code.sprint MT

TASK BOOKLET

- Qualifiers Round -

Secondary Category

2023

ICE Malta

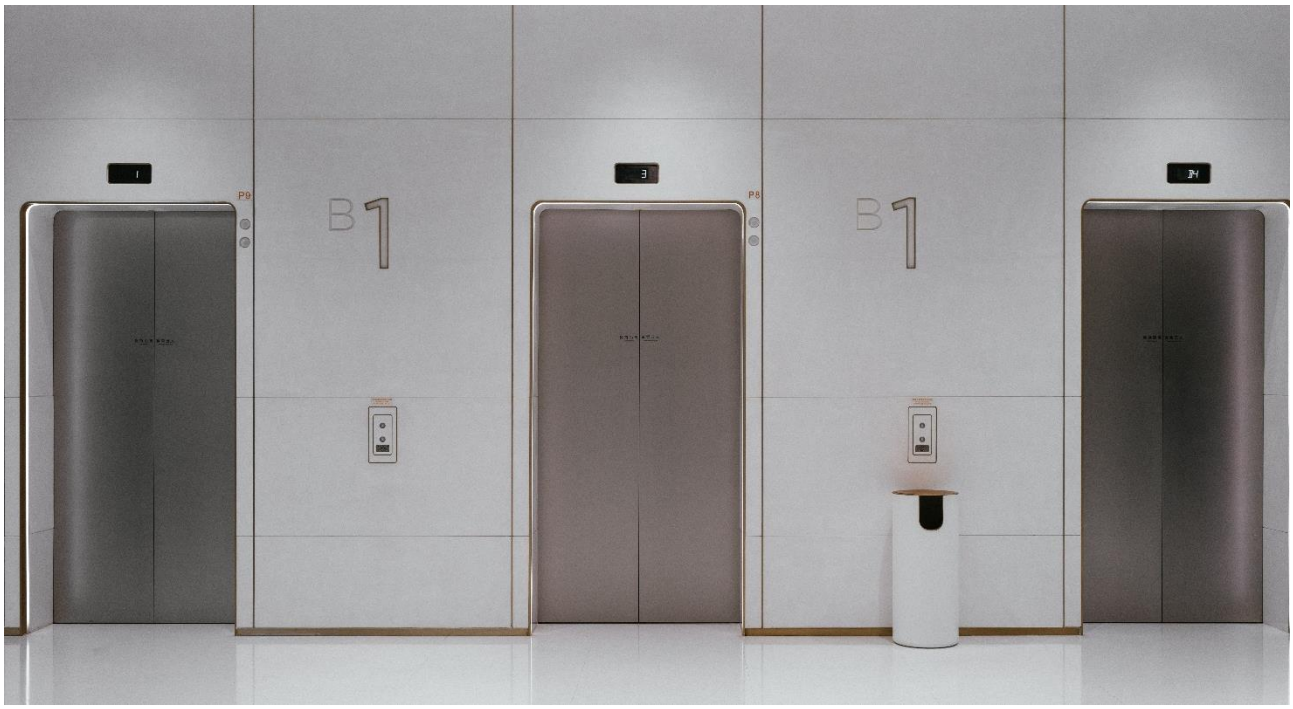
GOVERNMENT OF MALTA
MINISTRY FOR EDUCATION, SPORT, YOUTH, RESEARCH AND INNOVATION
DIRECTORATE FOR LEARNING AND ASSESSMENT PROGRAMMES

Qualifiers Round Schedule

- Task Analysis (10 min)
- Task Development (60 min)
- Interval (15 min)
- Continuation of Task Development (60 min)
- Task Submission (5 min)
- **TOP 10 FINALISTS ANNOUNCED!**

CodeLift: Rise to the Elevator Simulation Challenge!

In the bustling world we live in, elevators are essential for vertical transportation. To ensure their smooth operation, sophisticated software systems govern their movement, safety features, and user interactions.



In CodeLift, your task is to simulate an elevator software using Java. Design a text-based program that simulates a hotel elevator system to transport passengers between the following floors, considering a maximum passengers capacity limit of four (4) passengers:

- Garage Floors (-2 and -1): Entry/exit for hotel guests.
- Ground Floor (Reception): Hotel access point.
- Guest Room Floors (1-5): Transport to guest rooms.

