public class Car {

    public void move(String direction) {

        switch (direction) {

            case "forward":

                moveForward();

                break;

            case "backward":

                moveBackward();

                break;

            case "left":

                turnLeft();

                break;

            case "right":

                turnRight();

                break;

            case "honk":

                honkHorn();

                break;

            default:

                System.out.println("Invalid direction: " + direction);

        }

    }

    public void moveForward() {

        System.out.println("Car moves forward.");

    }

    public void moveBackward() {

        System.out.println("Car moves backward.");

    }

    public void turnLeft() {

        System.out.println("Car turns left.");

    }

    public void turnRight() {

        System.out.println("Car turns right.");

    }

    public void honkHorn() {

        System.out.println("Honk! Honk!");

    }

}

public class Main {

    public static void main(String[] args) {

        Car myCar = new Car();

        myCar.move("forward");

        myCar.move("left");

        myCar.move("backward");

        myCar.move("right");

        myCar.move("honk");

        myCar.move("stop"); // This will output "Invalid direction: stop"

    }

}

Objective : Create different forms of coffee like PlainCoffe, ChocolateCoffee, MilkCoffee etc  
with methods - getCost and getDescription.  
  
Extend CoffeeDecorator into all coffee types

interface Coffee{

    double getcost();

    String getDescription();

}

class plainCoffee implements Coffee {

    double getcost(){

        return 2.0;

    }

    String getDescription(){

        return "Pain Coffee";

    }

}

abstract class CoffeeDecorator implements Coffee{

    private final Coffee coffee;

    public CoffeeDecorator(Coffee coffee){

        this.coffee = coffee;

    }

    double getcost(){

        return coffee.getcost();

    }

    String getDescription(){

        return coffee.getDescription();

    }

}

class ChocolateCoffee extends CoffeeDecorator {

    public ChocolateCoffee(Coffee coffee){

        super(coffee);

    }

    double getcost();{

        return super.getcost() + 1.5;

    }

    public String getDescription() {

        return super.getDescription() + ", Chocolate";

    }

}

class MilkCoffee extends CoffeeDecorator {

    public MilkCoffee(Coffee coffee) {

        super(coffee);

    }

    public double getCost() {

        return super.getCost() + 1.0;

    }

    public String getDescription() {

        return super.getDescription() + ", Milk";

    }

}

public class Main{

    public static void main(String args[]){

    }

}