I selected Vehicle survey coding challenge for different reasons

* 1. It is a wide-scope problem, which allows me to use different OOP best practices and design patterns to implement extendable code
  2. It is a real-life problem

**Application Design**

The application is divided into different modules; every module is responsible for implementing part of the application as follows

1. **Data reader**: It reads records from the given data source, it is designed to support plugging different data readers; the current implementation reads from files. Also, to support different strategies for handling invalid records, data reader composes a handler, the current implementation stops with first invalid record. This module is responsible for the following
   1. Reads records from a file.
   2. Validates the loaded records.
2. **Record processor**: It processes the raw data loaded by the data reader. To store the processed records, the processor uses a composed collector; the application can be extended to support plugging different collectors like for example file-based collector. The current implementation uses memory-based collector. This processor does the following
   1. Processes chunk of records to detect the direction.
   2. Validates the sequence of records, so for example 2 successive records from sensor B should be considered as invalid data.
   3. Detects the change in reading day.
3. **Analyser**: it analyses vehicle records to generate the required reports. I used Command Pattern to implement the required reports, this module is designed to support ,this module does the following
   1. It uses a command factory to generate the command based on the required feature; this command is used to run the required report.
   2. Generates a summary from the generated result, like for example the max and min vehicle speed.
4. **Analysis result exporter**: it exports the result of analysis to files. Using Visitor design pattern, the application is ready to export different analysis results.