

# No Wait ECS Verification

## System Test 2<sup>nd</sup> Cycle

---

Team 2

200911388 박미관

200911412 이영준

200910793 임민우

# Index

---

- **Category-Partition Testing**
- **Pairwise Combination Testing**
- **Brute Force Testing**
- **Static Analysis**
  - **Find Bugs**
  - **PMD**
  - **Jdepend**
  - **Check Style**
  - **Jfeature**
  - **Clover**

---

# **Category-Partition Testing**

---

# Category-Partition Testing (1/12)

---

- Identify Independently Testable Units (13)

Group	Category
Simulator	내부 층 버튼 선택
	엘리베이터 호출
	비상구조 요청
	승차
	하차
Smart Phone	대기시간 표현 요청
	버튼 조작

# Category-Partition Testing (2/12)

---

- Identify Independently Testable Units (13)

Group	Category
Display	무게 초과 감지
	대기 시간 표시
	시뮬레이터 Status
	캐빈 내 인원
	이동 방향 표시
	층별 디스플레이

# Category-Partition Testing (3/12)

---

## □ Representative Values

Category	Values	#
내부 층 버튼 선택	같은 층을 선택	A01
	높은 층을 선택	A02
	낮은 층을 선택	A03
	선택하지 않는다	A04
엘리베이터 호출	UP	A05
	DOWN	A06
비상구조 요청	요청한다	A07
	요청하지 않는다	A08
승차	Invalid	A09
	0	A10
	1~20	A11
	21~	A12
하차	Invalid	A13
	0	A14
	1~20	A15
	21~	A16

# Category-Partition Testing (4/12)

---

## □ Representative Values

Category	Values	#
대기시간 표현 요청	요청한다	B01
	요청하지 않는다	B02
버튼 조작	요청한다	B03
무게 초과 감지	만원 표시	C01
	표시하지 않는다	C02
대기 시간 표시	Invalid	C03
	표시	C04
	표시하지 않는다	C05
시뮬레이터 Status	내부 선택된 층 표시	C06
	현재 엘리베이터 위치 표시	C07
캐빈 내 인원	0미만	C08
	0~20	C09
	20 초과	C10

# Category-Partition Testing (5/12)

---

## □ Representative Values

Category	Values	#
이동 방향 표시	위 아래 정지	C11 C12 C13
층별 디스플레이	호출버튼 활성화 캐빈 표시	C14 C15

※  $4 * 2 * 2 * 4 * 4 * 2 * 1 * 2 * 3 * 2 * 3 * 3 * 2 = 110,592$  Cases



# Category-Partition Testing (6/12)

---

## □ Error Constraints

Category	Values	Error
승차	Invalid 0 1~20 21~	Error
하차	Invalid 0 1~20 21~	Error
대기 시간 표시	Invalid 표시 표시하지 않는다	Error
캐빈 내 인원	0미만 0~20 20 초과	Error Error

※ 110,592 Cases -> 13,829 Cases

# Category-Partition Testing (7/12)

---

## □ Single Constraints

Category	Values	Single
승차	0	Single
	1~20	
	21~	Single
하차	0	Single
	1~20	
	21~	Single

※ 13,829 Cases -> 1,541 Cases

# Category-Partition Testing (8/12)

## □ Property Constraints

Category	Values	Property	If
내부 층 버튼 선택	같은 층을 선택 높은 층을 선택 낮은 층을 선택 선택하지 않는다	Sel_S Sel_U Sel_D Cancel	In In In In
엘리베이터 호출	UP DOWN	Up Down	
비상구조 요청	요청한다 요청하지 않는다	E_Call	In In
승차	1~20	In	Up, Down, SP_Call
하차	1~20	Out	Cancel, Sel_S, Sel_U, Sel_D, E_Call

# Category-Partition Testing (9/12)

---

## □ Property Constraints

Category	Values	Property	If
대기시간 표현 요청	요청한다 요청하지 않는다	W_Call W_N_Call	SP_Call SP_Call
버튼 조작	요청한다	SP_Call	
무게 초과 감지	만원 표시 표시하지 않는다		In In
대기 시간 표시	표시 표시하지 않는다		W_Call W_N_Call
시뮬레이터 Status	내부 선택된 층 표시 현재 엘리베이터 위치 표시		Sel_U, Sel_D
캐빈 내 인원	0~20		

# Category-Partition Testing (10/12)

---

## □ Property Constraints

Category	Values	Property	If
이동 방향 표시	위 아래 정지	Stop	Sel_U Sel_D Out
층별 디스플레이	호출버튼 활성화 캐빈 표시		Out Stop

※ 1,541 Cases -> 46 Cases

# Category-Partition Testing (11/12)

□ 46 Cases -> 7 Pass / 39 Fail

#	Test Case	설명	결과
1	A05.A11.A01.A08.A15	UP 버튼을 누르고 N명을 승차한 뒤 같은 층을 선택하고 N명을 하차한다.	Fail
2	A05.A11.A02.A08.A15	UP 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하고 N명을 하차한다.	Fail
3	A05.A11.A03.A08.A15	UP 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하고 N명을 하차한다.	Fail
4	A05.A11.A04.A08.A15	UP 버튼을 누르고 N명을 승차한 뒤 층 선택을 하지 않고 N명을 하차한다.	Fail
5	A05.A11.A01.A07.A15	UP 버튼을 누르고 N명을 승차한 뒤 같은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
6	A05.A11.A02.A07.A15	UP 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
7	A05.A11.A03.A07.A15	UP 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
8	A05.A11.A04.A07.A15	UP 버튼을 누르고 N명을 승차한 뒤 층 선택을 하지 않고 비상구조 요청한 뒤 하차한다.	Fail
9	A05.A11.C01	UP 버튼을 누르고 N명을 승차한 뒤 인원 초과 되면 만원 표시를 한다.	Fail
10	A05.A11.C02	UP 버튼을 누르고 N명을 승차한 뒤 인원 초과 되지 않으면 만원 표시를 하지 않는다.	Fail
11	A05.A11.A02.C06	UP 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
12	A05.A11.A03.C06	UP 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
13	A06.A11.A01.A08.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 같은 층을 선택하고 N명을 하차한다.	Fail
14	A06.A11.A02.A08.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하고 N명을 하차한다.	Fail
15	A06.A11.A03.A08.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하고 N명을 하차한다.	Fail
16	A06.A11.A04.A08.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 층 선택을 하지 않고 N명을 하차한다.	Fail
17	A06.A11.A01.A07.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 같은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
18	A06.A11.A02.A07.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
19	A06.A11.A03.A07.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
20	A06.A11.A04.A07.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 층 선택을 하지 않고 비상구조 요청한 뒤 하차한다.	Fail
21	A06.A11.C01	DOWN 버튼을 누르고 N명을 승차한 뒤 인원 초과 되면 만원 표시를 한다.	Fail
22	A06.A11.C02	DOWN 버튼을 누르고 N명을 승차한 뒤 인원 초과 되지 않으면 만원 표시를 하지 않는다.	Fail
23	A06.A11.A02.C06	DOWN 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
24	A06.A11.A03.C06	DOWN 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
25	B03.A11.A01.A08.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 같은 층을 선택하고 N명을 하차한다.	Fail
26	B03.A11.A02.A08.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 높은 층을 선택하고 N명을 하차한다.	Fail
27	B03.A11.A03.A08.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 낮은 층을 선택하고 N명을 하차한다.	Fail
28	B03.A11.A04.A08.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 층 선택을 하지 않고 N명을 하차한다.	Fail
29	B03.A11.A01.A07.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 같은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
30	B03.A11.A02.A07.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 높은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
31	B03.A11.A03.A07.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 낮은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
32	B03.A11.A04.A07.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 층 선택을 하지 않고 비상구조 요청한 뒤 하차한다.	Fail
33	B03.A11.C01	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 인원 초과 되면 만원 표시를 한다.	Fail
34	B03.A11.C02	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 인원 초과 되지 않으면 만원 표시를 하지 않는다.	Fail

# Category-Partition Testing (12/12)

---

#	Test Case	설명	결과
35	B03.A11.A02.C06	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 높은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
36	B03.A11.A03.C06	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 낮은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
37	B01.C04	스마트폰에서 대기 시간 표현 요청을 하면 대기 시간이 표시 된다.	Fail
38	B02.C05	스마트폰에서 대기 시간 표현 요청을 하지 않으면 대기 시간이 표시되지 않는다.	Fail
39	C07.*	현재 엘리베이터 위치가 표시된다.	Pass
40	C09.*	캐빈 내 인원이 표시된다.	Pass
41	A05.A11.A01.A08.A15.	UP 버튼을 누르고 N명을 승차한 뒤 같은 층을 선택하고 N명을 하차하면 이동 방향에 정지가 표시된다.	Fail
42	A02.C11	높은 층을 선택하면 이동 방향이 위가 된다.	Pass
43	A03.C12	낮은 층을 선택하면 이동 방향이 아래가 된다.	Pass
44	A15.C13	하차를 하면 이동 방향이 정지가 된다.	Pass
45	C13.C15	정지를 하면 캐빈이 표시 된다.	Pass
46	A15.C14	하차를 하면 호출 버튼이 활성화된다.	Pass

---

---

# **Pairwise Combination Testing**

---



# Pairwise Combination Testing (1/2)

---

## □ Pairewise Combination (1/2)

Category	Values	#
내부 층 버튼 선택	같은 층을 선택	A01
	높은 층을 선택	A02
	낮은 층을 선택	A03
	선택하지 않는다	A04
엘리베이터 호출	UP	A05
	DOWN	A06
버튼 조작	요청한다	B03

- Up -> 높은 층을 선택
- Down -> 낮은 층을 선택
- 버튼 조작 요청 -> 높은 층을 선택

# Pairwise Combination Testing (2/2)

□ 21 Cases -> 7 Pass / 14 Fail

#	Test Case	설명	결과
1	A05.A11.A02.A08.A15	UP 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하고 N명을 하차한다.	Fail
2	A05.A11.A02.A07.A15	UP 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
3	A05.A11.C01	UP 버튼을 누르고 N명을 승차한 뒤 인원 초과 되면 만원 표시를 한다.	Fail
4	A05.A11.C02	UP 버튼을 누르고 N명을 승차한 뒤 인원 초과 되지 않으면 만원 표시를 하지 않는다.	Fail
5	A05.A11.A02.C06	UP 버튼을 누르고 N명을 승차한 뒤 높은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
6	A06.A11.A03.A08.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하고 N명을 하차한다.	Fail
7	A06.A11.A03.A07.A15	DOWN 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
8	A06.A11.A03.C06	DOWN 버튼을 누르고 N명을 승차한 뒤 낮은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
9	B03.A11.A02.A08.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 높은 층을 선택하고 N명을 하차한다.	Fail
10	B03.A11.A02.A07.A15	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 높은 층을 선택하고 비상구조 요청한 뒤 하차한다.	Fail
11	B03.A11.A02.C06	스마트폰 버튼 조작을 하고 N명을 승차한 뒤 높은 층을 선택하면 내부 선택된 층이 표시된다.	Fail
12	B01.C04	스마트폰에서 대기 시간 표현 요청을 하면 대기 시간이 표시 된다.	Fail
13	B02.C05	스마트폰에서 대기 시간 표현 요청을 하지 않으면 대기 시간이 표시되지 않는다.	Fail
14	C07.*	현재 엘리베이터 위치가 표시된다.	Pass
15	C09.*	캐빈 내 인원이 표시된다.	Pass
16	A05.A11.A01.A08.A15.	UP 버튼을 누르고 N명을 승차한 뒤 같은 층을 선택하고 N명을 하차하면 이동 방향에 정지가 표시된다.	Fail
17	A02.C11	높은 층을 선택하면 이동 방향이 위가 된다.	Pass
18	A03.C12	낮은 층을 선택하면 이동 방향이 아래가 된다.	Pass
19	A15.C13	하차를 하면 이동 방향이 정지가 된다.	Pass
20	C13.C15	정지를 하면 캐빈이 표시 된다.	Pass
21	A15.C14	하차를 하면 호출 버튼이 활성화된다.	Pass

---

# **Brute Force Testing**

---

# Brute Force Testing (1/7)

□ 1<sup>st</sup> : 6 Cases -> 0 Pass / 6 Fail

#	Test Case	설명	결과
1	BruteForce #1	스마트 폰으로 호출시 2호기가 정지 상태이면 1호기의 위치에 상관없이 2호기가 1층으로 온다.	Fail
2	BruteForce #2	스마트 폰으로 호출시 1호기가 1층에 있어도 요청 실패 하지 않는다.	Fail
3	BruteForce #3	스마트 폰으로 호출시 1호기가 호출되어도 2호기가 호출되었다고 표시된다.	Fail
4	BruteForce #4	두 엘리베이터 모두가 이동 중에 다른 층에서 호출을 하여 층 선택 시 엘리베이터만 움직이고 사람 수는 변화가 없다.	Fail
5	BruteForce #5	1층에서 UP버튼을 누르고 높은 층 선택을 여러 번 반복하면 명령이 모두 실행되지 않는다.	Fail
6	BruteForce #6	Smart Phone UI에서 cabin1 Floor라는 항목이 중복되어 있다	Fail

□ 2<sup>nd</sup> : 7(6-2+3) Cases -> 1 Pass / 6 Fail

#	Test Case	설명	결과
1	BruteForce #1	스마트 폰으로 호출 기능이 동작하지 않는다.	Fail
2	BruteForce #2	두 엘리베이터 모두가 이동 중에 다른 층에서 호출을 하여 층 선택 시 엘리베이터만 움직이고 사람 수는 변화가 없다.	Fail
3	BruteForce #3	1층에서 UP버튼을 누르고 높은 층 선택을 여러 번 반복하면 명령이 모두 실행되지 않는다.	Fail
4	BruteForce #4	Smart Phone UI에서 cabin1 Floor라는 항목이 중복되어 있다.	Pass
5	BruteForce #5	비상구조 요청시 사람이 늘어나는 경우가 발생한다.	Fail
6	BruteForce #6	비상구조 요청시 사람이 타지 않고 엘리베이터만 움직이는 경우가 발생한다.	Fail
7	BruteForce #7	층 선택한 뒤 바로 비상구조 요청시 되지 않는 경우가 있다.	Fail

# Brute Force Testing (2/7)

## □ Case #01

- 스마트폰 호출 기능이 동작하지 않는다.

