICKET_PRICE) values (1002,'Shatabdi
Express','Delhi','Bangalore',2500);
Insert into TRAINS (TRAIN_NO,TRAIN_NAME,SOURCE,DESTINATION,TICKET_PRICE) values (1003,'Udyan
Express','Bangalore','Mumbai',1500);
Insert into TRAINS (TRAIN_NO,TRAIN_NAME,SOURCE,DESTINATION,TICKET_PRICE) values (1004,'Udyan
Express','Mumbai','Bangalore',1500);
Insert into TRAINS (TRAIN_NO,TRAIN_NAME,SOURCE,DESTINATION,TICKET_PRICE) values (1005,'Brindavan
Express','Bangalore','Chennai',1000);
Insert into TRAINS (TRAIN_NO,TRAIN_NAME,SOURCE,DESTINATION,TICKET_PRICE) values (1006,'Brindavan
Express','Chennai','Bangalore',1000);
```