**Yaniv Zamir 206593444**

**Alon Bletter 318650751**

**Classes And Enums**

**Class: GarageManager**

*Description*:

The GarageManager class serves as the main interface for managing the garage, including registering vehicles, changing repair statuses, performing maintenance tasks, and retrieving vehicle information.

**Class: VehicleRepairRecord**

*Description*:

The VehicleRepairRecord class represents a record of vehicle repair, including details such as owner name, contact information, repair status, and the associated vehicle. It provides functionality for creating and managing repair records.

**Enums**: eRepairStatus

*Description*:

An enumeration representing different repair statuses of vehicles in the garage.

Fixed: Represents the status of a vehicle that has been fixed.

Repairing: Represents the status of a vehicle that is currently being repaired.

Paid: Represents the status of a vehicle for which repair has been paid.

This enumeration is used to track the repair status of vehicles in the garage system.

**Class: Vehicle**

*Description*:

The Vehicle class represents a generic vehicle with common attributes such as license plate, model name, and energy status. It serves as the base class for specific vehicle types, providing common functionality for managing vehicles.

**Class: VehicleCreator**

*Description*:

The VehicleCreator class provides functionality for creating new instances of vehicles based on the specified vehicle type and license plate. It also provides methods for retrieving available vehicle types and validating vehicle types.

**Class: Car**

*Description:*

The Car class represents a type of vehicle with specific attributes such as color and number of doors. It inherits from the Vehicle class and implements functionality for creating and managing car objects.

**Enum**: eCarColor

*Description*:

An enumeration representing different colors that a car can have.

Blue: Represents the color blue.

White: Represents the color white.

Red: Represents the color red.

Yellow: Represents the color yellow.

**Enum**: eCarNumberOfDoors

*Description*:

An enumeration representing the number of doors that a car can have.

Two: Represents a car with two doors.

Three: Represents a car with three doors.

Four: Represents a car with four doors.

Five: Represents a car with five doors.

**Class: Motorcycle**

*Description*:

The Motorcycle class represents a type of vehicle with specific attributes such as license type and engine volume. It inherits from the Vehicle class and implements functionality for creating and managing motorcycle objects.

**Enum**: eLicenseType

*Description*:

An enumeration representing different types of motorcycle licenses.

A1: Represents the A1 motorcycle license type.

A2: Represents the A2 motorcycle license type.

AB: Represents the AB motorcycle license type.

B2: Represents the B2 motorcycle license type.

**Class: Truck**

*Description*:

The Truck class represents a type of vehicle with specific attributes such as cargo volume and whether it is carrying hazardous substances. It inherits from the Vehicle class and implements functionality for creating and managing truck objects.

**Class: Engine**

Description:

The Engine class is an abstract class representing a generic engine. It serves as the base class for specific engine types, providing common functionality for managing energy sources.

**Class: FuelEngine**

*Description*:

The FuelEngine class represents an engine that operates on fuel. It inherits from the Engine class and provides functionality for managing fuel engines, including refueling and monitoring fuel levels.

**Enum**: eFuelType

*Description*:

An enumeration representing different types of fuel for engines.

Soler: Represents Soler fuel type.

Octan95: Represents Octan 95 fuel type.

Octan96: Represents Octan 96 fuel type.

Octan98: Represents Octan 98 fuel type.

**Class: ElectricEngine**

*Description*:

The ElectricEngine class represents an electric engine, which is a type of engine that operates using electrical energy. It inherits from the Engine class and provides functionality for managing electric engines, including monitoring battery status and recharging.

**Class: Wheel**

*Description*:

The Wheel class represents a wheel of a vehicle with attributes such as manufacturer name, current air pressure, and maximum air pressure.

**Class: InvalidEngineTypeException**

*Description*:

The InvalidEngineTypeException class is a custom exception used to indicate invalid actions performed on a specific type of engine.

**Class: ValueOutOfRangeException**

*Description*:

The ValueOutOfRangeException class is another custom exception used to indicate when a value falls outside of a valid range.

**Class: ConsoleUI**

*Description*:

The ConsoleUI class in your application serves as the user interface for interacting with the garage management system.

