**1. Create a Vehicle Class**

* შექმენით კლასი Vehicle შემდეგი ფროფერთიებით:
  + brand (string)
  + model (string)
  + year (number)
  + mileage (number, default is 0)
* დაამატეთ მეთოდები:
  + drive(miles) – ზრდის ობიექტის არსებულ მილაჟს გადაცემული რაოდენობით.
  + getVehicleInfo() – აბრუნებს სტრინგს ფორმატით: "Brand: <brand>, Model: <model>, Year: <year>, Mileage: <mileage>".

**2. Inheritance with a Car Class**

* დააექსტენდეთ Vehicle კლასი და დაარქვით Car.
  + დაამატეთ ახალი ფროფერთიები: fuelType (string)
* შექმენით ახალი getVehicleInfo() მეთოდი რომელიც სტრინგში ჩაწერს fuelType ფროფერთისაც.

**3. Static Methods in a MathOperations Class**

* შექმენით კლასი MathOperations რომელიც მხოლოდ სტატიკურ მეთოდებს შეიცავს:
  + add(a, b) – აბრუნებს ჯამს.
  + subtract(a, b) – აბრუნებს სხვაობას.
  + multiply(a, b) – აბრუნებს ნამრავლს.
  + divide(a, b) – აბრუნებს განაყოფს, თუ b უდრის 0-ს აბრუნებს სტრინგს: “can not divide by 0”
* გამოიძახეთ ეს მეთოდები (ობიექტის შექმნის გარეშე).

**4. Encapsulation with Getters and Setters**

* შეცვალეთ Vehicle კლასი რომ შეიცავდეს private ფროფერთი #mileage - ს
* შექმენით getMileage და setMileage მეთოდები set ში შეამოწმეთ რომ მხოლოდ დადებითი რიცხვი მიენიჭოს #mileage -ს.

**5. Polymorphism with an ElectricCar Class**

* შექმენით Vehicle კლასის შვილობილი და დაარქვით ElectricCar რომელიც შეიცავს დამატებით ფროფერთის - batteryCapacity (ბატარეის ტევადობა).
* თავიდან აღწერეთ ამ კლასში drive(miles) მეთოდი რომ ყოველ 10 მილზე 1 ით შეამციროს batteryCapacity.