

2023

# Car Agency System



KemoMoh11

OOP Project

12/24/2023

# Car Agency System:

## OOP Masters Team:



Name: Akram Mohamed (Leader)	ID: 225186
Name: Mahmoud Maher	ID: 225200
Name: Arwa Hassan	ID: 225176
Name: Youmna Alaa	ID: 225023
Name: AbdElRahman Omar	ID: 225189

# Person Class:

```

Main.java  Person.java x Employee.java  Customer.java  Vehicle.java  Car.java  OrderDetails.java  FileRead.java  v  :
1  package Person;
   4 usages  2 inheritors
2  @ public class Person {
   2 usages
3      private String name;
   2 usages
4      private String phoneNumber;
   2 usages
5      private String address;
   2 usages
6      private String socialSecurityNumber;
7
8      2 usages
9      public Person(String name, String phoneNumber, String address , String socialSecurityNumber) {
10         this.name = name;
11         this.phoneNumber = phoneNumber;
12         this.address = address;
13         this.socialSecurityNumber = socialSecurityNumber;
14     }
15
16     10 usages
17     public String getName() { return name; }
18
19     6 usages
20     public String getPhoneNumber() { return phoneNumber; }
21
22     6 usages
23     public String getAddress() { return address; }
24
25     6 usages
26     public String getSocialSecurityNumber() { return socialSecurityNumber; }
27
28 }
29

```

# Employee Class:

```

Main.java  Person.java  Employee.java x Customer.java  Vehicle.java  Car.java  OrderDetails.java  FileRead.java  v  :
1  package Person.Employee;
2
3  import Person.Person;
4
5  11 usages
6  public class Employee extends Person {
   2 usages
7      private String employeeID;
8
9      2 usages
10     public Employee(String name, String phoneNumber, String address, String ssn, String employeeID) {
11         super(name, phoneNumber , address ,ssn);
12         this.employeeID = employeeID;
13     }
14
15     3 usages
16     public String getEmployeeID() { return employeeID; }
17

```

## Customer Class:

```
© Main.java © Person.java © Employee.java © Customer.java × © Vehicle.java © Car.java © OrderDetails.java © FileRead.java
1 package Person.Customer;
2
3 import Person.Person;
4
5 12 usages
6 public class Customer extends Person {
7     2 usages
8     private String customerID;
9
10    3 usages
11    public Customer(String name, String phoneNumber, String address, String ssn,String customerID) {
12        super(name, phoneNumber , address ,ssn);
13        this.customerID = customerID;
14    }
15
16    2 usages
17    public String getCustomerID() { return customerID; }
```

## Vehicle Class:

```
© Main.java © Person.java © Employee.java © Customer.java © Vehicle.java × © Car.java © OrderDetails.java © FileRead.java
1 package Vehicle;
2 2 usages 1 inheritor
3 public class Vehicle {
4     2 usages
5     private String brand;
6     2 usages
7     private String model;
8     2 usages
9     private int year;
10    2 usages
11    private String color;
12
13    1 usage
14    public Vehicle(String brand, String model, int year, String color) {
15        this.brand = brand;
16        this.model = model;
17        this.year = year;
18        this.color = color;
19    }
20
21    5 usages
22    public String getbrand() { return brand; }
23
24    5 usages
25    public String getModel() { return model; }
26
27    5 usages
28    public int getYear() { return year; }
29
30    5 usages
31    public String getColor() { return color; }
32 }
```

## Car Class:

```
© Main.java © Person.java © Employee.java © Customer.java © Vehicle.java © Car.java × © OrderDetails.java © FileRead.java
1 package Vehicle.Car;
2
3 import Vehicle.Vehicle;
4
5 public class Car extends Vehicle {
6     private double carPrice;
7
8     public Car(String brand, String model, int year, String color, double carPrice) {
9         super(brand, model, year, color);
10        this.carPrice = carPrice;
11    }
12
13    public double getCarPrice() { return carPrice; }
14 }
15
16
17
```

