internal class Wheel

private readonly string
r_ManufacturerName;
private float m_CurrentAirPressure;
private readonly float
r_MaxAirPressure;

internal Wheel(string
i_ManufacturerName, float
i_CurrentAirPressure, float
i_MaxAirPressure)
internal void InflateAction(float
i_AirToAdd)
internal string ManufacturerName
internal float CurrentAirPressure
internal float MaxAirPressure
public override string ToString()

public class VehicleCreator

public static VehicleInGarage
CreateVehicleInGrage(eRepairableCarType
i_RepairableCarType, string i_OwnerName, string
i_OwnerPhoneNumber, string i_ModeName, string
i_LicenseNumber)

internal class Vehicle

private readonly string r_ModelName;
private readonly string r_LicenseNumber;
internal float m_RemainingEnergyPercentage=0;
protected List<Wheel> m_Wheels;
protected float m_MaxAirPressure;

internal Vehicle(string i_ModeName, string i_LicenseNu
internal void AddWheels(string[]
i_ManufacturersOfWheelsName, float[]
i_CurrentAirPressureOfTheWheels)
internal string GetWheelsInfo()
internal string Model
internal string LicenseNumber
internal List<Wheel> Wheels
public override string ToString()

public class Garage

private readonly Dictionary<string, VehicleInGarage>
r_DictionryOfVehiclesInGarage = new Dictionary<string,
VehicleInGarage> { };
private List<eRepairableCarType>
m_ListOfVehiclesThatCanBeHandled = new
List<eRepairableCarType>();

public Garage()
public void AddVehicleToGarage(string
i_LicenseNumber, VehicleInGarage i_VehicleInGarage)
public string ListOfLicenseNumberOfAnyStatus()
public string
ListOfLicenseNumberOfASpecificStatus(eVehicleStatus
i_VehicleStatus)
public void UpdateVehicleStatus(eVehicleStatus
i_VehicleStatus, string i_LicenseNumber)
public void InflateTiresToMaximum(string
i_LicenseNumber)

Ex03.ConsoleUI

class GarageManager

readonly GarageLogic.Garage
r_Garage=new
GarageLogic.Garage();

public void StartManu()
internal void
AddVehicleToGarage()
internal void
AddMotorcycleToGarage(string
i_LicenseNumber)
internal void
AddCarToGarage(string
i_LicenseNumber)
internal void
AddTruckToGarage(string

public enum
eManuOption

One=1,
Two=2,
Three=3,
Four=4,
Five=5,
Six=6,
Seven=7,
Eight=8

static class Program

public static void Main()

internal class FuelBasedVehicle :
Vehicle

protected eFuelType m_FuelType;
protected float m_MaxAmountOfFuel;

internal FuelBasedVehicle(string
i_ModeName, string
i_LicenseNumber) :
base(i_ModeName,
i_LicenseNumber)
internal float CurrentAmountOfFuel
internal void Refueling(float
i_FuelToAdd, eFuelType i_FuelType)
public override string ToString()

internal class ElectricVehicle : Vehicle

protected float m_MaxTimeOfEngineOperation;

internal ElectricVehicle(string i_ModeName, string
i_LicenseNumber) : base(i_ModeName, i_LicenseNumb
internal float RemainingTime
internal void Recharging(float i_HoursToCharge)
public override string ToString()

internal class FuelTruck :
FuelBasedVehicle

private Truck m_Truck;
private const float
k_MaxAmountOfFuelOfTruck = 135;
private const eFuelType k_FuelType =
eFuelType.Soler;

internal FuelTruck(string i_ModeName,
string i_LicenseNumber)
internal void AddTruck(bool
i_DoesItContainDengerousMaterials,
float i_CargoTankVolume)
internal bool
DoesItContainDengerousMaterials
internal float CargoTankVolume
public override string ToString()

internal class FuelCar : FuelBasedVehicle

private Car m_Car;
private const float
k_MaxAmountOfFuelOfCar = 46;
private const eFuelType k_FuelTypeOfCar
= eFuelType.Octan95;

