Simulation of an Elevator

Aim: To simulate an elevator in java using threads.

Requirements:

- The elevator has to serve n floors, where n > 1.
- The elevator car can only hold 300kg.
- People inside the elevator weigh between 40kg and 100kg.
- Elevator will only open its door on a particular floor if any person needs to get down at that floor or anyone is waiting to get into elevator at that floor.
- It is possible that there are no people on a particular floor, the elevator can skip such floors.

Design:

We have chosen to simulate the elevator using the Model – View – Controller (MVC) paradigm of program design.

- Controller:
  - Update View
  - Handles movement of elevator
- View:
  - The display outside every elevator showing the floor being served by the elevator.
- Model:
  - Stores the main logic of elevator.
  - Processes all the input received from controller.
  - Decides whether to open the door at a particular floor
  - Informs when the lift is overweight.

Implementation:

- An elevator is an independent thread (Elevator.java).
- Every person is a distinguished thread (Person.java).
- Controller creates elevator and person threads (Controller.java)
- The program maintains 3 Linked Lists with nodes of type Person:
  - upList: stores all the people who wish to go in upward direction.
  - downList: stores all the people who wish to go in downward direction.
  - inList: stores all the people currently inside the elevator.

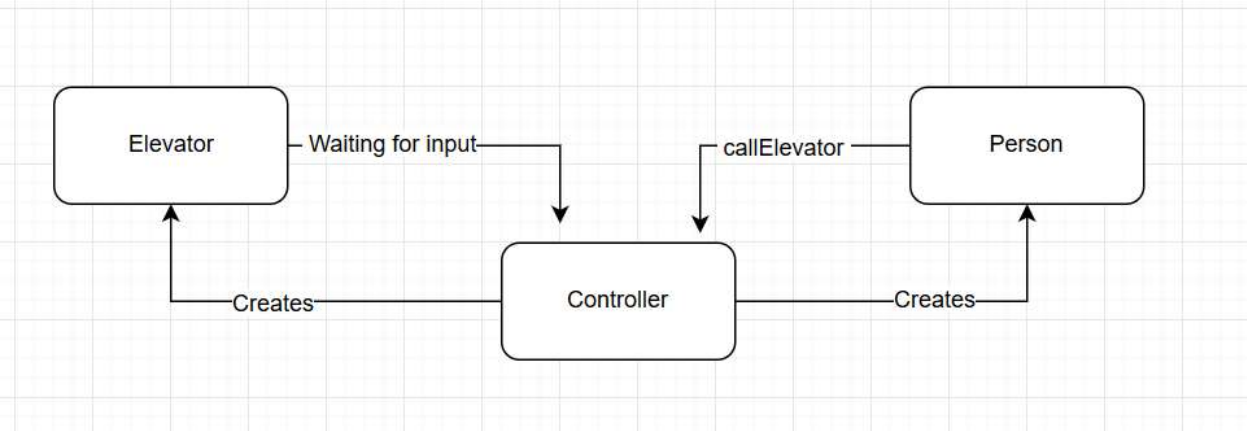

Fig. Initializing the environment

Flowchart:

