Class VehicleInfo:

base class that contains the basic vehicle information (that is common to all vehicles).

Class TruckInfo:

Class that contains the information for truck vehicle types.

Class CarInfo:

Class that contains the information for car vehicle types.

Enum eColor:

Enum that contains car color options.

Enum eNumOfDoosr:

Enum that contains car number of doors options.

Class MotorCycleInfo:

Class that contains the information for Motorcycle vehicle types.

Enum eLicenseType:

Enum that contains motorcycle license type options.

Class Vehicle:

base class that represents the basic vehicle structure, methods and information (that is common to all vehicles).

Class Wheel:

Class that represents the wheel information and methods to be

done on a wheel. (common for all vehicles)

Class FueledVehicle:

base class that represents the basic fueled vehicle structure, methods and information (that is common to all fueled vehicles).

Enum eFuelType:

Enum that contains fuel type options.

Class ElectricVehicle:

base class that represents the basic electric vehicle structure, methods and information (that is common to all electric vehicles).

Class FueledMotorCycle:

Class that represents fueled motorcycle, with all the relevant information and methods to be done on it.

Class ElectricMotorCycle:

Class that represents electric motorcycle, with all the relevant information and methods to be invoked.

Class FueledCar:

Class that represents fueled car, with all the relevant information and methods to be invoked.

Class ElectricCar:

Class that represents electric car, with all the relevant information and methods to be invoked.

Class FueledTruck:

Class that represents fueled truck, with all the relevant information and methods to be invoked.

Class VehicleFactory:

Class that is responsible for creating vehicle objects with specific types.

Enum eNumOfWheels:

Enum that contains number of wheels options, according to the vehicle type.

Enum eMaxWheelAirPressure:

Enum that contains maximum air pressure options, according to the vehicle type.

Enum eVehicleType:

Enum that contains vehicle type options.

Class Garage:

Class that is responsible for all the program functionality on the logical level.

Class GarageManager:

Inner class in the garage, that is responsible for all the program functionality on the logical level. Except of holding the vehicles information (garage holds it).

Class VehicleRecord:

Class that is responsible for holding the additional information of vehicles, that has relevance only in the garage.

Class FieldInfo:

Class that is responsible for holding all the relevant information inquiries for a specific vehicle.

Class LicensePlatesDTO:

Class that responsible for holding a list of license plates sorted by vehicle statuses in the garage.

Class VehcileInfoDTO:

Class that responsible for holding the relevant information about a vehicle

Class ValueOutOfRangeException:

Inherits from Exception. Responsible for out of range exceptions.

Class VehicleAlreadyExistsEsception:

Inherits from Exception. Logic according to the class name.

Class VehicleProperties:

API for UI to work with the vehicle properties

Class VehicleInfoProperties

API for UI to work with the vehicle Info properties

UI

Class Program:

Conventional entry point to the program.

Class GarageConsoleUI:

Wrapper class for the UI logic.

Class UserHandler:

Responsible for the communication with the user.

Class InputReader:

Responsible for getting input from the user.

Class InputValidator:

Responsible for validating input from the user.

Class OutputPrinter:

Responsible for most of the console prints.

