Programming and Data Structures
FInal Exam

You are provided with the implementation of the data structures shown in the UML diagram in figure 1 and the class hierarchy shown in the UML diagram in figure 2.

The given program manages a hotel using a list of guests stored in a TreeMap, a list of rooms stored in a LinkedList, and a list of staff stored in a HashMap as illustrated in the UML diagram of the class **`HotelManager`**.

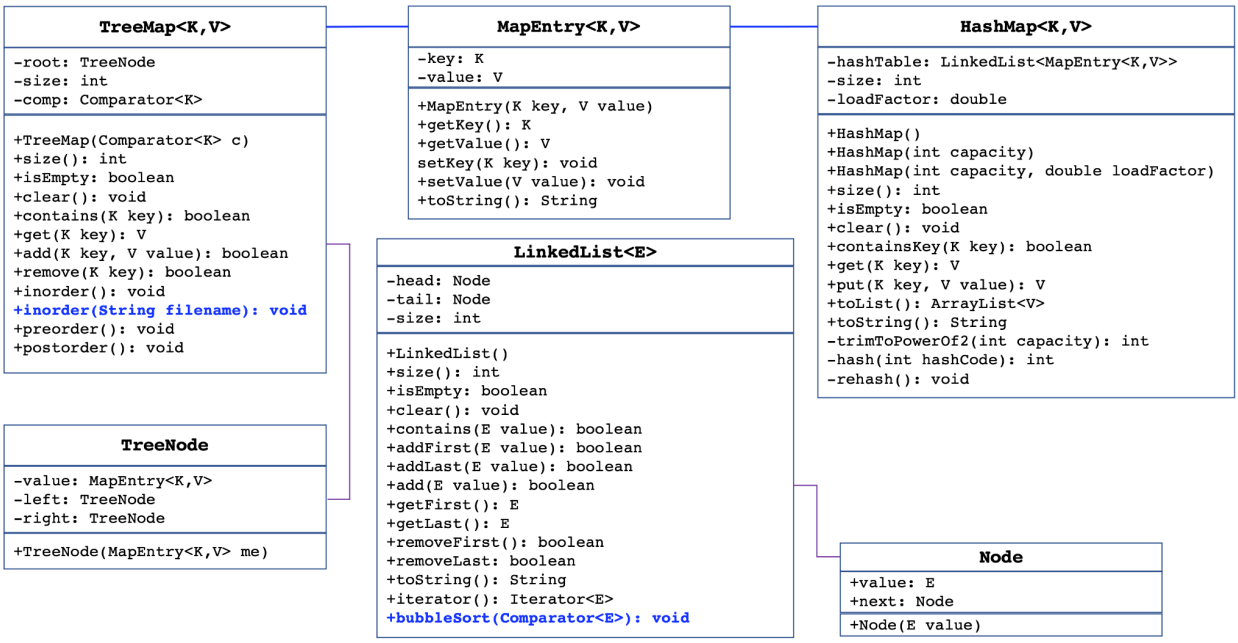

Figure 1: UML diagram of the provided data structures

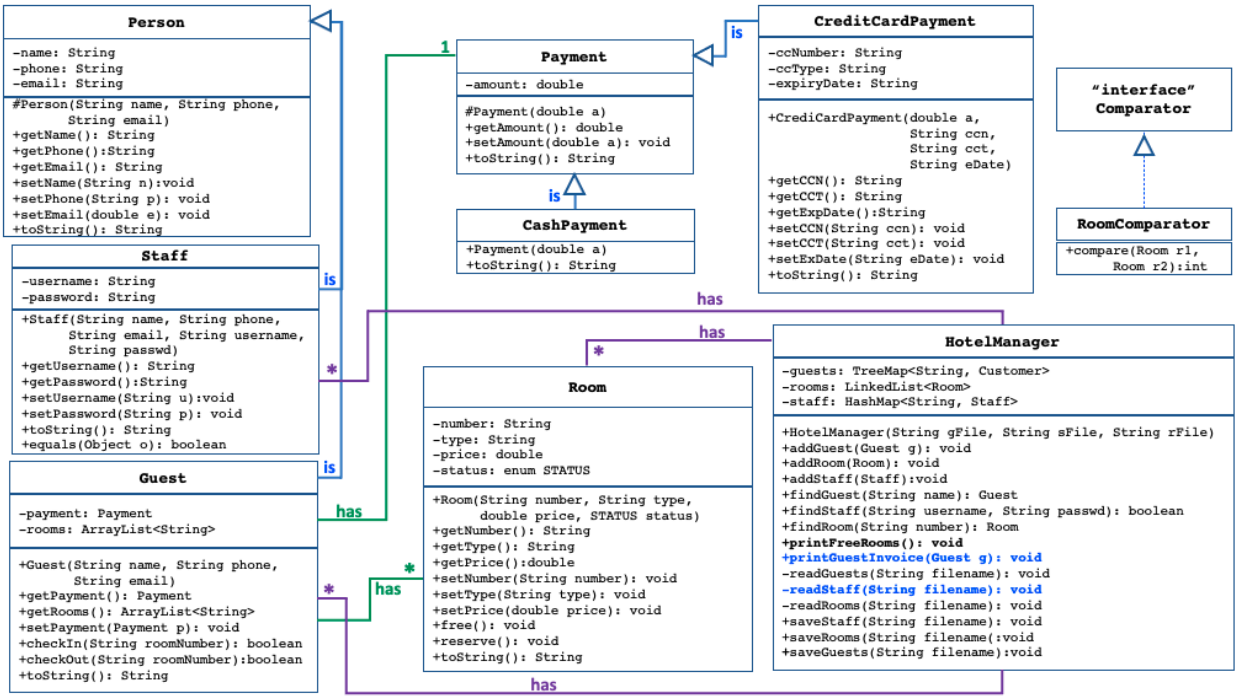

Figure 2: UML diagram of the provided program

All the classes in figure 1 and figure 2, the Test class, and the data files are provided. Download all the files from the following links:

(Java Files) (Data Files)

Take a few minutes to familiarize yourself with each class's attributes and interface and to understand the test class.

You are asked to add the definitions of five methods as described below:

1. **`checkEmail(String email)`** in the class **`Test`**. The method returns **`true`** if the email address format is valid or throws an exception of type **`Exception`** if not. The method must declare throwing an exception of type **`Exception`** and use regular expressions to check the email address format. Here is the regular expression for a valid email address:

```
([a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.(com|org|edu)
```

The method is invoked in the method **`guestOperations`** inside the class **`Test`**.

