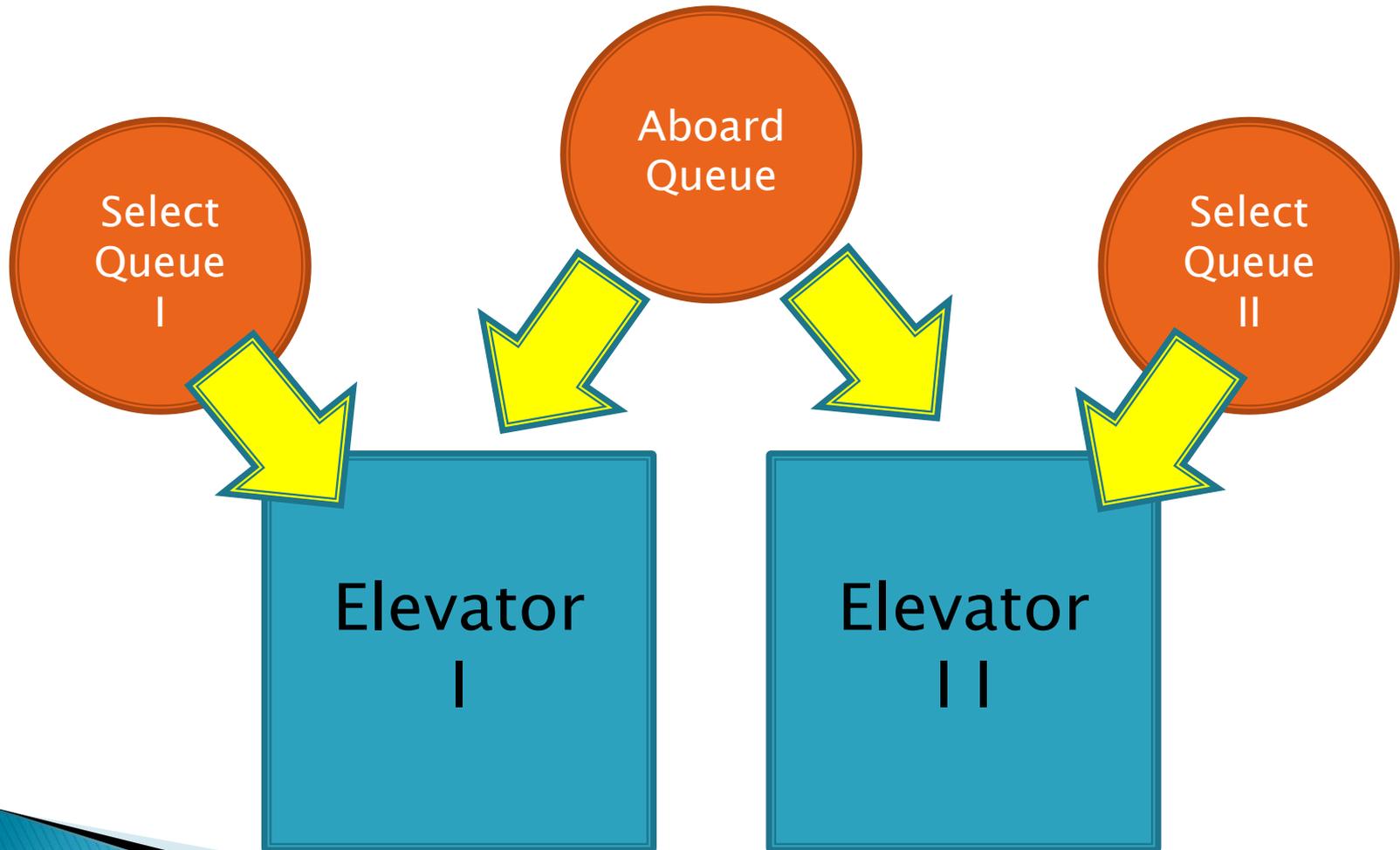


FLUXVATOR

200913215 이인구

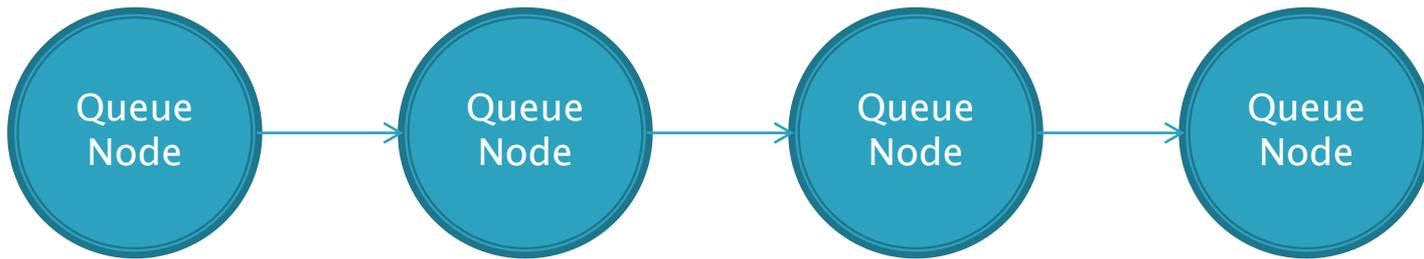
201013275 강태호

구현 중점 - 동적 큐 할당



구현 중점 - 큐

- ▶ Linked List -> 자바 기본 라이브러리 사용
기존 계획에 있던 prev, next 삭제.
- ▶ 자바 기본 제공 Iterator 사용

