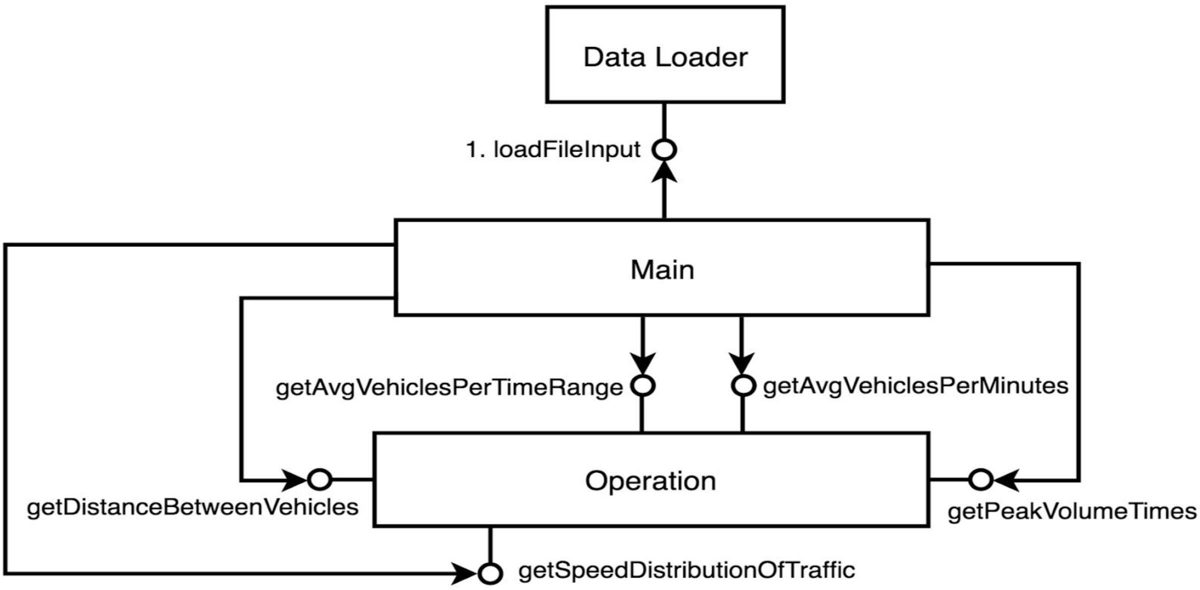
Explanation of Davinder Kumar

I found interesting both problems proposed but from the first reading I liked more the problem about the vehicles and I decided to solve it.

The component architecture shown below summarizes the functionality of the software.



Three components can be distinguished:

- **Data Loader**: provides a *loadFileInput* interface that allows to load the data from the input file into a data structure
- **Operation**: provides several interfaces that allow to analyze the data of the vehicles
- **Main**: this component load the data and execute the operations of analysis

The description of the individual interfaces is shown below.

- ➢ **loadFileInput**: By starting from the path of the input file this interface returns an array (of maps) of two elements. The maps contain the times of northbound and southbound.
- ➢ **getAvgVehiclePerTimeRange**: for each day calculate the number of vehicles in the given time range and return the average of the vehicles (morning and evening can be seen like a time range).
- ➢ **getAvgVehiclePerMinutes**: 1. scan each day for given intervals of minutes, 2. for each interval calculate the number of vehicles, 3. calculate the average for each day 4. the result is the average across all the day.
- ➢ **getPeakVolumeTimes**: 1. scan each day for given intervals of minutes, 2. for each interval calculate the number of vehicles, 3. across all the days calculate the averages in the same interval, 4. the result is a ranking created for the decreasing number of vehicles.
- ➢ **getSpeedDistributionOfTraffic**: 1. for each day calculate the average speed in the given time range, 2. the result is the average speed across all the days.

  The point 1 is calculated as follows: V = S/t

  S = 2.5 (wheelbase) t is the average time between the axles of the same vehicle on the sensor A

  Example: northbound $A_1A_2 \rightarrow t = A_2 – A_1$ southbound $A_1B_1A_2B_2 \rightarrow t = A_2 – A_1$
- ➢ **getDistanceBetweenVehicles**: 1. for each day calculate the average distance in the given time range, 2. the result is the average distance across all the days.

  The point 1 is calculated as follows: S = V*t

  V is calculated as in *getSpeedDistributionOfTraffic*

  t is the average time between the axles of the consecutive vehicles on the sensor A

  Example: northbound $A_1A_2A_3A_4 \rightarrow t = A_3 – A_2$ southbound: $A_1B_1A_2B_2\ A_3B_3A_4B_4 \rightarrow t = A_3 – A_2$

What I tried to do

By analyzing the input file with the "awk" filter I noticed that the times aren't mixed and therefore the AA pairs are separated by the ABAB sequence.

