View Problem

Problem Id : 12456 **Ticket - Requirement 1** **Reference Code :** ticket-1

Ticket - Requirement 1

Requirement 1:

Let's start off by creating two **Seat** objects and check whether they are equal.

1. Create a **Seat** Class with the following attributes:

Member Field Name	Type
seatNo	Integer
type	String
price	Double
passengerName	String
gender	String

2. Mark all the attributes as private

3. Create / Generate appropriate Getters & Setters

4. Add a default constructor and a parameterized constructor to take in all attributes in the given order:
   Seat (Integer seatNo, String type, Double price, String passengerName, String gender)

5. When the "Seat" object is printed, it should display the following details: **[Override the toString method]**

   Print format:

   Seat No:"seatNo"
   Type:"type"
   Price:"price"
   Passenger Name:"passengerName"
   Gender:"gender"

6. Two Seats are considered same if they have the same seatNo, and type. Implement the logic in the appropriate function. (Case – Insensitive)
   [Override the equals method]

The input format consists of Seat details separated by comma in the below order,
seatNo, type, price, passengerName, gender

The Input to your program would be details of two Seats, you need to display their details as given in "5th point(refer above)" and compare the two Seats and display if the Seats are same or different.

Note: There is an empty line between display statements. Print the empty lines in main function.
Display one digit after the decimal point for Double datatype.

Sample Input & Output 1:

Enter seat 1 detail:
10,Sleeper,600,Walter,Male
Enter seat 2 detail:
10,Sleeper,600,Walter,Male

Seat 1:
Seat No:10
Type:Sleeper
Price:600.0
Passenger Name:Walter
Gender:Male

Seat 2:
Seat No:10
Type:Sleeper
Price:600.0
Passenger Name:Walter
Gender:Male

Seat 1 is same as Seat 2

Sample Input & Output 2:

Enter seat 1 detail:
13,SemiSleeper,550,Lucas,Male
Enter seat 2 detail:
20,Sleeper,600,Alex,Female

Seat 1:
Seat No:13
Type:SemiSleeper
Price:550.0
Passenger Name:Lucas
Gender:Male

Seat 2:
Seat No:10
Type:Sleeper
Price:600.0
Passenger Name:Walter
Gender:Male

Seat 1 is same as Seat 2

Sample Input & Output 2:

Enter seat 1 detail:
13,SemiSleeper,550,Lucas,Male
Enter seat 2 detail:
20,Sleeper,600,Alex,Female

Seat 1:
Seat No:13
Type:SemiSleeper
Price:550.0
Passenger Name:Lucas
Gender:Male

Seat 2:
Seat No:20
Type:Sleeper
Price:600.0
Passenger Name:Alex
Gender:Female

Seat 1 and Seat 2 are different

View Problem

Problem Id : 12461 **Ticket - Requirement 2** **Reference Code :** ticket-2

Ticket - Requirement 2

Requirement 2:

Now we are gonna start creating a Ticket and add Seats to it. Start with creating a Ticket and use menu-driven approach to add, remove, display details of the Seats in the Ticket.

a)Create a Class **Seat** with the following attributes:

Member Field Name	Type
seatNo	Integer
type	String
price	Double
passengerName	String
gender	String

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Seat(Integer seatNo, String type, Double price, String passengerName, String gender).**Override **toString()** method to display the details.

b)Create a Class **Ticket** with the following attributes:

Member Field Name	Type
refId	String
dateOfJourney	java.util.Date
seatList	List

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Ticket(String refId, java.util.Date dateOfJourney, String seatList).** In constructor pass the seatList value as an empty list. Only one Ticket will be present at a time.

c) Create the following static method in Seat class,

Method Name	Description
public static Seat createSeat(String detail)	This method accepts a string which contains seat details separated by commas. Split the details and create a seat object from the details and return it.