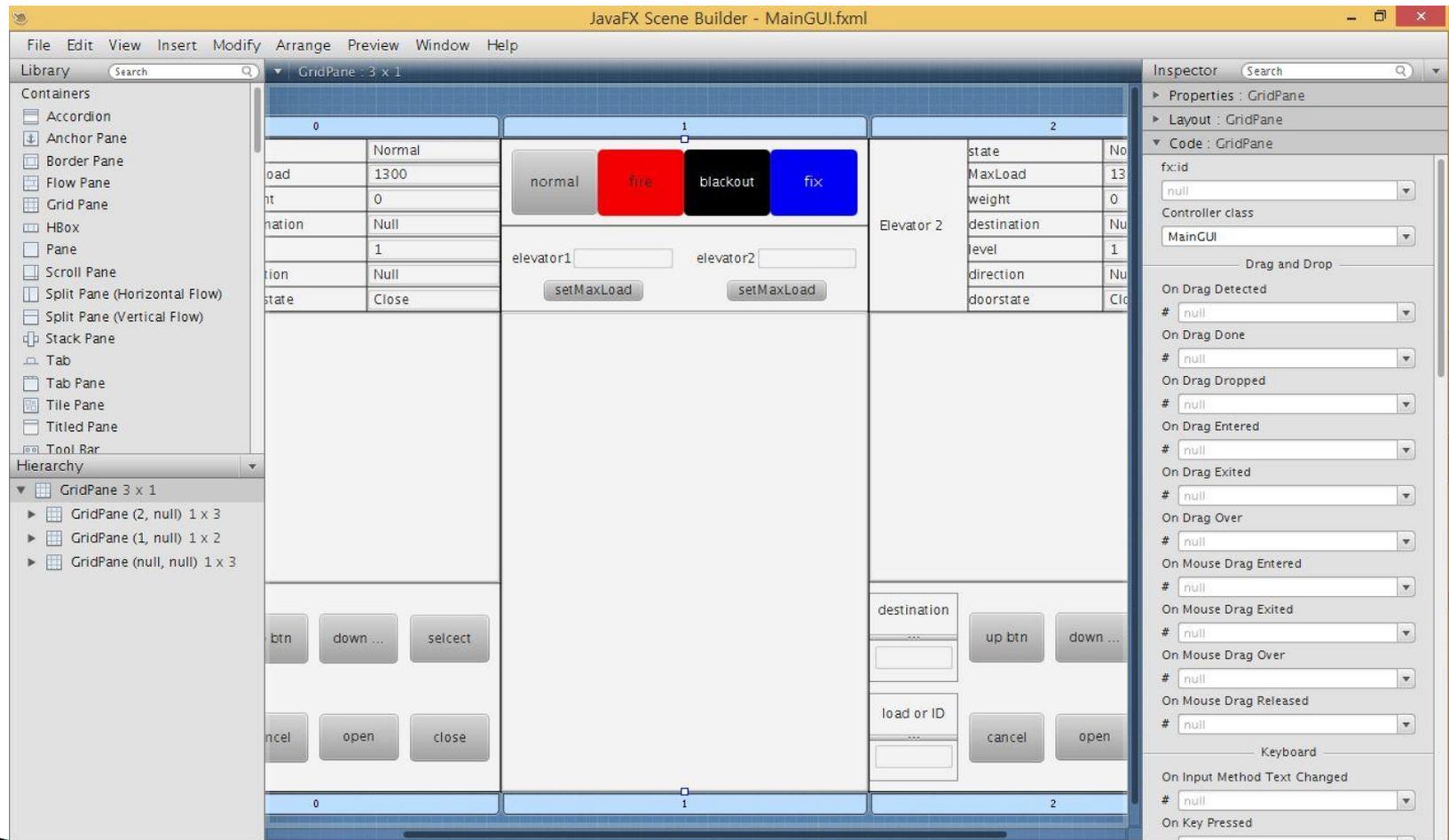


구현 중점 - 동기화

엘리베이터가
새로운 큐를
요구할 때

엘리베이터의
수행이 끝나고
수행한 큐를
제거 할 때

구현 중점 - GUI



DEMO

Aboard Request

The interface is divided into several sections:

- Elevator 1 Data:**

state	Normal
MaxLoad	1300
weight	250
destination	Null
level	7
direction	2
doorstate	Close
- Elevator 2 Data:**

state	Normal
MaxLoad	1300
weight	70
destination	Null
level	5
direction	2
doorstate	Close
- Control Buttons:** normal (grey), fire (red), blackout (black), fix (blue).
- Input Fields:** elevator1, elevator2, setMaxLoad (two buttons).
- Log:**

```
Elevator1's Door is closed  
Elevator1's destination is now8  
Elevator2's destination is now8  
Elevator1 moves 7floor  
Elevator2 moves 4floor  
Elevator1 moves 8floor  
Elevator2 moves 5floor  
Elevator1's Door is opened  
Now Elevator1's weight is 160  
Elevator1's Door is closed  
Elevator2 moves 6floor  
now Up request is added in Schedule  
Elevator1's destination is now10  
now Down request is added in Schedule  
Elevator1's destination is now10  
Elevator2's destination is now10  
Elevator1 moves 9floor  
Elevator1 moves 10floor  
Elevator2 moves 7floor  
Elevator1's Door is opened  
Now Elevator1's weight is 250  
Elevator1's Door is closed  
Elevator1's destination is now5  
Elevator2's destination is now5  
Elevator1 moves 9floor  
Elevator2 moves 6floor  
Elevator1 moves 8floor  
Elevator2 moves 5floor  
Elevator1 moves 7floor  
Elevator2's Door is opened  
Now Elevator2's weight is 70  
Elevator2's Door is closed
```
- Destination and Load Controls:**
