**EZEE Transport**

**Grade settings**: Maximum grade: 100  
**Disable external file upload, paste and drop external content**: Yes  
**Run**: Yes **Evaluate**: Yes  
**Automatic grade**: Yes **Maximum execution time**: 120 s **Maximum memory used**: 512 MiB **Maximum execution file size**: 1.25 GiB

**EEZEE TRANSPORT - Ticket Booking**

EEZEE TRANSPORT is one of the renowned travel service providers. They are in a need of system which helps customer book vehicle from Chennai to various cities and find appropriate travel cost. You being their software consultant have been approached by them to develop a software system which can be used for managing their business.

Provided **Booking** class with the below **private attributes** as a part of code skeleton

|  |  |
| --- | --- |
| customerName | String |
| mobileNumber | Long |
| destination | String |
| dateOfJourney | java.util.Date |

**Getter and setter** methods for all the above attributes have been provided as a part of code skeleton.

**Customer** class should be registered as a **bean** with the spring container via **XML file**.

**Provided Vehicle** class with the below **private attributes** as a part of code skeleton

|  |  |
| --- | --- |
| source | String |
| destinationMap | Map<String, Double> |

**Getter and setter** methods for all the above attributes have been provided as a part of code skeleton.

**Vehicle** class should be registered as a bean with the spring container via **XML file**.

The values for source attributes should **be injected via constructor based injection** and the default source must be "**Chennai**".

The Map should be configured in the bean.xml file with the below key-values should be **injected via setter based injection.**

|  |  |
| --- | --- |
| **Key -destination (String)** | **Value-** cost**(Double)** |
| Bangalore | 1500 |
| Delhi | 2500 |
| Mumbai | 2000 |
| Hyderabad | 1000 |
| Pune | 2500 |

**Note:** Only for the above mode of vehicle cost to be calculated as per the distance. **The keys are case sensitive**

**Main.java: main(String[] arg):void**

1. Using ApplicationContext, get TransportService bean and Booking bean.
2. Invoke the registerBooking(booking) method using the transportService bean.

**Technical Specifications:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Method Name** | **Input** | **Output** | **Exception** |
| TransportService | registerBooking | Booking booking | Void | ParseException |
| calculateTravelCost | String source, String destination, Date doj | Void |  |
| checkAvailableDateForTravel | Date doj | boolean:result |  |
| getCost | String destination | double: cost |  |
| bean.xml | Contains all the XML configurations related to Service |  |  |  |

**Business Rules & Validations:**

In **TransportService class**includes the following private attribute **and inject via setter based injection**.

**private Vehicle vehicle;**

**Getter and setter** methods for the above attribute has been provided as a part of code skeleton

**Service 1**: **registerBooking(booking): void**

            This method used to get booking information and calculate travel cost based on customer information.

1. Inject Booking bean using application context.
2. Get Booking information like customerName, mobileNumber, destination, date of journey (date format must be dd-MM-yyyy) from customer.
3. Invoke calculateTravelCost() method, Use booking bean to get the destination,dateOfJourney and use vehicle bean to get default source.

**Service 2:  calculateTravelCost(String source, String destination, Date doj) : void**

1. Invoke checkAvailableDateForTravel(doj).
2. If it returns true then display the **"Total Travel Cost is Rs. "**and invoke **getCost(destination)** method.
3. If it returns false then display "Invalid Travel Date".

**Service 3**: **checkAvailableDateForTravel(doj): boolean**

1. If DOJ is current date/next date, then return true.

**Service 4**: **getCost(destination): double**

1. Use  Vehicle Bean, get the travelcost from map if destination exists

**Limitations and Constraints:**

1.  **Booking**and**Vehicle class**should be in**com.cts.eezee.model**package.

2.  **TransportService**class should be in**com.cts.eezee.service**package**.**

3.   **Main class**should be in **com.cts.eezee** package.

4.   All of the above mentioned java classes to be configured as beans in **bean.xml** file

5.   **vehicle** should be declared as an **inner bean** in the **TransportService** class and should

      be injected by autowire inside **bean.xml.**

6.   **Vehicle** should be configured as bean inside bean.xml with the bean id as **"vehicle"**

7. The cost for each destination should be declared as a MAP in the **bean.xml** and

       should be injected using setter.

11. Use ONLY **bean.xml** for all configurations.

**Sample Input Output 1:**

Enter the Customer Name

**Antony Prakash**

Enter the Mobile Number

**9845623456**

Enter the Destination

**Delhi**

Enter the Date of Jouney <dd-MM-yyy>

**27-04-2020**

Total Travel Cost is Rs. 2500.0

**Sample Input Output 2:**

Enter the Customer Name

**Renita Colette**

Enter the Mobile Number

**9763458245**

Enter the Destination

**Delhi**

Enter the Date of Jouney <dd-MM-yyy>

**26-04-2020**

**Invalid Travel Date**