Functionality #1: Elevator Control Interface

1. The control panel prompts to enter the number of passengers who are entering the elevator whenever the elevator door opens. Inputting [X] indicates that no passengers are getting on the lift.
2. Upon entering the elevator, passengers will be asked to input their desired floor. The available floor options are: [-2] [-1] [0] [1] [2] [3] [4] [5].

Functionality #2: Floor Movement

1. The elevator moves between floors based on passengers' choice. The lift should follow the sequence of floors entered by the passengers.

   For example, if passenger 1 enters [3], passenger 2 enter [-2], and passenger 3 enters [4], the elevator will start by going to floor 3, then to floor -2 and then to floor 4.

2. When the elevator stops on a floor chosen by the passenger, it is assumed that the respective passenger/s exit the elevator automatically.

   For instance, let's imagine there are three passengers inside the elevator with the following floor choices: passengers 1 and 2 select floor [4], while passenger 3 chooses floor [2]. As the elevator reaches floor 4, passengers 1 and 2 will automatically exit the lift, resulting in an updated passenger count of one.

3. If there are no passengers entering the lift, i.e. if the user presses [X], the elevator proceeds to the next floor if there are still passengers in it.

4. When there are no passengers in the elevator, the elevator remains waiting at the last floor it stopped on, awaiting further instructions.

5. Each floor transition takes 1s (1000ms) to complete.

   Code Hint: the below code displays the string 'Before Delay', then it waits for 2 seconds and then displays the string 'After Delay'. This sample code can be used to create a visual of the elevator movement.

```
System.out.println("Before delay"); //display on screen

//time delay instructions
try {
   Thread.sleep(2000); //Delay for 2 seconds (2000 milliseconds)
} catch (InterruptedException e) {
   e.printStackTrace();
}
System.out.println("\fAfter delay"); //display on screen
```

- The `try … catch` routine is used to wait for a number of milliseconds.
- The escape character "`\f`" clears the screen to refresh screen visuals.

Functionality #3: Lift Capacity

1. Track the number of passengers currently inside the lift.
2. Limit the number of passengers in the lift to a maximum of four (4) at any point in time.

Functionality #4: Validation Processes

1. Validation is required on the instructions inputted by the user:
   - Number of passengers entering the elevator or [X] for no passengers.
   - Passenger's floor choice when entering the elevator.

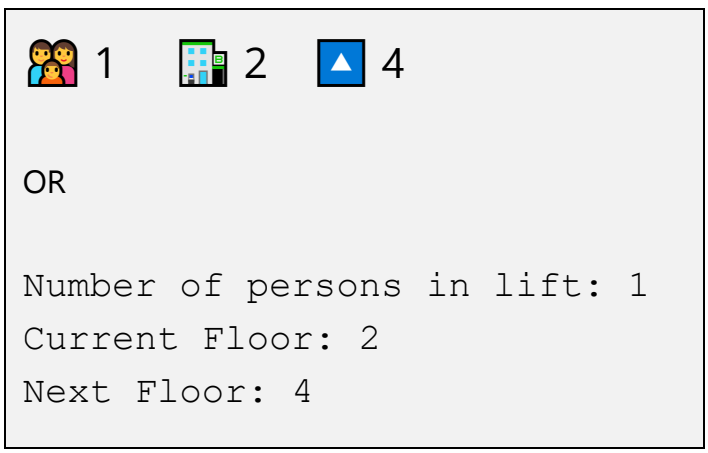
2. User instructions are not case sensitive.
3. A warning message should be prompted for invalid instructions.
4. Validation is required to avoid any possible runtime error.
5. Handle instances when exceeding the elevator's capacity.
6. Provide appropriate error messages and allow users to correct their input.

Functionality #5: User Interface.

Create a text-based interface that displays the current state of the elevator system and allows users to interact with it. Refer to screenshot 1, in page 5, for a comprehensive sample user interface.

