**Vehicle Identification**

About this document:

This document explains algorithm used to identify type of vehicle in given input xml file. It will starts with problem statement, input, output, data structures used and algorithm.

This documents also explains how to build project using ant script and how to run executable jar file.

Problem Statement:

Identify vehicle type on the basis of their components and report that lists each vehicle id and its type. The report should also provide a summary saying how many vehicles of each type are in the XML.

Please refer README\_PROBLEM\_DESCRIPTION.pdf

Input:

Absolute path of XML file which contains details of each components of vehicle.



Output:

The csv file ”vehicleIdentificationReport.csv “ which will be generated in the same folder of input XML file.



It contains no of vehicles of each type and their vehicle id.

Data Structure Used:

1. **vehicleList:** It is list of Vehicle objects which represent each vehicle in input XML file.
2. **bigWheelList**: It is list of String which represents id of vehicle which is of Big Wheel Type.
3. **bicycleList:** It is list of String which represents id of vehicle which is of bicycle Type.
4. **motorCycleList:** It is list of String which represents id of vehicle which is of motorcycle Type.
5. **handGliderList:** It is list of String which represents id of vehicle which is of hand glider Type.
6. **carList:** It is list of String which represents id of vehicle which is of car Type.
7. **invalidFormatList:** It is a list of String which represents id of vehicle which has invalid format.

Pre-requisites**:**

**“vehicle-identifier-config.xsd”** is xml schema which is generated according to the input xml file. It is used for reading and storing input data in efficient way and also for validating input xml file.

This xsd file should be present in the same folder where input xml file is present.



Algorithm:

1. Validate input xml file using “**vehicle-identifier-config.xsd”** schema.
   1. Throw exception in case of wrong input
   2. Populate **vehicleList** by reading given input xml file.
2. For each **vehicle** object present in **vehicleList**
   1. If vehicle has (material=plastic & PowerTrain=human & 3 wheels-front,rear left & right)

Add vehicle id into **bigWheelList**

* 1. If vehicle has (material=metal & PowerTrain=human & 2 wheels-front,rear)

Add vehicle id into **bicycleList**

* 1. If vehicle has (material=metal & PowerTrain=Internal Combustion & 2 wheels-front,rear)

Add vehicle id into **motorcycleList**

* 1. If vehicle has (material=plastic & PowerTrain= Bernoulli & 0 wheels )

Add vehicle id into **handGliderList**

* 1. If vehicle has (material=metal & PowerTrain= Internal Combustion & 4 wheels left & right-front left & right rear )

Add vehicle id into **carList**

* 1. Else add vehicle id into **invalidFormatList**

1. Print all strings from **invalidFormatList** saying **Invalid vehicle type**
2. **Print size of all list** except invalidFormatList which is **no of each type of vehicle** and **all id string from all list for each type of vehicle** in “vehicleIdentificationReport.csv”.

Build:

Assuming directory structure is present while building project using ant script “build\_vehicle\_identifier\_test.xml”

{ProjectName}

|\_\_src ( contains all source code )

|\_\_lib ( contains dependent jar files like junit-4.10.jar )

|\_\_config ( containg vehicle-identifier-config.xsd )

|\_\_buildScript ( containing build\_vehicle\_identifier\_test.xml )

To run build script

1. Set ANT\_HOME in environment variable.
2. Add ANT\_HOME into path in environment variable.
3. Open command prompt
4. cd “Path folder where build\_vehicle\_identifier\_test.xml is kept”

../{ProjectName}/buildScript

1. Run command “ant –f build\_vehicle\_identifier\_test.xml build”
2. vehicleIdentifier.jar is generated at “{ProjectName}\dist”

Run:

To run “vehicleIdentifier.jar”

1. Open command prompt
2. Cd {ProjectName}\dist
3. Run “java –jar vehicleIdentifier.jar {inputXMLFile} ” where inputXMLFIle contains details of vehicle and validated through vehicle-identifier-config.xsd schema file. This schema file must be present at same location where inputXMLFile is present.