**GarageLogic:**

**Vehicle Hierarchy:**

* **Class Vehicle**: An abstract class of a Vehicle object, holds the fields common to all vehicles (Model name, Percent of energy left, License plate, Wheels, Engine).
  + **Class Car**: An abstract base class for car objects, inherits from vehicle and maintains unique car fields:
    - **Enum eCarColor** - car colors (Yellow, Black, White, Silver).
    - **Enum eDoorsAmount** - doors amount (Two, Three, Four, Five).
    - **Class FuelCar**: inherits from Car. Creates car with Octan95 fuel engine, 5 wheels with max pressure 32, and max fuel capacity 48 liters.
    - **Class ElectricCar**: from Car. Creates car with electric engine, 5 wheels with max pressure 32, and max battery time 4.8 hours.
  + **Class MotorCycle**: An abstract base class for motorcycle objects, inherits from vehicle and maintains unique motorcycle fields: engine volume.
    - **Enum eLicenseType** - license types (A, A2, AB, B2).
    - **Class FuelMotorcycle**: inherits from MotorCycle. Creates motorcycle with Octan98 fuel engine, 2 wheels with max pressure 30, and max fuel capacity 5.8 liters.
    - **Class ElectricMotorcycle**: inherits from MotorCycle. Creates motorcycle with electric engine, 2 wheels with max pressure 30, and max battery time 3.2 hours.
  + **Class Truck**: inherits from vehicle and maintains unique truck fields: cargo volume and boolean for dangerous materials transportation. Creates truck with Soler fuel engine, 12 wheels with max pressure 27, and max fuel capacity 135 liters.

**Engine Hierarchy:**

* **Class Engine**: An abstract class of an engine object, holds the fields common to all engines (max energy amount, energy amount left).
  + **Class FuelEngine**: inherits from engine, in charge of refueling vehicle and maintains unique fuel engine fields:
    - **Enum eFuelType** - fuel types (Soler, Octan95, Octan96, Octan98).
  + **Class ElectricEngine**: inherits from engine, in charge of charging the vehicle.

