

Title: Seat/Client Assignment- Class Project

Initial First Pass – POC

Table of Contents

Goal	2
Initial Core Features	2
Rationale	2
INPUTS	3
1) Internal List of Clients	3
2) Room Matrix	3
PROCESSING.....	4
OUTPUTS.....	4
1) Seating Matrix.....	4
2) Clients List and their seating assignment.....	4
Project Criteria	4
Project Folder.....	5

Goal

To create an application which assigned clients to seats similar to a booking system for a movie theatre.

In this Initial First Pass, we are isolating and developing the most basic core features required for this project to build upon.

Initial Core Features

- contain a list of clients
- contain an internal matrix of a room which defines usable areas (where seating is possible)
- automatically display the merge of clients in list with available seats (randomly)
 - display matrix with seating arrangement
 - also list each client and their seating assignment

Rationale

The very core of the app is to marry a list of 1 to several clients to a matrix and be able to output the matrix and a list of clients and their seating arrangement.

As a proof of concept (POC) the input is provided via internal arrays and/or IO external file list which can be parse and loaded into an array.

The randomized processing is merely in lieu of manually assigning a client to a seat since in this iteration we are not taking any manual input, but the aspect of checking for the availability of a seat and the actual assignment will be the same.

The output is automatic at this point since the app doesn't require any user interactions in this iteration.

We will more than likely have a 'raw' unassigned matrix and then create a copy ready to use.

We will marry the list of clients to seating positions based on an association of IDs to seating.

The assignment process to the matrix can also create a list of clients and their seating.

INPUTS

1) Internal List of Clients

id	lastname	firstname
1	Mellor	Alexander
2	Schalk	Alexander
3	Storey	Alice
4	Montgomery	Andrew
5	Leighton	Angela
6	Torres	Brian Jesus Cedillo
7	Catzin	Chris
8	De Hoyos	Daniel
9	Loughran	David
10	Sobers	Deron F.
11	Hunter	Devlin Chase
12	Buchanan	Elbert
13	Silva	Issic
14	Guevara	Joel Reyes
15	Mills	John
16	Patiag	Jonathan
17	Chacko	Joseph
18	Williams	Lonnie
19	John	Marilyn
20	Kirk	Robert
21	Mao	Hannah Mao
22	Hand	Trey
23	Chmielewski	Victoria
24	Ubak	Yvonne
25	Gauthier	Claude

2) Room Matrix

Row/Column Array of objects which will contain the following info

- Initials of Client: "" ← starts as empty
- Client ID: "" ← starts as empty
- row/column reference (rows are letters A to ...) and columns are numbers 1 to ...
- canBeAssigned = true or false, as we are shaping a classroom, some row/column will never get assigned

IF the Client ID and Initials of Client are both empty, the 'seat' is considered "vacant"

Room Matrix Size

10 rows, 10 columns

PROCESSING

To randomly assigned a person to a row/column and ensure there are no duplication or overrides.

OUTPUTS

1) Seating Matrix

	Seat 1	Seat 2	Seat 3	Seat 4	Seat 5	to Seat 10
Row A	id: 01/CG					
Row B	vacant					
Row C						
To Row J						

Hint: research how you can encapsulate the printing of content is a pseudo-column like way to ensure your command prompt output looks like a grid.

2) Clients List and their seating assignment

Client	ID	Seat
Claude Gauthier	01	A-1
...		

Project Criteria

- Must be worked alone, no outside help, this is going to be work that will be evaluated for your MID and FINAL terms
- Demonstrate pragmatism, for example:
 - use an iterative approach in your development process, create versions/subversions of your app in stage
 - Use what you have mastered as a coder, keep it simple, when a version works, if you feel it could be better, then document and identify areas your code can be improved, perhaps with more advanced version of Java concepts and syntax and then, create a new version of your code
- By demonstrating pragmatism, you will deliver faster, albeit not always want you WISH you would have coded, but at least, it will work exactly as per minimum specifications. Your hiring manager can be made aware of your progress and help guide you towards your development, but only if you are able to document and clearly explain your thought process, your development strategy and the end goal you are trying to achieve.
- KEEP IT SIMPLE

Note: those who do provide help, you may be risking your own career over this. Do not help. It is up to all employees who require help, to see the manager's attention. If you are caught, there will be consequences, again, beware.

Project Folder

Folder: `javatraining/preterm/` ← folder

if you have any iteration, organize your code as per version, perhaps folder version naming convention for version of full projects such as version01, version02.. etc.. at the room of the preterm folder, have a **README.md** file which clearly indicates each version and the features you've added and any other comment which is relevant to the project's evolution. "missing features, bugs to fix, etc.."