**Enums**: eGarageMenuOptions

*Description*:

An enumeration representing different options available in the garage system menu.

RegisterNewVehicle: Represents the option to register a new vehicle.

DisplayRegisteredVehiclesLicensePlates: Represents the option to display the license plates of all registered vehicles.

ChangeVehicleRepairStatus: Represents the option to change the repair status of a vehicle.

InflateVehicleWheelsToMax: Represents the option to inflate the air pressure of a vehicle's wheels to the maximum.

RefuelVehicleTank: Represents the option to refuel a vehicle's tank.

RechargeVehicleBattery: Represents the option to recharge a vehicle's battery.

DisplayVehicleInformation: Represents the option to display detailed information about a vehicle.

ExitSystem: Represents the option to exit the garage system.

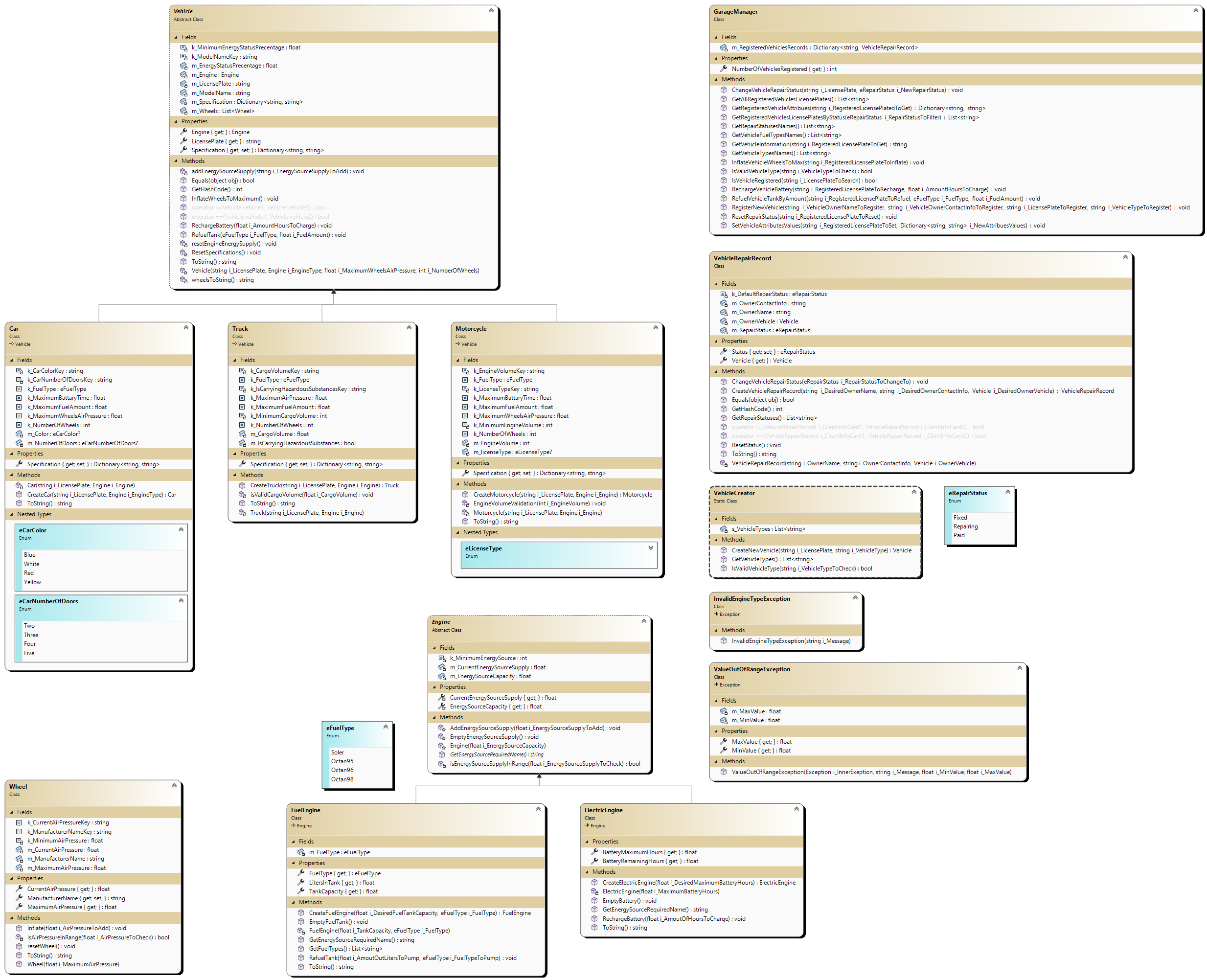
**Class: Program**

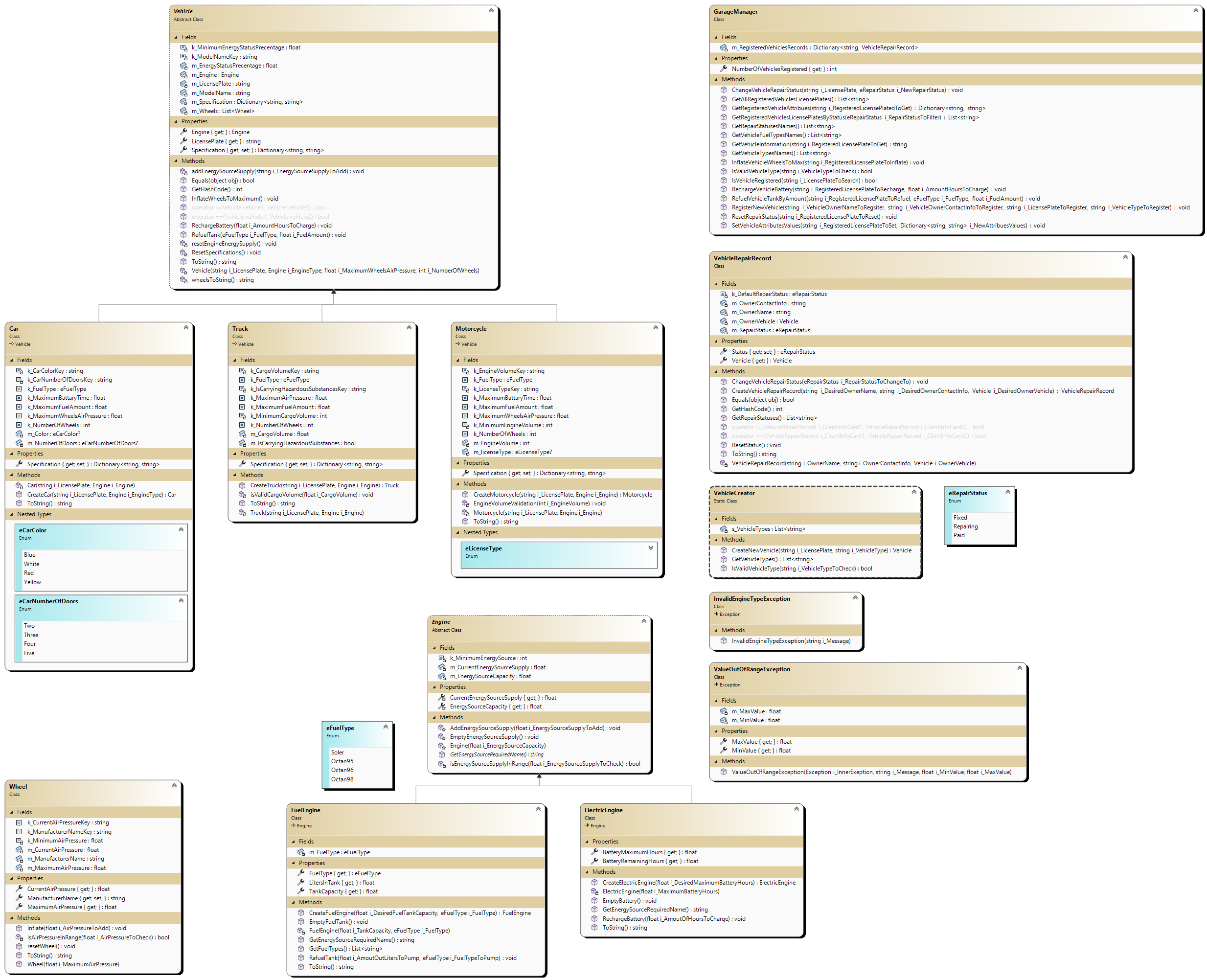
*Description*:

