**מגישים:**

אורי לרמן 205665508

אמיר נדב 305707044

**הטיפוסים שהגדרנו:**

* class VehicleFactory – מחלקה שתפקידה לייצר אובייקטים מטיפוס Vehicle, אחראי עם יצירת אובייקטים של כלי רכב עבור כל סוג נתמך.
* class VehiclesGarage – מחלקה שמחזיקה את מבנה הנתונים של המוסך. כל הרכבים מאוחסנים ב- Dictionary ובנוסף ישנם שלושה מערכי List עבור כל סטטוס שונה במוסך.
* abstract class Vehicle – מחלקה שמייצגת רכב כלשהו, מחזיקה את המידע המשותף לכל כלי הרכב (מספר רישוי למשל).
* enum eRepairState – מייצג את המצבים האפשריים עבור רכב במוסך.
* enum eVehicleWheels – מחזיק את האפשרויות למספרי גלגלים שונים בין רכבים.
* class Car – מכילה נפח אוויר/רמת דלק/רמת סוללה מקסימליים, צבע ומספר דלתות.
* enum eCarColor – מחזיק את האפשרויות לצבעים שונים עבור מכונית
* enum eNumberOfDoors – מס' הדלתות האפשרי במכונית.
* class Truck – מחזיקה נתונים על לחץ האוויר ורמת הדלק, האם תא המטען מקורר.
* enum eIsCooled – האם תא המטען של המשאית מקורר.
* class Motorcycle – מחזיקה דרגת רישיון ונפח מנוע של האופנוע
* enum eLicenseType – מחזיק את מספר האפשרויות שיש עבור רשיון נהיגה באופנוע
* class Wheel – מחזיקה את נפח האוויר המקסימלי, הנוכחי ויצרן הצמיג
* abstract class EnergyType – מייצגת סוג של מנוע
* class FuelEngine – סוג דלק, כמות מקסימלית, יחס דלק במיכל, בעלת מתודת מילוי
* enum eFuelType – מחזיק את סוגי הדלק האפשריים
* class ElectricityEngine – משך זמן מקסימלי ומשך זמן נותר, יחס זמן בסולל בעלת מתודת מילוי.
* Class VehicleInitialDetails – מייצג את כל הנתונים אותם יש לקלוט על מנת לייצר רכב, מכיל בתוכו מספר רב של אובייקטים
* Class CarInfo – נתונים אשר יש לקלוט עבור רכב
* Class TruckInfo וגם class MotorcycleInfo – מחזיקים את הנתונים הנ"ל
* Class EnergyTypeInfo – נתונים עבור מנוע כלשהו
* Class WheelInfo – נתונים אודות גלגל
* Class OwnerDetails – מייצגת את כל הנתונים עבור הבעלים של הרכב.
* Class GarageManagerForConsole – מנהלת את כל ממשק המשתמש, אחראית על קלט ופלט.

**דיאגרמה**

abstract class EnergyType

=

+abstract void CalculateRatio()

class ElectricityEngine

-readonly float MaxBattery

-float currentBattery

-float BatteryRatio

+void CalculateRatio()

+ctor

+ReCharge

class FuelEngine

-eFuelTypes fuelType

-float currentFuelAmount

-float fuelRatio

+void CalculateRatio()

+ctor

+Refuel(eFuelType, float)

class Truck

+const float MaxTruckPsi

+const float MaxTruckFuel

+const float MaxTruckBattery

-eLicenseTypes

-int EngineVolume

+ctor

+vehicleMaxPressure

class Motorcycle

+const float MaxMotorcyclePsi

+const float MaxMotorcycleFuel

+const float MaxMotorcycleBattery

-eLicenseTypes

-int EngineVolume

+ctor

+vehicleMaxPressure

class Car

+const float MaxCarPsi

+const float maxCarFuel

+const float MaxBatteryHours

-eCarColors

-eNumberOfDoors

+ctor

+vehicleMaxPressure

abstract class Vehicle

-readonly string m\_model

-readonly string m\_LicenseNumbrer

-eRepairState m\_RepairState

#list<wheel>

#OwnerDetails

#EnergyType

+ctor

+RepairState(Property)

+GetEnergySourece(Property – only get)

+fillAirToMax (string)

+LicenseNumber (Property)

+VehicleMaxPressure(**abstract property**)