Liav Turkia 323081950

Dan Barak 227367455

**List of types**

* GarageLogic (project)
  + ValueOutOfRangeException (class) – Exception type thrown when a value was not within a certain range of values
  + Garage (namespace) – Logic for the actual garage and vehicle registration
    - Garage (class) – manages the registration of vehicles to owners and their statuses
    - Owner (class) – Represents the information of an owner who can register a vehicle in the garage
    - VehicleRegistration (class) – Represents a registered vehicle in a garage, with the actual vehicle and the owner who registered it
      * eVehicleStatus (enum) – Potential status values for a registered vehicle in a garage, including None for use in the filtering
  + Vehicle (namespace) - Logic for the Vehicle class
    - Components (namespace) – Components that make up a Vehicle
      * Engine (abstract class) – Base class for an Engine component, which the vehicle uses
      * ElectricEngine (class) – Subclass of Engine that represents an engine that runs on electricity, and it’s associated data
      * FuelEngine (class) – Subclass of Engine that represents an engine that runs on fuel, and it’s associated data
        + eFuelType (enum) – Which type of fuel the FuelEngine uses
      * Wheel (class) – Represents a Wheel component for the vehicle, and also stores it’s specification and pressure (current and max)
    - Requirements (namespace) – Series of classes that are used to bind a requirement to user-provided data (to generate vehicles)
      * PropertyRequirement (abstract class) – Base class for a property requirement, which specifies a type of constraint that a required property needs on its value. Used to validate user input dynamically.
      * TypeRequirement (class) – A PropertyRequirement that requires the value to be of a certain type
      * MinimumRequirement (class) – A TypeRequirement that requires the value to be greater or equal to some minimum value
      * RangeRequirement (class) – A TypeRequirement that requires the value to be within a range [MinValue, MaxValue]
    - Types (namespace) – Collection of types that generate a type of vehicle, used in VehicleFactory to define new preset vehicle types
      * VehicleType (abstract class) – Base class for vehicle types, and contains a method to generate the preset vehicle (with an input of user supplied properties) and a method to get the required properties (to prompt the user) as every vehicle type needs different properties (for example, fuel vs electric engines require different input). The Engine component is independent of VehicleType, as well as tire specifications, however the number of wheels is based on VehicleType.
      * CarBasedType (class) – A VehicleType that generates a Car and requires the properties needed to generate Cars
      * MotorcycleBasedType (class) – A VehicleType that generates a Motorcycle and requires the properties needed to generate Motorcycle
      * TruckBasedType (class) – A VehicleType that generates a Truck and requires the properties needed to generate Truck
    - Vehicle (abstract class) – Base class for all types of Vehicles, and contains Engine and Wheel as components
    - Car (class) – A type of Vehicle that represents a car, with the corresponding car-specific member properties
      * eCarColor (enum) – Possible color that a Car can be
    - Motorcycle (class) – A type of Vehicle that represents a motorcycle, with the corresponding motorcycle-specific member properties
      * eLicenseType (enum) – Type of license that this motorcycle requires to drive
    - Truck (class) – A type of Vehicle that represents a Truck with the corresponding truck-specific member properties
    - VehicleFactory (class) – Factory type that generates Vehicles and requirements for different preset VehicleTypes. To add new vehicle types, this class is the only place that needs minor code additions
      * eVehicleType (enum) – The current defined vehicle types that VehicleFactory can generate
* ConsoleUI
  + Program (class) – Entry point to the program, creating the UI and Garage types.
  + ConsoleUI (class) – Class that handles the front-end console based UI to manage the garage, following the client specification



