מגיש 1: אברהם טלקר 305758880

מגיש 2: סהר חלצי 308235407

# הסבר מחלקות

## EX03.GarageLogic

1. Vehicle- מחלקה אבסטרקטית לייצוג רכב המכילה נתונים משותפים לגבי כלי רכב שונים
2. Wheel- מייצגת גלגל ברכב
3. Engine- מחלקה אבסטרקטית שמייצגת מנוע ומכילה נתונים משותפים לגבי מנועים שונים
4. FuelEngine- מחלקה המממשת מנוע הפועל על דלק. יורשת ממחלקת Engine
5. ElectricEngine- מחלקה הממשמת מנוע הפועל על חשמל יורשת ממחלקת Engine
6. Car- מחלקה המייצגת רכב מסוג מכונית. יורשת ממחלקת Vehicle.
7. Motorcycle- מחלקה המייצגת רכב מסוג אופנוע. יורשת ממחלקת Vehicle.
8. Truck – מחלקה המייצגת רכב מסוג משאית. יורשת ממחלקת Vehicle.
9. GarageManager- מחלקה המנהלת את המוסך. דרכה ניתן להוסיף רכבים חדשים ולטפל בהם. זוהי המחלקה המייחצנת את מרבית הפעולות עבור המשתמש.
10. OwnerInfo- מחלקה המייצגת נתונים על בעל הרכב
11. GarageEntity- מחלקה המייצגת ישות במוסך. מכילה את פרטי בעל הרכב ואת הרכב.
12. WheelsFactory- מחלקה סטטית האחראית על יצירת אובייקטים מסוג wheel
13. EngineFactory- מחלקה סטטית האחראית על יצרת אובייקטים מסוג Engine.
14. VehicleFactory- מחלקה סטטית האחראית על יצירת סוגי הרכבים השונים. המחלקה נעזרת בשאר מחלקות ה factory על מנת להרכיב את הרכב השלם.
15. ValueOutOfRangeException- מחלקה המייצגת exception המתאר חריגה מגבולות מוגדרים עבור טיפוס כלשהו.
16. Enum- EngineTypes- מייצג סוגי מנועים שונים
17. Enum- FuelTypes- מייצג סוגי דלק שונים
18. Enum- LicenseType- מייצג סוגי רישיונות שונים עבור אופנוע
19. Enum- VehicleColors – מייצג צבעים שונים עבור רכב מסוג מכונית
20. Enum- VehicleStatuses- מייצג את הסטאטוס של הרכב ב garage

## EX03.ConsoleUI

1. Program – מחלקת ה main של הפרויקט. מהווה entry point
2. ApplicationManager- אחראית על הפעלת ה garageManager ואחראית לתקשורת מול המשתמש.
3. MenusPrinter- אחראית על הדפסת התפריטים השונים
4. CustomConverter- אחראית על המורות שונות של קלט המשתמש לאובייקטים בתכנית.

**ElectricEngine**

+ Charge(float)

**GarageEntity**

- m\_VehicleStatus  
- m\_VehicleEntity  
- m\_OwnerContactInfo

+ GetVehicleInformationForm()

**GarageManager**

- m\_GarageVehicles

+ InsertVehicleToGarge()  
+ GetLicenseByVehicleStatus(eStatus)  
+ InflateWheels(string)  
+ ChangeVehicleStatus(string, estatus)  
+ FuelVehicle(string)  
+ ChargeVehicle(string, float)  
+ GetVehicleInformationForm(string)  
+ IsVehicleExist(string)  
- checkIfVehicleExist(string)  
-checkIfFillable()

**Wheel**

- m\_Manufacturer  
- m\_AirPressure  
- m\_MaximumAirPressue

+ Inflate(float)

**Vehicle**

- m\_model  
- m\_LicenseNumber  
- m\_Wheels  
- m\_Engine

+ InflateWheels()  
+ GetEngineType()

**Motorcycle**

- m\_LicenseType  
- m\_EngineVolume

**Truck**

- m\_IsTrunckCold  
- m\_TrunkVolume

**Car**

- m\_VehicleColor  
- m\_DoorsNumber

**Engine**

- m\_EnergyLeft  
- m\_EnergyCapacity

+ FillEnergy(float)

**FuelEngine**

- m\_FuelType

+ FillFuel(eFuelTypes, float)

**OwnerInfo**

- m\_Name  
- m\_PhoneNumber

**MenusPrinter (static)**

+ PrintMainMenu()  
+ PrintFuelTypesMenu()  
+ PrintColorsMenu()  
+ PrintLicenseTypesMenu()  
+PrintVehiclceTypeOptions()  
+PrintStatusFilterMenu()  
+PrintVehicleStatuses()

**VehicleFactory (static)**

+ CreateCar()

+ CreateMotorcycle()  
+ CreateTruck()

**WheelsFactory (static)**

+ CreateWheels()

**EngineFactory (static)**

+ CreateEngine()

- createFuelEngine()  
- createElectricEngine()

**CustomConverter (static)**

+ ConvertStringToIsColdBool()  
+ ConvertStringToFloat()  
+ ConvertStringToInt()  
+ ConvertInputVehicleTypeOption()  
+ConvertToValidNumberOfDoors()  
+ConvertStringColorToVehicleColor ()  
+ConvertStringToVehicleStatus()  
+ConvertStringToVehicleStatusFiltering()  
+ConvertStringToLicenseType()  
+ConvertInputToChoice()