The screenshot shows a simulation window titled "No Wait Elevator". It displays a 10-story building with elevator controls for each floor. The controls include buttons for "UP" and "DOWN", and a "정지" (Stop) button. The status of each floor is shown as "사람수" (Number of people) and "대기" (Waiting). A log on the right shows the elevator's current floor, direction, and passenger count. At the bottom, there are buttons for "1호기 SOS" and "2호기 SOS".

층	UP	DOWN	정지	사람수	대기
10층		DOWN		0	0
9층	UP	DOWN		0	0
8층	UP	DOWN		0	0
7층	UP	DOWN		0	0
6층	UP	DOWN		0	0
5층	UP	DOWN		0	0
4층	UP	DOWN		0	0
3층	UP	DOWN	정지	1	0
2층	UP	DOWN	정지	1	0
1층	UP			0	0

Log:

```

1호기가 1층에서 2층으로 이동 합니다.
2호기가 1층에서 3층으로 이동 합니다.
스마트폰 요청
스마트폰 요청
스마트폰 요청
스마트폰 요청
스마트폰 요청
    
```

현재 층	1호기	2호기
현재 층	2	3
방향	정지	정지
탑승객 수	0	0
무게 초과	정상	정상
SOS	1호기 SOS	2호기 SOS



# Brute Force Testing (4/7)

## □ Case #03

- 1층에서 UP버튼을 누르고 높은 층 선택을 여러 번 반복하면, 명령이 모두 실행되지 않는다.

The screenshot shows the 'No Wait Elevator' simulation interface. A red circle highlights a text box containing a list of elevator movements:

- 1호기가 1층에서 10층으로 이동 합니다.
- 2호기가 1층에서 9층으로 이동 합니다.
- 1호기가 1층에서 8층으로 이동 합니다.
- 2호기가 1층에서 7층으로 이동 합니다.
- 2호기가 1층에서 6층으로 이동 합니다.
- 2호기가 1층에서 5층으로 이동 합니다.
- 2호기가 1층에서 4층으로 이동 합니다.
- 2호기가 1층에서 3층으로 이동 합니다.
- 2호기가 1층에서 2층으로 이동 합니다.

The interface also shows a floor selection grid, passenger counts, and status indicators for two elevators.

층	1호기	2호기
10층		
9층		
8층	정지	
7층	정지	
6층		
5층		
4층	현재 층	7
3층	방향	정지
2층	탑승객 수	2
1층	무게 초과	정상

Buttons: 1호기 SOS, 2호기 SOS

# Brute Force Testing (5/7)

## Case #05

- 비상 구조 요청시, 사람이 늘어나는 경우가 발생한다.

No Wait Elevator

층	UP	DOWN	정지	사람수	대기
10층		DOWN		0	0
9층	UP	DOWN		0	0
8층	UP	DOWN	정지	1	0
7층	UP	DOWN		1	0
6층	UP	DOWN		0	0
5층	UP	DOWN		0	0
4층	UP	DOWN		0	0
3층	UP	DOWN		0	0
2층	UP	DOWN		0	0
1층	UP		정지	0	0

1호기가 1층에서 2층으로 이동 합니다.  
 1호기가 2층에서 8층으로 이동 합니다.  
 1호기 SOS요청  
 1호기 비상 정지  
 6층에서 비상 정지  
 6층에 1명 하차  
 1호기 비상 해제  
 1호기가 6층에서 7층으로 이동 합니다.  
 1호기가 7층에서 8층으로 이동 합니다.  
 1호기 SOS요청  
 1호기 정지 중입니다.

	1호기	2호기
현재 층	8	1
방향	정지	정지
탑승객 수	-1	0
무게 초과	정상	정상
SOS	<input type="button" value="1호기 SOS"/>	<input type="button" value="2호기 SOS"/>



# Brute Force Testing (6/7)

## □ Case #06

- 비상 구조 요청시, 사람이 타지 않고 엘리베이터만 움직이는 경우가 발생한다.

The screenshot shows the 'No Wait Elevator' simulation interface. It features a table of floor status and a log window. The floor status table is as follows:

Floor	UP	DOWN	Stop	Person Count	Waiting
10th		DOWN		0	0
9th	UP	DOWN		0	0
8th	UP	DOWN		0	0
7th	UP	DOWN		1	0
6th	UP	DOWN	정지	0	0
5th	UP	DOWN		0	0
4th	UP	DOWN		0	0
3th	UP	DOWN		0	0
2th	UP	DOWN		0	0
1th	UP		정지	0	0

The log window shows the following entries:

- 1호기 SOS요청
- 1호기 정지 종입니다.
- 1호기 SOS요청
- 1호기 정지 종입니다.
- 1호기가 7층에서 5층으로 이동 합니다. (highlighted in red)
- 1호기 SOS요청
- 1호기 정지 종입니다.
- 1호기 SOS요청
- 1호기 비상 정지
- 6층에서 비상 정지 |
- 6층에 0명 하차
- 1호기 비상 해제
- 1호기 SOS요청
- 1호기 비상 정지

Summary statistics at the bottom right:

	1호기	2호기
현재 층	6	1
방향	정지	정지
탑승객 수	0	0
무게 초과	정상	정상

Buttons for '1호기 SOS' and '2호기 SOS' are located at the bottom right.

# Brute Force Testing (7/7)

## □ Case #07

- 층 선택한 뒤 바로 비상 구조 요청이 되지 않는 경우가 있다.

Floor	UP	DOWN	정지	사람수	대기	이벤트
10층.		DOWN		0	0	스마트폰 요청
9층.	UP	DOWN		0	0	1호기가 1층에서 7층으로 이동 합니다.
8층.	UP	DOWN		0	0	1호기 SOS요청
7층.	UP	DOWN	정지	1	0	1호기 비상 정지
6층.	UP	DOWN		0	0	2층에서 비상 정지
5층.	UP	DOWN		0	0	2층에 1명 하차
4층.	UP	DOWN		0	0	1호기 비상 해제
3층.	UP	DOWN		0	0	2호기가 1층에서 7층으로 이동 합니다.
2층.	UP	DOWN	정지	1	0	2호기 SOS요청
1층.	UP			0	0	2호기 정지 중입니다.

현재 층	1호기	2호기
2	2	7
방향	정지	정지
탑승객 수	0	0
무게 초과	정상	정상

SOS

1호기 SOS      2호기 SOS

---

# Static Analysis

---

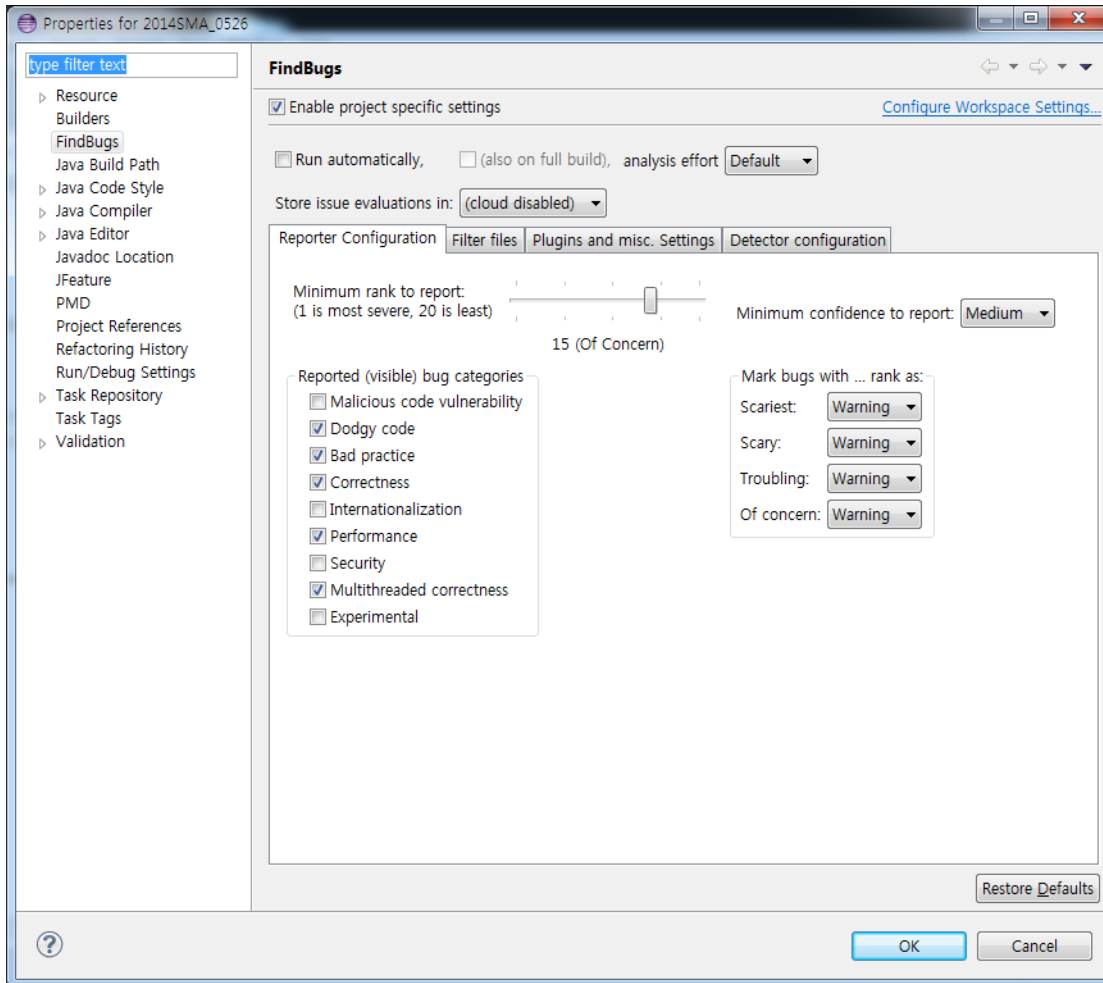
- Find bugs
- PMD
- Jdepend
- Check Style
- Jfeature
- Clover

