**Project Report**

GreenWheel Manufacturing System randomly generates existing car models and also include new cars using parts from existing cars in the inventory. After generation of vehicles the cars are the distributed to current 50 GreenWheel dealers.

Following things are kept in mind while designing the system

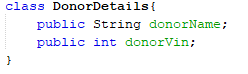
1. Existing cars are randomly generated
2. New car only takes maximum of two parts from same vehicle
3. Dealers are randomly and vehicles are added to their inventory
   1. A dealer contains min 5 vehicles.
   2. A dealer contains max of 10 vehicles.
4. No External libraries is allowed
5. No Database is maintained

**System Design**

**Classes**

* DonorDetails :

Keeps record of donor Vehicle.

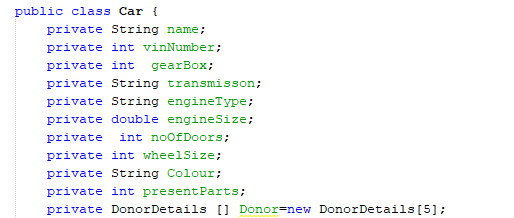


* Car :

Maintains all the needed information about the car and contains array of object of

Donor Details (Donor).

Contains a constructor, getters and setters for performing required operations.



* CarGeneration:

Maintains records of existing cars and randomly generates new vehicles using existing car parts. Uses 4 different Array of object Car.

* **Existing:**

Creates 10 model existing cars which are randomly generated

* **newModels:**

Creates 4 new model Cars using 10 **Existing** array of object.

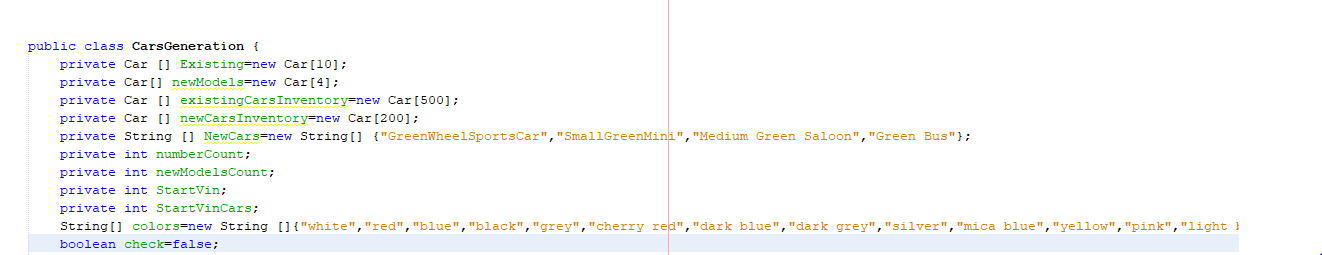
* **existingCarsInventory:**

Generates 500 existing cars using **Existing** array of object with different car vin number and color.

* **newCarsInventory:**

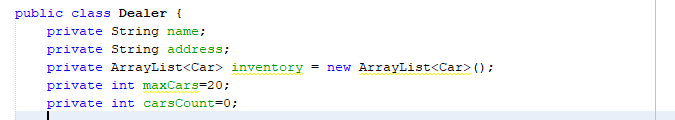
Generates 200 new cars using **newModels** array of object with different car vin number and color.

Contains methods AddCars (), GenerateNewModels (), GenerateExistingCars (), GeneratesNewCars () and Methods for displaying existing and new cars.



* Dealer:

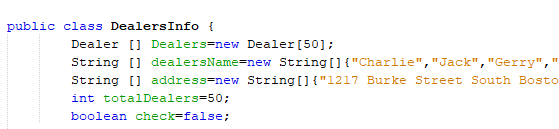
Maintains information about dealer name, address and cars present in the inventory.



* DealerInfo:

Maintains Information about all the dealer, randomly generates all dealers and distribute vehicles among the dealers.

Contains methods GenerateDealers (), DistributeExistingCars (), DistributeNewCars () and DisplayDealer ().



* ProductCreation:
* Contains Main Method
* Main method initialize Objects of CarsGeneration and DealersInfo
* Randomly Creates Existing Cars models and adds to Cars Generation.
* Calls functions of CarsGeneration to perform required operation like generating news car, show existing and new vehicles etc.
* Calls functions of DealerInfo to generate dealers, distribute cars and show vehicles in dealer inventory.

**Reason for design Choices:**

The following design pattern is to ensure simplicity,removes complications and keep in mind the real world scenerio of the system

1. DonorDetails

Contains Array of Objects of Size 5.

Object[0] contains the donordetails of gearBox.

Object[1] contains the donordetails of transmission.

Object[2] contains the donordetails of engine.

Object[3] contains the donordetails of doors.

Object[4] contains the donordetails of wheelSize.

1 class is made instead of 5 just to remove reduncency by creating 5 classes storing the same information.

Donor Array of DonorDetails present in Car to store information about the donor of specific part.

1. **Existing** and **newModels**  in CarGeneration

Existing contains 10 cars that are random generated and use as a sample for the generation of existing cars.

newModels contains 4 cars that are random generated using parts of existing vehicles and used as a sample for the generation of existing cars.

The reason behind choosing this design is that different cars with one model contains same specifcation with Vin and colour.

1. Creation of **CarsGeneration**

CarsGeneration is the most important class performing most of the functions under one roof. CarsGeneration method are called in main method and also used in DealersInfo for adding existing and new cars to inventories of dealers.