

Assignment 0 - Object Oriented Programming

Yanir Cohen

Netanel Levine

1. - <https://www.youtube.com/watch?v=sigiJAJWUVg>
- <https://github.com/00111000/Elevator-Scheduling-Simulator>
- https://www.researchgate.net/publication/31595225_Estimated_Time_of_Arrival_ETA_Based_Elevator_Group_Control_Algorithm_with_More_Accurate_Estimation
2. The main difference between Online algorithm and Offline algorithm is that the Offline algorithm relies on collecting the entire input before starting any activity/calculations/decisions making, allowing it to see the entire picture before running. While the Online algorithm receives the input throughout the entire time of the program. Therefore it needs to adjust itself all the time according to the input of the exact time. Which means that the Online algorithm needs to consider all the time that it might receive a new input which might affect the efficiency of its last decision.

Offline Algorithm:

Our offline algorithm is similar to the Look algorithm with some modifications.

1. First we will divide it into 2 arrays , one for all the calls which are going up and one for all the calls which are going down.
2. If there are 2 calls in the same direction the elevator will go to the furthest in the zone and will pick up both (furthest means the call whose source floor is further from the destination), calculates the travel time between picking them both at once or picking one taking him to the destination and returning and picking the other up.
Note: all that assuming there isn't any other available elevator.
3. If there are any other calls in the route to the destination pick them up unless they are in the opposite direction.
4. When an elevator has reached the destination, go to the next and nearest call source floor and check 2.

3.

Online Algorithm

1. First divide into 3 cases: flag 0 , flag 1 and flag 2.

Flag 0 will be designed for a case in which the building has only one Elevator.

Flag 1 is designed for a building with slow Elevators and a high amount of floors.

Flag 2 is designed for a building with fast Elevators or a decent amount of floors .

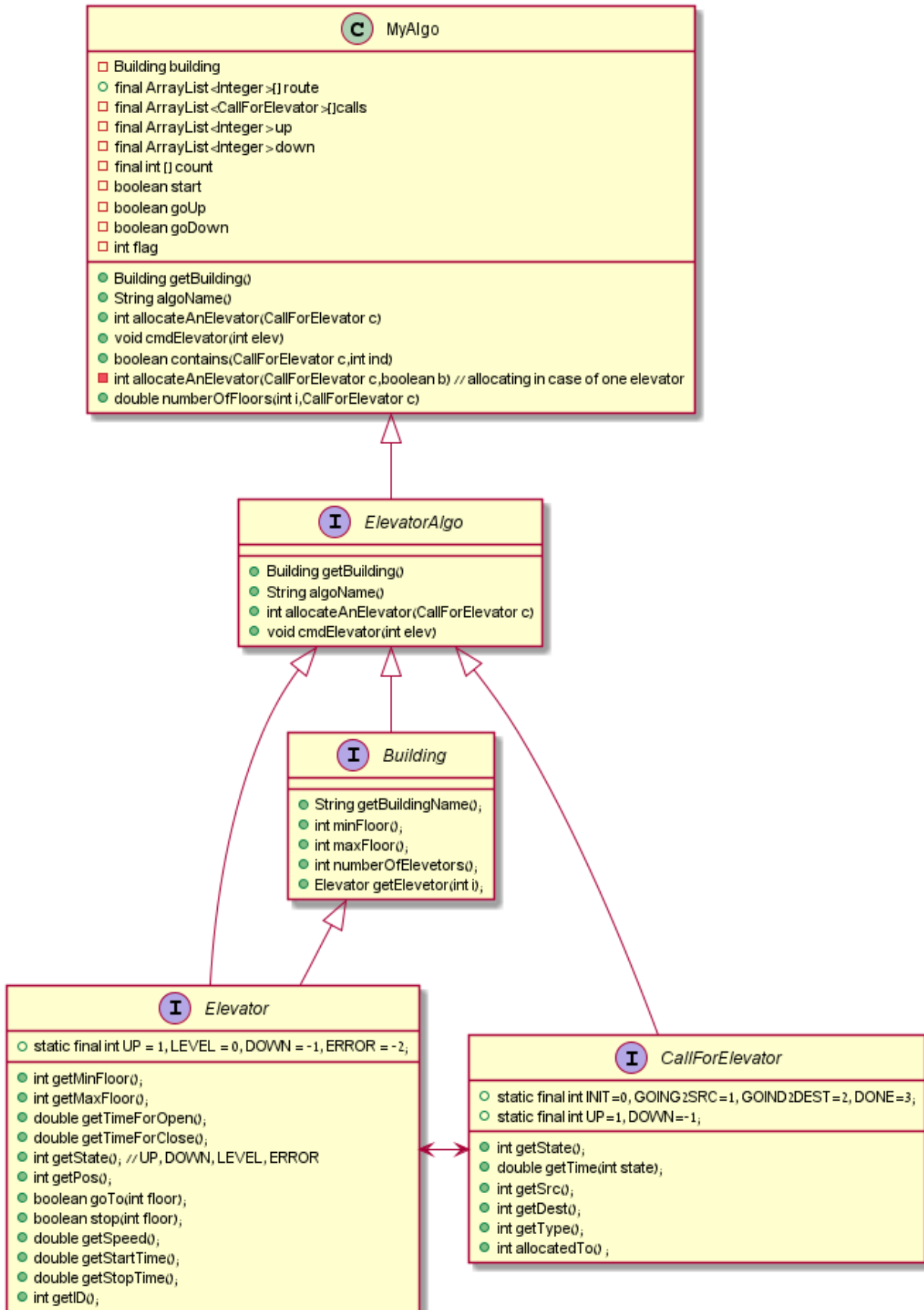
2. Flag 0 will check if there is a stop in between the calls and pick them up otherwise it will complete its' existing route.

3. Flag 1 will have an elevator waiting for long travels such as 60 floors
Meaning we will go over the route of each elevator in the building and will choose the elevator by which one will finish its route the fastest after adding the call.

This will be done by calculating the floors in each route and dividing by the speed, then adding to the one with the lowest time to finish.

However there will be an elevator which waits and picks up all the long calls and waits then travels again and again from top to bottom.

4. Flag 2 will go from the fastest to reach the call and finish it.
Meaning we will go over the route of each elevator in the building and will choose the elevator by which one will finish its route the fastest after adding the call.



5. Tests will be to our functions.

1. Calculate the expected time we want by parameters of the building and fail it if we did not reach the expected time.
2. See that our function assigns the call to the Elevator which will finish it the fastest after adding the call.
3. Amount of calls should be equally divided between the elevators taking into consideration speed.
4. Route size should not be over 10 in short scenarios after the scenario ended.
5. Check different scenarios for each flag, and see that all the elevators go according to the