

# Programming II

## Exercise

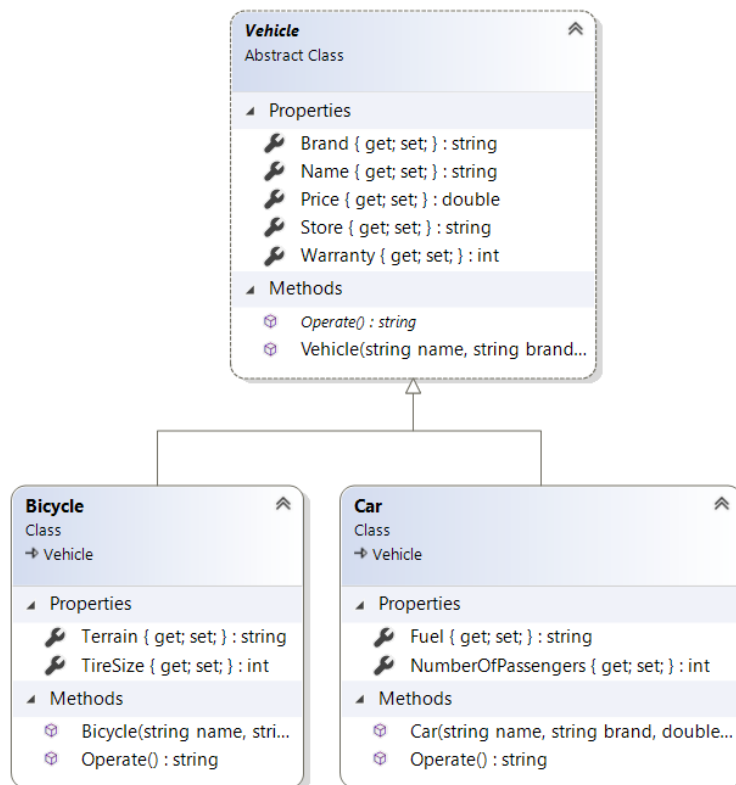
You will be creating a Windows Application that will allow users to create a vehicle. It can be a Bike or a Car.

After the vehicle gets created it is added to a List of vehicles.

## Details

**STEP 1** Create a “Windows Forms App” project called **VehicleApp**;

**STEP 2** Implement the following 3 classes in the project:



## Vehicle Class

This class is an *Abstract* class used to define general information about the vehicle. It has two derived classes: **Bicycle** and **Car**

**public Vehicle(string name, string brand, double price, int warranty, string store)** – This constructor takes 5 arguments;

**public abstract string Operate()** – Abstract method. It should be implemented in the derived classes;

## Bicycle Class

This class is derived from the abstract **Vehicle** class.

**public Bicycle(string name, string brand, double price, int warranty, string store, int tireSize, string terrain)** – This constructor takes 7 arguments;

**public string Operate()** – This method should return the message "You need to pedal!".

## Car Class

This class is derived from the abstract **Vehicle** class.

**public Car(string name, string brand, double price, int warranty, string store , int numberOfPassengers, string fuel)** – This constructor takes 7 arguments;

**public string Operate()** – This method should return the message "You turn on the ignition!".

### STEP 3 Create the following GUI:

The screenshot shows a Windows application window titled "Vehicles". Inside the window, there is a "Vehicle Information" section. This section contains several input fields: "Name:", "Brand:", "Price:", "Warranty:", and "Store:". To the right of these fields, there are two conditional sections. The "For Cars" section includes a "Passengers:" spin box (set to 0) and a "Fuel:" dropdown menu. The "For Bikes" section includes a "Tire Size:" spin box (set to 0) and a "Terrain:" dropdown menu. To the right of these sections is a "Type of Vehicle" section with two radio buttons: "Car" (which is selected) and "Bicycle". At the bottom right of the window, there is a "Create Vehicle" button.

### Details about how the GUI should work:

- Provide proper names to the GUI controls
- Declare and create a private field called **vehicles** in the **Form** class of type **List<Vehicle>**. This list will store the **Vehicle** object once it gets created.
- Use a ComboBox for **Fuel** with the following options: *Gas, Diesel, Electric, Hybrid*.
- Use a ComboBox for **Terrain** with the following options: *Paved, Unpaved*.
- If **"Car"** radio button is selected, the **"For Cars"** groupbox should be enabled and the **"For Bikes"** groupbox should be disabled;
- If **"Bicycle"** radio button is selected, the **"For Cars"** groupbox should be disabled and the **"For Bikes"** groupbox should be enabled;
- When the user clicks on **"Create Vehicle"** the following needs to happen:
  - o A **new vehicle** will be created, and the information filled in the GUI will be used to set the vehicle properties;
  - o Use a **try..catch** block to handle any invalid input, like when the user enters text in the Warranty or Price – Show a message to let the user knows about the input problem;
  - o Add the newly created vehicle to the **vehicles** List. This List should store the list of all vehicles getting created;
  - o Clear all input controls and set the selected type of vehicle to **"Car"**;
  - o Use **MessageBox.Show()** to show the total number of vehicles stored so far in the **vehicles** List;