Challenge OOP - Car Factory

Task

As a programmer in a software development company you have been asked to complete a small Java program for a Car Classification **system.**

Part 1

Create a Java project. The **project solution** *and* **package folder** should both be named **p2<your student id> e.g. p23048201 (check you have done this correctly before continuing)**

Create the following classes as per the UML Class diagram (figure 1).

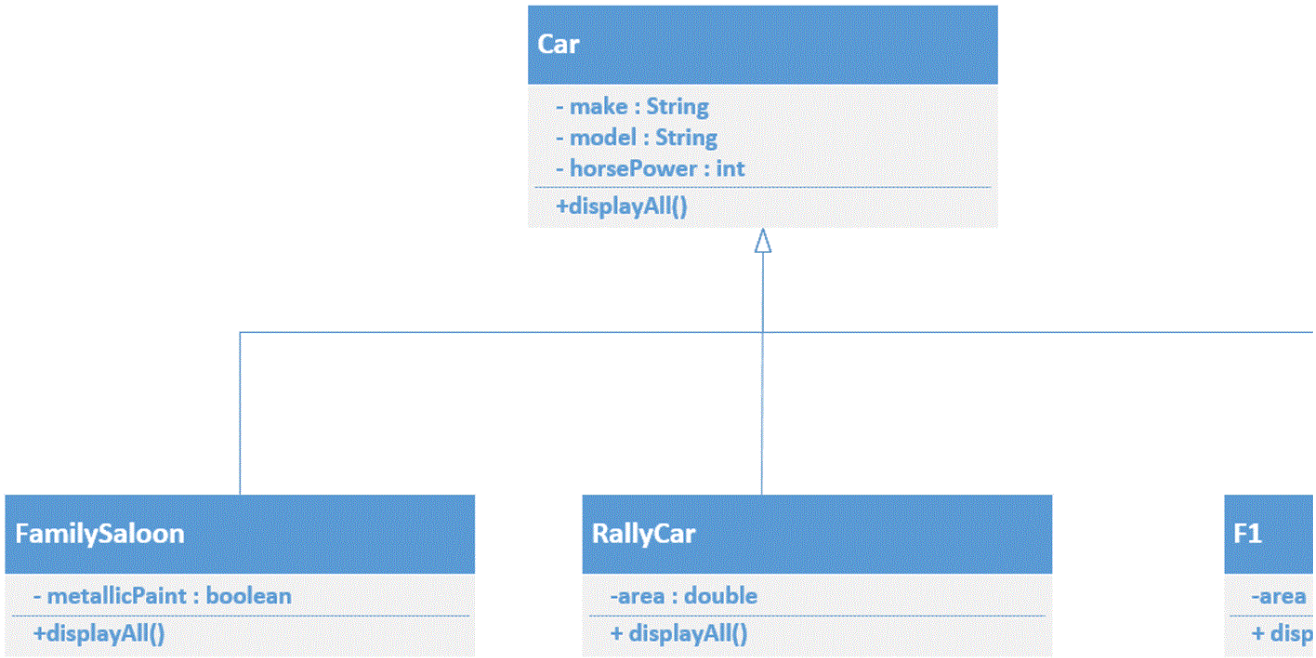

Figure 1 – UML class diagram

Implementation details

- The acceptable range for the ***horsepower*** value is between 0 and 1000 (inclusive). Any attempt to set the value outside this range should result in the value being set to ***-999***.

- **displayAll() :** The displayAll method for each class should display to screen all the instance variables for the particular class to screen. For example, for a ***FamilySaloon*** it should display the ***make, model, horsePower*** and An example output format for the **FamilySaloon** is shown in figure 2.

```
Family Saloon
Make     :Audi
Model    :A3
HorsePower       :130
metallicPaint    :true
```

Figure 2 – example output format from the displayAll() method

Part 2

Create a ***CarFactory*** class. Add your **name** and **student number** to the JavaDoc Code for this class.

Within the main method of the ***CarFactory*** class create several of the car objects and store each car object in an array. Use the array to help output the details for each object via the **displayAll**() method for each object. The Test data and expected output is shown in figure 3.

```
All cars

Family saloon
Make : Audi
Model : A3
HorsePower : 130
Metallic Paint : true


F1
Make : Ferrari
Model : Maranello
HorsePower : 2330
Area : 1.24


F1
Make : McLaren
Model : MCL2016
HorsePower : 2320
Area : 1.12


Rally car
Make : Subaru
Model : Impreza
HorsePower : 143
Area : 1.98


Family saloon
Make : Subaru
Model : Legacy
HorsePower : 123
Metallic Paint : false
```

Figure 3 - Test data to be used and expected output format

Add additional functionality to the **CarFactory** class to search the car objects:

1. To find the car(s) with the least ***horsePower***. Output only the ***model*** of the car(s).
2. Output **all details** of the car(s) of make
3. Output the average ***horsePower*** for **all** the car objects. (Display to two decimal places)

How to compress a Java file (you will need to do this for the assessment)


34 Compressing Java files