## OrderDetails Class:

```
© Main.java © Person.java © Employee.java © Customer.java © Vehicle.java © Car.java © OrderDetails.java × © FileRead.java
1 package Order;
2
3 import Person.Employee.*;
4 import Person.Customer.*;
5 import Vehicle.Car.*;
6 public class OrderDetails {
7     private Customer customer;
8     private Employee employee;
9     private Car car;
10    private double agreedOnPrice;
11    private boolean rented;
12
13    public OrderDetails(Customer customer, Employee employee, Car car, double agreedOnPrice, boolean rented) {
14        this.customer = customer;
15        this.employee = employee;
16        this.car = car;
17        this.agreedOnPrice = agreedOnPrice;
18        this.rented = rented;
19    }
20
21    public Customer getCustomer() { return customer; }
22
23
24
25    public Employee getEmployee() { return employee; }
```

```

29 > 9 usages
    public Car getCar() { return car; }
32
    2 usages
33 > public boolean isRented() { return rented; }
36
    2 usages
37 > public double getAgreedOnPrice() { return agreedOnPrice; }
40 }

```

## FileRead Class:

```

Main.java  Person.java  Employee.java  Customer.java  Vehicle.java  Car.java  OrderDetails.java  FileRead.java x v ⋮
1 package FileHandler;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5
6 import Order.OrderDetails;
7 import Person.Employee.*;
8 import Person.Customer.*;
9 import Vehicle.Car.*;
10
11 5 usages
12 public class FileRead {
13     1 usage
14     @ public static Car[] readCars() {
15         final int size = 50;
16
17         int i = 0;
18         Car[] cars = new Car[size];
19         try (BufferedReader buff = new BufferedReader(new FileReader(fileName: "cars.txt"))) {
20             String data;
21
22             while ((data = buff.readLine()) != null && i < size) {
23                 String[] extractedData = data.split(regex: ",");
24                 if (extractedData.length == 5) {
25                     cars[i] = new Car(extractedData[0], extractedData[1], Integer.parseInt(extractedData[2]), extractedData[3], Double.
26                     i++;
27                 } else {
28                     throw new Exception();
29                 }
30             }
31         }
32     }
33 }

```

```

33     catch (Exception e) {
34         System.out.println("An error has occurred while reading cars: " + e.getMessage());
35     }
36
37     return cars;
38 }
39
40 1 usage
41 @ public static Employee[] readEmployees() {
42     final int size = 50;
43
44     int i = 0;
45     Employee[] employees = new Employee[size];
46     try (BufferedReader buff = new BufferedReader(new FileReader( fileName: "employees.txt"))) {
47         String data;
48
49         while ((data = buff.readLine()) != null && i < size) {
50             String[] extractedData = data.split( regex: " ");
51             if (extractedData.length == 5) {
52                 employees[i] = new Employee(extractedData[0], extractedData[1], extractedData[2], extractedData[3], extractedData[4]);
53                 i++;
54             }
55         } catch (Exception e) {
56             System.out.println("An error has occurred while reading employees: " + e.getMessage());
57         }
58
59         return employees;
60     }
61
62 1 usage
63 @ public static Customer[] readCustomers() {
64     final int size = 50;
65
66     int i = 0;
67     Customer[] customers = new Customer[size];
68     try (BufferedReader buff = new BufferedReader(new FileReader( fileName: "customers.txt"))) {
69         String data;
70
71         while ((data = buff.readLine()) != null && i < size) {
72             String[] extractedData = data.split( regex: " ");
73
74             if (extractedData.length == 5) {
75                 customers[i] = new Customer(extractedData[0], extractedData[1], extractedData[2], extractedData[3], extractedData[4]);
76                 i++;
77             }
78         }
79     } catch (Exception e) {
80         System.out.println("An error has occurred while reading customers: " + e.getMessage());
81     }
82
83     return customers;
84 }
85
86
87

```

```

92
1 usage
93 @ public static OrderDetails[] readOrderDetails() {
94     final int size = 50;
95
96     int i = 0;
97     OrderDetails[] orderDetails = new OrderDetails[size];
98     try (BufferedReader buff = new BufferedReader(new FileReader( fileName: "orderDetails.txt"))) {
99         String data;
100
101         while ((data = buff.readLine()) != null && i < size) {
102             String[] extractedData = data.split( regex: ",");
103
104             if (extractedData.length == 17) {
105                 orderDetails[i] = new OrderDetails(new Customer(extractedData[7],extractedData[8],extractedData[9],extractedData[10],
106                     extractedData[11],extractedData[12],extractedData[13],extractedData[14],extractedData[15],extractedData[16]));
107                 i++;
108             }
109         } catch (Exception e) {
110             System.out.println("An error has occurred while reading customers: " + e.getMessage());
111         }
112
113         return orderDetails;
114     }
115 }
116

