

~\OneDrive\Documents\Desktop\java programing\import java.util.java

```
1 import java.util.ArrayList;
2 import java.util.List;
3 import java.util.Scanner;
4
5 class Car {
6     private String carId;
7     private String brand;
8     private String model;
9     private double basePricePerDay;
10    private boolean isAvailable;
11
12    public Car(String carId, String brand, String model, double basePricePerDay) {
13        this.carId = carId;
14        this.brand = brand;
15        this.model = model;
16        this.basePricePerDay = basePricePerDay;
17        this.isAvailable = true;
18    }
19
20    public String getCarId() {
21        return carId;
22    }
23
24    public String getBrand() {
25        return brand;
26    }
27
28    public String getModel() {
29        return model;
30    }
31
32    public double calculatePrice(int rentalDay) {
33        return basePricePerDay * rentalDay;
34    }
35
36    public boolean isAvailable() {
37        return isAvailable;
38    }
39
40    public void rent() {
41        isAvailable = false;
42    }
43
44    public void returnCar() {
45        isAvailable = true;
46    }
47 }
48
49 class Customer {
50     private String customerId;
51     private String name;
```

```
52
53     public Customer(String customerId, String name) {
54         this.customerId = customerId;
55         this.name = name;
56     }
57
58     public String getCustomerId() {
59         return customerId;
60     }
61
62     public String getName() {
63         return name;
64     }
65 }
66
67 class Rental {
68     private Car car;
69     private Customer customer;
70     private int days;
71
72     public Rental(Car car, Customer customer, int days) {
73         this.car = car;
74         this.customer = customer;
75         this.days = days;
76     }
77
78     public Car getCar() {
79         return car;
80     }
81
82     public Customer getCustomer() {
83         return customer;
84     }
85
86     public int getDays() {
87         return days;
88     }
89 }
90
91 class CarRentalSystem {
92     private List<Car> cars;
93     private List<Customer> customers;
94     private List<Rental> rentals;
95
96     public CarRentalSystem() {
97         cars = new ArrayList<>();
98         customers = new ArrayList<>();
99         rentals = new ArrayList<>();
100     }
101
102     public void addCar(Car car) {
103         cars.add(car);
104     }
105 }
```

```
106     public void addCustomer(Customer customer) {
107         customers.add(customer);
108     }
109
110     public void rentCar(Car car, Customer customer, int days) {
111         if (car.isAvailable()) {
112             car.rent();
113             rentals.add(new Rental(car, customer, days));
114         } else {
115             System.out.println("Car is not available for rent.");
116         }
117     }
118
119     public void returnCar(Car car) {
120         car.returnCar();
121         Rental rentalToRemove = null;
122         for (Rental rental : rentals) {
123             if (rental.getCar() == car) {
124                 rentalToRemove = rental;
125                 break;
126             }
127         }
128         if (rentalToRemove != null) {
129             rentals.remove(rentalToRemove);
130             System.out.println("Car returned successfully.");
131         } else {
132             System.out.println("Car was not rented.");
133         }
134     }
135
136     public void menu() {
137         Scanner scanner = new Scanner(System.in);
138         while (true) {
139             System.out.println("==== Car Rental System =====");
140             System.out.println("1. Rent a Car");
141             System.out.println("2. Return a Car");
142             System.out.println("3. Exit");
143             System.out.print("Enter your choice: ");
144
145             int choice = scanner.nextInt();
146             scanner.nextLine(); // Consume the newline character
147
148             if (choice == 1) {
149                 System.out.println("\n== Rent a Car ==\n");
150                 System.out.println("Enter your name: ");
151                 String customerName = scanner.nextLine();
152
153                 System.out.println("\nAvailable Cars:");
154                 for (Car car : cars) {
155                     if (car.isAvailable()) {
156                         System.out.println(car.getCarId() + " - " + car.getBrand() + " " +
car.getModel());
157                     }
158                 }
```

```

159
160     System.out.print("\nEnter the car ID you want to rent: ");
161     String carId = scanner.nextLine();
162
163     System.out.print("Enter the number of days for rental: ");
164     int rentalDays = scanner.nextInt();
165     scanner.nextLine(); // Consume the newline character
166
167     // Create or retrieve the customer
168     Customer newCustomer = null;
169     for (Customer c : customers) {
170         if (c.getName().equalsIgnoreCase(customerName)) {
171             newCustomer = c;
172             break;
173         }
174     }
175     if (newCustomer == null) {
176         newCustomer = new Customer("CUS" + (customers.size() + 1),
customerName);
177         addCustomer(newCustomer);
178     }
179
180     // Find the car by ID
181     Car selectedCar = null;
182     for (Car car : cars) {
183         if (car.getCarId().equals(carId) && car.isAvailable()) {
184             selectedCar = car;
185             break;
186         }
187     }
188
189     if (selectedCar != null) {
190         double totalPrice = selectedCar.calculatePrice(rentalDays);
191         System.out.println("\n== Rental Information ==\n");
192         System.out.println("Customer ID: " + newCustomer.getCustomerId());
193         System.out.println("Customer Name: " + newCustomer.getName());
194         System.out.println("Car: " + selectedCar.getBrand() + " " +
selectedCar.getModel());
195         System.out.println("Rental Days: " + rentalDays);
196         System.out.printf("Total Price: $%.2f\n", totalPrice);
197
198         System.out.print("\nConfirm rental (Y/N): ");
199         String confirm = scanner.nextLine();
200
201         if (confirm.equalsIgnoreCase("Y")) {
202             rentCar(selectedCar, newCustomer, rentalDays);
203             System.out.println("\nCar rented successfully.");
204         } else {
205             System.out.println("\nRental canceled.");
206         }
207     } else {
208         System.out.println("\nInvalid car selection or car not available for
rent.");
209     }

