Trains Timetable

Develop an application to manage trains timetable and monitor when trains stop in the railway stations.
All classes MUST be in package "timetable".

R1 - Define Paths

The System works with class Timetable.
The first phase defines trains' railway distances.
The method **createPath()** has these parameters: the code (e.g. IC2345) and the category; then it creates a **Path** object. Train's category can be "Intercity", "Eurostar", "Interregional", or "Regional". The class **Path** implements methods **getCode()**, and **getCategory()**. Trains type can be *ordinary* or *extraordinary*, and it can be defined and read with methods **setExtraordinary()** which has a boolean parameter, and **isExtraordinary()** which returns a boolean value. The default value is *ordinary*.
It is possible to know all paths dfined in the timetable with method **getPaths()** returning a Collection of Path objects.
Moreover, the method **getPath()** receives a path code abd returns the corresponding object.

R2 - Train Stops

Each Path is made of different Train Stops, each one related to one railway station.
The class Path provides the method **addStop()** which has these parameters: name of the railway station, the arrival time in hours and minutes; this method returns the corresponding **TrainStop** object.
Class TrainStop has methods **getStation()**, **getHour()**, **getMinutes()**.
The list of all train stops in a path can be obtained through the method **getTrainStops()** in class Path: it returns a list of TrainStop objects ordered by arrival time.

R3 - Trains

Trains follow a predefined path and stop at the railway station at a particular moment.
The class Timetable has a method **newTrain()** that, once reived parameters like code, path and the date, returns the new Train object created.
If path code does not correspond to a path defined, then **InvalidPath** exception is generated..
Clas Train has methods **getPath()** and **getDate()** to read these information.
Starting form a Path it is possible to obtain the trains list which have passed through this Path, using the method **getTrains()** of Path class: this method returns the trains list ordered by date in descendant order.

R4 - Passages

A train can pass through a railway station at a certain moment.
The class Train has method **registerPassage()** receiving station name, hour and minutes; this method returns the **Passage** object created.
If the station name does not correspond to one of the train stops of the path, then a **InvalidStop** exception is generated.

The class **Passage** has methods **getStation()**, **getHour()**, **getMinutes()**.
Then the method **delay()** returns the delay in minutes of the train with respect to the timetable of the TrainStop.

R5 - Statistics

Class Train allows obtaining some statistics.
Method **arrived()** returns true if the train has stopped at the last railway station.
Methods **maxDelay()**, **minDelay()**, **totalDelay()** give information on delays.
Class Path has also methods **maxDelay()**, **minDelay()**, **totalDelay()** which return values related to total delays of all trains passed by this path.