HOTEL PMS

HOTEL PROPERTY MANAGEMENT SYSTEM

Nikoloz Naskidashvili - nikoloz.naskidashvili.1@gmail.com

General Description

A Hotel Property Management System (PMS) is a comprehensive software solution designed to streamline hotel operations, manage bookings, handle guest information, and facilitate efficient guest services.

Hotel PMS has following features:

1. Rooms for guests
2. Ability to add room to a hotel
3. Ability to remove room from a hotel
4. Ability to display (print) hotel room information

HOTEL PMS STRUCTURE

We will have the following classes in the software:

1. Room - The room itself
2. HPMS - Hotel Property Management System itself
3. HPMSTester - Hotel Property Management System tester class

Class Room	
Integer	roomNumber
Integer	floor
String	maxGuests
String	pricePerNight

Class HPMS	
List <Room>	rooms
void addRoom(Room)	
void removeRoom(Room)	
void getRooms()	
void displayRooms()	

Class Room

This class should have all the fields shown in the table above. This class can be implemented as following:

```
package midterm.nikoloz_naskidashvili_1.task3;

public class Room {
   private Integer roomNumber;
   private Integer floor;
   private String maxGuests;
   private String pricePerNight;

   /**
   * Room default Constructor
   */
   public Room() {

   }

   /**
    * Room Constructor
    * @param roomNumber - room number
    * @param floor - floor number
    * @param maxGuests - max guests
    * @param pricePerNight - price per night
    */
   public Room(Integer roomNumber, Integer floor, String maxGuests, String
pricePerNight) {
       this.roomNumber = roomNumber;
       this.floor = floor;
       this.maxGuests = maxGuests;
       this.pricePerNight = pricePerNight;
   }

   /**
    * Get room number
    * @return room number
    */
   public Integer getRoomNumber() {
       return roomNumber;
   }

   /**
    * Set room number
    * @param roomNumber - room number
    */
```

```java
    public void setRoomNumber(Integer roomNumber) {
        this.roomNumber = roomNumber;
    }

    /**
     * Get floor number
     * @return floor number
     */
    public Integer getFloor() {
        return floor;
    }

    /**
     * Set floor number
     * @param floor - floor number
     */
    public void setFloor(Integer floor) {
        this.floor = floor;
    }

    /**
     * Get max guests
     * @return max guests
     */
    public String getMaxGuests() {
        return maxGuests;
    }

    /**
     * Set max guests
     * @param maxGuests - max guests
     */
    public void setMaxGuests(String maxGuests) {
        this.maxGuests = maxGuests;
    }

    /**
     * Get price per night
     * @return price per night
     */
    public String getPricePerNight() {
        return pricePerNight;
    }

    /**
     * Set price per night
     * @param pricePerNight - price per night
```

```
     */
    public void setPricePerNight(String pricePerNight) {
        this.pricePerNight = pricePerNight;
    }

    /**
     * Get the information of the room.
     * @return The information of the room.
     */
    public String toString() {
        String result = "Room number: " + roomNumber + "\n";
        result += "Floor: " + floor + "\n";
        result += "Max guests: " + maxGuests + "\n";
        result += "Price per night: " + pricePerNight;
        return result;
    }
}
```

* Note: pay attention that in addition to a constructor method, class has getter and setter methods for each of the private fields. By default, all fields are private to prevent others from altering or overriding our fields.

What's more, our class has *toString()* method, which conveniently returns room information as a String object.

Class HPMS

The Hotel Property Management System has all methods described above in the table and attribute rooms to store the hotel rooms.

It also implements the helper method *displayRooms()*, which prints all information about hotel room state.

```
package midterm.nikoloz_naskidashvili_1.task3;

import java.util.ArrayList;

public class HPMS {
    ArrayList <Room> rooms = new ArrayList<>();

    /**
     * Adds a new room to the system.
     * @param room The room to be added.
     */
    public void addRoom(Room room) {
        rooms.add(room);
```

```
    }

    /**
     * Removes a room from the system.
     * @param room The room to be removed.
     */
    public void removeRoom(Room room) {
        rooms.remove(room);
    }

    /**
     * Returns a list of all rooms in the system.
     * @return List of all rooms in the system.
     */
    public ArrayList<Room> getRooms() {
        return rooms;
    }

    /**
     * Displays all rooms in the system.
     */
    public void displayRooms() {
        if (rooms.isEmpty()) {
            System.out.println("No rooms available.");
        } else {
            for (Room room : rooms) {
                System.out.println("------------");
                System.out.println("HOTEL STATE:");
                System.out.println("------------");
                System.out.println(room.toString());
                System.out.println("------------");
            }
        }
    }
}
```

Class HPMSTester

In this class we first initialize the HPMS class instance. Display empty rooms. After that we create rooms and add them to our hotel. Finally we display rooms and also remove a few rooms to ensure all the methods are working correctly.

```java
package midterm.nikoloz_naskidashvili_1.task3;

public class HPMSTester {
    /**
     * Main method showcasing task 3 solution.
     */
    public static void main(String[] args) {
        // Create HPMS object
        HPMS hpms = new HPMS();

        // Display rooms
        hpms.displayRooms();

        // Test getters and setters
        Room room = new Room();
        room.setRoomNumber(1);
        room.setFloor(1);
        room.setMaxGuests("2");
        room.setPricePerNight("100");

        System.out.println("Room created by getters and setters:");
        System.out.println(room.toString());

        // Create rooms
        Room room1 = new Room(1, 1, "2", "100");
        Room room2 = new Room(2, 2, "3", "150");

        // Add rooms to HPMS
        hpms.addRoom(room1);
        hpms.addRoom(room2);

        // Display rooms
        hpms.displayRooms();

        // Remove room
        hpms.removeRoom(room1);

        // Display rooms
        hpms.displayRooms();
    }
}
```