```

## FileWrite Class:

```

Person.java Employee.java Customer.java Vehicle.java Car.java OrderDetails.java FileRead.java FileWrite.java
1 package FileHandler;
2
3 import Order.OrderDetails;
4 import Person.Customer.*;
5 import Person.Employee.*;
6 import Vehicle.Car.*;
7
8
9 import java.io.BufferedWriter;
10 import java.io.IOException;
11 import java.io.FileWriter;
12
13 5 usages
14 public class FileWrite {
15     1 usage
16     @ public static void saveCustomer(Customer customer) {
17         try (BufferedWriter writer = new BufferedWriter(new FileWriter( fileName: "customers.txt", append: true))) {
18             writer.write( str: customer.getName() + "," + customer.getPhoneNumber() + "," + customer.getAddress()
19                 + "," + customer.getSocialSecurityNumber()
20                 + "," + customer.getCustomerID());
21             writer.newLine();
22         } catch (IOException e) {
23             System.out.print(e.getMessage());
24         }
25     }
26 }

```



```

no usages
25 @
26 public static void saveEmployee(Employee employee) {
27     try (BufferedWriter writer = new BufferedWriter(new java.io.FileWriter( fileName: "employees.txt", append: true))) {
28         writer.write( str: employee.getName() + "," + employee.getPhoneNumber() + "," + employee.getAddress()
29             + "," + employee.getSocialSecurityNumber()
30             + "," + employee.getEmployeeID());
31         writer.newLine();
32     } catch (IOException e) {
33         System.out.print(e.getMessage());
34     }
35 }

1 usage
36 @
37 public static void saveCar(Car car) {
38     try (BufferedWriter writer = new BufferedWriter(new java.io.FileWriter( fileName: "cars.txt", append: true))) {
39         writer.write( str: car.getbrand() + "," + car.getModel() + "," + car.getYear() + ","
40             + car.getColor() + "," + car.getCarPrice());
41         writer.newLine();
42     } catch (IOException e) {
43         System.out.print(e.getMessage());
44     }
45 }

2 usages
46 @
47 public static void saveOrder(OrderDetails order) {
48     try (BufferedWriter writer = new BufferedWriter(new java.io.FileWriter( fileName: "orderdetails.txt", append: true))) {
49         writer.write( str: order.getAgreedOnPrice() + "," + (order.isRented() ? "Yes" : "No") + ",");
50
51         writer.write( str: order.getEmployee().getName() + "," + order.getEmployee().getPhoneNumber() + "," + order.getEmployee().getA
52             + "," + order.getEmployee().getSocialSecurityNumber()
53             + "," + order.getEmployee().getEmployeeID() + ","");
54
55         writer.write( str: order.getCustomer().getName() + "," + order.getCustomer().getPhoneNumber() + "," + order.getCustomer().getA
56             + "," + order.getCustomer().getSocialSecurityNumber()
57             + "," + order.getCustomer().getCustomerID() + ","");
58
59         writer.write( str: order.getCar().getbrand() + "," + order.getCar().getModel() + "," + order.getCar().getYear() + ","
60             + order.getCar().getColor() + "," + order.getCar().getCarPrice());
61
62         writer.newLine();
63     } catch (IOException e) {
64         System.out.print(e.getMessage());
65     }
66 }