2. **`readStaff(String filename)`** in the class **`HotelManager`**. The method reads the information of the hotel staff from **`filename`** and stores it in the hashmap **`staff`**. The staff username is used as the key and the Staff object as the value. This method is invoked by the **`HotelManager`** class constructor to read the staff information from the file **`staff.txt`**.

3. **`printGuestInvoice(Guest guest)`** in the class **`HotelManager`**. The method finds **`guest`** in the treemap **`guests`** and prints her/his list of checked in rooms and the total price.

4. **`bubbleSort(Comparator<E> comp)`** in the class **`LinkedList`**. The method sorts the nodes of the linked list using a bubble sort algorithm and uses the Comparator object **`comp`** to compare the values of the nodes. Determine the time complexity of **`bubbleSort`** using Big-O notation and write it as a comment before the method header. Note that **`bubbleSort`** is invoked by the method **`printFreeRooms()`** with a comparator object of type **`RoomComparator`**. Use the bubble sort algorithm provided in the file **`Sort.java`** and modify it to work with a linked list and a comparator object.

5. **`inorder(String filename)`** in the class **`TreeMap`**. The method writes the values of the nodes of the treemap to **`filename`** using the inorder traversal. (Hint: Modify the **`inorder()`** method to print to a **`PrintWriter`** object instead of **`System.out`**). Determine the time complexity of **`inorder(String)`** and add it as a comment before the method header. Note that **`inorder(String)`** is invoked by the method **`saveGuests()`** in the **`HotelManager`** class.

