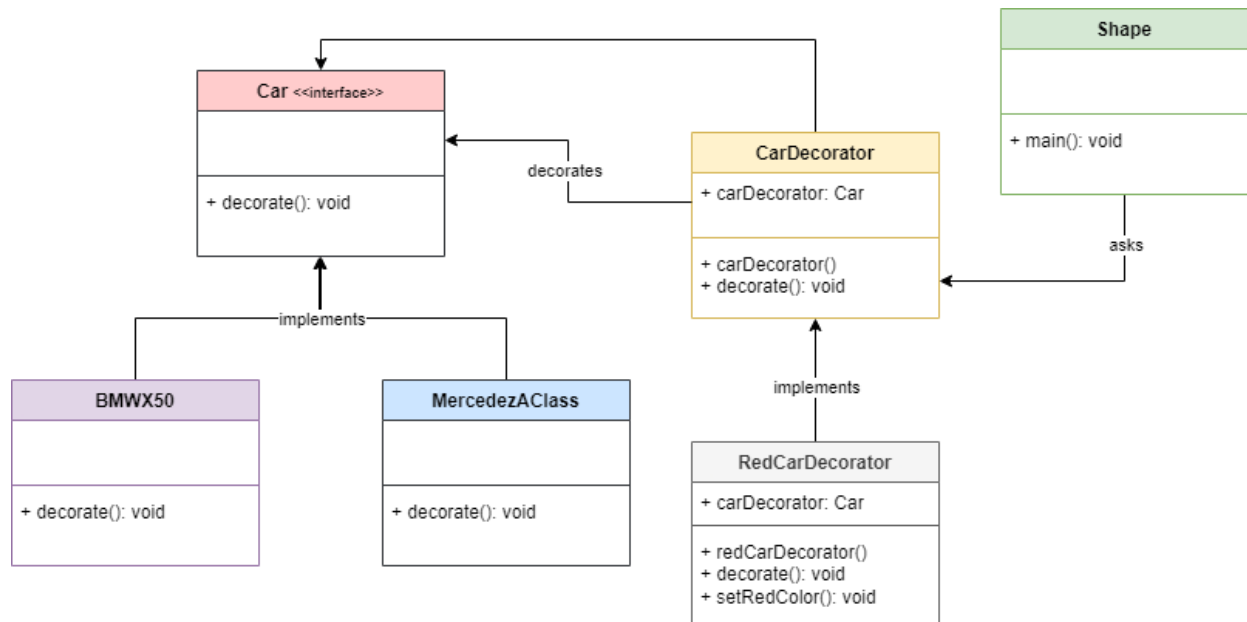


Assignment 10: Decorator Design Pattern

What is Decorator Design Pattern?

Decorator is a structural design pattern that lets you attach new behaviours to objects by placing these objects inside special wrapper objects that contain the behaviours.

Structure (Class Diagram)



Implementation (Code)

```

interface Car {
    void decorate();
}

class BMW50 implements Car {
    public void decorate() {
        System.out.println("Car: BMW50");
    }
}

class MercedesAClass implements Car {
    public void decorate() {
        System.out.println("Car: MercedesAClass");
    }
}

abstract class carDecorator implements Car {
    protected Car decoratedCar;
    public carDecorator(Car decoratedCar){
        this.decoratedCar = decoratedCar;
    }
    public void decorate(){
        decoratedCar.decorate();
    }
}

class RedCarDecorator extends carDecorator {
    public RedCarDecorator(Car decoratedCar) {

```

```
        super(decoratedCar);
    }
    public void decorate() {
        decoratedCar.decorate();
        setRedColor(decoratedCar);
    }
    private void setRedColor(Car decoratedCar){
        System.out.println("Car Color: Red");
    }
}

public class Main {
    public static void main(String[] args) {

        Car mercedesAClass = new MercedesAClass();
        Car bmwX50 = new BMWX50();

        Car redMercedesAClass = new RedCarDecorator(new MercedesAClass());
        Car redBMW50 = new RedCarDecorator(new BMWX50());

        System.out.println("\nBMW X-50 with white Color");
        bmwX50.decorate();

        System.out.println("\nMercedes A-Class with white Color");
        mercedesAClass.decorate();

        System.out.println("\nBMW X-50 of red color");
        redMercedesAClass.decorate();

        System.out.println("\nMercedes A-Class of red color");
        redBMW50.decorate();
    }
}
```

Output

```
BMW X-50 with white Color
Car: BMWX50

Mercedes A-Class with white Color
Car: MercedesAClass

BMW X-50 of red color
Car: MercedesAClass
Car Color: Red

Mercedes A-Class of red color
Car: BMWX50
Car Color: Red
```

Applicability

1. Use the Decorator pattern when you need to be able to assign extra behaviours to objects at runtime without breaking the code that uses these objects.
2. Use the pattern when it's awkward or not possible to extend an object's behaviour using inheritance.