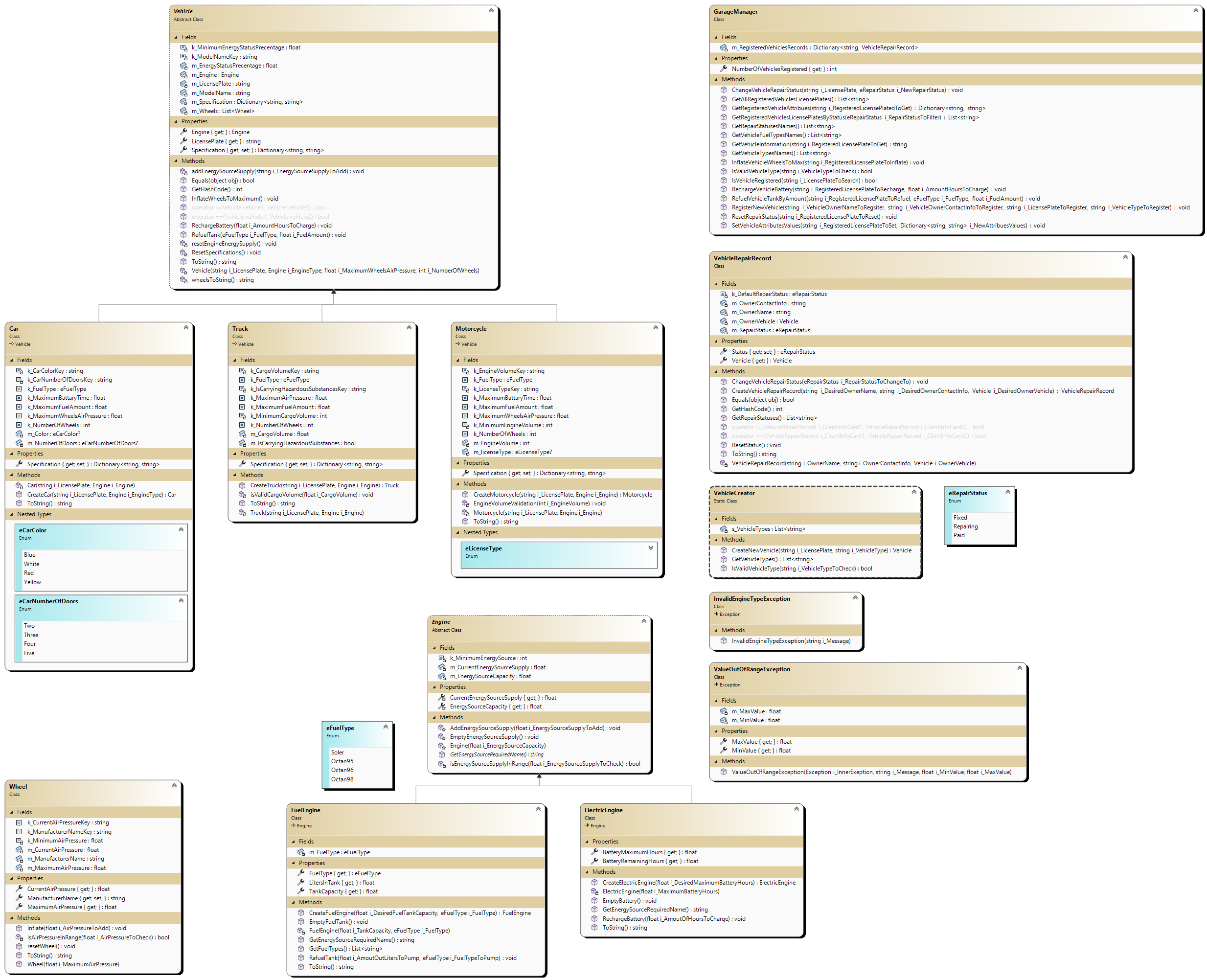
The Program class serves as the entry point for the application.