The seat details should be given as a comma-separated value in the below order,

seatNo, type, price, passengerName, gender

d) Create the following methods in Ticket class,

Method Name	Description
public void addSeatToTicket(Seat seat)	This method accepts a seat object and adds the seat to the seat list of the current Ticket.
public Boolean removeSeatFromTicket (Integer seatNo)	This method will get the seatNo of the seat and delete the seat with the specified seatNo from the current Ticket. If a seat with the given seatNo found, delete the seat and return **true**. If a seat with the seatNo is not found return **false**.
public void displaySeats()	This method will display the seat list in the current Ticket. If the seat list is empty display "**No seats to show**", else display "Seats in [ticket refId]" and display all the seat details in the specified format. Where [ticket refId] specifies the refId of the ticket.

After deletion, if true is returned print "**Seat successfully deleted**", else print "**Seat not found in ticket**".

Note: The above print statements should be present in the main method.

When the "seat" object is printed, it should display the following format

Print format:

System.out.format("%-8s %-12s %-5s %-15s %s\n","Seat No","Type","Price","Passenger Name","Gender");

Display 1 digit after decimal point in Double.

```
Enter your choice:
1
20,Sleeper,1550,Cecile,Female
Seat successfully added
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit
Enter your choice:
3
Seats in BLR-MR12 CN-EG01/U16
Seat No  Type       Price Passenger Name  Gender
19       Seater     800.0 Natasha        Female
22       SemiSleeper 850.0 Jane          Female
20       Sleeper    1550.0 Cecile        Female
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit
Enter your choice:
2
Enter the seat number to be deleted:
15
Seat not found in ticket
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit
Enter your choice:
2
Enter the seat number to be deleted:
20
Seat successfully deleted
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit
Enter your choice:
3
Seats in BLR-MR12 CN-EG01/U16
Seat No  Type       Price Passenger Name  Gender
19       Seater     800.0 Natasha        Female
22       SemiSleeper 850.0 Jane          Female
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit
Enter your choice:
4
```

Sample Input and Output 1:

Enter the reference id of the ticket:
BLR-MR12 CN-EG01/U16
Enter the date of journey of the ticket:
05-06-2018
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit
Enter your choice:
3
No seats to show
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit
Enter your choice:
1
19,Seater,800,Natasha,Female
Seat successfully added
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit
Enter your choice:
1
22,SemiSleeper,850,Jane,Female
Seat successfully added
1.Add Seat
2.Delete Seat
3.Display Seats
4.Exit

View Problem

Problem Id : 12466 **Ticket - Requirement 3** **Reference Code :** ticket-3

Ticket - Requirement 3

Requirement 3:

In this requirement, you need to validate the RefId of the Ticket.

a)Create a class **Main** with the following static methods:

Method Name	Description
static Boolean validateRefId(String refId)	Validate the RefId based on the rules given below. Returns **true** if RefId is valid else return **false**

b) While validating refId follow the below rules. The format of the refId id is given below

DCC-VC ACC-VC/SEATNO

1. The **DCC**(Dept. City Code) and **ACC**(Arv. City Code) should contains only alphabets(uppercase) with the length of 2 to 4.Follw
2. The **VC**(Venue Code) can contain either alphabets(uppercase) or numbers or both with the length of 3 to 4.
3. There should be space in between DCC-VC and ACC-VC .
4. RefId may/may not include seat No. Seat no can be included with forward slash('/') and with the below conditions –
   The first character of SEATNO can contain one of the alphabet of [U,L,W,S] followed by two digit number.

Followed by '-'.

Example: **BLR-MR12 CN-EG01/U16** is a valid refId id.

Note: Print "**RefId is valid**" if refId is valid else print "**RefId is invalid**".
All the above print statements are present in the main method.

