Breakdown of Classes

Uditi Madan

This ElevatorSimulation gives a comprehensive simulation of elevator operations, incorporating floors, elevators, and passengers class objects. The main method reads parameters from a properties file to initialize the simulation, which is then executed. The elevator load and unload, transit, and the arrival of fresh passengers are handled by the simulation iteratively through ticks. It also calculates arrival-conveyance times and reports the statistics of average, longest, and shortest time between passenger arrival and conveyance. I tried to encapsulate a realistic elevator behavior, including considerations for passenger arrivals, elevator movement, and load balancing.

The main `ElevatorSimulation` code simulates the operation of elevators in a building over a specified duration. Here's an explanation of the key components and logic:

`ElevatorSimulation` Class

1. **Attributes:**

   - `floor`: Represents the building with floors and elevators.
   - `tick`: A variable that tracks the simulation time.
   - `duration`: Specifies the total duration of the simulation.
   - `passengerProbability`: Represents the probability of a new passenger appearing during each tick.
   - `arrivalTimes` and `conveyanceTimes`: Array lists to store arrival and conveyance times for passengers (to be used for reporting).

2. **Main Method:**

   - The `main` method reads the configuration from the provided property file.
   - It initializes the simulation parameters such as the number of floors, elevator capacity, duration, etc.

- The loaded properties are then used to create an instance of `ElevatorSimulation`.

3. **Simulation Methods:**
   - `runSimulation`: Runs the simulation for the specified duration.
     - During each tick, it handles elevator unload/load, elevator travel, and new passengers.
     - Elevators unload passengers at their current floor and load passengers waiting at that floor.
     - Elevators move up one floor if they are not at capacity.
     - New passengers are generated with a specified probability.
     - Arrival and conveyance times for passengers are recorded.
   - `handleElevatorUnloadAndLoad`: Handles the unloading and loading of passengers for each elevator.
   - `handleElevatorTravel`: Moves elevators up by one floor if they are not at capacity. It also checks for pending requests in the current direction, and change direction if there are no pending requests. This approach mimics the behavior of a real elevator system that aims to conserve energy and serve requests efficiently.
   - `handleNewPassengers`: Generates new passengers based on the specified probability.
   - `generateRandomPassenger`: Creates a new passenger with a random start and destination floor.
4. **Results Reporting:**
   - The simulation results are reported at the end of the simulation.
   - This includes the average, longest, and shortest times between passenger arrival and conveyance to the final destination.

`Floor` Class

1. **Attributes:**
   - `numFloors`: Represents the total number of floors.

- `floorQueues`: A list of queues representing passengers waiting on each floor.
- `elevators`: A list of elevators on the floor.

2. **Methods:**

- `handleArrival`: Adds a passenger to the queue of the floor they want to depart from.
- `getNumFloors`: Returns the total number of floors.

`Elevator` Class

1. **Attributes:**

- `currentFloor`: Represents the current floor of the elevator.
- `capacity`: Maximum passenger capacity of the elevator.
- `passengers`: List of passengers currently in the elevator.

2. **Methods:**

- `loadPassenger`: Adds a passenger to the elevator.
- `unloadPassengers`: Removes passengers whose destination floor is the current floor.
- `moveToFloor`: Moves the elevator to a specified floor.
- `isAtCapacity`: Checks if the elevator is at its maximum capacity.
- `getConveyedPassengers`: Returns an array of passengers whose destination is the current floor.

`Passenger` Class

1. **Attributes:**

- `arrivalTime`: Represents the time when the passenger arrives.
- `startFloor`: The floor where the passenger initially appears.
- `destinationFloor`: The floor to which the passenger wants to travel.
- `conveyanceTime`: The time it takes for the passenger to be conveyed to the destination.

2. **Methods:**

    - `getArrivalTime`: Returns the time when the passenger arrives.
    - `getStartFloor`: Returns the floor where the passenger initially appears.
    - `getDestinationFloor`: Returns the floor to which the passenger wants to travel.
    - `getConveyanceTime`: Returns the time it takes for the passenger to be conveyed to the destination.