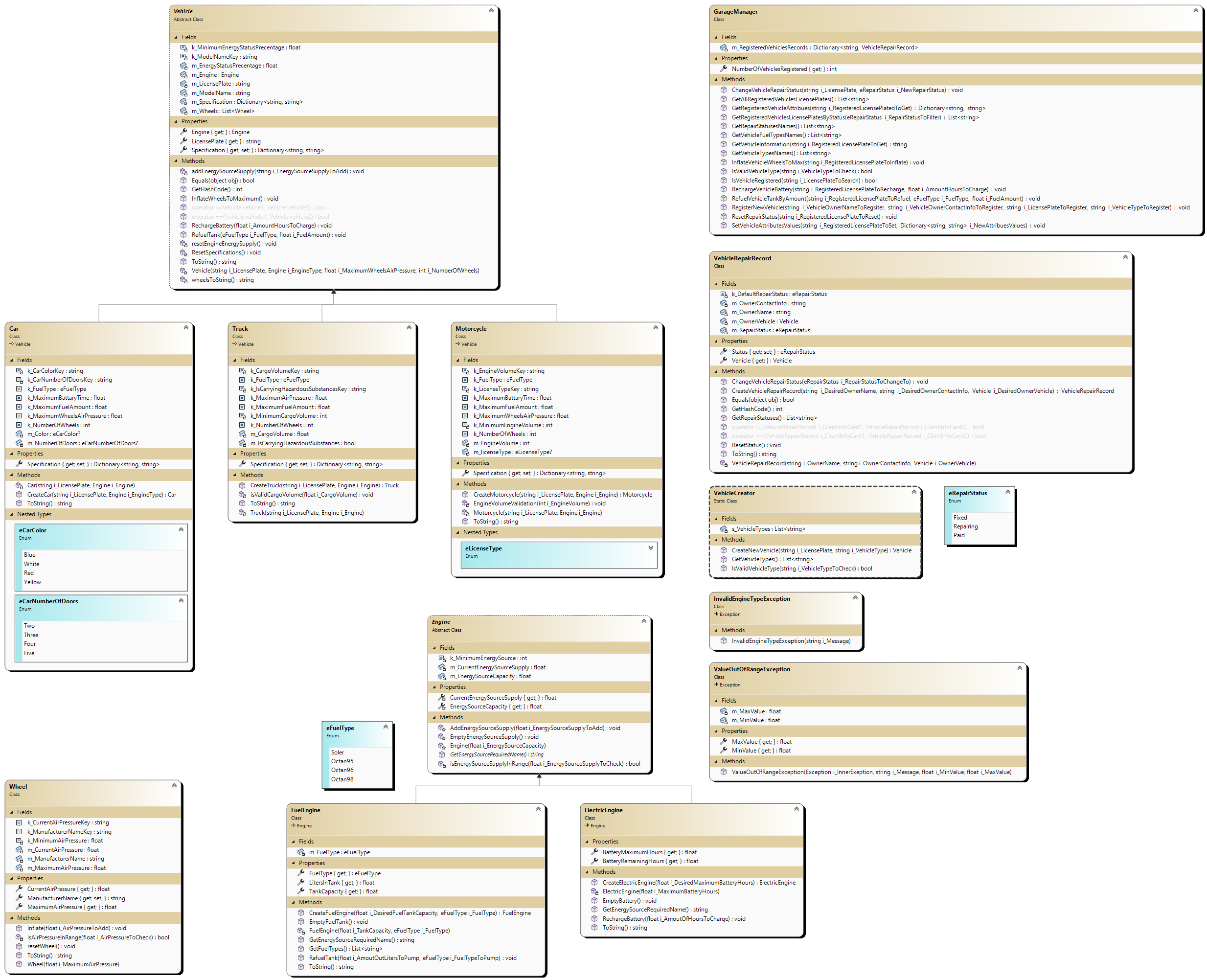
(Zoom in to see the diagram)

**GarageLogic UML Diagram**









**ConsoleUI UML Diagram**