internal FuelCar(string i_ModeName,
string i_LicenseNumber)
internal void AddCar(eCarColor i_Color,
eNumberOfDors i_NumberOfDors)
internal eCarColor Color
internal eNumberOfDors NumberOfDors
public override string ToString()

internal class FuelMotorcycle :
FuelBasedVehicle

private Motorcycle m_Motorcycle;
private const float k_MaxAmountOfFuel
=6.4f;
private const eFuelType k_FuelType =
eFuelType.Octan98;

internal FuelMotorcycle(string
i_ModeName, string i_LicenseNumber)
internal void
AddMotorcycle(eLicenseType
i_LicenseType, int i_EngainVolume)
internal eLicenseType LicenseType
internal int EngineVolume
public override string ToString()

internal class ElectricMotorcycle :
ElectricVehicle

private Motorcycle
m_Motorcycle;
private const float
k_MaxTimeOfEngineOperation
OfMotorcycle = 2.6f;

internal
ElectricMotorcycle(string
i_ModeName, string
i_LicenseNumber)
internal void
AddMotorcycle(eLicenseType
i_LicenseType, int
i_EngainVolume)
internal eLicenseType
LicenseType
internal int EngineVolume

internal class ElectricCar :
ElectricVehicle

private Car m_Car;
private const float
k_MaxTimeOfEngineOperation
OfCar= 5.2f;

internal ElectricCar(string
i_ModelName, string
i_LicenseNumber)
internal void AddCar(eCarColor
i_Color, eNumberOfDors
i_NumberOfDors)
internal eCarColor Color
internal eNumberOfDors
NumberOfDors
public override string ToString()

internal struct Car

private readonly eCarColor
r_CarColor;
private readonly
eNumberOfDors
r_NumberOfDors
public static float
s_MaxAirPressure = 33;

internal Car(eCarColor
i_CarColor, eNumberOfDors
i_NumberOfDors)
internal eCarColor Color
internal eNumberOfDors
NumberOfDors
internal string InfoAboutCar()

internal struct Motorcycle

private readonly
eLicenseType
r_LicenseType;
private readonly int
r_EngineVolume;
public static float
s_MaxAirPressure = 33;

internal
Motorcycle(eLicenseType
i_LicenseType, int
i_Volume)
internal eLicenseType
LicenseType
internal int EngineVolume
internal string
InfoAboutMotorcycle()

internal struct Truck

private bool
m_DoesItContainDengerou
sMaterials;
private readonly float
r_CargoTankVolume;
public static float
s_MaxAirPressure = 26;

internal Truck(bool
i_DoesItContainDengerous
Materials, float i_Volume)
internal float
CargoTankVolume
internal bool
DoesItContainDengerousM
aterials
internal string
InfoAboutTruck()

public class VehicleInGarage

private readonly Vehicle r_Vehicle;
private readonly string r_OwnerName;
private readonly string
r_OwnerPhoneNumber;
private eVehicleStatus m_VehicleStatus;

internal VehicleInGarage(Vehicle i_Vehicle,
string i_OwnerName, string
i_OwnerPhoneNumber, eVehicleStatus
i_VehicleStatus)
internal eVehicleStatus VehicleStatus
internal Vehicle Vehicle
public override string ToString()

class
ValueOutOfRangeExcep
tion : Exception

float MaxValue;
float MinValue;

public
ValueOutOfRangeExcep
tion(float i_MaxValue,
float i_MinValue)
public
ValueOutOfRangeExcep
tion(string i_Message,
float i_MaxValue,float
i_MinValue)

public enum
eRepairableCarType

ElectricMotorcycle = 1,
ElectricCar = 2,
FuelBasedMotorcycle =
3,
FuelBasedCar = 4,
FuelBasedTrack = 5

public enum
eVehicleStatus

InRepair=1,
Repaired=2,
PayedFor=3

public enum
eLicenseType

A1=1,
A2=2,
AA=3,
B1=4

public enum
eNumberOfDors

Two = 2,
Three = 3,
Four = 4,
Five = 5

public enum
eFuelType

Octan98 = 1,
Octan96 = 2,
Octan95 = 3,
Soler = 4

public enum
eCarColor

White=1,
Black=2,
Yellow=3,
Red=4