

**Due Date: 06.01.2026, 23:55**

## **SEDS562**

### **Homework 2: Hotel Reservation System**

This homework will cover the topics given below:

- Stacks
- Queues
- Lists
- File I/O
- Generics
- Arrays

You are expected to implement a “Hotel Reservation System” application using Java.

Assume that there is an online hotel reservation platform that manages room bookings for various types of accommodations, including single rooms, double rooms, suites, and deluxe rooms. The platform handles customer reservations and organizes room allocations based on availability.

The information to be provided for each room reservation is outlined in the following table. The properties are to be emphasized in bold.

Rooms:

<b>Property</b>	<b>Description</b>
<b>Room Type</b>	The type of accommodation (Single, Double, Suite, Deluxe)
<b>Room Number</b>	The unique identifier for the room
<b>Availability</b>	Indicates whether the room is available or booked

Reservations:

<b>Property</b>	<b>Description</b>
<b>Reservation ID</b>	The unique identifier for the reservation
<b>Customer Name</b>	The name of the customer making the reservation
<b>Room Type</b>	The type of room reserved

**Your application is expected to perform the following operations:**

1. For each type of room in the hotel (single, double, suite, deluxe), create 5 rooms. Each room should have properties as room number, type (single, double, suite, deluxe), and availability (available or not). You are expected to give the numbers of the rooms. This means that the room number of the first room is 1; the room number of the next room

is 2 and the room number of the last room is 20. Therefore, there will be 20 rooms created in total.

2. For each different food category, create **a pile of rooms** to manage the availability of rooms. In the pile, room having the **smallest** room ID is at the top of the pile. (Hint: Room number 1 will be at the top of the pile) The room put onto the pile last will be first to be out. (There will be 4 different piles of rooms.) While giving room number, follow the order of single, double, suite and deluxe. Give first 5 numbers to single rooms, give numbers 6 to 10 to double rooms, 11 to 15 to suite rooms and 16 to 20 to deluxe rooms. All rooms are available at the beginning.
3. Read reservation information from the **reservations.txt**, which includes reservation ID, customer name, and room type. Create **a waiting line of reservations for each room type**. (There will be 4 different waiting lines of reservations for each room type.) Use these waiting lines of reservations to manage the waiting reservations. The reservation that is put first in the waiting line, will be the one to come out first. When a customer requests a reservation and the desired room type is not available, add the reservation to this line. If the room is available for the reservation, hold this booked room in a list.
4. After all reservations in the **reservations.txt** are processed (booked or added to the waiting line of reservations), make all booked odd numbered rooms available. Then, this time process the reservations in the waiting lines of reservations. When a room becomes available, remove the first reservation from the waiting line and assign the room as booked. If a customer waits for a suite type room and this room is available now, make the reservation for that room. When all necessary processes are done, waiting lines of reservations should change at the end. As in the first process, rooms will be reserved starting from smallest numbered room. (If there are two single rooms available numbered as 1 and 3, 1 will be reserved first.)
5. Manage the final situation of rooms. Create **two lists** of the situations of rooms. If a room is booked, add this room to the **Unavailable Rooms**, if room is empty, add this room to the **Available Rooms**.
6. Print on the console, the **contents** of
  - **four** different piles of rooms in the desired order (before and after processing the reservations)
  - **four** waiting lines of reservations for each room type after reading **reservations.txt** file and after the change in the odd numbered rooms separately.
  - Unavailable Rooms list and Available Rooms list

Single pile of room

Double pile of room

Suit pile of room

Deluxe pile of room

\*\*\*\*\*  
(after reading and processing **reservations.txt**)

Single waiting line of reservations  
Double waiting line of reservations  
Suit waiting line of reservations  
Deluxe waiting line of reservations

\*\*\*\*\*  
(situation after all booked odd numbered rooms available, piles should be ready to managing reservations in the waiting line of reservations but not processing reservations yet, piles should contain available rooms to process waiting lines)

Single pile of room  
Double pile of room  
Suit pile of room  
Deluxe pile of room

\*\*\*\*\*  
(situation after processing waiting line for all booked odd numbered rooms available)

Single waiting line of reservations  
Double waiting line of reservations  
Suit waiting line of reservations  
Deluxe waiting line of reservations  
\*\*\*\*\*

Unavailable Rooms

\*\*\*\*\*

Available Rooms

### Submission Rules

All submissions must:

- be performed via **Microsoft Teams**,
- be exported as an Eclipse Project and saved in ZIP format,
- include all necessary data files (if any TXT, CSV, JSON, etc.) in the right directory,
- follow a specific naming convention such that *StudentID*\_SEDS562\_HW2.

**Eclipse Project:** *StudentID*\_SEDS562\_HW2

**Exported Archive File:** *StudentID*\_SEDS562\_HW2.zip