# 1. Find Bugs (1/7)

---

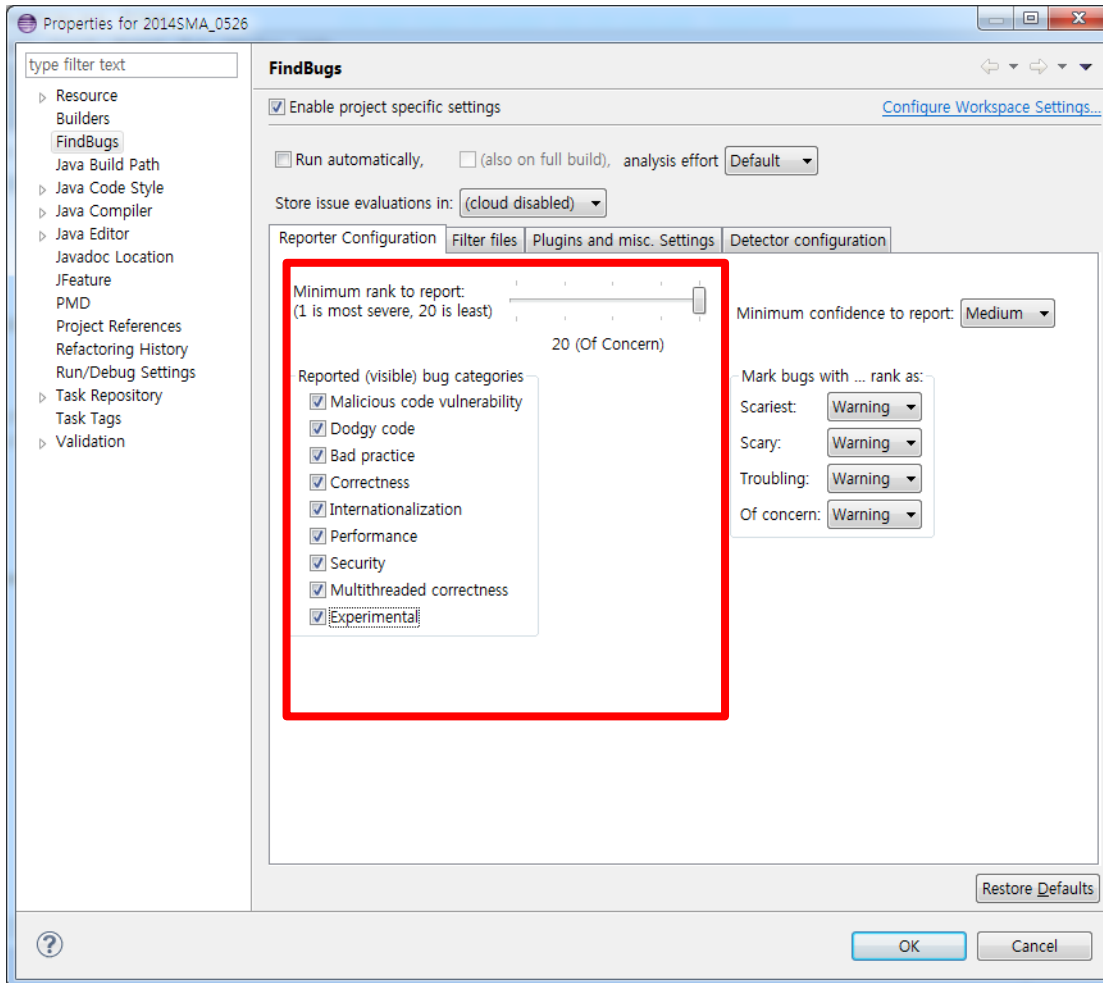
- ❖ 정적 분석 도구
- ❖ Java 소스에서 버그를 찾기 위한 오픈 소스 프로그램
- ❖ Java Byte Code에서 잠재적으로 버그가 일어날 부분을 식별
- ❖ 잠재적인 에러를 4등급으로 분류
  1. Scariest (rank 1~4)
  2. Scary (rank 5~9)
  3. Troubling (rank 1~14)
  4. concern (rank 15~20)

# 1. Find Bugs (2/7)



- Find Bugs 기본 설정
- ✓ 검출된 Bugs 없음

# 1. Find Bugs (3/7)



- Find Bugs 기본 설정
- ✓ 검출된 Bugs 없음
  
- 모든 카테고리 설정

# 1. Find Bugs (4/7)

---

## □ Results (7)

- ▲ 📁 2014SMA\_0526 (7)
  - ▲ 🚩 Of Concern (7)
    - ▲ 🚩 Normal confidence (7)
      - ▲ 🚩 Switch statement found where default case is missing (1)
        - 🚩 Switch statement found in window.MainView.check(int) where default case is missing [Of Concern(19), Normal confidence]
      - ▲ 🚩 Should be a static inner class (1)
        - 🚩 Should window.ReservationView\$ReservationLog be a `_static_` inner class? [Of Concern(18), Normal confidence]
      - ▲ 🚩 Unread field (5)
        - 🚩 Unread field: nwec.Cabin.movingState [Of Concern(18), Normal confidence]
        - 🚩 Unread field: nwec.Cabin.smartphoneCall [Of Concern(18), Normal confidence]
        - 🚩 Unread field: nwec.Floor.humanList [Of Concern(18), Normal confidence]
        - 🚩 Unread field: nwec.Human.INCABIN [Of Concern(18), Normal confidence]
        - 🚩 Unread field: nwec.Human.INFLOOR [Of Concern(18), Normal confidence]

# 1. Find Bugs (5/7)

---

- Switch문에서 Default문을 쓰지 않음 – MainView.java

```
public String check(int direction){
    String str="";
    switch (direction) {
        case 1:
            str="위";
            break;
        case -1:
            str="아래";
            break;
        case 0:
            str="정지";
            break;
    }
    return str;
}
```



# 1. Find Bugs (6/7)

---

- 내부 Class는 Static으로 선언해야 한다. – ReservationView.java

```
public class ReservationLog extends JTextArea{  
  
    public ReservationLog(){  
  
    public void append(String str) {  
        // TODO Auto-generated method stub  
        super.append(str+"\n");  
        super.setCaretPosition(super.getDocument().getLength());  
    }  
}
```

---

# 1. Find Bugs (7/7)

---

## □ 사용되지 않은 객체 (8)

### ☛ Unread field (5)

☛ Unread field: nwec.Cabin.movingState [Of Concern(18), Normal confidence]

☛ Unread field: nwec.Cabin.smartphoneCall [Of Concern(18), Normal confidence]

☛ Unread field: nwec.Floor.humanList [Of Concern(18), Normal confidence]

☛ Unread field: nwec.Human.INCABIN [Of Concern(18), Normal confidence]

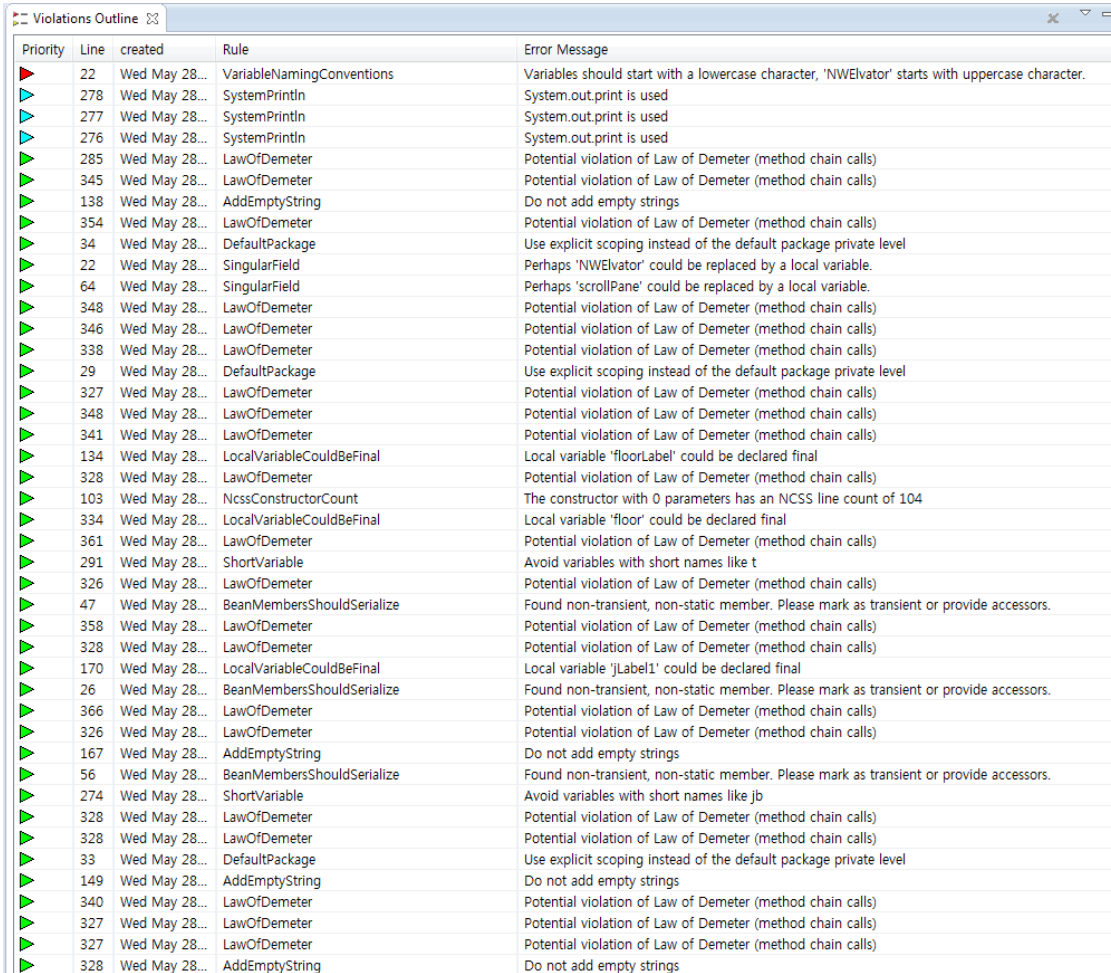
☛ Unread field: nwec.Human.INFLOOR [Of Concern(18), Normal confidence]

## 2. PMD (1/18)

---

- ❖ 정적 분석 도구
- ❖ Java 소스를 분석하여 프로그램의 불필요한 부분을 찾아내고 성능을 높이도록 도와주는 오픈 소스 도구
- ❖ Java 소스코드 분석기로 잠재적인 결함을 찾아준다.
- ❖ Eclipse, Jbuilder 등의 IDE와 연동 가능

# 2. PMD (2/18)



Priority	Line	created	Rule	Error Message
▶	22	Wed May 28...	VariableNamingConventions	Variables should start with a lowercase character, 'NWElevator' starts with uppercase character.
▶	278	Wed May 28...	SystemPrintIn	System.out.print is used
▶	277	Wed May 28...	SystemPrintIn	System.out.print is used
▶	276	Wed May 28...	SystemPrintIn	System.out.print is used
▶	285	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	345	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	138	Wed May 28...	AddEmptyString	Do not add empty strings
▶	354	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	34	Wed May 28...	DefaultPackage	Use explicit scoping instead of the default package private level
▶	22	Wed May 28...	SingularField	Perhaps 'NWElevator' could be replaced by a local variable.
▶	64	Wed May 28...	SingularField	Perhaps 'scrollPane' could be replaced by a local variable.
▶	348	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	346	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	338	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	29	Wed May 28...	DefaultPackage	Use explicit scoping instead of the default package private level
▶	327	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	348	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	341	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	134	Wed May 28...	LocalVariableCouldBeFinal	Local variable 'floorLabel' could be declared final
▶	328	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	103	Wed May 28...	NcssConstructorCount	The constructor with 0 parameters has a NCSS line count of 104
▶	334	Wed May 28...	LocalVariableCouldBeFinal	Local variable 'floor' could be declared final
▶	361	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	291	Wed May 28...	ShortVariable	Avoid variables with short names like t
▶	326	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	47	Wed May 28...	BeanMembersShouldSerialize	Found non-transient, non-static member. Please mark as transient or provide accessors.
▶	358	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	328	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	170	Wed May 28...	LocalVariableCouldBeFinal	Local variable 'jLabel1' could be declared final
▶	26	Wed May 28...	BeanMembersShouldSerialize	Found non-transient, non-static member. Please mark as transient or provide accessors.
▶	366	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	326	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	167	Wed May 28...	AddEmptyString	Do not add empty strings
▶	56	Wed May 28...	BeanMembersShouldSerialize	Found non-transient, non-static member. Please mark as transient or provide accessors.
▶	274	Wed May 28...	ShortVariable	Avoid variables with short names like jb
▶	328	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	328	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	33	Wed May 28...	DefaultPackage	Use explicit scoping instead of the default package private level
▶	149	Wed May 28...	AddEmptyString	Do not add empty strings
▶	340	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	327	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	327	Wed May 28...	LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
▶	328	Wed May 28...	AddEmptyString	Do not add empty strings

□ Default 설정으로 실행하면  
너무 많은 결과가 나온다.

□ 중요하다고 생각되는  
Rule만 적용

## 2. PMD (3/18)

### □ Cabin.java (19)

Line	created	Rule	Error Message
9	Wed May 28 02:27:19 KST 2014	CyclomaticComplexity	The class 'Cabin' has a Cyclomatic Complexity of 4 (Highest = 32).
148	Wed May 28 02:27:19 KST 2014	CyclomaticComplexity	The method 'getWaitingTime' has a Cyclomatic Complexity of 16.
201	Wed May 28 02:27:19 KST 2014	CyclomaticComplexity	The method 'goToFloor' has a Cyclomatic Complexity of 32.
201	Wed May 28 02:27:19 KST 2014	NPathComplexity	The method goToFloor() has an NPath complexity of 1440
240	Wed May 28 02:27:19 KST 2014	ShortVariable	Avoid variables with short names like o2
240	Wed May 28 02:27:19 KST 2014	ShortVariable	Avoid variables with short names like o1
253	Wed May 28 02:27:19 KST 2014	ShortVariable	Avoid variables with short names like o1
253	Wed May 28 02:27:19 KST 2014	ShortVariable	Avoid variables with short names like o2
10	Wed May 28 02:27:19 KST 2014	TooManyMethods	This class has too many methods, consider refactoring it.

