Title

Car showroom.

Background

N/A

Program Specifications

There are three types of car in the showroom:

AUDI:
- Available colors: *WHITE, YELLOW, ORANGE*
- Price: 5500, 3000, 4500 $
- Sold on: *FRIDAY SUNDAY MONDAY*

MERCEDES:
- Available colors: *GREEN, BLUE, PURPLE*
- Price: 5000, 6000, 8500 $
- Sold on: *TUESDAY, SATURDAY, WEDNESDAY*

BMW:
- Available colors: *PINK, RED, BROWN*
- Price: 2500, 3000, 3500 $
- Sold on: *MONDAY,SUNDAY, THURSDAY*

Write a program to help salesmen to check that the customer's needs match the current cars showroom is selling.
- Given that customers can choose unpainted car to get $100 discount.
- Customers can choose other options which will be added to the final price of the car.
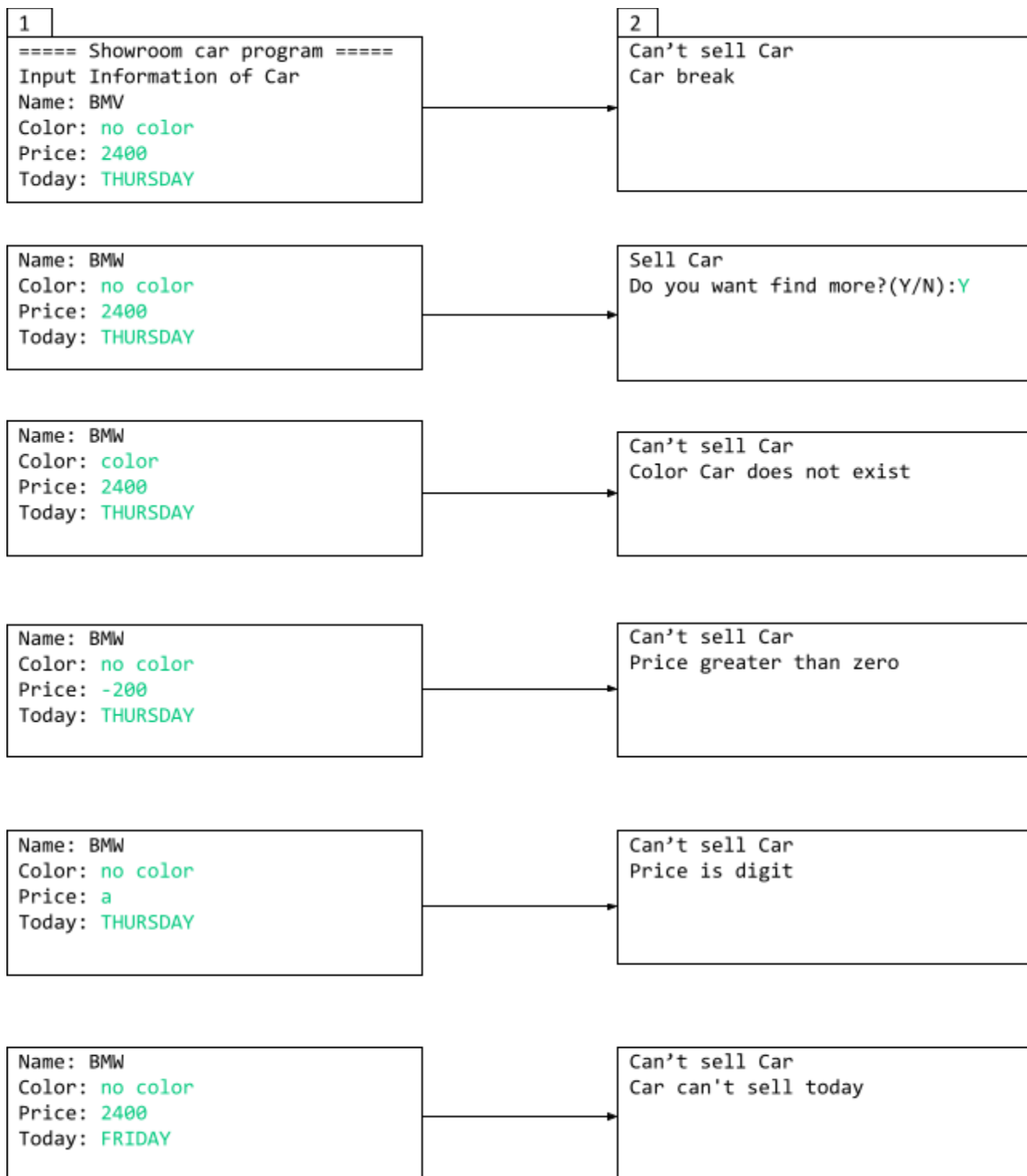
Function details:

Function 1: Display Gui And Input Data.
- User runs the program. The program prompts user to input data.

- Auto next **Function 2**.

Function 2: Perform function
- Program displays information of customer's request, and check if they match the cars in showroom?

- Display message and exit program.

- Sort by price

Expectation of User interface:

```
1
===== Showroom car program =====
Input Information of Car
Name: BMV
Color: no color
Price: 2400
Today: THURSDAY
```

```
2
Can't sell Car
Car break
```

```
Name: BMW
Color: no color
Price: 2400
Today: THURSDAY
```

```
Sell Car
Do you want find more?(Y/N):Y
```

```
Name: BMW
Color: color
Price: 2400
Today: THURSDAY
```

```
Can't sell Car
Color Car does not exist
```

```
Name: BMW
Color: no color
Price: -200
Today: THURSDAY
```

```
Can't sell Car
Price greater than zero
```

```
Name: BMW
Color: no color
Price: a
Today: THURSDAY
```

```
Can't sell Car
Price is digit
```

```
Name: BMW
Color: no color
Price: 2400
Today: FRIDAY
```

```
Can't sell Car
Car can't sell today
```

Guidelines

Student must implement methods

- checkCar

- getPrices

- getColors

- getDaySells

in startup code.

Hint

- Create ExceptionCar class which inherits Exception class; pass the message content to the constructor.

- Create Car Enum:

    o Create methods getPrices, getColors, getDaySells which return the List of sale information.

    o Create method Car getCar(String car)

- Create Enum Day:

    o List all days in a week

    o Create method Day getDay(String day), if day is not a Day Enum then return null.

- Create Enum Color:

    o List all colors of car in showroom, and "no color" option.

    o Create method Color getColor(String color), if color is not a Color Enum then return null.

- Create method checkCar whose input is the information of customer's request. It validates the information and return the Car Enum if matched, if not it throws the ExceptionCar.

- In Main, use scanner to enter the car information

    o If name of car is not in Car Enum, enter null value using getCar method.

    o If color of car is not in Color Enum, enter null value using getColor method.

    o If the day is not in Day Enum, enter null value using getDay method.

Function 1: Validate car sale information

    o Method name: public Car checkCar(Car car, String color, Day day, String price) thows ExceptionCar

        ● Input:

            car: Enum Car

            color: Enum Color

            day: Enum Day

            price: Prices of car

        ● Return:

            Enum Car

            ExceptionCar