 - Left: destination (10), up btn, down ..., select; load or ID (90), cancel, open, close.
 - Right: destination (5), up btn, down ..., select; load or ID (70), cancel, open, close.

Select Level

The interface is divided into several sections:

- Elevator 1 (Left):** A table showing its current state: state: Normal, MaxLoad: 1300, weight: 20, destination: Null, level: 5, direction: 1, doorstate: Close.
- Elevator 2 (Right):** A table showing its current state: state: Normal, MaxLoad: 1300, weight: 0, destination: Null, level: 1, direction: Null, doorstate: Close.
- Control Buttons (Top Center):** Four buttons: 'normal' (grey), 'fire' (red), 'blackout' (black), and 'fix' (blue).
- Input Fields (Middle Center):** Two text boxes labeled 'elevator1' and 'elevator2', each with a 'setMaxLoad' button below it.
- Log (Center):** A text area displaying the following sequence of events:

```
now Up request is added in Schedule
Elevator1's destination is now2
Elevator1 moves 2floor
Elevator1's Door is opened
Now Elevator1's weight is 50
Elevator1's Door is closed
Elevator1's Select Level request is added in Schedule
Elevator1's destination is now5
Elevator1's destination is now5
Elevator1 moves 3floor
Elevator1 moves 4floor
Elevator1 moves 5floor
Elevator1's Door is opened
Now Elevator1's weight is 20
Elevator1's Door is closed
```
- User Controls (Bottom):** Two panels for each elevator. The left panel has a 'destination' dropdown (value: 5), 'up btn', 'down ...', and 'select' buttons. The right panel has a 'load or ID' dropdown (value: 30), 'cancel', 'open', and 'close' buttons.