- **CyclomaticComplexity** - 순환복잡도는 메소드 내의 결정포인트(decision point)들의 수로 결정된다. 결정포인트란 일반적으로 if, while, for 및 case labels들을 의미하며, 1-4는 낮은 복잡도, 5-7은 중간 복잡도, 8-9는 높은 복잡도 그리고 11이상은 매우 높은 복잡도를 의미한다.
- **NPathComplexity** - 메소드의 복잡성은 많은 양의 비순환적인 실행 문(acyclic execution paths)을 의미한다. 일반적인 한계점은 한 메소드당 200이며, 이 한계점을 넘는 복잡성을 줄일 필요가 있다.
- **ShortVariable** - 너무 짧은 클래스 멤버 변수, 지역변수, 전달 인수는 피하자.
- **Toomanymethods** - 클래스가 너무 많은 메소드를 가지고 있다 (최대 10개)

## 2. PMD (4/18)

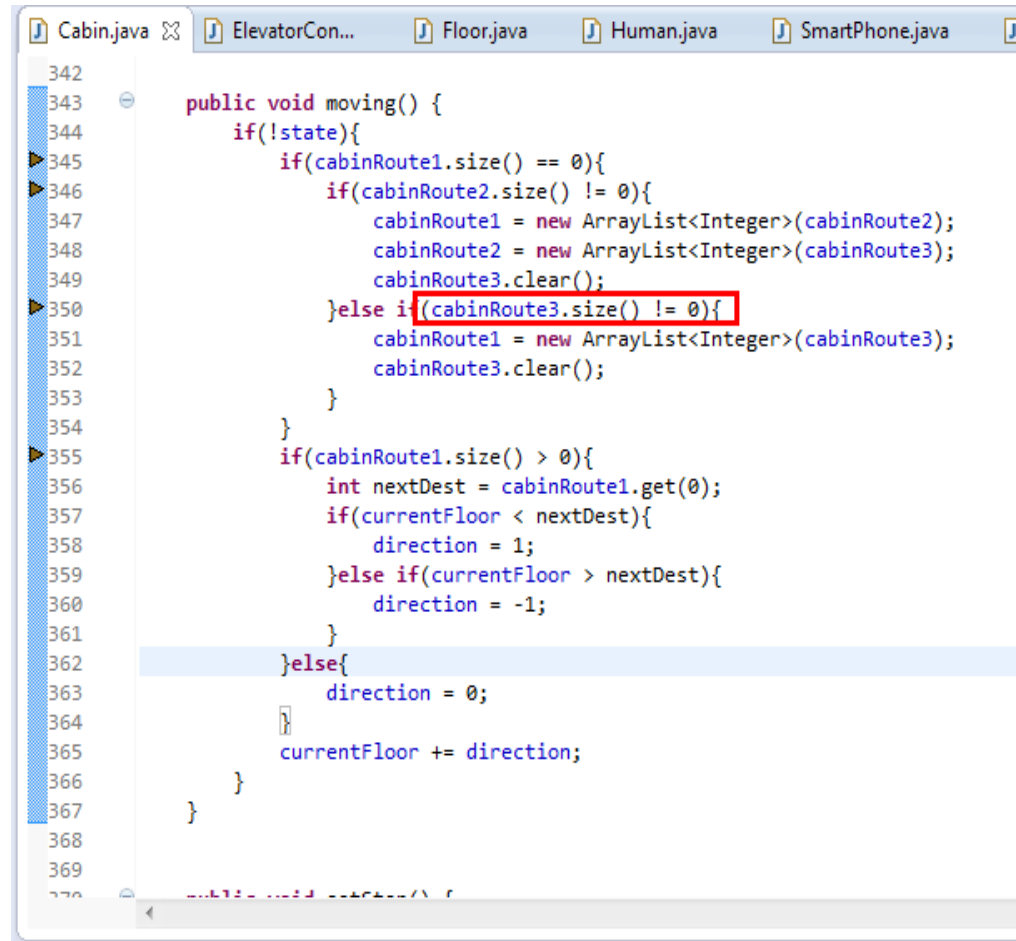
### □ Cabin.java (19)

108	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
313	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
112	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
182	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
381	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
352	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
377	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
178	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
273	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
347	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
379	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
357	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
110	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
348	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
186	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
285	Wed May 28 02:27:19 KST 2014	UseCollectionIsEmpty	Substitute calls to size() == 0 (or size() != 0) with calls to isEmpty()
220	Wed May 28 02:27:19 KST 2014	UselessParentheses	Useless parentheses.
155	Wed May 28 02:27:19 KST 2014	UselessParentheses	Useless parentheses.
164	Wed May 28 02:27:19 KST 2014	UselessParentheses	Useless parentheses.
209	Wed May 28 02:27:19 KST 2014	UselessParentheses	Useless parentheses.

- **UseCollectionIsEmpty** - size() 와 0을 비교하는 것은 중복되는 행동이기 때문에 isEmpty 메소드를 이용하라
- **Useless parentheses** - 불필요한 문장은 삭제되어야 한다.

## 2. PMD (5/18)

- Cabin.java
- UseCollectionIsEmpty
- size와 0을 비교하는 건 중복
- isEmpty() method 이용 가능



```
342
343 public void moving() {
344     if(!state){
345         if(cabinRoute1.size() == 0){
346             if(cabinRoute2.size() != 0){
347                 cabinRoute1 = new ArrayList<Integer>(cabinRoute2);
348                 cabinRoute2 = new ArrayList<Integer>(cabinRoute3);
349                 cabinRoute3.clear();
350             }else if(cabinRoute3.size() != 0){
351                 cabinRoute1 = new ArrayList<Integer>(cabinRoute3);
352                 cabinRoute3.clear();
353             }
354         }
355     }
356     if(cabinRoute1.size() > 0){
357         int nextDest = cabinRoute1.get(0);
358         if(currentFloor < nextDest){
359             direction = 1;
360         }else if(currentFloor > nextDest){
361             direction = -1;
362         }
363     }else{
364         direction = 0;
365     }
366     currentFloor += direction;
367 }
368
369
370 public void restStep() {
```

## 2. PMD (6/18)

---

### □ ElevatorControl.java (2)

Priority	Line	created	Rule	Error Message
▶	39	Wed ...	UselessParentheses	Useless parentheses.
▶	40	Wed ...	UselessParentheses	Useless parentheses.

- UselessParentheses – 불필요한 문장은 삭제되어야 한다.



## 2. PMD (7/18)

- ElevatorControl.java
- UselessParentheses
- SelectCabin = -1



```
Cabin.java ElevatorCon... Floor.java Human.java SmartPhone.java Reservati
28         if(floorList.get(i).getWaitingNum(>0){
29             System.out.println(i+"층에 사람이 있어요~");
30             floorList.get(i).setButton(true);
31         }
32     }
33 }
34
35 public int goToFloor(int curFloor, int targetFloor){
36     int selectCabin = -1;
37     int shortTime = -1;
38     for(int i =0; i< cabinList.size(); i++){
39
40         int time = cabinList.get(i).getWaitingTime(curFloor, targetFloor);
41
42         if(selectCabin == -1
43            || (shortTime > 0 && time < shortTime)
44            || (shortTime <= 0 && time <= 0 && time > shortTime)
45           ){
46
47             selectCabin = i;
48             shortTime = time;
49
50         }
51     }
52     cabinList.get(selectCabin).goToFloor(curFloor, targetFloor);
53     System.out.println("EC -goToFloor "+shortTime+", " + selectCabin);
54     return selectCabin;
55 }
```

## 2. PMD (8/18)

---

### □ Floor.java (1)

Priority	Line	created	Rule	Error Message
▶	17	Wed ...	UnusedPrivateField	Avoid unused private fields such as 'humanList'.

- **UnusedPrivateField** - private로 접근 제한된 클래스 멤버 변수들 중에 정의 되었거나 값이 할당되었지만 사용되지 않은 경우를 경고한다.

## 2. PMD (9/18)

---

### □ Human.java (3)

Priority	Line	created	Rule	Error Message
▶	7	Wed ...	AvoidFieldNameMatchingMethodName	Field INCABIN has the same name as a method
▶	7	Wed ...	UnusedPrivateField	Avoid unused private fields such as 'INCABIN'.
▶	8	Wed ...	UnusedPrivateField	Avoid unused private fields such as 'INFLOOR'.

- **AvoidFieldNameMatchingMethodName** - 클래스 명과 그 클래스에 포함된 필드 명이 같다면 매우 혼동된다.
- **UnusedPrivateField** - `private`로 접근 제한된 클래스 멤버 변수들 중에 정의 되었거나 값이 할당되었지만 사용되지 않은 경우를 경고한다.

## 2. PMD (10/18)

---

### □ Simulator.java (9)

Priority	Line	created	Rule	Error Message
▶	35	Wed ...	AvoidFieldNameMatchingTypeName	It is somewhat confusing to have a field name matching the declaring class name
▶	10	Wed ...	ClassWithOnlyPrivateConstructorsShouldBeFinal	A class which only has private constructors should be final
▶	10	Wed ...	CyclomaticComplexity	The class 'Simulator' has a Cyclomatic Complexity of 4 (Highest = 22).
▶	244	Wed ...	CyclomaticComplexity	The method 'update' has a Cyclomatic Complexity of 22.

- **AvoidFieldNameMatchingMethodName** - 클래스 명과 그 클래스에 포함된 필드 명이 같다면 매우 혼동된다.
- **ClassWithOnlyPrivateConstructorsShouldBeFinal** - 클래스가 오직 private 생성자만을 포함한다면 내부 클래스(inner class)가 호출하지 않는 이상, 이 클래스는 final을 갖고 있어야 한다.
- **CyclomaticComplexity** - 순환복잡도는 메소드 내의 결정포인트(decision point)들의 수로 결정된다. 결정포인트란 일반적으로 if, while, for 및 case labels들을 의미하며, 1-4는 낮은 복잡도, 5-7은 중간 복잡도, 8-9는 높은 복잡도 그리고 11이상은 매우 높은 복잡도를 의미한다.

## 2. PMD (11/18)

---

### □ Simulator.java (9)

▶	171	Wed ...	EmptyStatementNotInLoop	An empty statement (semicolon) not part of a loop
▶	165	Wed ...	EmptyStatementNotInLoop	An empty statement (semicolon) not part of a loop
▶	244	Wed ...	NPathComplexity	The method update() has an NPath complexity of 20736
▶	38	Wed ...	NonThreadSafeSingleton	Singleton is not thread safe
▶	272	Wed ...	UnconditionalIfStatement	Do not use if statements that are always true or always false

- **EmptyStatementNotInLoop** - 필요없는 문장 (;)이 있음
- **NPathComplexity** - 메소드의 복잡성은 많은 양의 비순환적인 실행 문(acyclic execution paths)을 의미한다. 일반적인 한계점은 한 메소드당 200이며, 이 한계점을 넘는 복잡성을 줄일 필요가 있다.
- **UnconditionalIfStatement** - 조건절에 비교문이 들어가지 않고 단순히 true 또는 false가 되는 경우 if문을 사용하지 말자

## 2. PMD (12/18)

### □ MainView.java (7)

Priority	Line	created	Rule	Error Message
▶	103	Wed May 28...	CyclomaticComplexity	The constructor 'MainView' has a Cyclomatic Complexity of 10.
▶	19	Wed May 28...	CyclomaticComplexity	The class 'MainView' has a Cyclomatic Complexity of 6 (Highest = 10).
▶	291	Wed May 28...	ShortVariable	Avoid variables with short names like t
▶	274	Wed May 28...	ShortVariable	Avoid variables with short names like jb
▶	271	Wed May 28...	ShortVariable	Avoid variables with short names like e
▶	19	Wed May 28...	TooManyFields	Too many fields
▶	62	Wed May 28...	UnusedPrivateField	Avoid unused private fields such as 'simulatorLog'.

