18COA256: Object-oriented Programming Coursework Assignment

Dr Hossein Nevisi Semester 2

Task

You are to develop an administration tool for a used car dealer in Java. The new system will allow the staff to perform many operations which are needed for their daily business transactions. Each car is of a certain type; it has a number plate, model, size (only available and applicable for vans), colour, mileage, accident history, transmission type (either manual or automatic transmission), price and it is either *available* or *sold*; the date of arrival at the car dealer and the selling date are recorded. The date format is yyyy-mm-dd, e.g. 2019-01-15 for the 15th January 2019.

The cars are categorised by their types consisting of *body type* (e.g. Hatchback), *number of doors* (e.g. 5), *number of seats* (e.g. 5). In the requested implementation, “Van” as a body type has the additional attribute *size* (small or large).

The system can be accessed by three types of users: admin, staff and customer. Admin users have the rights to create login credentials for staff and customers, and there is no restriction for them to access anything in the system. The staff are allowed to search, add and modify the car records as well as calculating the sales revenue whereas customer users are only allowed to search for cars, and they don’t have access to the accident history of the cars.

The program should include at least the car types listed in Appendix A. An example list of cars is shown in Appendix B.

Functionality Three types of roles need to be created: "Admin Role", "Staff Role" and "Customer Role". Once the users successfully log into the system, they should be able to access the below functionalities according their roles.

- **Admin Role**
  - Creates login credentials for staff and customers as below:
    - Creates username, password and sets their role.
    - Passwords should not be stored as they are. Any algorithm for encoding is accepted, and it does not matter how complex it is.
  - Has all functionality that the staff role has.
- **Staff Role**
  - Login ... if the entered credential is correct, allows user to use the available functionality.
  - Add cars ... allows to add a list of cars provided in a file; each line in the file describes one car by the following attributes, separated by commas: *number plate, model, car type, size (only available and applicable for vans), colour, mileage, accident history, transmission type, price, arrival date, selling date*. The selling date is optional; if it is left out, then the car is not yet sold. If the arrival date is also left out, then the current date is used as the arrival date. (An example file is given in Appendix B.)
  - Add car ... allows to add a single car with the parameters *number plate, model, car type, size (only available and applicable for vans), colour, mileage, accident history, transmission type, price, arrival date, selling date*. Again if the selling date is left out, then the car is not yet sold. If the arrival date is also left out, then the current date is used as the arrival date.

  - Sell car ... changes the status of a car to sold by saving the date the car was sold.
  - Print cars ... outputs a file of all cars grouped into sold and unsold and sorts the former by the selling date and the latter by the day of their arrival. The format is the same as the input format with the exception that every car must have an arrival date.
  - Search ... for each of the following options, outputs a list of all available cars (all car details must be included)
    * that have the given model and transmission type or have the given colour,
    * whose number of seats is between a user-specified minimum and maximum,
    * which are Vans with a user-specified size.
  - Calculate revenue ... calculates and outputs the revenue of the specified day or month, that is the sum of the selling prices of the cars sold on that day or in that month.

- **Customer Role**
  - can do the same search as staff users, but not having access to accident history information.

The program should be able to avoid and detect errors. Examples: (i) The user should be notified if they want to sell a car that is already sold. (ii) Cars with a same plate number must not be added more than once.

User Interface You should provide a graphical user interface (GUI) or command-line interface (CLI). A GUI scores more points than a CLI. A good interface is intuitive, quick and easy to use.

Design and Implementation

Before you start programming, design your software and structure it in an object-oriented way. Identify classes, abstract classes, enums and interfaces, their attributes and methods, and think about how they interact with each other before you produce any code.

Note that while this is a fairly small program and could be implemented in a single class, your task is to show that you can design and implement a program in an object-oriented way and identify ways to apply object-oriented concepts in a useful way.

Note that this is an individual exercise and that you must not share any code.

You can use an IDE such as Eclipse or BlueJ to develop the program. But the code must compile and run from the command line or in Eclipse. If you use another IDE / GUI builder, be careful to use one that allows you to compile the program outside of it.

Writing Understandable Code

Your code must be understandable for readers. For this purpose, you should use

- meaningful names for your classes, objects, methods and variables
- comments properly in your code
- indentation appropriately

Documentation and Submission

You are expected to submit

- a documentation,
- all source files (i.e. java files) as well as all other files needed to compile and run the program (e.g. libraries),

- a jar file `cardealer.jar` that is executed by the command `java -jar cardealer.jar`,
- a text file `cars-output.txt` to store the output of the "Print cars" function (one car per line, see the "Print cars" function),
- a text file `cars-import.txt` to import a list of cars (see the "Add cars" function), and
- a text file `cars-users.txt` to store the login credentials (username, password and role). The passwords must be encrypted (see the Functionality section). The following credentials(username, password, role) should be applied to your system:
  - admin1, apple, admin
  - staff1, orange, staff
  - customer1, lemon, customer

All files need to be zipped into a single zip file called

`coursework_cardealer_<your first name>_<your last name>.zip`

and submitted electronically via Learn.