Here are some suggestions for the user interface:

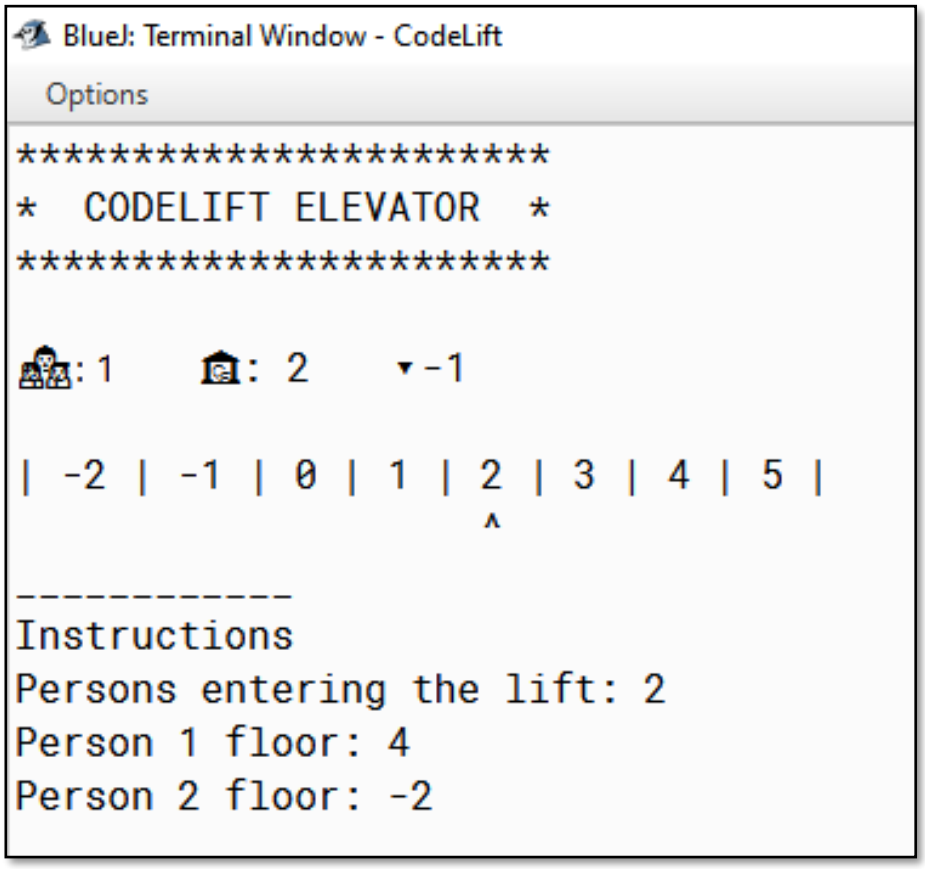
1. **Display the Current State**: Show the number of passengers inside, current floor of the elevator, and the next floor level movement. For example:



2. **Elevator Movement Visualization:** Create a visual representation of the elevator moving between floors. In the sample visualisation shown below the lift is represented using the ' ^ ' sign.

```
| -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 |
                    ^
```

Make sure that this visualisation is updated during the elevator movement, considering 1 second (1000 milliseconds) of delay from one floor to another.



Screenshot 1: Sample user-interface

Name the class containing the main method **RunApp.**

Submit your program in a folder named **CodeLift**

Assessment Rubric

<table>
<tr><td>Overall Program Functionality</td><td>User-Friendly Interface</td><td>Proper use of Comments</td><td>Proper Conventions
(Camel case, meaningful var names etc.)</td><td>Name of Folder & Class/es</td><td>User Input</td><td>Suitable Prompts / Messages displayed</td></tr>
<tr><td>Update of Elevator Status</td><td>Update Elevator Movement Visualisation</td><td>Animate Elevator movement between floors</td><td>Elevator movement follows passengers' floor sequence</td><td>Proper use of Data Structures</td><td colspan="2" rowspan="2">Maximum Score: 38
+ 2 for every extra feature.</td></tr>
<tr><td colspan="5">Validations</td></tr>
<tr><td>Number of passengers entering elevator or [x] for none</td><td>Floor Number.</td><td>Elevator Capacity Limit</td><td>Avoid Runtime errors</td><td>Instruction Case Sensitivity</td><td>Modular Code</td><td>Code Efficiency</td></tr>
<tr><td colspan="7">0 – Not Satisfactorily | 1- Partly Satisfactorily | 2- Entirely Satisfactorily</td></tr>
</table>


eSkills
eSkills Malta Foundation


LeoVegas


me direct