FACULTY OF COMPUTING & INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY

كلية الحاسبات وتقنية المعلومات
جامعة الملك عبدالعزيز

Vehicle Service System

Assignment 3

CPCS203 Programming-II - Term – Spring 2021

Assigned Date: 28/03/2021
Delivery Date: 15/04/2021 @ 11:00 PM

Instructions

- This program must ONLY be submitted on the Blackboard.
- This project worth 10% of the overall module marks (100%).
- NO assignment will be accepted after 11:59 pm for any reason.
- For discussion schedule, check the captain name, date and time on the BlackBoard.
- Further information is provided in the course syllabus.

Objectives

- Practice class inheritance and polymorphism.
- Learn how to use and implement abstract classes and interfaces.
- Learn to use File I/O (Reading/Writing from/to files).

How to submit your assignment?

- Submit your assignment on the Blackboard ONLY.
- Make sure to add your names / IDs / Section / Your name / Assignment number at the beginning of your program

Files provided with assignment

- Input file samples:

  input.txt: which contains all customers and vehicle details that needs to be registered into the system. It also contains all the commands to generate vehicle services (sale or rental). These commands are read from the file and processed by the system.

- Output files:

  output.txt: This output file displays all the registered record for the customers, rental and sale vehicles.

  Note: Please check the format of each of these files and make sure you follow this format in your assignment solution.

System Description

This project is related to vehicle service agents by developing a computerized mechanism for selling and renting different kind of vehicles. This will help in reducing manual and paperwork and therefore expected to enhance the overall service experience both for the customers and the agents. The vehicle service system provides different vehicle types for sale or rent.

The system you are required to develop is called **Vehicle Store System.** At the initial stage, the system will register all the available vehicle information either for renting or sale, and the customers' details from **input.txt.** Information read from **input.txt** are written with all the details into an output file, called **output.txt**.

After populating the data, the system will be ready for serving customers according to available vehicle. For a rent service, the customer id and vehicle license details are read from input file **input.txt**. After processing the service, fare will be calculated and the details is written to the output file **output.txt**.

For a more detailed description of the system and commands, please follow the next three steps which will explain how to develop the system.

Step 1: Add customer, rental and sale Records

All customer and vehicle details (either for rent or sale) will be added into the system before making any services. The first line read from ***input.txt*** contains two integers, which determine the number of registered vehicles and customers. For example, the input file has 10 vehicles and 20 customers. In the following, we describe the format of each command.

1.1 Command: Add_Customer_Record

This command is used to add all information of a customer. This includes customer id, name, nationality, gender (M or F) and phone number. Check the following example and table.

Command Example
Add_Customer_Record 20005 Meshal_ALi Saudi M 50123456

Field name	Type	Example
id	int	20005
name	String	Meshal_ALi
natioanlity	String	Saudi
Gender	Char	M
phone	int	50123456

1.2 Command: Add_Rental

This command will add details of a rental vehicle into the system.

Command Example
Add_Rental GH7000 Mercedez CClass 180000.00 300 5

Field name	Type	Example
License	String	GH7000
Make	String	Mercedez
Model	String	CClass
Price	double	180000.00
Rent_per_day	double	300
max number of renters	int	5

1.3 Command: Add_Sale

This command will add details of the vehicles for sale into the system.

Command Example
Add_Sale HI2000 Nissan Sunny 45000.00 5

Field name	Type	Example
License	String	HI2000
Make	String	Nissan
Model	String	Sunny
Price	double	45000.00
Discount rate	double	5

Step 2: Process Services

The services details are provided in ***input.txt*** followed by the commands used to fill in the vehicles and customers arrays. More detailed description of booking process has been explained in the following subsections:

2.1 Command: Ass_Customer_to_Rent

This command is used to assign customer to a rented vehicle. The command contains the necessary information such as the customer id, the vehicle license number and the number of days to rent the vehicle.

Command Example
Ass_Customer_to_Rent GH7000 20007 7

Field name	Type	Example
license	String	GH7000
Customer id	int	20007
Number od days	int	7

Consider the following notes when issuing a service:

Important Notes
• The system will read the ***license number*** as a string. You need to search for the vehicle object associated with the given license number.
• The system will read the ***customer id*** as integer. You need to search for the customer object associated with the given id.
• The system will check the vehicle maximum renters' capacity before assigning the vehicle to the customer.
• You must calculate the ***price*** for each renting process

2.2 Command: Quit

The command quit will exit the vehicle service system.