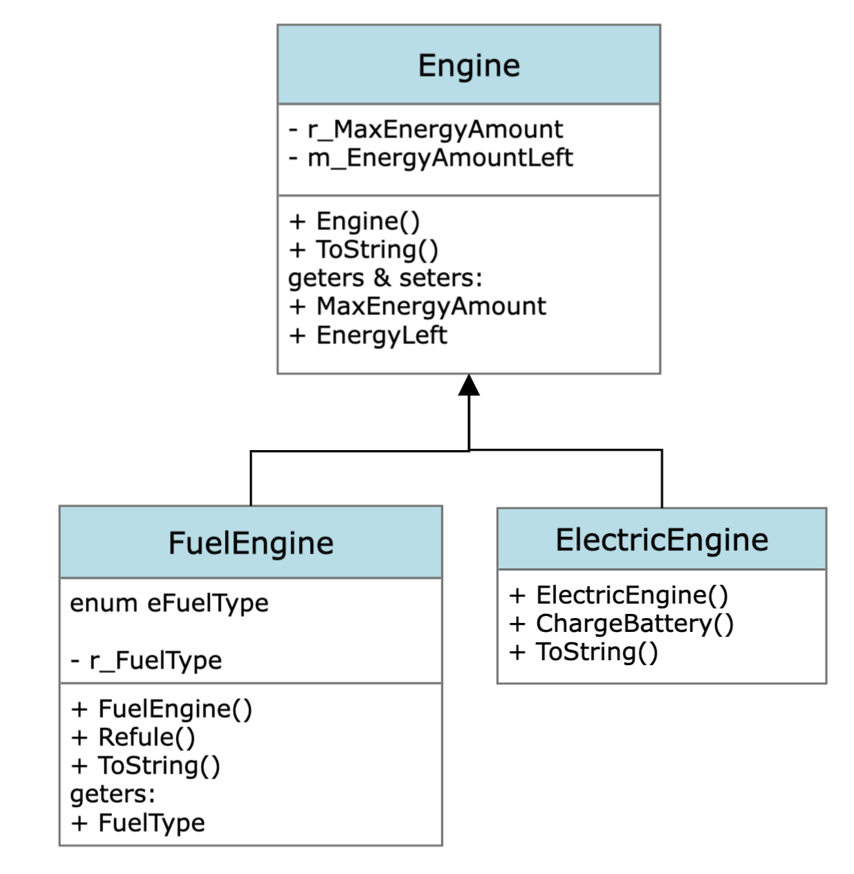
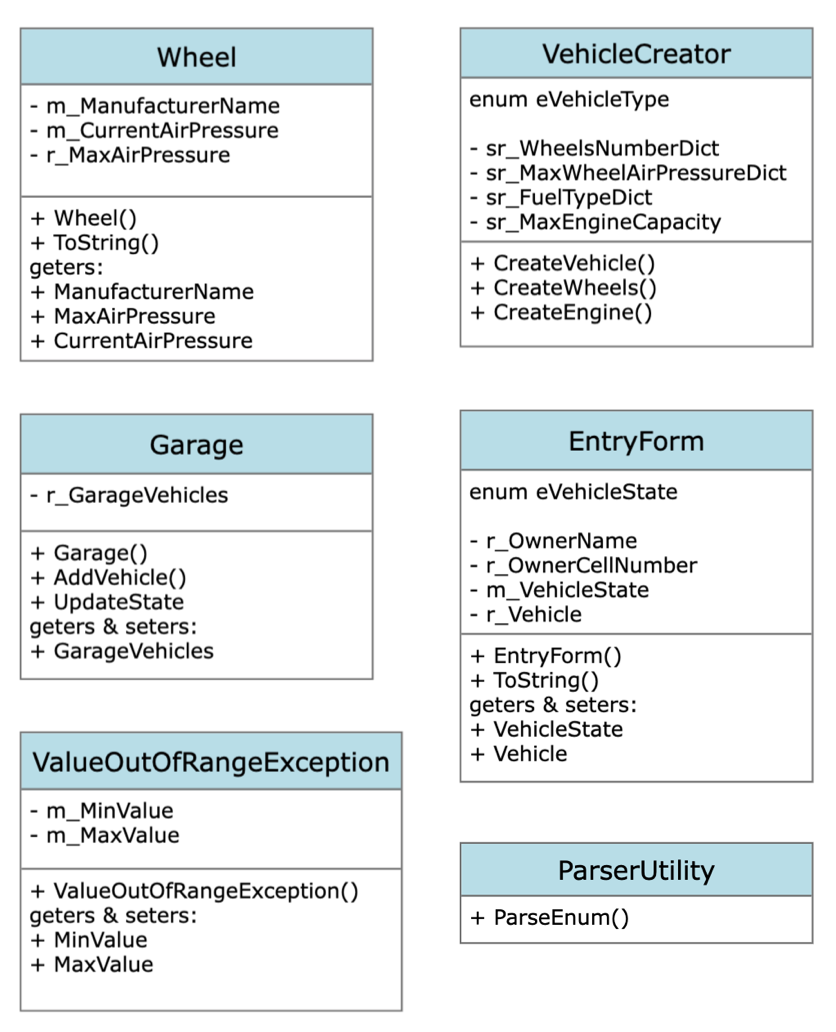
**Other Core Classes:**

