

Discussion-8: Classes and Object-Oriented Concepts in Java

Discussion Topic:

A car dealership has hired your software development firm to create a new car inventory program. You will first need to develop a class that represents the inventory of the dealership. What components and attributes will need to be included in your Automobile Class? Why? Illustrate a default and parameterized constructor for your class.

My Post:

Hello class,

To develop a class for the new car inventory program is essential to categorize the data that needs to be managed and to define the program functionalities required by the customers.

To categorize the data class attributes should be defined, for a car inventory program the attributes should cover the essential information about each car, such as its make, model, year, price, and other relevant details.

To define the program functionalities, class methods such as assessors and mutators should be implemented such as setting and getting the current value of the car, descriptions, and calculating the depreciation of the car (number of days on the lot). Below I made a list of attributes and methods for the class.

Attributes:

- make: The manufacturer of the car (e.g., Toyota, Honda).
- model: The specific model of the car (e.g., Corolla, Accord).
- year: The year the car was manufactured.
- price: The initial purchase price of the car.
- mileage: The mileage of the car.
- vin: The Vehicle Identification Number, a unique identifier for the car.
- color: The color of the car.
- initialValue: The car's initial value.
- currentValue: The current value of the car after depreciation.
- daysOnLot: The number of days on the lot

Methods:

- setMake(String make): Sets the make of the car.
- getMake(): Returns the make of the car.
- setModel(String model): Sets the model of the car.
- getModel(): Returns the model of the car.
- setYear(int year): Sets the year of the car.
- getYear(): Returns the year of the car.
- setPrice(double price): Sets the price of the car.
- getPrice(): Returns the price of the car.
- setMileage(int mileage): Sets the mileage of the car.
- getMileage(): Returns the mileage of the car.
- setVin(String vin): Sets the VIN of the car.

- `getVin()`: Returns the VIN of the car.
- `setColor(String color)`: Sets the color of the car.
- `getColor()`: Returns the color of the car.
- `setInitialValue()`: Sets the car's initial value.
- `getInitialValue()`: Returns the car's initial value.
- `setDaysOnLot()`: Sets the number on the lot.
- `getDaysOnLot()`: Returns the number on the lot.
- `calcCurrentValue()`: Calculates the current value of the car based on the initial value and the number of days on the lot.
- `printInfo()`: Outputs the car's information.

Example of the class with a default and parameterized constructor:

```
public class Automobile {
    // Attributes
    private String make;
    private String model;
    private int year;
    private double price;
    private int mileage;
    private String vin;
    private String color;
    private double initialValue;
    private double currentValue;
    private int daysOnLot;

    // Default constructor
    public Automobile() {
        this.make = "unknown";
        this.model = "unknown";
        this.year = 0;
        this.price = 0.0;
        this.mileage = 0;
        this.vin = "unknown";
        this.color = "unknown";
        this.initialValue = 0.0;
        this.daysOnLot = 0;
        this.currentValue = 0.0;
    }

    // Parametrized constructor
    public Automobile(String make, String model, int year, double price, int mileage, String vin,
        String color, double initialValue, int daysOnLot) {
        this.make = make;
        this.model = model;
        this.year = year;
        this.price = price;
        this.mileage = mileage;
        this.vin = vin;
        this.color = color;
        this.initialValue = initialValue;
        this.daysOnLot = daysOnLot;
        calcCurrentValue()
    }
}
```

```

// Accessor methods
public String getMake() {
    return make;
}
public String getModel() {
    return model;
}

~~~~~

public int getDaysOnLot() {
    return daysOnLot;
}

// Mutator methods
public void setMake(String make) {
    this.make = make;
}
public void setModel(String model) {
    this.model = model;
}

~~~~~

public void setDaysOnLot(int daysOnLot) {
    this.daysOnLot = daysOnLot;
}

public void calcCurrentValue() {
    double dailyDepreciationRate = 0.001; // Example daily depreciation rate of 0.1%
    this.currentValue = this.initialValue * Math.pow((1 - dailyDepreciationRate),
this.daysOnLot);
}

public void printInfo() {
    System.out.println("Make: " + make);
    System.out.println("Model: " + model);
    System.out.println("Year: " + year);
    System.out.println("Price: $" + price);
    System.out.println("Mileage: " + mileage + " miles");
    System.out.println("VIN: " + vin);
    System.out.println("Color: " + color);
    System.out.println("Initial Value: $" + initialValue);
    System.out.println("Current Value: $" + currentValue);
    System.out.println("Days on Lot: " + daysOnLot);
}
}

```

Explanation of why I chose to implement the above attributes, constructors, and methods:

The attributes represent the essential details about each car. The default constructor initializes the attributes with default values to ensure the object can be instantiated without parameters. The parameterized constructor allows initializing the attributes with specific values. The methods provide access to set and get the attributes and functionalities like calculating the current value and printing information.

-Alex