Writing the five methods above will result in modifying the following classes only:

`TreeMap.java`, **`LinkedList.java`**, **`HotelManager.java`**, and **`Test.java`**

You must submit all the project Java files listed below:

`-MapEntry.java`,

`-HashMap.java`,

`-TreeMap.java`,

`-LinkedList.java`,

`-Person.java`,

`-Guest.java`,

`-Staff.java`,

`-Payment.java`,

`-CashPayment.java`,

`-CreditCardPayment.java`,

`-Room.java`,

`-RoomComparator.java`,

`-HotelManager.java`, and

`-Test.java`.

Javadoc comments **<u>are not</u>** required.

Sample Program Outputs are provided below to test the added methods.

-- Test 1: bubbleSort method (Free rooms sorted by price)--

```
Enter username or guest:
guest

Select an operation:
1: View free rooms
2: Check in a room
3: Print invoice
4: Quit
1
Room#           Type            Price/night
103             Standard        $180.00
105             Standard        $180.00
311             Standard        $180.00
490             Standard        $180.00
329             Accessible      $350.00
515             Joint           $350.00
104             Suite           $680.00
211             Connecting      $700.00
412             Deluxe          $700.00
415             Deluxe          $700.00
215             Suite           $800.00
312             Suite           $800.00
```

-- Test #2: checkEmail and printGuestInvoice methods --

```
Enter username or guest:
guest

Select an operation:
1: View free rooms
2: Check in a room
3: Print invoice
4: Quit
1
Room#           Type            Price/night
103             Standard        $180.00
105             Standard        $180.00
311             Standard        $180.00
490             Standard        $180.00
329             Accessible      $350.00
515             Joint           $350.00
104             Suite           $680.00
211             Connecting      $700.00
412             Deluxe          $700.00
415             Deluxe          $700.00
215             Suite           $800.00
312             Suite           $800.00

Select an operation:
1: View free rooms
2: Check in a room
3: Print invoice
4: Quit
2
Enter the room number:
215
Enter the number of nights:
2
Enter your full name:
Alexandra Newman
Enter your phone number:
6102274100
```

```
Enter your email:
anewman.gmail.com
Invalid email address.
Try again.
Enter your email:
anewman@gmail.com
Enter the type of payment (cash/credit):
cash

Select an operation:
1: View free rooms
2: Check in a room
3: Print invoice
4: Quit
3
Enter your full name:
Alexandra Newman
Invoice for Alexandra Newman
Room#           Type            Price
215             Suite           $800.00
Total:                          $1600.00

Select an operation:
1: View free rooms
2: Check in a room
3: Print invoice
4: Quit
4
```

-- Test #3: inorder method (guest file updated after Test #2)--

```
Enter username or guest:
guest

Select an operation:
1: View free rooms
2: Check in a room
3: Print invoice
4: Quit
3
Enter your full name:
Alexandra Newman
Invoice for Alexandra Newman
Room#           Type            Price
215             Suite           $800.00
Total:                          $1600.00

Select an operation:
1: View free rooms
2: Check in a room
3: Print invoice
4: Quit
4
```

```
-- Test #4: readStaff method (Staff login)--
Enter username or guest:
lub110
enter password:
zxcvGH23@q
Login Failed. Try again.
enter username:
lub110
enter password:
zxcvGH23@qwe

Select an operation:
1: Check out a room
2: Print customer invoice
3: Quit
2
Enter the customer name:
Marie Charles
Invoice for Marie Charles
Room#           Type            Price
411             Connecting      $700.00
102             Standard        $180.00
Total:                          $880.00

Select an operation:
1: Check out a room
2: Print customer invoice
3: Quit
1
Enter the guest full name:
Marie Charles
Enter the room number:
411

Select an operation:
1: Check out a room
2: Print customer invoice
3: Quit
2
```

```
Enter the customer name:
Marie Charles
Invoice for Marie Charles
Room#          Type           Price
102            Standard       $180.00
Total:                        $880.00

Select an operation:
1: Check out a room
2: Print customer invoice
3: Quit
3
```