[All text in bold corresponds to input]
Sample Input and Output 1:

Enter the RefId to be validated:
BLR-MR12 CN-EG01
RefId is valid

Sample Input and Output 2:

Enter the RefId to be validated:
MYSOR-A12 TMK-MN25
RefId is invalid

View Problem

Problem Id : 12471 **Ticket - Requirement 4** **Reference Code :** ticket-4

Ticket - Requirement 4

Requirement 4:

In this requirement develop a feature in which you can search a List of Seats by seatNo, and gender.

a) Create a Class Seat with the following attributes:

Member Field Name	Type
seatNo	Integer
type	String
price	Double
passengerName	String
gender	String

String passengerName,
String gender)

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Seat(Integer seatNo, String type, Double price, String passe**

b) Create a class **SeatBO** with the following methods,

Method Name	Description
public List<Seat> findSeat(List<Seat> seatList,Integer seatNo)	This method accepts a list of seats and seatNo as arguments and returns a list of seats that matches with the given seatNo.
public List<Seat> findSeat(List<Seat> seatList,String gender)	This method accepts a list of seats and gender as arguments and returns a list of seats that matches with the given gender.

The mail details should be given as a comma-separated value in the below order,
seatNo, type, price, passengerName, gender

Print format:

System.out.format("%-8s %-12s %-5s %-15s %s\n","Seat No","Type","Price","Passenger Name","Gender");

Note: The seat lists are displayed in the main method.

If any other choice is selected, display "**Invalid choice**"

If the search detail is not found, display "**No such seat is present**"

Display one digit after the decimal point for Double Datatype.

Sample Input and Output 1:

Enter the number of seats:

7

10,Sleeper,600,Walter,Male

13,SemiSleeper,550,Lucas,Male

11,Seater,500,Brandon,Male

12,Sleeper,600,Will,Male

15,SemiSleeper,550,Caitlin,Female

16,Seater,500,Sara,Female

20,Sleeper,600,Alex,Female

Enter a search type:

1.By Seat No

2.By Gender

1

Enter the Seat No:

12

Seat No	Type	Price	Passenger Name	Gender
12	Sleeper	600.0	Will	Male

Sample Input and Output 2:

Enter the number of seats:

7

10,Sleeper,600,Walter,Male

13,SemiSleeper,550,Lucas,Male

11,Seater,500,Brandon,Male

12,Sleeper,600,Will,Male

15,SemiSleeper,550,Caitlin,Female

16,Seater,500,Sara,Female

20,Sleeper,600,Alex,Female

Enter a search type:

1.By Seat No

2.By Gender

2

Enter the Gender:

Male

Seat No	Type	Price	Passenger Name	Gender
10	Sleeper	600.0	Walter	Male
13	SemiSleeper	550.0	Lucas	Male
12	Sleeper	600.0	Will	Male

Sample Input and Output 3:

Enter the number of seats:

7

10,Sleeper,600,Walter,Male

13,SemiSleeper,550,Lucas,Male

11,Seater,500,Brandon,Male

12,Sleeper,600,Will,Male

15,SemiSleeper,550,Caitlin,Female

16,Seater,500,Sara,Female

20,Sleeper,600,Alex,Female

Enter a search type:

1.By Seat No

2.By Gender

3

Invalid choice

Problem Id : 12476 **Ticket - Requirement 5** **Reference Code :** ticket-5

Ticket - Requirement 5

Requirement 5:

In this requirement, you need to sort the list of seats based on seatNo, and passengerName.

a) Create a Class Seat with the following attributes:

Member Field Name	Type
seatNo	Integer
type	String
price	Double
passengerName	String
gender	String

Seat(Integer seatNo, String type, Double price, String passengerName, String gender)

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Seat(Integer seatNo,**

b) Create the following static methods in the Seat class,

Method Name	Description
static Seat createSeat(String detail)	This method accepts a String. The seat detail separated by commas is passed as the argument. Split the details and create a seat object and returns it.

The seat details should be given as a comma-separated value in the below order,
seatNo, type, price, passengerName, gender

c) The Seat class should implement the **Comparable** interface which sorts the Seat list based on seatNo. While comparing, all the seatNo attributes in the list are unique.

d) Create a class **PassengerNameComparator** which implements Comparator interface and sort the Seat list based on passengerName. While comparing, all the passengerName attributes in the list are uni

Get the number of Seats and seat details and create a seat list. Sort the Seats according to the given option and display the list.