Cancel Request

The interface displays the following data:

Elevator	state	MaxLoad	weight	destination	level	direction	doorstate
Elevator 1	Normal	1300	10	Null	9	1	Close
Elevator 2	Normal	1300	0	Null	8	1	Close

Control buttons: normal, fire (red), blackout, fix. Input fields for elevator1 and elevator2, and setMaxLoad buttons.

Log text:
now Up request is added in Schedule
Elevator1's destination is now12
now Up request is added in Schedule
Elevator1's destination is now9
Elevator2's destination is now12
Elevator1 moves 2floor
Elevator1 moves 3floor
Elevator2 moves 2floor
Elevator1 moves 4floor
Elevators up 12 level request is canceled
Elevator1's destination is now9
Elevator2's destination is now9
Elevator2 moves 3floor
Elevator1 moves 5floor
Elevator1 moves 6floor
Elevator2 moves 4floor
Elevator2 moves 5floor
Elevator1 moves 7floor
Elevator1 moves 8floor
Elevator2 moves 6floor
Elevator2 moves 7floor
Elevator1 moves 9floor
Elevator1's Door is opened
Now Elevator1's weight is 10
Elevator1's Door is closed
Elevator2 moves 8floor

Destination and load controls for Elevator 1 (9, 10) and Elevator 2 (12, 1).

Fire Control

Elevator 1	state	Fire	<input type="button" value="normal"/> <input type="button" value="fire"/> <input type="button" value="blackout"/> <input type="button" value="fix"/>	Elevator 2	state	Fire
	MaxLoad	1300			MaxLoad	1300
	weight	0			weight	0
	destination	Null			destination	Null
	level	2			level	2
	direction	2			direction	2
	doorstate	Close			doorstate	Open
					<input type="text" value="elevator1"/> <input type="text" value="elevator2"/> <input type="button" value="setMaxLoad"/> <input type="button" value="setMaxLoad"/>	
<pre> now Up request is added in Schedule Elevator1's destination is now2 Elevator1 moves 4floor now Up request is added in Schedule Elevator1's destination is now2 Elevator2's destination is now2 Elevator1 moves 3floor Elevator1 moves 2floor Elevator2 moves 2floor Elevator1's Door is opened Now Elevator1's weight is 50 Elevator1's Door is closed Elevator1's destination is now3 Elevator2's destination is now3 Elevator1 moves 3floor Elevator2 moves 3floor Elevator1's Door is opened Elevator2's Door is opened Now Elevator1's weight is 80 Elevator1's Door is closed Now Elevator2's weight is 0 Now Elevators' state is fire Elevator1's destination is now2 Elevator2's destination is now2 Elevator1 moves 2floor Elevator2 moves 2floor Elevator1's Door is opened Now Elevator1's weight is 0 Elevator2's Door is openedElevator1's Door is closed Now Elevator2's weight is 0 Elevator2's destination is now2 Elevator2's Door is closed </pre>			<pre> requestID : 3 level : 2 load : 0 </pre>			
destination <input type="text" value="3"/>	<input type="button" value="up btn"/> <input type="button" value="down ..."/> <input type="button" value="select"/>	destination <input type="text"/>	<input type="button" value="up btn"/> <input type="button" value="down ..."/> <input type="button" value="select"/>			
load or ID <input type="text" value="30"/>	<input type="button" value="cancel"/> <input type="button" value="open"/> <input type="button" value="close"/>	load or ID <input type="text"/>	<input type="button" value="cancel"/> <input type="button" value="open"/> <input type="button" value="close"/>			

Blackout Control

The interface displays two elevator control panels, Elevator 1 and Elevator 2, with a central log and control buttons.