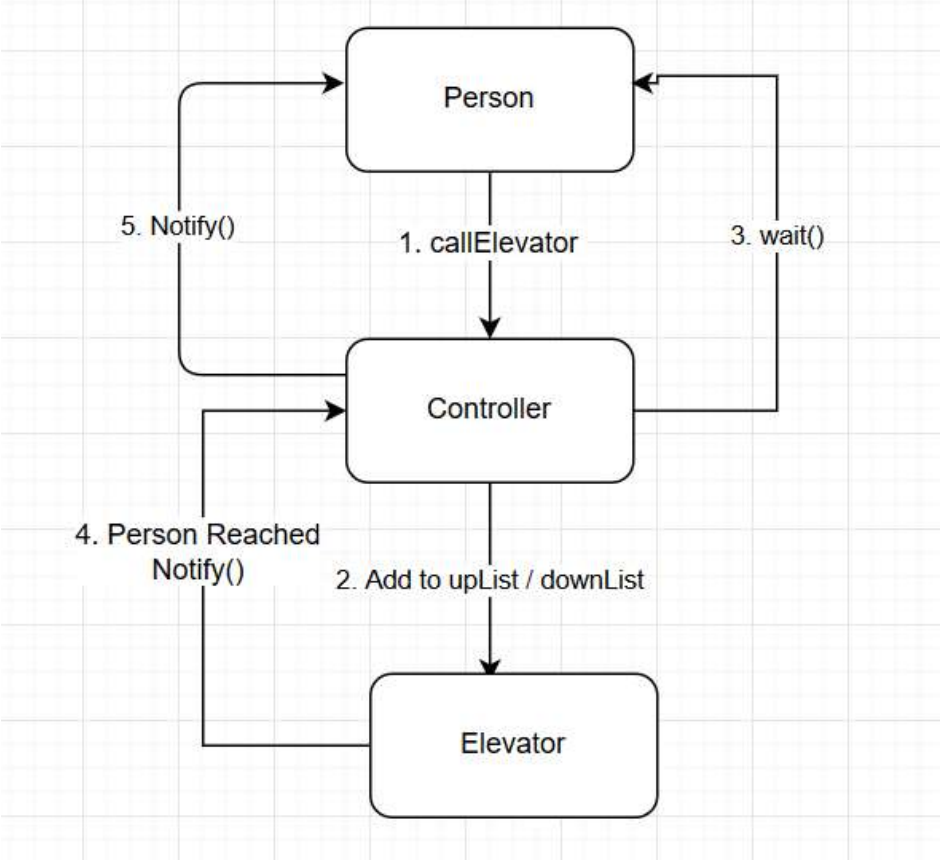

Fig. Flow of execution

Pseudocode:

1. Controller:

   Create elevator thread and start it.

   Create n people (n taken from user).

2. Person:

   Synchronize on the controller(common to all people)

   Call elevator using this controller

   Get added in one of the lists(upList/downList)

   Wait and allow other threads to add themselves to elevators lists.

3. Elevator:

   Direction of lift = up
   Check upList, if not Null serve the people going up.
   For every floor :
   Check if anyone wants to get down
   Check if anyone wants to enter the elevator
   Check downList, if not null serve the people going down.
   For every floor :
   Check if anyone wants to get down
   Check if anyone wants to enter the elevator

Note:

upList stores all the person threads in ascending order of their start floors.

downList stores all the person threads in descending order of their start floors.

Class Diagram:

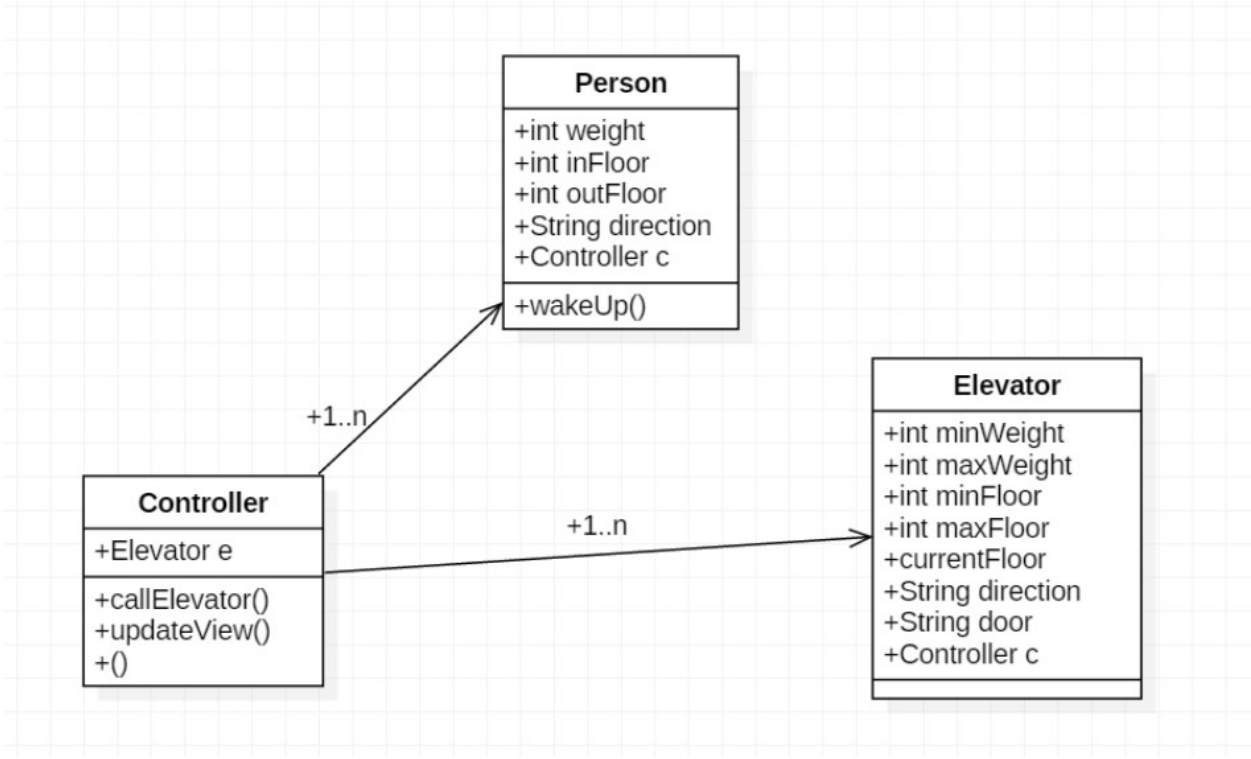

Fig. class diagram