A third-party company runs a business that helps people to sell and rent their cars. Suppose you are developing a program for the company to manage its inventory of cars. Suppose the design of classes has been done by other parties and given to you, as shown in Figure 1. **`SaleCar`** and **`RentalCar`** are the classes for cars for sale and rental, respectively. As for the cars for sale, they are provided with the contact of the owner, so that potential buyers can directly refer to the owner for further inquiries. Besides that, the company arranges the cars into three separated lists for easy managment. First is the list of cars for sale, second is the list of cars for rental consisting of MPV (multi-purpose vechicle) cars, and the third list is also cars for rental but consisting of regular cars.

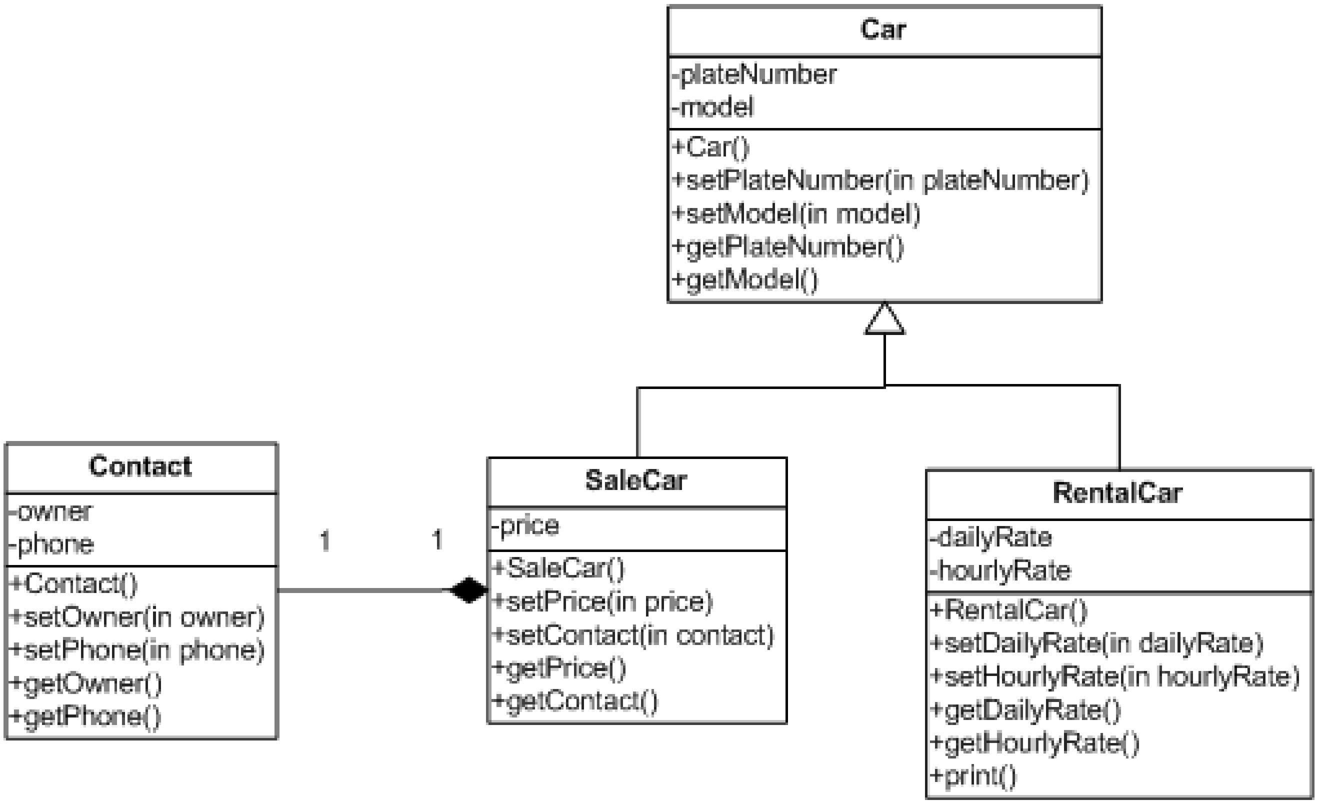

Figure 1: The class diagram

The implementation of all the classes have been provided in the source code file, **`sbt3.cpp`**. Along with the classes, two functions have been given in the source code file, named **`searchSaleCar`** and **`searchRentalCar`**. These functions are used for searching for a car in a list of cars for sale and rental, respectively. If the car is found, the functions return the index of the car in the list, otherwise they return -1.

Without making any changes to the code regarding the declaration and definition of the classes and the given functions, edit the source code filem, `sbt3.cpp`, to accomplish the following task requirements:

1. Complete the definition of function `readSaleCars`. This function reads a list of cars for sale from a text file. **(7 marks)**

2. Complete the definition of function `readRentalCars`. This function reads a list of cars for rental from a text file. **(4 marks)**

3. Read the list of cars for sale from the file **`"sale_car.txt"`** and store them into the array `saleCars`. **(2 marks)**

4. Read the list of cars for rental from the file **`"rental_mpv.txt"`** and store them into the array `mpvRentalCars`. **(2 marks)**

5. Read the list of cars for rental from the file **`"rental_regular.txt"`** and store them into the array `regularRentalCars`. **(2 marks)**

6. Search for a car in the the three lists. If the searched car is found, print in which list the car belongs to (i.e., list of cars for sale, list of mpv cars for rental or list of regular cars for rental). Also, if the car is for sale, print the model and the owner of the car. If the car is for rental (i.e., either an mpv or a regular car for rental), print all the information of the car. However, if there is no car found, prompt the user with an appropriate message. **(18 marks)**

Figure 2 shows the output of the program using different input. Note that the bold texts indicate input from the user.

```
Enter the plate number of the car to search for => JKW9067
```

```
The car is in the list of cars for sale.
Model : PROTON WIRA
Owner : YAHYA BIN ABDUL HANIZ
```

(a) Example run 1

```
Enter the plate number of the car to search for => WMV3392

The car is in the list of mpv cars for rental.
Plate Number: WMV3392
Model       : NISSAN GRAND LIVINA
Daily Rate  : RM 290
Hourly Rate : RM 25
```

(b) Example run 2

```
Enter the plate number of the car to search for => JJV7266

The car is in the list of regular cars for rental.
Plate Number: JJV7266
Model       : PERODUA KANCIL
Daily Rate  : RM 50
Hourly Rate : RM 5
```

(c) Example run 3

```
Enter the plate number of the car to search for => ABC1234

The car is not found.
```

(d) Example run 4

Figure 2: Expected output of the program