```

# Main:

```

Main.java x Person.java Employee.java Customer.java Vehicle.java Car.java OrderDetails.java FileRead.java
1  import FileHandler.FileRead;
2  import FileHandler.FileWrite;
3  import Order.OrderDetails;
4  import Person.Employee.*;
5  import Person.Customer.*;
6  import Vehicle.Car.*;
7
8  import java.util.Scanner;
9
10 public class Main {
11     20 usages
12     static Scanner scanner = new Scanner(System.in);
13     3 usages
14     static Employee[] employees = FileRead.readEmployees();
15     1 usage
16     static Customer[] customers = FileRead.readCustomers();
17     18 usages
18     static Car[] cars = FileRead.readCars();
19     16 usages
20     static OrderDetails[] orderDetails = FileRead.readOrderDetails();
21
22     5 usages
23     public static void login() {
24         int choice;
25         boolean found = false;
26
27         System.out.println("Are you an employee or a customer?\n1- Employee\n2- Customer");
28
29         try {
30             choice = scanner.nextInt();
31             clearScanner();
32
33             if (choice == 1) {
34                 System.out.print("Please state your name: ");
35                 String name;
36                 name = scanner.nextLine();
37
38                 for (var employee : employees) {
39                     if (employee != null && employee.getName() != null && employee.getName().equalsIgnoreCase(name)) {
40                         System.out.println("Hello, " + employee.getName() + "!\nLogging you in at the moment, please wait...");
41                         Thread.sleep( time: 3000);
42                         try {
43                             loginEmployee(employee);
44                         } catch (Exception e) {
45                             System.out.println("An error has occurred: " + e.getMessage() + "\nPlease try again in a moment...");
46                             Thread.sleep( time: 5000);
47                             login();
48                             return;
49                         }
50
51                         found = true;
52                         break;
53                     }
54                 }
55
56                 if (!found) {
57                     System.out.println("We could not find your name in our list, you may not be an employee?\nLet's try again..\n");
58                     login();
59                 }
60             }
61         }
62     }
63 }
```

```

54         } else if (choice == 2) {
55             System.out.print("Please enter your social security number: ");
56             String ssn = scanner.next();
57
58             for (var customer : customers) {
59                 if (ssn.equals(customer.getSocialSecurityNumber())) {
60
61                     System.out.println("Welcome, " + customer.getName() + "! We will log you in shortly...");
62
63                     Thread.sleep(time: 3000);
64
65                     try {
66                         loginCustomer(customer);
67                     } catch (Exception e) {
68                         System.out.println("An error has occurred: " + e.getMessage() + "\nPlease try again in a moment...");
69                         clearScanner();
70                         Thread.sleep(time: 5000);
71                         login();
72                         return;
73                     }
74
75                     found = true;
76                     break;
77                 }
78             }
79
80             if (!found) {
81                 System.out.println("We could not find your name in our list, please try again and check your input");

```

```

82             }
83         } else {
84             throw new Exception("You have entered the wrong choice...Please try again");
85         }
86     } catch (Exception e) {
87         System.out.println("Error while logging in: " + e.getMessage());
88         clearScanner();
89         login();
90     }
91 }
92
93 1 usage
94 public static void showAllCars() {
95     for (int i = 0; i < cars.length; i++) {
96         if (cars[i] != null) {
97             System.out.println((i + 1) + ". " + cars[i].getbrand() + " " + cars[i].getModel() +
98                 " " + cars[i].getYear() + ", Color: " + cars[i].getColor() + ", Price: $" + cars[i].getCarPrice());
99         }
100     }
101 }
102
103 1 usage
104 public static void showAllOrderDetails() {
105     final String line = "-----";
106     for (int i = 0; i < orderDetails.length; i++) {
107         if (orderDetails[i] != null) {
108             System.out.println(line + "\n" + line + "\nOrder #" + (i + 1) + "\n" + line + "\nCar\n" + line + "\nCar brand: " + order
109         }