Your coursework should be submitted with appropriate documentation which should include

- an explanation of how your code should be run, how to use the interface and how the functionality can be used (user instructions),
- a list of all packages that have been used to develop the program,
- a list of functionality not completed in your application,
- a class diagram showing all classes, their attributes, methods, relationships, etc.
- a description of the object-oriented design and the diagram
- test plans (how you have evaluated and tested the program).

Marking

Your software will be marked with respect to functionality as well as object-oriented design. The following is a very rough guide how the assignment will be marked.

- Object-oriented design, documentation and writing understandable code: 34%
  - Object-oriented design (must be reflected in your documentation): 20%
  - Documentation: 10%
  - Writing understandable code: 4%
- User interface: 10%
- Functionality: 56%
  - Create user credential and login: 10%
  - Implementation of roles and access levels: 8%
  - Add car (3%) and Sell car (3%) functions: 6%
  - Add cars (6%) and Print cars functions (8%): 14%
  - Search (12%) and Calculate revenue (6%) functions : 18%

The following requirements are absolutely essential. If any of them is not fulfilled, the coursework will not pass.

- The code must compile and run.
- The jar file containing the program must run calling `java -jar cardealer.jar`.
- The implemented functionality is usable via an interface.
- There is sufficient documentation on how to use the interface.

Please be aware:

- The source files (i.e. java files) and all other files necessary for compiling and running the program (libraries etc.) must be provided. The code must also compile.
- The code must compile on the command line or in Eclipse. If you use another IDE / GUI builder, be careful to use one that allows you to compile the program outside of it.
- If the code is not submitted or does not compile and if there is no working jar file, the program cannot be tested and will therefore not pass.

A Car Types

The program should include at least the car types in the following box where each type is given as a comma-separated list in the form: `body types, number of doors, number of seats`. If the car body type is "Van", there should be the extra attribute `size` which determines the van is small or large.

```
SUV, 5, 5
MPV, 5, 7
Van, 5, 3
Hatchback, 5, 4
Saloon, 5, 5
Coupe, 2, 2
```

B Example List

The following comma-separated list of cars is in the correct format. The *number plate* is followed by *model*, *car type*, *size* which must only be available for vans, *colour*, *mileage*, *accident history*, *transmission type*, *price*, *arrival date* and *selling date*. If there is no selling date, then the car has not been sold. If the arrival date is also left out, then the current date is used as the arrival date.
The first car with the number plate YB18NBA is a manual grey Honda Jazz. Its mileage is 20000, and its price is £8000. It has no accident history, arrived on 1st September 2018 at the car dealer, and has been sold on 4th October 2018.

```
YB18NBA, Honda Jazz, Hatchback, grey, 20000, none, manual, 8000, 2018-09-01,
2018-10-04
MX67TCP, Mercedes Citan, Van, small, white, 35000, front damage, manual, 9500,
2018-11-04
AA13QWE, Ford Transit, Van, small, black, 60000, none, automatic, 7300, 2018-11-01
BA66IHN, Lexus IS, Saloon, white, 23000, indent on passenger door, automatic,
18100, 2019-01-02
NB17GFG, Lexus IS, Saloon, dark blue, 6000, none, manual, 26100, 2018-10-04,
2018-10-14
MA67TTP, Mercedes Citan, Van, small, white, 35900, front windscreen damage,
automatic, 10000, 2018-08-04
TT66AAA, Lexus NX, SUV, dark blue, 12000, none, automatic, 31000
UU54IOI, Ford SMax, MPV, red, 70000, none, manual, 11000, 2017-01-13
VV10BBB, Vauxhall Movano, Van, large, white, 60000, rear windscreen damage,
manual, 5000, 2019-01-10, 2019-03-01
PO13FDF, Vauxhall Insignia, Saloon, green, 45000, none, automatic, 10800,
```

```
2018-08-10, 2019-03-05
P013FDF, Vauxhall Insignia, Saloon, red, 40000, none, manual, 10200, 2018-08-10
PD64HAH, Vauxhall Movano, Van, large, silver, 60000, none, automatic, 8800
HN11PHN, BMW 3, Saloon, black, 49000, door damage, manual, 9500, 2019-03-01,
2019-03-05
UU54IOI, Ford SMax, MPV, red, 70000, none, automatic, 3800
PK14EMD, Mercedes EClass, Coupe, yellow, 25000, none, automatic, 10400,
2018-03-08, 2019-03-02
HN19PDH, Jaguar FType, Coupe, silver, 3000, none, automatic, 52000, 2018-03-01
```