- **CyclomaticComplexity** - 순환복잡도는 매소드 내의 결정포인트(decision point)들의 수로 결정된다. 결정포인트란 일반적으로 if, while, for 및 case labels들을 의미하며, 1-4는 낮은 복잡도, 5-7은 중간 복잡도, 8-9는 높은 복잡도 그리고 11이상은 매우 높은 복잡도를 의미한다.
- **ShortVariable** - 너무 짧은 클래스 멤버 변수, 지역변수, 전달 인수는 피하자.
- **TooManyFields** - 클래스가 너무 많은 수의 클래스 멤버 변수들을 갖고 있다면, 가능하면 연관성 있는 필드들은 그룹화된 오브젝트로 묶어야한다.
- **UnusedPrivateField** - private로 접근 제한된 클래스 멤버 변수들 중에 정의 되었거나 값이 할당되었지만 사용되지 않은 경우를 경고한다

## 2. PMD (13/18)

### □ ReservationView.java (11)

Priority	Line	created	Rule	Error Message
▶	121	Wed ...	ShortVariable	Avoid variables with short names like e
▶	123	Wed ...	ShortVariable	Avoid variables with short names like jb
▶	111	Wed ...	UnnecessaryConstructor	Avoid unnecessary constructors - the compiler will generate these for you
▶	17	Wed ...	UnusedImports	Avoid unused imports such as 'nwec.Cabin'
▶	12	Wed ...	UnusedImports	Avoid unused imports such as 'javax.swing.JOptionPane'
▶	4	Wed ...	UnusedImports	Avoid unused imports such as 'java.awt.Dimension'
▶	31	Wed ...	UnusedPrivateField	Avoid unused private fields such as 'humanList'.
▶	36	Wed ...	UnusedPrivateField	Avoid unused private fields such as 'reservationList'.
▶	33	Wed ...	UnusedPrivateField	Avoid unused private fields such as 'curStateJScrollPane'.
▶	34	Wed ...	UnusedPrivateField	Avoid unused private fields such as 'reservationLog'.
▶	28	Wed ...	UnusedPrivateField	Avoid unused private fields such as 'desFloor'.

- **ShortVariable** - 너무 짧은 클래스 멤버 변수, 지역변수, 전달 인수는 피하자.
- **UnnecessaryConstructor** - 이 룰은 생성자가 불필요한 경우를 경고한다. 생성자가 불필요한 경우는 클래스 내에 하나의 **public** 생성자외에는 어떤 한 것도 들어있지 않은 경우이다.
- **TooManyFields** - 클래스가 너무 많은 수의 클래스 멤버 변수들을 갖고 있다면, 가능하면 연관성 있는 필드들은 그룹화된 오브젝트로 묶어야한다.
- **Unusedimports** - 사용되지 않는 **import** 문장을 피해야 한다
- **UnusedPrivateField** - **private**로 접근 제한된 클래스 멤버 변수들 중에 정의 되었거나 값이 할당되었지만 사용되지 않은 경우를 경고한다

## 2. PMD (14/18)

---

### □ SmartLogView.java (1)

Priority	Line	created	Rule	Error Message
▶	9	Wed ...	ClassWithOnlyPrivateConstructorsShouldBeFinal	A class which only has private constructors should be final

- **ClassWithOnlyPrivateConstructorsShouldBeFinal** - 클래스가 오직 `private` 생성자만을 포함한다면 내부 클래스(inner class)가 호출하지 않는 이상, 이 클래스는 `final`을 갖고 있어야 한다.



## 2. PMD (15/18)

---

### □ SmartPhoneView.java (2)

Priority	Line	created	Rule	Error Message
▶	90	Wed May 28 02:27:19 KST 2014	ConstructorCallsOverridableMethod	Overridable method 'init' called during object construction
▶	95	Wed May 28 02:27:19 KST 2014	ShortVariable	Avoid variables with short names like t

- **ConstructorCallsOverridableMethod** - 생성자가 오버라이드 가능한 (overridable) toString 같은 메소드들을 호출한다면, 불완전하게 생성된 오브젝트를 호출하는 위험이 있을 수 있으며, 이를 알아차리기 힘들다
- **ShortVariable** - 너무 짧은 클래스 멤버 변수, 지역변수, 전달 인수는 피하자.

## 2. PMD (16/18)

---

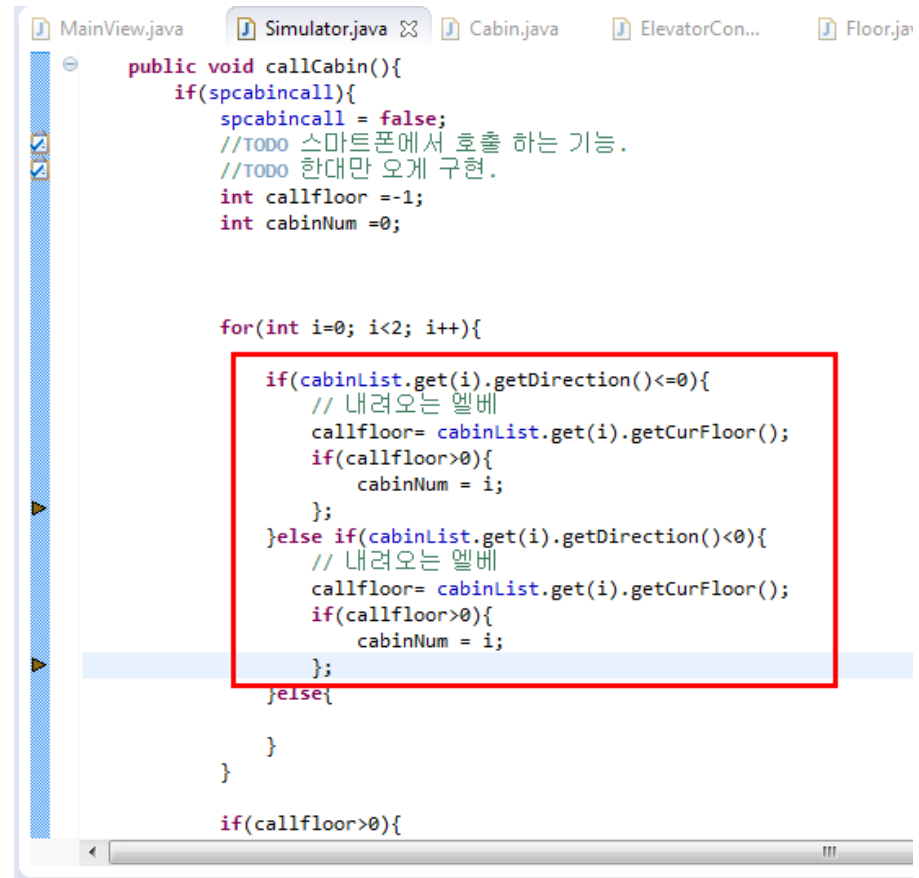
### □ 중복 코드 검사

➤ 4개의 중복 코드 발견 -> 2개 제외

Spans	Source
▶ 7	src.nwec.Simulator src.nwec.Simulator
▶ 4	src.nwec.Simulator src.nwec.Simulator
▶ 6	src.nwec.Cabin src.nwec.Cabin
▶ 11	src.window.ReservationView src.window.SimulatorLogView

## 2. PMD (17/18)

- 중복 코드 검사
- Simulator.java



```
public void callCabin(){
    if(spcabincall){
        spcabincall = false;
        //TODO 스마트폰에서 호출 하는 기능.
        //TODO 한대만 오게 구현.
        int callfloor = -1;
        int cabinNum = 0;

        for(int i=0; i<2; i++){

            if(cabinList.get(i).getDirection()<=0){
                // 내려오는 엘베
                callfloor= cabinList.get(i).getCurFloor();
                if(callfloor>0){
                    cabinNum = i;
                };
            }else if(cabinList.get(i).getDirection()<0){
                // 내려오는 엘베
                callfloor= cabinList.get(i).getCurFloor();
                if(callfloor>0){
                    cabinNum = i;
                };
            }
        }

        if(callfloor>0){
```

## 2. PMD (18/18)

### □ 중복 코드 검사 (1/4)

```
    }  
}  
  
public class ReservationLog extends JTextArea{  
  
    public ReservationLog(){  
  
        public void append(String str) {  
            // TODO Auto-generated method stub  
            super.append(str+"\n");  
            super.setCaretPosition(super.getDocument().getLength());  
        }  
    }  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // TODO Auto-generated method stub  
        JButton jb = (JButton) e.getSource();  
  
        inputBtn=Integer.parseInt(jb.getName());  
  
        if(inputBtn==11){  
  
            simulator.setFloorWaitingHuman(curFloor, humanList);  
            this.setVisible(false);  
            //  
            //  
            //  
            <
```

ReservationView.java

```
package window;  
  
import javax.swing.JTextArea;  
  
/**  
 * 로그 출력용 클래스.  
 * @author minwookim  
 */  
public class SimulatorLogView extends JTextArea{  
    private static SimulatorLogView simulatorLog = new SimulatorLogView  
  
    private SimulatorLogView(){  
  
        /**  
         * 전역적으로 사용되는 로그 출력을 위해 singleton 패턴을 이용.  
         * @return simulatorLog Instance.  
         */  
        public static SimulatorLogView getInstance(){  
            return simulatorLog;  
        }  
  
        /**  
         * append 함수의 확장으로 줄내림과 함께 스크롤을 맨 아래로 내리는  
         */  
        public void append(String str) {  
            // TODO Auto-generated method stub  
            super.append(str+"\n");  
            super.setCaretPosition(super.getDocument().getLength());  
        }  
    }  
}
```

SimulatorLogView.java

# 3. JDepend (1/2)

---

- 정적 분석 도구
- Java 패키지간의 의존성에 대한 수치들을 알려주는 도구
- 패키지 별로 의존성 측정 가능
- 수치화된 정보를 텍스트 & 그래프로 표현

# 3. JDepend (2/2)

**Packages**

- nwec
- window

패키지 종류

**Dependencies**

Selected object(s) **현재 선택한 패키지**

Package	CC(con...)	AC(abst...)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
nwec	9	0	1	1	0.00	0.50	0.50	⚠
window	9	0	1	1	0.00	0.50	0.50	⚠

Packages with cycle **사이클을 가지고 있는 패키지**

Package	CC(con...)	AC(abst...)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
nwec	9	0	1	1	0.00	0.50	0.50	⚠
window	9	0	1	1	0.00	0.50	0.50	⚠

Depends upon - efferent dependencies **선택한 패키지에서 사용하고 있는 패키지**

Package	CC(con...)	AC(abst...)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
nwec	9	0	1	1	0.00	0.50	0.50	⚠
window	9	0	1	1	0.00	0.50	0.50	⚠

Used by - afferent dependencies **선택한 패키지를 사용하고 있는 패키지**

Package	CC(con...)	AC(abst...)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
nwec	9	0	1	1	0.00	0.50	0.50	⚠
window	9	0	1	1	0.00	0.50	0.50	⚠

**Metrics**

Instability ->

Abstractnes

# 4. Check Style (1/7)

---

- 정적 분석 도구
- Java 소스코드의 기술형식 코딩 규약에 준하고 있는지 확인
- 오픈 소스

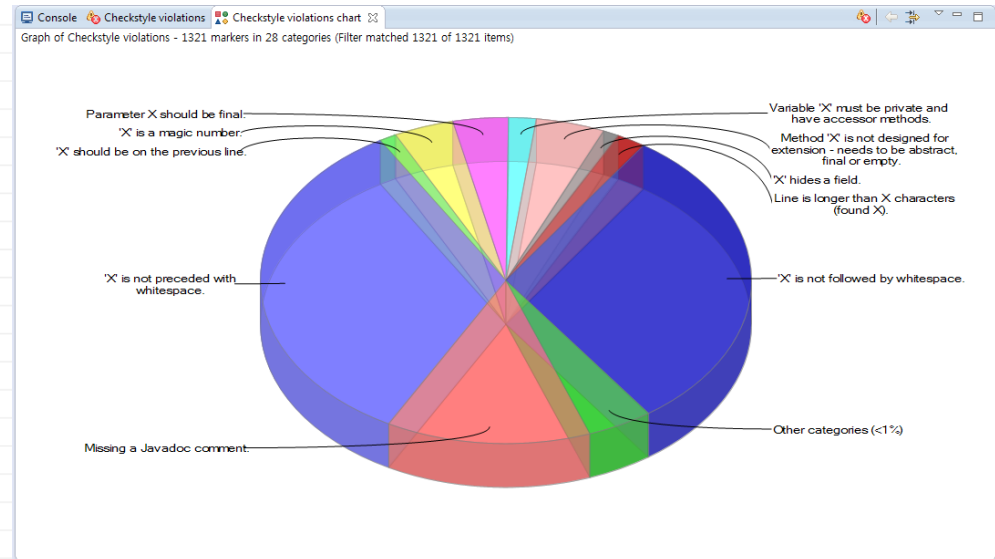
# 4. Check Style (2/7)

## □ 29개의 룰에 대해 1321개의 위반 발견

Console Checkstyle violations

Overview of Checkstyle violations - 1321 markers in 28 categories (Filter matched 1321 of 1321 items)

Checkstyle violation type	Marker count
Missing a Javadoc comment.	178
Missing package-info.java file.	2
switch without "default" clause.	2
Array brackets at illegal position.	13
Name 'X' must match pattern 'X'.	8
'X' is not preceded with whitespace.	443
'X' should be alone on a line.	2
Method length is X lines (max allowed is X).	1
'X' should be on the previous line.	15
'X' is a magic number.	52
Parameter X should be final.	48
'X' should be on the same line.	1
'X' construct must use '{}'. 'X' is not preceded with whitespace.	1
Variable 'X' must be private and have accessor methods.	24
Method 'X' is not designed for extension - needs to be abstract, final ...	60
First sentence should end with a period.	1
Conditional logic can be removed.	1
Must have at least one statement.	3
'X' is preceded with whitespace.	6
'X' hides a field.	14
Unused import - X.	3
Class X should be declared as final.	2
Expected X tag for 'X'.	4
Empty statement.	2
Line is longer than X characters (found X).	25
Avoid nested blocks.	2





# 4. Check Style (3/7)

---

## □ Switch문에서 Default문을 쓰지 않음

Details of Checkstyle violation "switch without "default" clause." - 2 occurrences

Resource	In Folder	Line	Message
⚠ MainView.java	/2014SMA/src/window	309	switch without "default" clause.
⚠ Cabin.java	/2014SMA/src/nwec	266	switch without "default" clause.