Elevator 1 Status:

state	Normal
MaxLoad	1300
weight	0
destination	Null
level	1
direction	2
doorstate	Close

Elevator 2 Status:

state	Normal
MaxLoad	1300
weight	0
destination	Null
level	1
direction	2
doorstate	Open

Control Buttons: normal (grey), fire (red), blackout (black), fix (blue). Below these are input fields for 'elevator1' and 'elevator2', and 'setMaxLoad' buttons.

Log:

```

Elevator 1 moves 3floor
Elevator2 moves 3floor
Elevator1's Door is opened
Elevator2's Door is opened
Now Elevator1's weight is 80
Elevator1's Door is closed
Now Elevator2's weight is 0
Now Elevators' state is fire
Elevator1's destination is now2
Elevator2's destination is now2
Elevator1 moves 2floor
Elevator2 moves 2floor
Elevator1's Door is opened
Now Elevator1's weight is 0
Elevator2's Door is openedElevator1's Door is closed

Now Elevator2's weight is 0
Elevator2's destination is now2
Elevator2's Door is closed
Now Elevators' state is normal
Now Elevators' state is blackout
Elevator1's destination is now1
Elevator2's destination is now1
Elevator1 moves 1floor
Elevator2 moves 1floor
Elevator1's Door is opened
Elevator2's Door is opened
Now Elevator1's weight is 0
Elevator1's Door is closed
Now Elevator2's weight is 0
Elevator2's Door is closed
Now Elevators' state is normal
    
```

Control Panels:

Elevator 1: destination (dropdown: 3), up btn, down ..., select; load or ID (input: 30), cancel, open, close.

Elevator 2: destination (dropdown), up btn, down ..., select; load or ID (input), cancel, open, close.

Fix State

The screenshot displays a software interface for controlling two elevators. The interface is divided into several sections:

- Elevator 1 Panel (Top Left):** A table showing the current state of Elevator 1:

state	fix
MaxLoad	1300
weight	0
destination	Null
level	1
direction	Null
doorstate	Close
- Elevator 2 Panel (Top Right):** A table showing the current state of Elevator 2:

state	fix
MaxLoad	1300
weight	0
destination	Null
level	1
direction	Null
doorstate	Close
- Control Buttons (Top Center):** Four buttons labeled 'normal' (grey), 'fire' (red), 'blackout' (black), and 'fix' (blue).
- Input Fields (Middle Center):** Two text input fields labeled 'elevator1' and 'elevator2', each with a 'setMaxLoad' button below it.
- Status Area (Middle):** A text area displaying the message: "Elevators' state is fix. Elevators' state is not normal!".
- Bottom Control Panel (Bottom):** Two identical panels for each elevator. Each panel includes:
 - A 'destination' input field with a dropdown arrow and a value of '5'.
 - Buttons: 'up btn', 'down ...', and 'select'.
 - A 'load or ID' input field with a dropdown arrow and a value of '60'.
 - Buttons: 'cancel', 'open', and 'close'.