Step 3: Print all the information

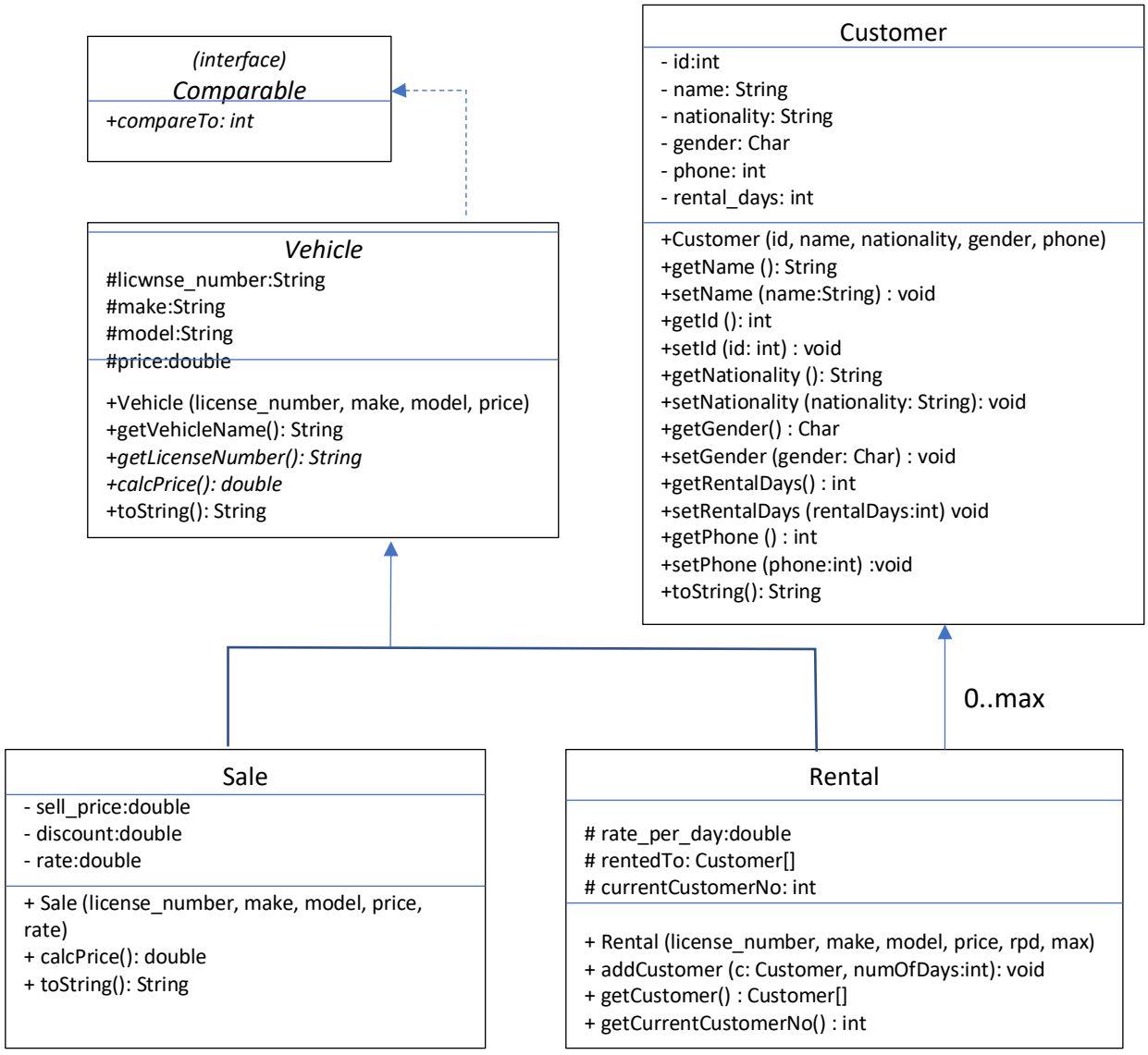
The system will print detailed information of the sale and rental vehicles.

Command Example
Print_Sale: use the overridden toString method to print all information about the vehicles for sale. You need to print the sale vehicle sorted according to the license number. **Hint:** You need to implement the ***compareTo*** method to make the sale vehicle comparable. You may want to use the ***Array.sort*** method.

Command Example
Print_Rental: use the overridden toString method to print all information about the vehicles for rent with records of all customers rented that vehicle if any.

1.2 UML Class Diagram

In addition to the main class, you should create all classes as shown in the following UML diagram. Note that you should write appropriate constructor, setter, and getter methods for all classes. (You don't need to follow the same given arguments). Be aware of the visibility (public-private) for each attribute/method and the use of abstract classes and interfaces.



Important Notes:

- Use of class & object, arrays of Object, and passing object to method
- Use of Files, Reading/Writing from/on files
- Your program output must be exactly same as given sample output files.
- Your display should be in a readable form.
- Organize your code in separated methods.
- Document your code with comments.
- Use meaningful variables.
- Use dash lines between each method.
- **Delayed submission will not be accepted and there will not be any extension of the project.**

Deliverables:

- You should submit one zip file containing all java codes: BA1887415P2_EasyRent.java where BA is your section, 1887415 your ID and P2 is program 2.
- **NOTE: your name, ID, and section number should be included as comments in all files!**

Input and Output Format

Your program must generate output in a similar format to the sample run provided.
Sample input: See sample input file.
Sample output: See sample output files.

Good Luck!