I tried to consider even the case of mixed times but failing to solve some cases (traffic) I decided to do without it. However if I can find a solution for the mixed times, I will have to modify a small part of the software (upstream) while the rest will continue to work.

How to execute the software

I created a script to make it easier to run the software.

The script must be present in the same path of the input file.

To run the script just do "java -jar vehicleChallenge.jar" and follow the instructions.

You can also import the project on Eclipse(file→import→general→project from folder or archive→Directory).

Testing

The test classes are two:

1. *DataLoaderTest*: used to verify the correct loading of data from the input file
2. *OperationTest*: used to test the class Operation

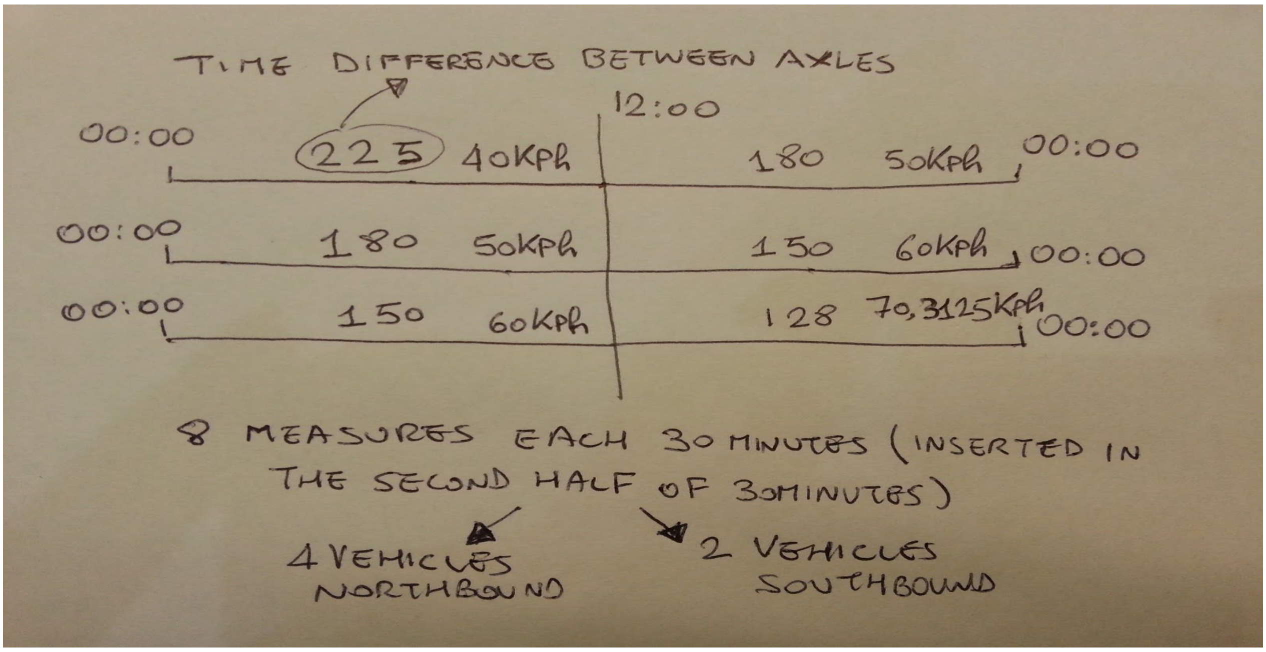
In the *OperationTest* I created a setup shown in the following image.

I will describe only the first day because the same reasoning is valid for others.

FIRST DAY

The time difference between the axles is of 225 in the first part of the day and 180 in the second one.

If the Wheelbase is 2.5 we have 3.6*(2.5/0.225) = 40 kph for the first part and 50 kph for the second one.



8 MEASURES EACH 30 MINUTES (INSERTED IN THE SECOND HALF OF 30MINUTES)

In OperationTest I was tempted to create two separate test classes: one to test the number of vehicles and peak volume time and the other to test the speed and distance between the vehicles.

However I discarded this solution because I thought it was not great.

Therefore I tried to understand if it is possible to create a separate setup only for the number of vehicles and for the peak volume times but I have not found a clear solution.

I wanted to minimize the number of lines in test-case.

Melbourne, 05/07/2018
Kind regards
Davinder Kumar