SetMaxLoad

<table border="1"> <tr><td>state</td><td>Normal</td></tr> <tr><td>MaxLoad</td><td>300</td></tr> <tr><td>weight</td><td>110</td></tr> <tr><td>destination</td><td>Null</td></tr> <tr><td>level</td><td>6</td></tr> <tr><td>direction</td><td>1</td></tr> <tr><td>doorstate</td><td>Close</td></tr> </table>	state	Normal	MaxLoad	300	weight	110	destination	Null	level	6	direction	1	doorstate	Close	<div style="display: flex; justify-content: space-around;"> normal fire blackout fix </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <input type="text" value="elevator1 300"/> <input type="text" value="elevator2"/> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> setMaxLoad setMaxLoad </div>	<table border="1"> <tr><td>state</td><td>Normal</td></tr> <tr><td>MaxLoad</td><td>1300</td></tr> <tr><td>weight</td><td>0</td></tr> <tr><td>destination</td><td>Null</td></tr> <tr><td>level</td><td>5</td></tr> <tr><td>direction</td><td>1</td></tr> <tr><td>doorstate</td><td>Close</td></tr> </table>	state	Normal	MaxLoad	1300	weight	0	destination	Null	level	5	direction	1	doorstate	Close
state	Normal																													
MaxLoad	300																													
weight	110																													
destination	Null																													
level	6																													
direction	1																													
doorstate	Close																													
state	Normal																													
MaxLoad	1300																													
weight	0																													
destination	Null																													
level	5																													
direction	1																													
doorstate	Close																													
<pre> now Up request is added in Schedule Elevator1's destination is now6 now Up request is added in Schedule Elevator1's destination is now5 Elevator2's destination is now5 Elevator1 moves 2floor Elevator1 moves 3floor Elevator2 moves 2floor Elevator1 moves 4floor Elevator1 moves 5floor Elevator2 moves 3floor Elevator1's Door is opened Now Elevator1's weight is 60 Elevator1's Door is closed Elevator1's destination is now6 Elevator2's destination is now6 Elevator1 moves 6floor Elevator2 moves 4floor Elevator1's Door is opened Elevator2 moves 5floor Now Elevator1's weight is 110 Elevator1's Door is closed You cannot change maxload over currentLoad Now Elevator1's maxLoad is 300 </pre>																														
<table border="1"> <tr><td>destination</td><td>...</td></tr> <tr><td>5</td><td>up btn</td><td>down ...</td><td>select</td></tr> <tr><td>load or ID</td><td>...</td></tr> <tr><td>60</td><td>cancel</td><td>open</td><td>close</td></tr> </table>	destination	...	5	up btn	down ...	select	load or ID	...	60	cancel	open	close		<table border="1"> <tr><td>destination</td><td>...</td></tr> <tr><td>6</td><td>up btn</td><td>down ...</td><td>select</td></tr> <tr><td>load or ID</td><td>...</td></tr> <tr><td>50</td><td>cancel</td><td>open</td><td>close</td></tr> </table>	destination	...	6	up btn	down ...	select	load or ID	...	50	cancel	open	close				
destination	...																													
5	up btn	down ...	select																											
load or ID	...																													
60	cancel	open	close																											
destination	...																													
6	up btn	down ...	select																											
load or ID	...																													
50	cancel	open	close																											

Unit Test – Pass

Test Case Name	1. testEnQueue
Objective	-QueueAlgorithm의 enqueue기능을 시행하였을때 실제로 요청이 저장되는지 확인 -Queue 내의 searchForNode 기능 작동 확인 -Queue 내의 findClosestNodeFromPosition 기능 작동 확인 1) queueID 1, requestID 1, level 1, load 10 을 넣어 enqueue 2) 앞에 넣은 큐의 selectForNode에 같은 enqueue에 넣어 준 값을 넣어 그중 load 값이 처음에 넣은 10과 같은지 확인 3) findClosestNodeFromPosition에서 엘리베이터의 현재 위치를 enqueue에 넣은 요청에서의 목적지와 동일하다고 가정하게 parameter를 주고 역시 리턴된 요청의 load 값 비교
Estimated Result	2) 의 결과 10, 3)의 결과 10으로 assertEquals 통과
Actual Result	PASS

Test Case Name	2. testMakeNode
Objective	-Queue에서의 MakeNode 작동 확인 -makeNode기능을 세번 호출하여 총 세개의 요청을 Queue 에 넣은 뒤 Queue의 size가 3과 동일한지 확인 -앞에서 넣은 요청 중 level 5, requestID 3으로 넣은 요청에 대해 searchForNode를 호출하여 결과 일치하는지 확인
Estimated Result	Queue size 3, 검색한 결과값의 level 5, requestID 3
Actual Result	PASS

Test Case Name	3. testSearchNode
Objective	-앞의 테스트에서 이용되긴 하였으나 searchForNode 메소드 정상 작동 확인 -먼저 존재하지 않는 Node인 level 5, requestID 4인 Node 에 대한 검색 시행, null과 같은지 비교 -다음으로 존재하는 Node인 level 5, requestId 3 인 Node 에 대한 검색 시행, 결과노드의 requestId 3 값 맞는지 확인
Estimated Result	assertNotEquals 통과, requestID 비교값 3
Actual Result	PASS