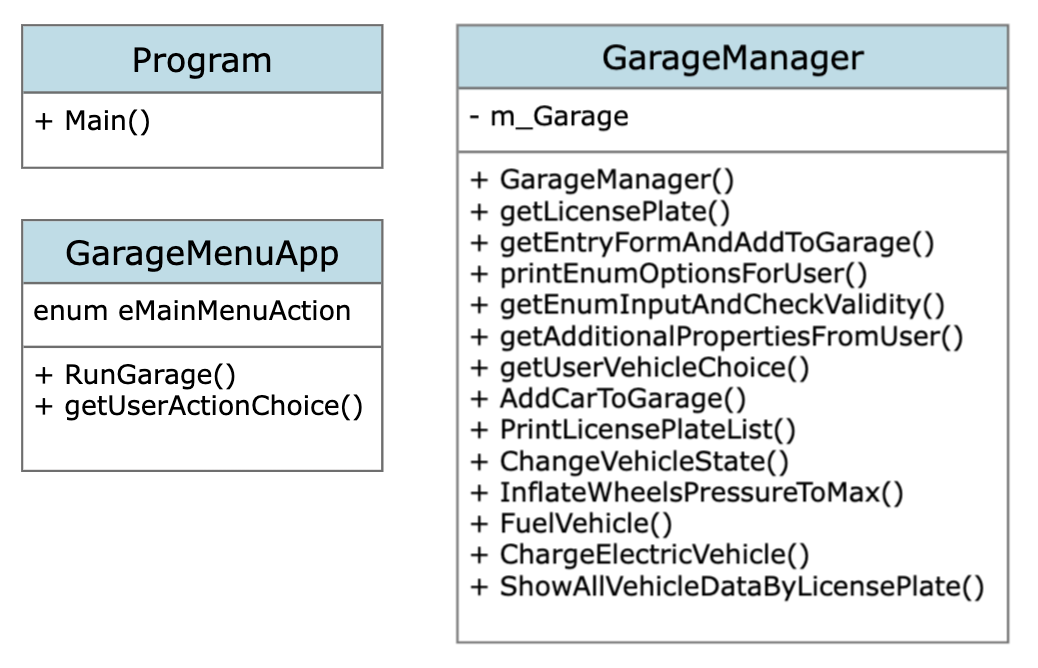
* **Class Wheel**: Holds fields common to all wheels: manufacturer name, current air pressure, max air pressure.
* **Class VehicleCreator**: An abstract class for creating vehicles using factory pattern. Contains static method CreateVehicle that creates specific vehicle instances (FuelCar, ElectricCar, FuelMotorcycle, ElectricMotorcycle, Truck) based on string type parameter. Also provides SupportedTypes list.
* **Class EntryForm**: A class for entry form object, meant to fill when a new customer comes to the garage. Holds the fields relevant for an entry form: owner name, owner phone number, owner's vehicle and:
  + **Enum eVehicleState** - vehicle states (UnderRepair, Fixed, Paid).
* **Class Garage**: Has a dictionary field that holds keys of license plates and values of entry forms. Provides methods for vehicle management including adding vehicles, changing states, inflating wheels, refueling, and charging.
* **Class GarageDataLoader**: A static class responsible for loading vehicle data from external files (such as Vehicles.db). Parses file format and creates appropriate vehicle instances using VehicleCreator.
* **Class ValueOutOfRangeException**: inherits from Exception class and meant to alarm when something is out of range. Contains MinValue and MaxValue properties.

**ConsoleUI:**

* **Class GarageManager**: Has a field of garage object and maintains all the functionality possible in the garage by using the GarageLogic project's classes. Provides methods for all garage operations including loading from file, adding vehicles, displaying information, and handling user interactions.
* **Class GarageMenuApp**: A static class for maintaining the garage menu, responsible for the flow of the program, initializes the GarageManager and activates it by calling its methods. Has also:
  + **Enum eMainMenuAction** - menu options (LoadFromFile, AddCar, ShowLicensePlates, ChangeVehicleState, InflateWheelsToMax, FuelVehicle, ChargeVehicle, ShowVehicleInfo, Exit).
* **Class Program**: initializes the project and has the main method that starts the garage application.

sGarageLogic:



ConsoleUI:

Adi Algazi 208000588

Tomer Agai 211906326