```

```

111 public static void showCar(int i) {
112     if (cars[i] != null) {
113         System.out.println(cars[i].getbrand() + " " + cars[i].getModel() +
114             " " + cars[i].getYear() + ", Color: " + cars[i].getColor() + ", Price: $" + cars[i].getCarPrice());
115     }
116 }
117
118
119 1 usage
120 public static void loginEmployee(Employee employee) {
121     System.out.println("Now, what (else) do you want to do?\n1- Add a new car\n2- Add a new Customer\n3- Show all order details\n4-
122     try {
123         int choice = scanner.nextInt();
124         clearScanner();
125
126         if (choice == 1) {
127             String brand, model, color;
128             int year;
129             double price;
130             System.out.println("Please enter all the details of your new car in this order: brand, model, year, color, price");
131
132             brand = scanner.next();
133             model = scanner.next();
134             year = scanner.nextInt();
135             clearScanner();
136             color = scanner.next();
137             price = scanner.nextDouble();
138             clearScanner();
139
140             FileWrite.saveCar(new Car(brand, model, year, color, price));
141             System.out.println("Car added successfully!");
142         } else if (choice == 2) {
143             System.out.println("Please enter the customer information in this order: name, phone number, address, social security n
144             String name, phoneNumber, address, socialSecurityNumber, id;
145             name = scanner.next();
146             phoneNumber = scanner.next();
147             address = scanner.next();
148             socialSecurityNumber = scanner.next();
149             id = scanner.next();
150
151             FileWrite.saveCustomer(new Customer(name, phoneNumber, address, socialSecurityNumber, id));
152             System.out.println("Customer added successfully!");
153         } else if (choice == 3) {
154             System.out.println("Sure, here's your list of the latest 50 orders:");
155             showAllOrderDetails();
156         } else if (choice == 4) {
157             System.exit( code: 0);
158         } else {
159             throw new Exception("User made an invalid choice.Try again...");
160         }
161     } catch (Exception e) {
162         System.out.println("An error has occurred: " + e.getMessage());
163         clearScanner();
164     }
165     loginEmployee(employee);
166 }
167
168 4 usages
169 public static void loginCustomer(Customer customer) {
170     System.out.println("Please choose which car you'd like to buy/rent: ");
171     showAllCars();

```

```

170     int carChoice = scanner.nextInt();
171     clearScanner();
172
173     try {
174         showCar(carChoice - 1);
175         System.out.println("Do you want to purchase this car?\n1- Yes\n2- No");
176         int choice = scanner.nextInt();
177         clearScanner();
178
179         if (choice == 1) {
180             System.out.println("Do you want to buy or rent the car?\n1- Buy\n2- Rent");
181             choice = scanner.nextInt();
182             clearScanner();
183
184             if (choice == 1) {
185                 FileWrite.saveOrder(new OrderDetails(customer, employees[0], cars[carChoice - 1], cars[carChoice - 1].getCarPrice(),
186                 System.out.println("Your purchase has completed successfully. Thank you for using our service!");
187                 System.exit(code: 0);
188             } else if (choice == 2) {
189                 System.out.print("Please enter for how many months you're going to rent this car: ");
190                 do {
191                     try {
192                         int months = scanner.nextInt();
193                         if (months > 1 && months != cars[carChoice - 1].getCarPrice()) {
194                             double price = cars[carChoice - 1].getCarPrice() * 0.05 * months;
195                             System.out.println("Your car rent is going to cost $" + (int)(price/months) + " per month\nDo you want to
196                             choice = scanner.nextInt();
197                             clearScanner();
198                             if (choice == 1) {
199                                 FileWrite.saveOrder(new OrderDetails(customer, employees[0], cars[carChoice - 1], price, rented: true));
200                                 System.out.println("Your purchase has completed successfully. Thank you for using our service!");
201                             } else if (choice == 2) {
202                                 System.out.print("Enter your new month to calculate the rent with: ");
203                             } else if (choice == 3) {
204                                 System.out.println("Alright, going back to main menu...\n");
205                                 Thread.sleep(time: 3000);
206                                 loginCustomer(customer);
207                                 break;
208                             } else {
209                                 throw new Exception("User gave an invalid number of months");
210                             }
211
212                         } else {
213                             throw new Exception("User gave an invalid number of months");
214                         }
215                     } catch (Exception e) {
216                         System.out.print("An error has occurred please enter your months again: ");
217                         clearScanner();
218                     }
219                 } while (true);
220             } else {
221                 throw new Exception("User made an invalid choice. Try again...");
222             }
223
224         } else if (choice == 2) {
225             System.out.println("Alright, going back to main menu...\n");
226

