if you are stuck and need help you can call me for help!.Your IDE is your best helper to achieve this task.

ATKLASS : 5AX4 👍

Lab Test 1: Duration 1 Hour

Objectives

- **Implement the Car Class in C#** with the specified attributes, constructors, properties, and methods.
- **Write a Driver Program** (with a Main method in a Program class) that:
    - Creates multiple instances of the Car class.
    - Demonstrates the use of all three constructors.
    - Displays car details and computes the average fuel efficiency for each car.
    - Identifies the car with the highest and lowest average fuel efficiency OR Sorts the cars by price in ascending order and displays the sorted list.

Detailed Requirements

1. UML Diagram

Elements to Include:

- **Attributes (Fields):**
    - **carId: int** – A unique, automatically assigned identifier for each car.
    - **model: string** – The model name of the car.
    - **manufacturer: string** – The car's manufacturer.
    - **price: double** – The price of the car.
    - **fuelEfficiencies: double[]** – An array storing various fuel efficiency measurements (e.g., MPG).
- **Constructors:**
    - **Default Constructor:**
        - Sets `model` to `"Unknown"`, `manufacturer` to `"Unknown"`, `price` to `0.0`, and initializes `fuelEfficiencies` as an empty array.
    - **Parameterized Constructor #1:**
        - Accepts `model`, `manufacturer`, and `price`, and initializes `fuelEfficiencies` as an empty array.
    - **Parameterized Constructor #2:**

        - Accepts `model`, `manufacturer`, `price`, and a `fuelEfficiencies` array.
- **Properties:**
    - **CarId:** Read-only property for the car's unique ID.
    - **Model:** Read/write property for the car's model.
    - **Manufacturer:** Read/write property for the car's manufacturer.
    - **Price:** Read/write property for the car's price.
    - **FuelEfficiencies:** Read/write property for the array of fuel efficiency values.
- **Methods:**
    - **ToString(): string** – Returns a formatted string displaying the car's ID, model, manufacturer, price, and list of fuel efficiencies.
    - **GetAverageFuelEfficiency(): double** – Computes and returns the average fuel efficiency. If the array is empty, return `0.0`.

Sample UML Diagram Representation:

csharp
Copy

```
+-------------------------------------------------------------+
|                            Car                              |
+-------------------------------------------------------------+
| - carId: int                                                |
| - model: string                                             |
| - manufacturer: string                                        |
| - price: double                                             |
| - fuelEfficiencies: double[]                                |
+-------------------------------------------------------------+
| + Car()                                                     |
| + Car(model: string, manufacturer: string, price: double)       |
| + Car(model: string, manufacturer: string, price: double,       |
|       fuelEfficiencies: double[])                           |
| + CarId: int { get; }                                         |
| + Model: string { get; set; }                                 |
| + Manufacturer: string { get; set; }                          |
| + Price: double { get; set; }                                 |
| + FuelEfficiencies: double[] { get; set; }                      |
| + GetAverageFuelEfficiency(): double                        |
| + ToString(): string                                        |
+-------------------------------------------------------------+
```

---

2. C# Implementation

Car Class:

- **Attributes:**
    - Define the private fields for `carId`, `model`, `manufacturer`, `price`, and `fuelEfficiencies`.
- **Constructors:**
    - **No-Argument Constructor:**
      Sets `model` and `manufacturer` to `"Unknown"`, `price` to `0.0`, and initializes `fuelEfficiencies` as an empty array.
    - **Parameterized Constructor #1:**
      Accepts `model`, `manufacturer`, and `price`; initializes `fuelEfficiencies` as an empty array.
    - **Parameterized Constructor #2:**
      Accepts `model`, `manufacturer`, `price`, and a `fuelEfficiencies` array.
- **Properties:**
    - Provide a read-only property for `CarId` and read/write properties for the other attributes.
- **Methods:**
    - **ToString():**
      Format and return a string that shows all car details.
    - **GetAverageFuelEfficiency():**
      Calculate and return the average value from the `fuelEfficiencies` array. Return `0.0` if the array is empty.

Driver Program (Program Class with Main Method):

Create an array of Car objects:

```
Car[] garage = new Car[3];
```

- **Instantiate Each Element Using Different Constructors:**
    - One using the default constructor.
    - One using the constructor with `model`, `manufacturer`, and `price`.

    - One using the constructor with all parameters (including a `fuelEfficiencies` array).
- **For Each Car:**
    - Print the car's details using the `ToString()` method.
    - Calculate and display the average fuel efficiency using `GetAverageFuelEfficiency()`.
- **Extra Task – Identify the Best and Sort by Price:**
    - Determine which car has the highest and lowest average fuel efficiency and display its details. OR Sort the array of Car objects by `price` in ascending order and print the sorted list.

Submission Guidelines

Provide the following C# source files:

- **Car.cs** – Contains the definition of the Car class.
- **Program.cs** – Contains the driver program with the Main method that demonstrates all required functionalities.
- You may include both of these classes in one! Good luck1]