package factorymethod;

import java.util.Scanner;

public interface Car {

String getModel();

void setWheel(String wheel);

String getWheel();

void setEngine(String engine);

String getEngine();

void setColor(String color); // New method for color

String getColor(); // New method for color

}

package factorymethod;

public class HatchbackCar implements Car {

String model;

String wheel;

String engine;

String color; // New property for color

HatchbackCar(String model, String wheel, String engine, String color) {

this.model = model;

this.wheel = wheel;

this.engine = engine;

this.color = color;

}

public String getModel() {

return model;

}

public void setWheel(String wheel) {

this.wheel = wheel;

}

public String getWheel() {

return wheel;

}

public void setEngine(String engine) {

this.engine = engine;

}

public String getEngine() {

return engine;

}

public void setColor(String color) {

this.color = color;

}

public String getColor() {

return color;

}

}

package factorymethod;

public class SedanCar implements Car {

String model;

String wheel;

String engine;

String color; // New property for color

SedanCar(String model, String wheel, String engine, String color) {

this.model = model;

this.wheel = wheel;

this.engine = engine;

this.color = color;

}

public String getModel() {

return model;

}

public void setWheel(String wheel) {

this.wheel = wheel;

}

public String getWheel() {

return wheel;

}

public void setEngine(String engine) {

this.engine = engine;

}

public String getEngine() {

return engine;

}

public void setColor(String color) {

this.color = color;

}

public String getColor() {

return color;

}

}

package factorymethod;

public class SUVCar implements Car {

String model;

String wheel;

String engine;

String color; // New property for color

SUVCar(String model, String wheel, String engine, String color) {

this.model = model;

this.wheel = wheel;

this.engine = engine;

this.color = color;

}

public String getModel() {

return model;

}

public void setWheel(String wheel) {

this.wheel = wheel;

}

public String getWheel() {

return wheel;

}

public void setEngine(String engine) {

this.engine = engine;

}

public String getEngine() {

return engine;

}

public void setColor(String color) {

this.color = color;

}

public String getColor() {

return color;

}

}

package factorymethod;

public interface CarFactory {

Car buildCar(String model, String wheel, String engine, String color); // Updated method

}

package factorymethod;

public class HatchbackCarFactory implements CarFactory {

@Override

public Car buildCar(String model, String wheel, String engine, String color) {

return new HatchbackCar(model, wheel, engine, color);

}

}

package factorymethod;

public class SedanCarFactory implements CarFactory {

@Override

public Car buildCar(String model, String wheel, String engine, String color) {

return new SedanCar(model, wheel, engine, color);

}

}

package factorymethod;

public class SUVCarFactory implements CarFactory {

@Override

public Car buildCar(String model, String wheel, String engine, String color) {

return new SUVCar(model, wheel, engine, color);

}

}

package factorymethod;

public class TestFactoryPattern {

CarFactory carBuilder;

Car car;

public static void main(String[] args) {

TestFactoryPattern client = new TestFactoryPattern();

client.buildCarMethod();

}

public void buildCarMethod() {

Scanner scanner = new Scanner(System.in);

System.out.println(&quot;Choose the type of car to build (Hatchback, Sedan, SUV): &quot;);

String carType = scanner.nextLine();

System.out.println(&quot;Enter model: &quot;);

String model = scanner.nextLine();

System.out.println(&quot;Enter wheel brand: &quot;);

String wheel = scanner.nextLine();

System.out.println(&quot;Enter engine brand: &quot;);

String engine = scanner.nextLine();

System.out.println(&quot;Enter color: &quot;);

String color = scanner.nextLine();

switch (carType.toLowerCase()) {

case &quot;hatchback&quot;:

carBuilder = new HatchbackCarFactory();

break;

case &quot;sedan&quot;:

carBuilder = new SedanCarFactory();

break;

case &quot;suv&quot;:

carBuilder = new SUVCarFactory();

break;

default:

System.out.println(&quot;Invalid car type!&quot;);

return;

}

car = carBuilder.buildCar(model, wheel, engine, color);

System.out.println(&quot;Car is assembled and painted now. Model of car is: &quot; + car.getModel() + &quot;,

Color: &quot; + car.getColor());

}

}