List are unique

Print format:

System.out.format("%-8s %-12s %-5s %-15s %s\n","Seat No","Type","Price","Passenger Name","Gender");

Display one digit after decimal point for Double datatype.

Sample Input and Output 1:

Enter the number of the seats:

6

25,Sleeper,900,Carter,Female
19,Seater,800,Natasha,Female
11,SemiSleeper,850,Christine,Female
14,Sleeper,900,Strange,Male
08,Seater,800,Bruce,Male
12,SemiSleeper,850,Steve,Male

Enter a type to sort:
1.Sort by Seat No
2.Sort by Passenger Name

1

Seat no	Type	Price	Passenge Name	Gender
8	Seater	800.0	Bruce	Male
11	SemiSleeper	850.0	Christine	Female
12	SemiSleeper	850.0	Steve	Male
14	Sleeper	900.0	Strange	Male
19	Seater	800.0	Natasha	Female
25	Sleeper	900.0	Carter	Female

Sample Input and Output 2:

Enter the number of the seats:

6

25,Sleeper,900,Carter,Female
19,Seater,800,Natasha,Female
11,SemiSleeper,850,Christine,Female
14,Sleeper,900,Strange,Male
08,Seater,800,Bruce,Male
12,SemiSleeper,850,Steve,Male

Enter a type to sort:
1.Sort by Seat No
2.Sort by Passenger Name
1

Seat no	Type	Price	Passenge Name	Gender
8	Seater	800.0	Bruce	Male
11	SemiSleeper	850.0	Christine	Female
12	SemiSleeper	850.0	Steve	Male
14	Sleeper	900.0	Strange	Male
19	Seater	800.0	Natasha	Female
25	Sleeper	900.0	Carter	Female

Sample Input and Output 2:

Enter the number of the seats:
6
25,Sleeper,900,Carter,Female
19,Seater,800,Natasha,Female
11,SemiSleeper,850,Christine,Female
14,Sleeper,900,Strange,Male
08,Seater,800,Bruce,Male
12,SemiSleeper,850,Steve,Male
Enter a type to sort:
1.Sort by Seat No
2.Sort by Passenger Name
2

Seat No	Type	Price	Passenger Name	Gender
8	Seater	800.0	Bruce	Male
25	Sleeper	900.0	Carter	Female
11	SemiSleeper	850.0	Christine	Female
19	Seater	800.0	Natasha	Female
12	SemiSleeper	850.0	Steve	Male
14	Sleeper	900.0	Strange	Male

View Problem

Problem Id : 12481 **Ticket - Requirement 6** **Reference Code :** ticket-6

Ticket - Requirement 6

Requirement 6:

In this requirement, given a list of employees, you need to find gender wise count of passengers.

a)Create a Class **Seat** with the following attributes:

Member Field Name	Type
seatNo	Integer
type	String
price	Double
passengerName	String
gender	String

Seat(Integer seatNo, String type, Double price, String passengerName, String gender)

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Seat(Integer seatNo,**

b) Create the following static method in Seat class,

Method Name	Description
public static Seat createSeat(String detail)	This method accepts a string which contains seat details separated by commas. Split the details and create a seat object from the details and return it.
public static Map<String,Integer> genderWiseCount(List<Seat> seatList)	This method accepts a list of seats and returns a treemap with gender wise count of passengers occupying those seats.

The seat details should be given as a comma-separated value in the below order,

seatNo, type, price, passengerName, gender

Print format:

```
System.out.format("%-15s %s\n","Gender","No. of Seats");
```

Sample Input and Output 1:

```
Enter the number of seats
5
31,SemiSleeper,1500,Laurel,Female
21,Seater,1450,Oliver,Male
20,Sleeper,1550,Cecile,Female
35,SemiSleeper,1500,Joe,Male
15,Seater,1450,Lisa,Female
Gender          No. of Seats
Female          3
Male            2
```