## □ Method의 길이가 너무 길다 (150줄 이상)

Console ⚠ Checkstyle violations ⌵

Details of Checkstyle violation "Method length is X lines (max allowed is X)." - 1 occurrences

Resource	In Folder	Line	Message
⚠ MainView.java	/2014SMA/src/window	103	Method length is 164 lines (max allowed is 150).

# 4. Check Style (4/7)

- 문장이 너무 길다
- 80자 이상

Resource	In Folder	Line	Message
SmartPhoneVie...	/2014SMA/src/window	110	Line is longer than 80 characters (found 87).
ReservationVie...	/2014SMA/src/window	142	Line is longer than 80 characters (found 118).
ReservationVie...	/2014SMA/src/window	145	Line is longer than 80 characters (found 93).
ReservationVie...	/2014SMA/src/window	44	Line is longer than 80 characters (found 84).
MainView.java	/2014SMA/src/window	327	Line is longer than 80 characters (found 87).
MainView.java	/2014SMA/src/window	338	Line is longer than 80 characters (found 88).
MainView.java	/2014SMA/src/window	346	Line is longer than 80 characters (found 88).
MainView.java	/2014SMA/src/window	280	Line is longer than 80 characters (found 85).
MainView.java	/2014SMA/src/window	281	Line is longer than 80 characters (found 107).
MainView.java	/2014SMA/src/window	285	Line is longer than 80 characters (found 82).
Simulator.java	/2014SMA/src/nwec	180	Line is longer than 80 characters (found 120).
Simulator.java	/2014SMA/src/nwec	183	Line is longer than 80 characters (found 86).
Simulator.java	/2014SMA/src/nwec	212	Line is longer than 80 characters (found 116).
Simulator.java	/2014SMA/src/nwec	273	Line is longer than 80 characters (found 101).
Simulator.java	/2014SMA/src/nwec	275	Line is longer than 80 characters (found 110).
Simulator.java	/2014SMA/src/nwec	304	Line is longer than 80 characters (found 105).
Simulator.java	/2014SMA/src/nwec	141	Line is longer than 80 characters (found 81).
ElevatorControl...	/2014SMA/src/nwec	37	Line is longer than 80 characters (found 82).
ElevatorControl...	/2014SMA/src/nwec	17	Line is longer than 80 characters (found 84).
Cabin.java	/2014SMA/src/nwec	129	Line is longer than 80 characters (found 85).
Cabin.java	/2014SMA/src/nwec	179	Line is longer than 80 characters (found 152).
Cabin.java	/2014SMA/src/nwec	183	Line is longer than 80 characters (found 151).
Cabin.java	/2014SMA/src/nwec	187	Line is longer than 80 characters (found 151).
Cabin.java	/2014SMA/src/nwec	273	Line is longer than 80 characters (found 84).
Cabin.java	/2014SMA/src/nwec	285	Line is longer than 80 characters (found 84).

# 4. Check Style (5/7)

- 변수가 **private**으로 설정되어 있지 않고 접근 메소드가 없다.

Resource	In Folder	Line	Message
SmartPhoneVie...	/2014SMA/src/window	29	Variable 'cabin1_F' must be private and have acces...
SmartPhoneVie...	/2014SMA/src/window	30	Variable 'cabin1CurPos' must be private and have ...
SmartPhoneVie...	/2014SMA/src/window	31	Variable 'cabin2_F' must be private and have acces...
SmartPhoneVie...	/2014SMA/src/window	32	Variable 'cabin2CurPos' must be private and have ...
SmartPhoneVie...	/2014SMA/src/window	33	Variable 'callEle' must be private and have accessor...
MainView.java	/2014SMA/src/window	23	Variable 'smart' must be private and have accessor ...
MainView.java	/2014SMA/src/window	25	Variable 'buidingDisplay' must be private and have...
MainView.java	/2014SMA/src/window	26	Variable 'floorState' must be private and have acce...
MainView.java	/2014SMA/src/window	28	Variable 'upButtons' must be private and have acce...
MainView.java	/2014SMA/src/window	29	Variable 'downButtons' must be private and have a...
MainView.java	/2014SMA/src/window	30	Variable 'cabin1Pos' must be private and have acce...
MainView.java	/2014SMA/src/window	31	Variable 'cabin2Pos' must be private and have acce...
MainView.java	/2014SMA/src/window	32	Variable 'humanNum' must be private and have ac...
MainView.java	/2014SMA/src/window	33	Variable 'waitingNum' must be private and have ac...
MainView.java	/2014SMA/src/window	34	Variable 'isweight' must be private and have access...
MainView.java	/2014SMA/src/window	35	Variable 'sosBtn' must be private and have accesso...
MainView.java	/2014SMA/src/window	38	Variable 'sideDisplay' must be private and have acc...
MainView.java	/2014SMA/src/window	41	Variable 'reservationView' must be private and hav...
Cabin.java	/2014SMA/src/nwec	17	Variable 'movingState' must be private and have a...
Cabin.java	/2014SMA/src/nwec	18	Variable 'currentFloor' must be private and have ac...
Cabin.java	/2014SMA/src/nwec	19	Variable 'sosState' must be private and have access...
Cabin.java	/2014SMA/src/nwec	20	Variable 'humanNum' must be private and have ac...
Cabin.java	/2014SMA/src/nwec	21	Variable 'overWeight' must be private and have acc...
Cabin.java	/2014SMA/src/nwec	26	Variable 'smartphoneCall' must be private and hav...

# 4. Check Style (6/7)

---

- ❑ 블록 안에 아무 코드도 들어있지 않다.

Console Checkstyle violations

Details of Checkstyle violation "Must have at least one statement." - 3 occurrences

Resource	In Folder	Line	Message
ReservationVie...	/2014SMA/src/window	130	Must have at least one statement.
Simulator.java	/2014SMA/src/nwec	172	Must have at least one statement.
Cabin.java	/2014SMA/src/nwec	98	Must have at least one statement.

- ❑ Import 된 패키지를 사용하지 않는다.

Console Checkstyle violations

Details of Checkstyle violation "Unused import - X." - 3 occurrences

Resource	In Folder	Line	Message
ReservationVie...	/2014SMA/src/window	4	Unused import - java.awt.Dimension.
ReservationVie...	/2014SMA/src/window	12	Unused import - javax.swing.JOptionPane.
ReservationVie...	/2014SMA/src/window	17	Unused import - nwec.Cabin.

# 4. Check Style (7/7)

---

- **;(세미콜론) 전까지 아무런 코드가 없다. 빈 문장이다.**

Console Checkstyle violations

Details of Checkstyle violation "Empty statement." - 2 occurrences

Resource	In Folder	Line	Message
Simulator.java	/2014SMA/src/nwec	165	Empty statement.
Simulator.java	/2014SMA/src/nwec	171	Empty statement.

- **Else 다음에 블록이 없다.**

Console Checkstyle violations

Details of Checkstyle violation "'X' construct must use '{}'. " - 1 occurrences

Resource	In Folder	Line	Message
Cabin.java	/2014SMA/src/nwec	93	'else' construct must use '{}'.

## 5. JFeature (1/13)

---

- 정적 분석 도구
- Junit의 Test Case 메소드와 통합
- 요구 사항의 수정 사항이 생겼을 때, 즉시 뷰를 제공하므로 빠른 편집이 가능
- 요구 사항의 반영여부를 Coverage형태로 쉽게 확인

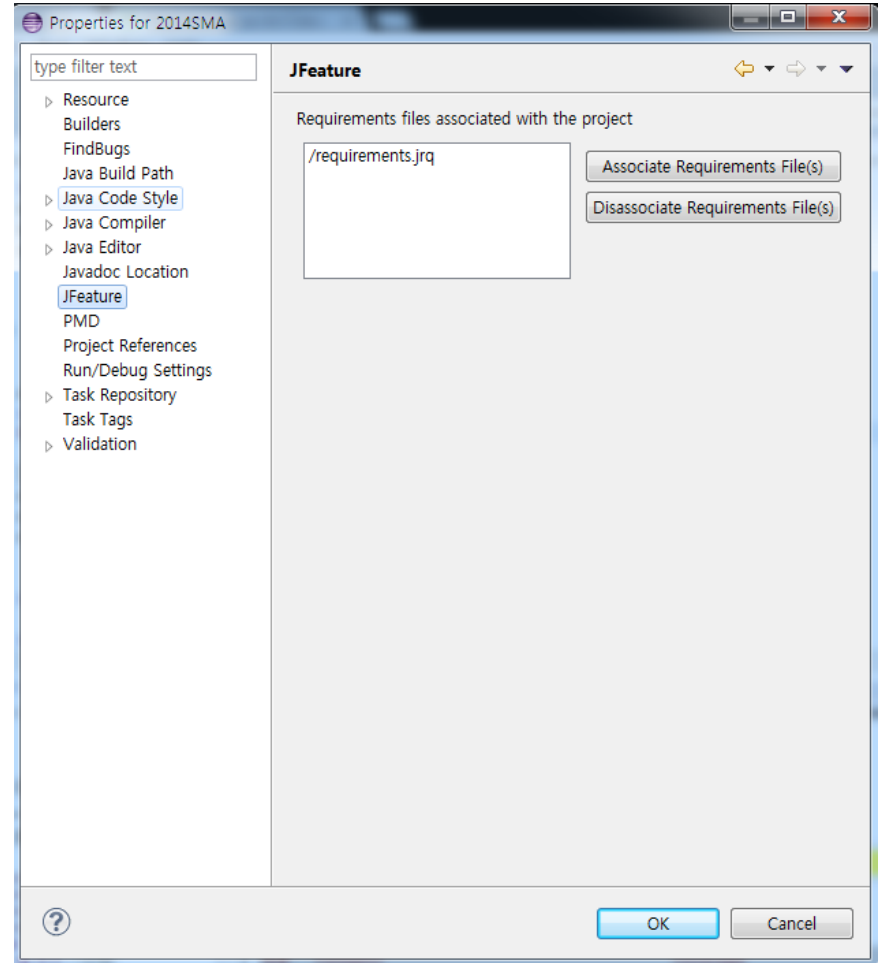
# 5. JFeature (2/13)

## □ 요구 사항 기록 (15)

Requirement ID	Category	Sub Category	Title	Target Release	Prior...	M...	Test Methods
REQ_001	엘리베이터	이동	엘리베이터 이동		3	No	nwec.NwecTest.testCabinSecond...
REQ_002	엘리베이터	비상정지	엘리베이터 비상 정지		3	No	nwec.NwecTest.testSOScall
REQ_008	시뮬레이터	무게 초과	엘리베이터 무게 초과 표시		3	No	
REQ_009	시뮬레이터	이동 방향 표시	엘리베이터 이동 방향 표시		3	No	nwec.NwecTest.testSetDestination
REQ_010	시뮬레이터	층 별 디스플레이	층 별 디스플레이		3	No	nwec.NwecTest.testCabinSecond...
REQ_011	시뮬레이터	사람 승/하차	사람 승/하차		3	No	nwec.NwecTest.testCabinSecond...
REQ_012	시뮬레이터	비상 구조 요청	비상 구조 요청		3	No	nwec.NwecTest.testSOScall
REQ_013	스마트폰	대기 시간 표현	스마트폰 대기 시간 표현		3	No	
REQ_014	스마트폰	엘리베이터 위치 ...	현재 엘리베이터 위치 표시		3	No	
REQ_015	스마트폰	버튼 조작	스마트폰 버튼 조작하여 엘리베이터 호출		3	No	
REQ_005	시뮬레이터	호출	N층에서 엘리베이터 호출		3	No	
REQ_003	엘리베이터	대기 시간 계산	엘리베이터 대기 시간 계산		3	No	
REQ_004	엘리베이터	무게 초과 감지	엘리베이터 무게 초과 감지		3	No	
REQ_006	시뮬레이터	층 선택	내부에서 층 선택		3	No	nwec.NwecTest.testCabinSecond...
REQ_007	시뮬레이터	Status 표현	시뮬레이터 Status 표현		3	No	

# 5. JFeature (3/13)

- Jfeature와 Junit과의 연결
- [프로젝트] - [설정] - [Jfeature]



