**Important Instruction:**

1. **Please read the document thoroughly before you code.**
2. **Import the given skeleton code into your Eclipse.**
3. **You have to create the input file for the methods.**
4. **Refer/Use the solution file only when you are not able to complete the case study within 3.5 hours**
5. **Business Scenario:**

A leading travel service company in Japan has planned for laying 2 major rail route in the country. The routes are termed as TR1 and TR2. The details of each train and its routes are stored in a data file (TrainRoutesandFares.dat). The train stations are represented as numbers from 1 to 20. Trains running in one route cannot run in another route. The station numbers for each route is given below.

For TR1 the station numbers are from 1 to 10 and TR2 the station numbers are from 11 to 20.

**Develop an application for the below 2 requirements.**

**Requirement 1:**

Given the source station number, destination station number and date of travel as input, the application should return all the train details which is running strictly from the source to the destination stations and not the trains passing through the stations. *Also note that only special train runs on Sundays also.* *Other trains runs only From Monday to Saturday.*

\* Train Details object is provided with the skeleton code, the attribute names, getters / setters and equals method should not be modified should be used as it is.

**Requirement 2:**

This is to return all the train details running only on Sundays. It is expected to return a Map of TreeSet (TreeSet to hold the train numbers) with Map key denoting Sunday (integer 1 denotes Sunday)

1. **Skeleton File for Development:**

Import the below attached skeleton code into your eclipse project and implement the required functionalities



1. **Technical Specifications:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ClassName** | **Method Name** | **Input Parameters** | **Sample Input** | **Output Parameters** |
| TrainServiceManager | getTrainDetails | **String dataFile** – path of the folder along with the filename where the data feed is located. | **"C\:\\CoreJava\\TrainRoutesAndFares.dat"** | List<TrainDetailsVO> |
| **int source –** Source stop | **2** |
| **int destination –** destination stop | **9** |
| **String dateOfTravel –** Date of travel (in dd-mm-yyyy format) | **“31-05-2015”** |
| getTrainSchedule | **String dataFile** – path of the folder along with the filename where the data feed is located. | **"C\:\\CoreJava\\TrainRoutesAndFares.dat”** | Map<Integer,TreeSet> |

**Input File:** Create an input file with the name “TrainRoutesandFares.dat” and store the sample train records(each line is a record). Use the delimiter “,” to separate the field values.

The record format is given below

<<TrainNumber>>,<<RouteNumber>>,<<SourceStationNumber>>,<<DestinationStationNumber>>,<<Special>>

Field Constraints:

TrainNumber –5 digit number

RouteNumber – String of 3 character – (either TR1 or TR2)

SourceStationNumber/DestinationStationNumber – For TR1: 01 to 10 and for TR2:11 to 20

Special – Single Character (Y or N) – “Y” denotes that the train runs on Sundays also.

For example (the data file with two train details)

**12345|TR1|01|10|Y**

**16549|TR2|11|17|N**

### Requirement 1: Method Description: getTrainDetails()

|  |  |  |  |
| --- | --- | --- | --- |
| **ClassName** | **Method Name** | **Parameters** | **Output Parameters** |
| TrainServiceManager | getTrainDetails | **String dataFile** – Path of the folder along with the filename where the data feed is located. | List<TrainDetailsVO> |
| **int source –** Source station number |
| **int destination –** Destination station number |
| **String dateOfTravel –** Date of travel (in dd-mm-yyyy format) |

### Method Parameter and Validation:

1. dataFile: The absolute path(with the file name) of the data file (TrainRouteAndFares.dat). The file should be available
2. sourceStop: Source Station number. It should be 1 to 20
3. destinationStop: Destination Station number. It should be 1 to 20.
4. dateOfTravel: Date of travel in dd-mm-yyyy format. Date of travel should be always greater than current date.

Note that source and destination station number should NOT be same.

If any of the above validation fails please throw “TrainServiceException”

**Method Logic (to be implemented):**

For the given source, destination, and date of trave, return the train details which runs only from the source to the destination. Please note only special trains runs on Sundays

**Return Value:**

Place all the train details in the TrainDetails object and return the List of TrainDetailsVO.

### Requirement 2: Method Description: getTrainSchedule()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ClassName** | **Method Name** | **Input Parameters** | **Sample Input** | **Output Parameters** |
| TrainServiceManager | getTrainSchedule | **String dataFile** – path of the folder along with the filename where the data feed is located. | **"C\:\\CoreJava\\TrainRoutesAndFares.dat”** | Map<Integer,TreeSet> |

### Method Parameter and Validation:

1. dataFile: The absolute path(with the file name) of the data file (TrainRouteAndFares.dat). The file should be available. For any errors please through “TrainServiceException”

**Method Logic (to be implemented):**

Select all the trains that run on Sunday and return a Map with the key for Sunday(i.e, 1) and the value as TreeSet (train numbers running on Sunday (Special Trains))

**Return Value:**

Map with the key for Sunday (i.e, 1) and the value as TreeSet (train numbers running on Sunday (Special Trains))

1. **Solution:**

**The attached code is one of the possible solution meeting the given requirement. Please refer/use this only if you are not able solve the above given problem scenario within 3.5 hours.**

****

**Note: You will not find the solution file in the actual assessment.**