Elevator Simulation System - Documentation

Author: Akshay Kumar Boda

College: Sree Dattha Engineering & Science College

Course: B.Tech, 3rd Year

# 1. What is the Project?

This project is a simulation of a basic elevator (lift) system in a multi-story building. It is designed using Java programming language and applies object-oriented principles to mimic real-world elevator operations.

# 2. Why this Project?

Elevators are common in buildings, and simulating one in code helps understand core programming concepts such as:  
- Object-oriented design (classes, enums, encapsulation)  
- Real-time decision making  
- Control flow and state management  
- User interaction and input validation  
This project serves as a foundation for learning and applying Java in a meaningful and practical scenario.

# 3. Class: Direction.java

Package: elevator

## Description:

Defines an enum for elevator movement directions: UP, DOWN, and IDLE.

## Enum Values:

- UP: Elevator is moving upward  
- DOWN: Elevator is moving downward  
- IDLE: Elevator is not moving

# 4. Class: Elevator.java

Package: elevator

## Description:

Contains the core logic of the elevator system, including request handling, floor processing, direction control, and movement.

## Constructor:

- Elevator(): Initializes the elevator's current floor, direction, and storage for requests and destinations.

## Fields:

- MIN\_FLOOR / MAX\_FLOOR: Range of valid floors (0 to 10)  
- processingTime: Delay in milliseconds for moving between floors  
- currentFloor: Current floor of the elevator  
- currentDirection: Current direction of travel (UP, DOWN, IDLE)  
- requestedPathsMap: Stores pending requests as start -> destination list  
- currentFloorDestinations: Boolean array for marking destination stops

## Methods:

- setProcessingTime(int): Sets time delay between floors  
- getCurrentFloor(): Returns current floor  
- getrequestedPathMap(): Returns request map  
- getCurrentFloorDestinations(): Returns destination map  
- start(): Main loop that processes elevator operation  
- lunchTimeElevatorRush(): Generates 30 random requests  
- callElevator(int, int): Adds a request to the elevator  
- processFloor(int): Handles boarding/unboarding at each floor  
- moveElevator(): Controls direction logic and calls movement functions  
- moveUP(): Moves elevator one floor up with delay  
- moveDown(): Moves elevator one floor down with delay  
- isInvalidFloor(int): Checks if the floor number is out of range

# 5. Class: ElevatorChallenge.java

Package: elevator

## Description:

This is the main class that runs the program and handles user interaction.

## Methods:

- automaticElevator(): Runs elevator in auto mode using 30 random requests  
- manualElevator(): Takes user input for start and destination and runs elevator  
- main(): Entry point, calls both manual and automatic modes