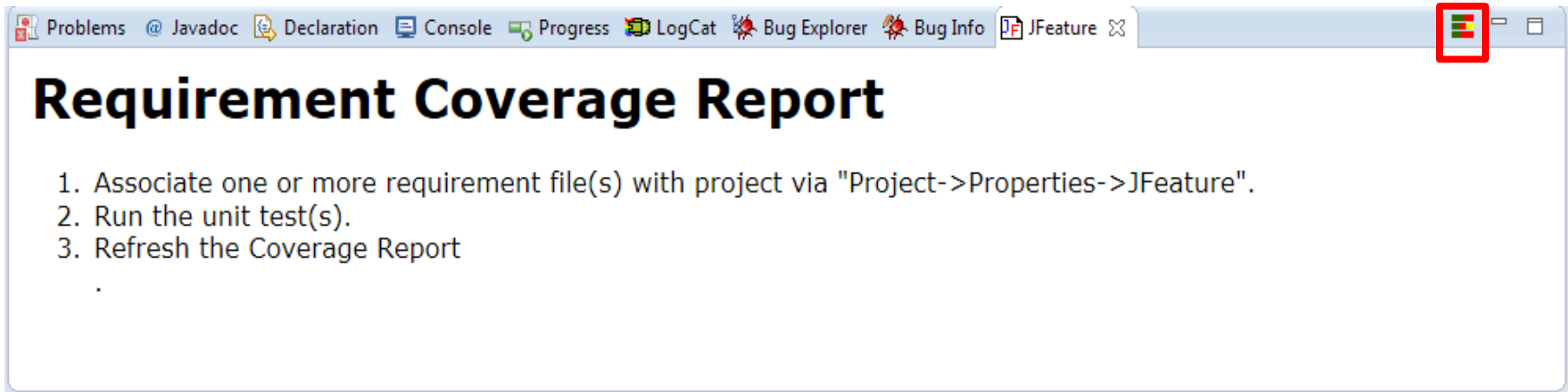


# 5. JFeature (4/13)

---

## □ Jfeature 결과 확인

### ➤ Junit 실행



The screenshot shows an IDE window with a tab labeled 'JFeature'. The window title bar includes icons for Problems, Javadoc, Declaration, Console, Progress, LogCat, Bug Explorer, Bug Info, and JFeature. The main content area displays the following text:

## Requirement Coverage Report

1. Associate one or more requirement file(s) with project via "Project->Properties->JFeature".
2. Run the unit test(s).
3. Refresh the Coverage Report

A red square highlights the window control buttons (minimize, maximize, close) in the top right corner of the IDE window.

# 5. JFeature (5/13)

## □ Jfeature 결과

**Requirement Coverage Report**

**Requirement Coverage Summary**

Summary (15)	7 (46.67%)	8 (53.33%)
--------------	------------	------------

**Requirement Coverage Details**

Sr#	Category	Coverage
1.	엘리베이터 (4)	2 (50%) / 2 (50%)
2.	스마트폰 (3)	3 (100%)
3.	시뮬레이터 (8)	5 (62.5%) / 3 (37.5%)

Report generated on 2014 23:29:31 KST

Powered By JFeature

**All Categories**

- 엘리베이터 (50%)
- 스마트폰 (0%)
- 시뮬레이터 (62.5%)

**All Requirements**

- 엘리베이터 이동 (100%)
- 엘리베이터 비상 중지 (100%)
- 엘리베이터 무게 초과 표시 (0%)
- 엘리베이터 이동 방향 표시 (100%)
- 층별 디스플레이 (100%)
- 사람 승/하차 (100%)
- 비상 구조 요청 (100%)
- 스마트폰 대기 시간 표현 (0%)
- 현재 엘리베이터 위치 표시 (0%)
- 스마트폰 버튼 조작하여 엘리베... (0%)
- N층에서 엘리베이터 호출 (0%)
- 엘리베이터 대기 시간 계산 (0%)
- 엘리베이터 무게 초과 방지 (0%)
- 내부에서 층 선택 (100%)
- 시뮬레이터 Status 표현 (0%)

**Number of Requirements**: 15

**Unique Test Methods**: 6

**Requirements:Test Methods Ratio**: 2:1

**Missing Test Methods**: None

**Unmapped Test Methods**: 7

# 5. JFeature (6/13)

## □ Jfeature 결과 분석

### □ All Categories

- 모든 카테고리에 대한 내용을 볼 수 있다.
- 카테고리 별로 테스트 성공률을 볼 수 있다.

### □ All Requirements

- 모든 요구사항에 대한 내용을 볼 수 있다.
- 요구 사항 별로 테스트 성공률을 볼 수 있다.

Home

#### All Categories

엘리베이터 (50%)  
스마트폰 (0%)  
시뮬레이터 (62.5%)

#### All Requirements

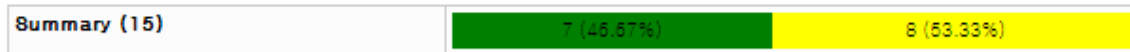
엘리베이터 이동 (100%)  
엘리베이터 비상 정지 (100%)  
엘리베이터 무게 초과 표시 (0%)  
엘리베이터 이동 방향 표시 (100%)  
층 번호 디스플레이 (100%)  
사람 승/하차 (100%)  
비상 구조 요청 (100%)  
스마트폰 대기 시간 프론 (0%)  
현재 엘리베이터 위치 표시 (0%)  
스마트폰 버튼 조작하여 엘리베... (0%)  
N층에서 엘리베이터 호출 (0%)  
엘리베이터 대기 시간 계산 (0%)  
엘리베이터 무게 초과 경고 (0%)  
내부에서 층 선택 (100%)  
시뮬레이터 Status 프론 (0%)

# 5. JFeature (7/13)

## □ Jfeature 결과 분석

Requirement Coverage Report

Requirement Coverage Summary



Number of Requirements	15
Unique Test Methods	6
Requirements:Test Methods Ratio	2:1
Missing Test Methods	None
Unmapped Test Methods	7

Requirement Coverage Details

Br#	Category	Coverage
1.	엘리베이터 (4)	
2.	스마트폰 (3)	
3.	시뮬레이터 (8)	

- 커버리지 범위에 대한 내용을 **Successful**, **broken**, **no coverage** 3가지로 나누어서 비율로 표시 한다.

# 5. JFeature (8/13)

---

## □ Jfeature 결과 분석

### □ Number of Requirements

- 요구 사항 수

### □ Unique Test Methods

- 테스트 method 수

### □ Requirements : Test Methods Ratio

- 요구사항에 대한 테스트 method의 비율
- Method가 높을 경우 해당 요구사항이 제대로 테스트 되지 못하였음을 의미

### □ Missing Test Methods

- 등록된 method중 수행되지 못한 method가 있음을 뜻함

### □ Unmapped Test Methods

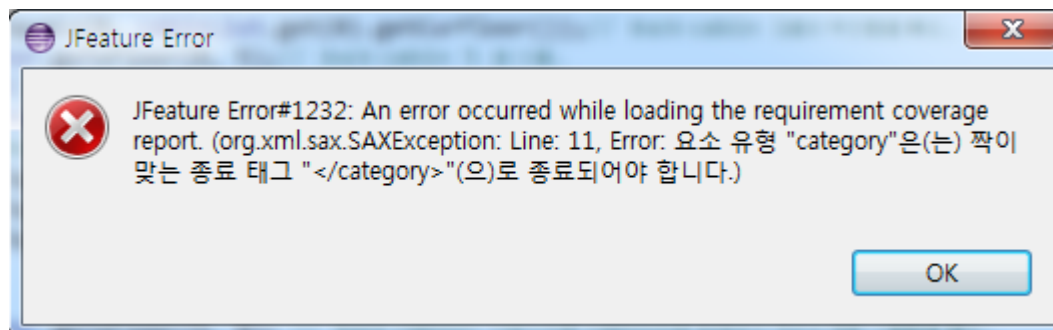
- 요구사항과 매칭되지 못한 테스트의 수

Number of Requirements	15
Unique Test Methods	6
Requirements:Test Methods Ratio	2:1
Missing Test Methods	None
Unmapped Test Methods	7

# 5. JFeature (9/13)

---

- Jfeature 에러 해결
- 실행이 되지 않음.



# 5. JFeature (10/13)

- Jfeature 에러 해결
- 한글을 ""안에 넣어서 해결

Requirement ID	Category	Sub Category	Title	Target Release	Prior...	M...	Test Methods
REQ_001	"엘리베이터"	"이동"	"엘리베이터 이동"		3	No	nwec.NwecTest.testCabinSecond...
REQ_002	"엘리베이터"	"비상정지"	"엘리베이터 비상 정지"		3	No	nwec.NwecTest.testSOScall
REQ_008	"시뮬레이터"	"무게 초과"	"엘리베이터 무게 초과 표시"		3	No	
REQ_009	"시뮬레이터"	"이동 방향 표시"	"엘리베이터 이동 방향 표시"		3	No	nwec.NwecTest.testSetDestination
REQ_010	"시뮬레이터"	"층 별 디스플레이"	"층 별 디스플레이"		3	No	nwec.NwecTest.testCabinSecond...
REQ_011	"시뮬레이터"	"사람 승/하차"	"사람 승/하차"		3	No	nwec.NwecTest.testCabinSecond...
REQ_012	"시뮬레이터"	"비상 구조 요청"	"비상 구조 요청"		3	No	nwec.NwecTest.testSOScall
REQ_013	"스마트폰"	"대기 시간 표현"	"스마트폰 대기 시간 표현"		3	No	
REQ_014	"스마트폰"	"엘리베이터 위치 ..."	"현재 엘리베이터 위치 표시"		3	No	
REQ_015	"스마트폰"	"버튼 조작"	"스마트폰 버튼 조작하여 엘리베이터 호..."		3	No	
REQ_005	"시뮬레이터"	"호출"	"N층에서 엘리베이터 호출"		3	No	
REQ_003	"엘리베이터"	"대기 시간 계산"	"엘리베이터 대기 시간 계산"		3	No	
REQ_004	"엘리베이터"	"무게 초과 감지"	"엘리베이터 무게 초과 감지"		3	No	
REQ_006	"시뮬레이터"	"층 선택"	"내부에서 층 선택"		3	No	nwec.NwecTest.testCabinSecond...
REQ_007	"시뮬레이터"	"Status 표현"	"시뮬레이터 Status 표현"		3	No	

# 5. JFeature (11/13)

- Jfeature 에러 해결
- 한글 깨짐

The screenshot shows the JFeature application window with the 'Requirement Coverage Report' open. The interface includes a sidebar with 'All Categories' and 'All Requirements', a main report area with a summary bar and details table, and a summary table on the right.

### Requirement Coverage Report

#### Requirement Coverage Summary

Summary (15)	7 (46.67%)	8 (53.33%)
--------------	------------	------------

#### Requirement Coverage Details

Br#	Category	Coverage
1.	"?설-?출력???" (4)	2 (50%) / 2 (50%)
2.	"?사-?출력?" (3)	3 (100%)
3.	"?설??출력?" (8)	5 (62.5%) / 3 (37.5%)

Number of Requirements	15
Unique Test Methods	6
Requirements:Test Methods Ratio	2:1
Missing Test Methods	None
Unmapped Test Methods	7

Report generated on 2014 00:27:36 KST

Powered By JFeature

**All Categories**

- "?설-?출력???" (50%)
- "?사-?출력?" (0%)
- "?설??출력?" (62.5%)

**All Requirements**

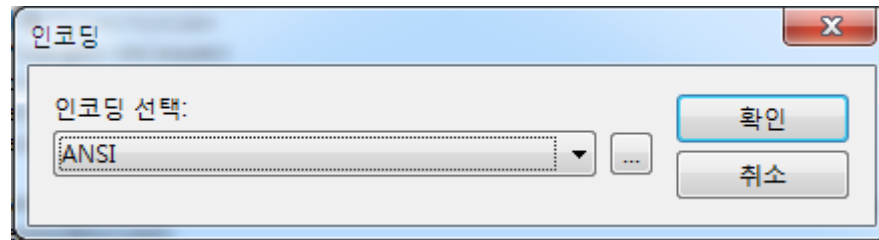
- "?설-?출력???" (100%)
- "?설-?출력???" (100%)
- "?설-?출력???" (0%)
- "?설-?출력???" (100%)
- "?설-?출력???" (100%)
- "?설-?출력???" (100%)
- "?사-?출력???" (0%)
- "?설-?출력???" (0%)
- "?사-?출력???" (0%)
- "N?출력???" (0%)
- "?설-?출력???" (0%)
- "?설-?출력???" (0%)
- "?설-?출력???" (0%)
- "?대??출력?" (100%)
- "?설??출력?" (0%)