Test Case Name	4. testfindClosestNodeFromPos
Objective	-Queue에서 현재엘레베이터의 위치와 방향을 넣었을 때 가장 가까운 요청 검색하는 기능 -엘리베이터가 3층에서 올라간다고 가정하고 5층으로 내리는 요청과 4층에서 올라가는 요청 정의 -가장 가까운 findClosestNodeFromPos 호출하고 가장 가까운 요청에 대한 목적지 비교
Estimated Result	가장 가까운 목적지 층 결과값 4
Actual Result	PASS

Test Case Name	5. testOpenNClose
Objective	-Elevator 클래스 내 문을 여는 메소드 OpenDoor와 CloseDoor 두 메소드 작동 확인 -문을 열고 문의 현재상태 비교 -문을 닫고 문의 현재상태 비교
Estimated Result	문을 열었을 때 문의상태 1, 문을 닫았을 때 0
Actual Result	PASS

Test Case Name	6. testClearResult
Objective	-Queue 클래스 내에 Queue 내용을 모두 비우는 메소드 ClearNode 메소드 작동 확인 -ClearNode 호출 후 Queue가 비었는지 확인
Estimated Result	isEmpty() true 리턴
Actual Result	PASS

Test Case Name	7. testCompClosestNode
Objective	-엘리베이터의 현재위치에 대해 가장 가까운 요청 검색 되는지 확인 -현재 엘리베이터의 위치를 1으로 놓고 5층 상승 요청, 4층 내림 요청 두개 생성 -두 요청에 대해 더 가까운 요청 판별 후 그 요청의 목적지 확인
Estimated Result	결과로 된 요청의 목적지는 4가 되어 할 것이다.
Actual Result	PASS

Test Case Name	8. testArrivalCalibration
Objective	-Elevator 내의 arrivalCalibration 적용 되는지 판별
Estimated Result	
Actual Result	PASS

Unit Test – Error

Test Case Name	9. testHandleAboard
Objective	-QueueAlgorithm의 탑승요청 handleAboard() 메소드 기능 확인 -11층에서 상승요청을 의미하는 요청 전달 -탑승요청 큐에 해당 요청 존재하는지 확인
Estimated Result	해당 요청 존재
Actual Result	ERROR

Test Case Name	10. testEmergencyDestination
Objective	-비상 프로토콜을 위한 자동 비상 목적지 설정, 정차요청 기능 작동 확인
Estimated Result	엘리베이터의 목적지 변경
Actual Result	ERROR

Test Case Name	11. testFire
Objective	-화재기능
Estimated Result	큐내 모든요청 삭제됨
Actual Result	ERROR

Test Case Name	12. testBlack
Objective	-정전기능
Estimated Result	큐내 모든요청 삭제됨
Actual Result	ERROR

Test Case Name	13. testHandleSelect
Objective	-QueueAlgorithm의 하차요청 handleSelect() 메소드 기능 확인 -6층에서 정차를 하도록 하는 요청 전달 -탑승요청 큐에 해당 요청 존재하는지 확인
Estimated Result	해당 요청 존재
Actual Result	ERROR

Test Case Name	14. testHandleCancel
Objective	-존재하는 요청 삭제하는 기능인 HandleCancel() 메소드 작동 확인 -한 개의 요청을 넣은 뒤 취소요청 (HandleCancel) 적용 후 해당요청 존재 여부 확인
Estimated Result	해당 요청 없음
Actual Result	ERROR

Test Case Name	15. testSetNextDest
Objective	-setNextDestinationByComparison() 메소드 작동 확인 - 엘리베이터 1에대한 메소드 호출 후 엘리베이터의 목적지 확인
Estimated Result	목적지 변화
Actual Result	ERROR

Test Case Name	16. testAdd
Objective	-목적지에 도착하여 arrivecalibration 중 해당 하중 내리기/ 태우기 기능 작동 확인
Estimated Result	
Actual Result	ERROR

Test Case Name	17. testDepart
Objective	-엘리베이터 출발 기능 작동 확인
Estimated Result	
Actual Result	ERROR

Q&A