```

```
210         } else if (choice == 2) {
211             System.out.println("\n== Return a Car ==\n");
212             System.out.print("Enter the car ID you want to return: ");
213             String carId = scanner.nextLine();
214
215             Car carToReturn = null;
216             for (Car car : cars) {
217                 if (car.getCarId().equals(carId) && !car.isAvailable()) {
218                     carToReturn = car;
219                     break;
220                 }
221             }
222
223             if (carToReturn != null) {
224                 Customer customer = null;
225                 for (Rental rental : rentals) {
226                     if (rental.getCar() == carToReturn) {
227                         customer = rental.getCustomer();
228                         break;
229                     }
230                 }
231
232                 if (customer != null) {
233                     returnCar(carToReturn);
234                     System.out.println("Car returned successfully by " +
customer.getName());
235                 } else {
236                     System.out.println("Car was not rented or rental information is
missing.");
237                 }
238             } else {
239                 System.out.println("Invalid car ID or car is not rented.");
240             }
241         } else if (choice == 3) {
242             break;
243         } else {
244             System.out.println("Invalid choice. Please enter a valid option.");
245         }
246     }
247     System.out.println("\nThank you for using the Car Rental System!");
248 }
249 }
250
251 public class Main {
252     public static void main(String[] args) {
253         CarRentalSystem rentalSystem = new CarRentalSystem();
254
255         Car car1 = new Car("C001", "Toyota", "Camry", 70.0);
256         Car car2 = new Car("C002", "Honda", "Accord", 70.0);
257         Car car3 = new Car("C003", "Mahindra", "Thar", 150.0);
258         rentalSystem.addCar(car1);
259         rentalSystem.addCar(car2);
260         rentalSystem.addCar(car3);
261     }
```

```
262         rentalSystem.menu();  
263     }  
264 }  
265
```

==== Car Rental System ====

1. Rent a Car
2. Return a Car
3. Exit

Enter your choice: 1

== Rent a Car ==

Enter your name:

abhishek kumar yadav

Available Cars:

C001 - Toyota Camry

C002 - Honda Accord

C003 - Mahindra Thar

Enter the car ID you want to rent: C003

Enter the number of days for rental: 15

== Rental Information ==

Customer ID: CUS1

Customer Name: abhishek kumar yadav

Car: Mahindra Thar

Rental Days: 15

Total Price: \$2250.00

Confirm rental (Y/N): Y

Car rented successfully

.

==== Car Rental System ====

1. Rent a Car
2. Return a Car
3. Exit

Enter your choice: 2

== Return a Car ==

Enter the car ID you want to return: C003

Car returned successfully.

Car returned successfully by abhishek kumar yadav