LabWork JAVA 3

EEng3

You will do one project for the entire labwork. This project will contain the necessary classes, including the main class. Each part will have to be tested before moving on to the next.

Objectives :
The objective of this labwork is to create classes and methods that will allow the analysis of the regularity of TGVs in recent years.
These and other data are freely available on the website www.data.gouv.fr and https://ressources.data.sncf.com/
Retrieve on Espace.ecam.fr the files MonthlyRegularity.java, ReadTrainFile.java, DisplayHisto.java and regularite-monthly-tgv.csv. *Copy the three classes to the src folder of your project*
Copy the csv file into the root of your project (and not in src).

What does the csv file contain? (You can open this file with Excel which will interpret it or with notepad ++ which displays it directly).
How many lines does it contain?
For the TGV from Lyon to Paris, are there several lines in the file? Why ?

Part 1 : Station

Create the Station class as follows.

Station
- idStation : integer - name : String
+ Station(idStationP : integer, nameP : String) + getIdStation() : integer + getName() : String

Caution: Be sure to **exactly** follow the names of classes, attributes, methods specified.

Test by creating an instance of your class.

Part 2 : MonthlyRegularity

The MonthlyRegularity class is provided. What does this class contain?
What does this class correspond to the data in the csv file?
What does the toString method do in the MonthlyRegularity class?
Add a toString method to the station class.

The ReadTrainFile class is provided.

What does the getFileTrain method return?

Test your project by displaying the number of rows in the result array of the getFileTrain method. (8811)

Part 3 : TrainTreatment

Create the proposed TreatmentTrain class below.
You will realize and test the methods one by one.

TrainTreatment
- tabMR : array of MonthlyRegularity
+ TrainTreatment(tabMRParam : array of MonthlyRegularity) + display() + nbOfMR(idStationFrom : int ; idStationTo : int) : int + minDate() : String + maxDate : String + pcRegularity(idStationFrom : int ; idStationTo : int) : double + histoMonthlyLate() : array of int

TrainTreatment : Constructor with one parameter

display : displays each MonthlyRegularity then the total number of MonthlyRegularity

nbOfMR : calculates the total number of MonthlyRegularity (number of lines of the file) between the two stations whose id passed in parameters.
Test with 0 (Paris) and 19 (Lyon). Result: 82

minDate : calculates the smallest date of the MRs in the table. This method returns a string associated with the month and year found (month + "" + year)
Result: 9 2011

maxDate : same. Result 6 2018

pcRegularity : percentage of trains not canceled and not delayed for two given stations. Test with 0 and 19. Result: 93.16%

histoMonthlyLate: returns the histogram of the number of delays (the number of delays is included in 0 and 220) in the form of an array of integers

The DisplayHisto class allows you to graphically display an integer array as a histogram. Use: construct with an array of integers in parameter.

A little bit more ? :

Calculate the average regularity percentage. Be careful, some lines in the table have 0 trains. The result may vary depending on the calculation method.
Show the couple of stations with the lowest regularity percentage