```

```

225     } else if (choice == 2) {
226         System.out.println("Alright, going back to main menu...\n");
227         Thread.sleep(time: 3000);
228         loginCustomer(customer);
229     } else {
230         throw new Exception("User made an invalid choice.Try again...");
231     }
232 } catch (Exception e) {
233     System.out.println("Error occurred: " + e.getMessage() + "\nTrying again...\n");
234     clearScanner();
235     loginCustomer(customer);
236 }
237 }
238
13 usages
239 > public static void clearScanner() { scanner.nextLine(); }
242
243 > public static void main(String[] args) { login(); }
246

```

## Out Put:

- If I entered as Employee:

```
Are you an employee or a customer?
1- Employee
2- Customer
1
Please state your name: Mahmoud Maher
Hello, Mahmoud Maher!
Logging you in at the moment, please wait...
Now, what (else) do you want to do?
1- Add a new car
2- Add a new Customer
3- Show all order details
4- Exit
```

- If I Added a New Car:

```
Are you an employee or a customer?
1- Employee
2- Customer
1
Please state your name: Mahmoud Maher
Hello, Mahmoud Maher!
Logging you in at the moment, please wait...
Now, what (else) do you want to do?
1- Add a new car
2- Add a new Customer
3- Show all order details
4- Exit
1
Please enter all the details of your new car in this order: brand, model, year, color, price
Bmw
X6
2020
White
95000
Car added successfully!
Now, what (else) do you want to do?
1- Add a new car
2- Add a new Customer
3- Show all order details
4- Exit
```

- If I Add New Customer:

```
Hello, Mahmoud Maher!  
Logging you in at the moment, please wait...  
Now, what (else) do you want to do?  
1- Add a new car  
2- Add a new Customer  
3- Show all order details  
4- Exit  
2  
Please enter the customer information in this order: name, phone number, address, social security number, id  
Akram  
01067405635  
Tanta  
30412151602399  
A2
```

- If I Show All Order Details:

```
-----  
Order #9  
-----  
Car  
-----  
Car brand: Bmw  
Car model: X6  
Car year: 2020  
Car color: White  
Car is rented: true  
Agreed on price: 28500.0  
-----  
Employee  
-----  
Employee name: Mahmoud Maher  
Employee phone number: 012112362565  
Employee address: El-Nozha  
Employee ID: A456  
-----  
Customer  
-----  
Customer name: Mahmoud Maher  
Customer phone number: 1212121221  
Customer address: Salam  
Customer social security number: 30412170100655  
-----
```



- If I entered as a customer:

```
Are you an employee or a customer?  
1- Employee  
2- Customer  
2  
Please enter your social security number: 30412151602399  
Welcome, Akram! We will log you in shortly...  
Please choose which car you'd like to buy/rent:  
1. Toyota Camry 2022, Color: Blue, Price: $25000.0  
2. Bmw X6 2020, Color: White, Price: $95000.0
```

- If I Choose to buy a Car:

```
Are you an employee or a customer?  
1- Employee  
2- Customer  
2  
Please enter your social security number: 30412151602399  
Welcome, Akram! We will log you in shortly...  
Please choose which car you'd like to buy/rent:  
1. Toyota Camry 2022, Color: Blue, Price: $25000.0  
2. Bmw X6 2020, Color: White, Price: $95000.0  
2  
Bmw X6 2020, Color: White, Price: $95000.0  
Do you want to purchase this car?  
1- Yes  
2- No  
1  
Do you want to buy or rent the car?  
1- Buy  
2- Rent  
1  
Your purchase has completed successfully. Thank you for using our service!
```

- If I Choose to Rent a Car:

2

Please enter your social security number: 30412151602399

Welcome, Akram! We will log you in shortly...

Please choose which car you'd like to buy/rent:

1. Toyota Camry 2022, Color: Blue, Price: \$25000.0
2. Bmw X6 2020, Color: White, Price: \$95000.0

2

Bmw X6 2020, Color: White, Price: \$95000.0

Do you want to purchase this car?

- 1- Yes
- 2- No

1

Do you want to buy or rent the car?

- 1- Buy
- 2- Rent

2

Please enter for how many months you're going to rent this car: 12

Your car rent is going to cost \$3800 per month

Do you want to confirm rent?

- 1- Yes
- 2- No, Enter another month
- 3- Cancel

1

Your purchase has completed successfully. Thank you for using our service!