# 5. JFeature (12/13)

---

- Jfeature 에러 해결
- 인코딩 방법을 UTF-8 -> ANSI로 변환



# 5. JFeature (13/13)

## □ Jfeature 에러 해결

**Requirement Coverage Report**

**Requirement Coverage Summary**

Summary (15)	7 (46.67%)	8 (53.33%)
--------------	------------	------------

**Requirement Coverage Details**

Sr#	Category	Coverage
1.	엘리베이터 (4)	2 (50%)
2.	스마트폰 (3)	3 (100%)
3.	시뮬레이터 (8)	5 (62.5%)

Report generated on 2014-05-29 23:29:31 KST

Powered By **JF JFeature**

**Summary Table:**

Number of Requirements	15
Unique Test Methods	6
Requirements:Test Methods Ratio	2:1
Missing Test Methods	None
Unmapped Test Methods	7

**All Categories**

- 엘리베이터 (50%)
- 스마트폰 (0%)
- 시뮬레이터 (62.5%)

**All Requirements**

- 엘리베이터 이동 (100%)
- 엘리베이터 비상 중지 (100%)
- 엘리베이터 무게 초과 표시 (0%)
- 엘리베이터 이동 방향 표시 (100%)
- 층별 디스플레이 (100%)
- 사람 승/하차 (100%)
- 비상 구조 요청 (100%)
- 스마트폰 대기 시간 표현 (0%)
- 현재 엘리베이터 위치 표시 (0%)
- 스마트폰 버튼 조작하여 엘리베... (0%)
- N층에서 엘리베이터 호출 (0%)
- 엘리베이터 대기 시간 계산 (0%)
- 엘리베이터 무게 초과 방지 (0%)
- 내부에서 층 선택 (100%)
- 시뮬레이터 Status 표현 (0%)

## 6. Clover (1/8)

---

- Atlassian이라는 애자일 지원 도구 전문업체가 개발
- 소스코드에 직접 계측코드를 삽입하여 Code Coverage를 측정
- 분석 결과 중 하나로 클래스의 복잡도와 Code Coverage 분석 결과를 함께 보여주는 그래프

# 6. Clover (2/8)

## □ 초기 화면

Your Clover evaluation license expires in 29 days.

Show: All classes

Elem	Cov%	Av Me Cpx	Cpx
2014SMA	0.0%	2.2	274.0
nwec	0.0%	2.1	218.0
Cabin.java	0.0%	2.9	109.0
ElevatorControl.java	0.0%	3.2	13.0
Floor.java	0.0%	1.0	16.0
Human.java	0.0%	1.2	11.0
NwecTest.java	0.0%	1.3	19.0
Simulator.java	0.0%	2.8	42.0
SmartPhone.java	0.0%	1.0	8.0
window	0.0%	2.8	56.0
MainView.java	0.0%	3.9	31.0
ReservationView.java	0.0%	3.2	13.0
SimulatorLogView.java	0.0%	1.0	3.0
SmartPhoneView.java	0.0%	1.8	9.0

**Metrics for: 2014SMA**

Structure	Test Executions
Packages: 2	Executed Tests: 0
Files: 11	Passes: 0
Classes: 12	Fails: 0
Methods: 124	Errors: 0
Statements: 708	
Branches: 240	

**Source**

LOC:	2,099	NC LOC:	1,338
Total Cmp:	274	Cmp Density:	0.4
Avg Method Cmp:	2.2		

# 6. Clover (3/8)

## □ Coverage Explorer – 프로젝트의 통계

Your Clover evaluation license expires in 29 days.

Show: All classes

Elem	Cov%	Av Me Cpx	Cpx
2014SMA	45.7%	2.2	274.0
└─ nwec	65.3%	2.1	218.0
└─ Cabin.java	44.6%	2.9	109.0
└─ ElevatorControl.java	89.7%	3.2	13.0
└─ Floor.java	73.0%	1.0	16.0
└─ Human.java	71.9%	1.2	11.0
└─ NwecTest.java	100.0%	1.3	19.0
└─ Simulator.java	84.2%	2.8	42.0
└─ SmartPhone.java	12.5%	1.0	8.0
└─ window	1.8%	2.8	56.0
└─ MainView.java	0.0%	3.9	31.0
└─ ReservationView.java	0.0%	3.2	13.0
└─ SimulatorLogView.java	100.0%	1.0	3.0
└─ SmartPhoneView.java	0.0%	1.8	9.0

Settings

### Metrics for: 2014SMA

Structure	Test Executions
Packages: 2	Executed Tests: 52
Files: 11	Passes: 51
Classes: 12	Fails: 1
Methods: 124	Errors: 0
Statements: 708	
Branches: 240	

### Source

LOC:	2,099	NC LOC:	1,338
Total Cmp:	274	Cmp Density:	0.4
Avg Method Cmp:	2.2		

# 6. Clover (4/8)

```
112 public Cabin(){
112     movingState = 0;
112     currentFloor = 0;
112     sosState = false;
112     humanNum = 0;
112     overWeight = false;
112     cabinRoute1 = new ArrayList<Integer>();
112     cabinRoute2 = new ArrayList<Integer>();
112     cabinRoute3 = new ArrayList<Integer>();
112     state = false;
112     direction=0;
112     smartphoneCall = false;
112 }

210 public int getCurFloor() {
210     return currentFloor;
210 }

0 public void setFloor(int floor) {
0     this.currentFloor = floor;
0 }

0 public boolean isSOS() {
0     // SOS = elevator.getEmergency();
0     return sosState;
0 }

0 public void setSOS() {
0     // elevator.setEmergency();
0     sosState = true;
0 }

28 public int getHumanNum() {
28     return humanNum;
28 }
```

- Java Editor
- 초록색 : Test를 합격 하거나 Main Method처럼 외부 Test를 하는 라인
- 노란색 : 실패한 Test Coverage
- 회색 : 제외된 코드
- 빨간색 : Coverage가 없는 Code
- 물결표시 붉은선 : 부분 분기 Coverage

# 6. Clover (5/8)

## □ Coverage Dashboard - 프로젝트에 대한 개요



# 6. Clover (6/8)

## □ Test Run Explorer – 최근에 실행된 테스트를 탐색

The screenshot displays the Test Run Explorer and Coverage Contribution views in an IDE. The Test Run Explorer shows a list of tests with their status (PASS or FAIL), start time, and duration. The Coverage Contribution view shows the percentage of code covered by the tests for various classes and methods.

**Tests run:**

Test	Start	Rslt	Time	Msg
NwecTest.testGetCabinList()	4. 5. 30 오전 3:19	PASS	0.0s	
NwecTest.testGetCabinList()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetCabinList()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetCabinList()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetFloorCount()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetFloorCount()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetFloorCount()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetFloorCount()	4. 5. 30 오전 3:19	PASS	0.0s	
NwecTest.testGetFloorList()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetFloorList()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetFloorList()	4. 5. 30 오전 3:19	PASS	0.0s	
NwecTest.testGetFloorList()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetSimulatorInstance()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetSimulatorInstance()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetSimulatorInstance()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testGetSimulatorInstance()	4. 5. 30 오전 3:19	PASS	0.0s	
NwecTest.testHumanNumInCabin()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testHumanNumInCabin()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testHumanNumInCabin()	4. 5. 30 오전 3:21	FAIL	0.002s	expected: <1> but was: <0>
NwecTest.testHumanNumInCabin()	4. 5. 30 오전 3:19	PASS	0.001s	
NwecTest.testHumanNumInFloor()	4. 5. 30 오전 3:21	PASS	0.001s	
NwecTest.testHumanNumInFloor()	4. 5. 30 오전 3:21	PASS	0.001s	
NwecTest.testHumanNumInFloor()	4. 5. 30 오전 3:21	PASS	0.001s	
NwecTest.testHumanNumInFloor()	4. 5. 30 오전 3:19	PASS	0.0s	
NwecTest.testSetDestination()	4. 5. 30 오전 3:21	PASS	0.001s	
NwecTest.testSetDestination()	4. 5. 30 오전 3:21	PASS	0.0s	
NwecTest.testSetDestination()	4. 5. 30 오전 3:21	PASS	0.002s	
NwecTest.testSetDestination()	4. 5. 30 오전 3:19	PASS	0.001s	

**Coverage Contribution:**

Class	Contrib%	Uniq%
Cabin	28.7%	0.0%
closeCabin	100.0%	0.0%
getCainDirection	25.0%	0.0%
getCurFloor	100.0%	0.0%
getDestination	53.8%	0.0%
getDirection	100.0%	0.0%
getHumanNum	100.0%	0.0%
getState	100.0%	0.0%
getWaitingTime	22.4%	0.0%
goToFloor	32.1%	0.0%
moving	58.1%	0.0%
openCabin	87.5%	0.0%
ElevatorControl	79.3%	0.0%
getCabinList	100.0%	0.0%
goToFloor	100.0%	0.0%
searchFloor	62.5%	0.0%
Floor	16.2%	0.0%
getHumanNum	100.0%	0.0%
getWaitingNum	100.0%	0.0%
setHumanNum	100.0%	0.0%
Human	25.0%	0.0%
Human	100.0%	0.0%
waitElevator	60.0%	0.0%
Simulator	42.1%	2.7%
callCabin	60.6%	9.1%
goToFloor	66.7%	0.0%
update	63.9%	0.0%



# 6. Clover (7/8)

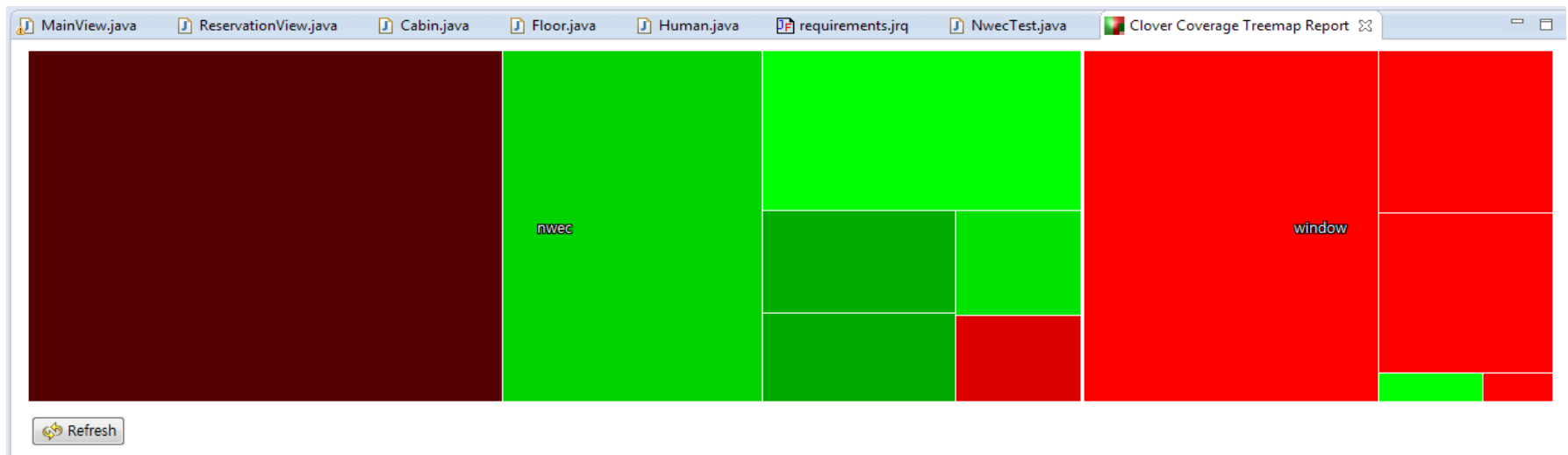
## □ Test Contributions – Unit Test와 선택한 자바소스파일에 대한 Coverage 확인

The screenshot displays the Clover Dash... window in an IDE, showing test contribution results. The interface is split into two panes:

- All contributing tests:** A tree view showing the following tests:
  - testCabinSecondCall() - 14. 5. 30 오전 3:21 (4 instances)
  - testSOScall() - 14. 5. 30 오전 3:19 (1 instance)
  - testSOScall() - 14. 5. 30 오전 3:21 (2 instances)
  - testHumanNumInCabin() - 14. 5. 30 (5 instances)
  - testHumanNumInFloor() - 14. 5. 30 (3 instances)
  - testDestination() - 14. 5. 30 오전 3:1 (2 instances)
  - testDestination() - 14. 5. 30 오전 3:2 (3 instances)
  - testSetDestination() - 14. 5. 30 오전 (4 instances)
  - testFloorInit() - 14. 5. 30 오전 3:21 (2 instances)
- Contributing tests at cursor:** A tree view showing the selected test method:
  - testHumanNumInCabin() - 14. 5. 30 오전 3: (4 instances)

# 6. Clover (8/8)

## □ Coverage Treemap Report



- 복잡성과 Code Coverage에 의한 Class와 Package를 동시에 비교하여 보여준다.
- 크기 = 복잡성
- 색상 = Coverage의 수준 (초록 : 잘 되었음, 빨강 : 잘 안되었음)

---

# Thanks

---