SafeHome

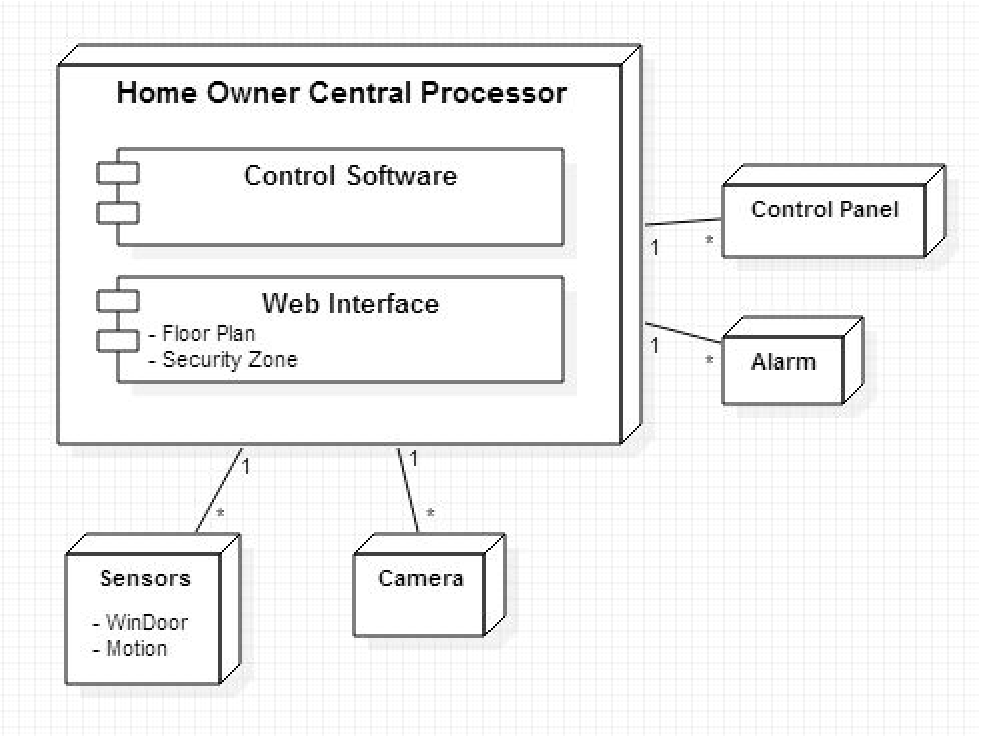
TEAM 5
Sumin Han
Joonhoo Oh

FINAL IMPLEMENTATION

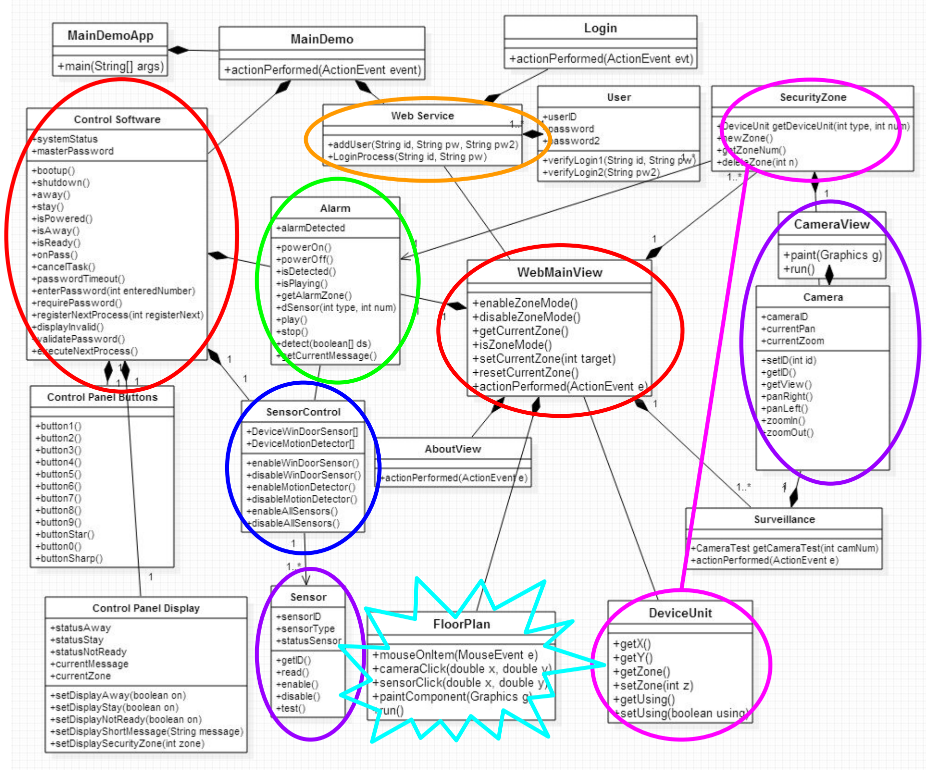
INDEX

1. Deployment Diagram
2. Class Diagram
3. Code Statistics
4. Testing Method (a. JUnit, b. Integration)
5. Testing Result (a. JUnit, b. Integration)

1. Deployment Diagram



2. Class Diagram



3. Code Statistics

LOC(Lines Of Code): 1,539 *...(+528 MainDemoTest.java)*

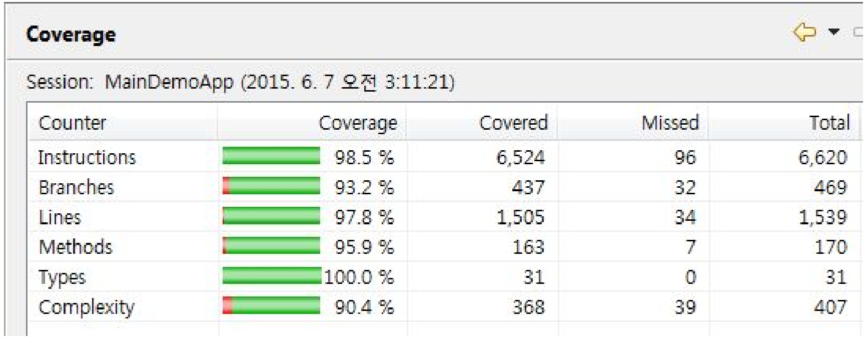
# of files: 23

# of branches: 437

# of classes: 24

# of methods: 170

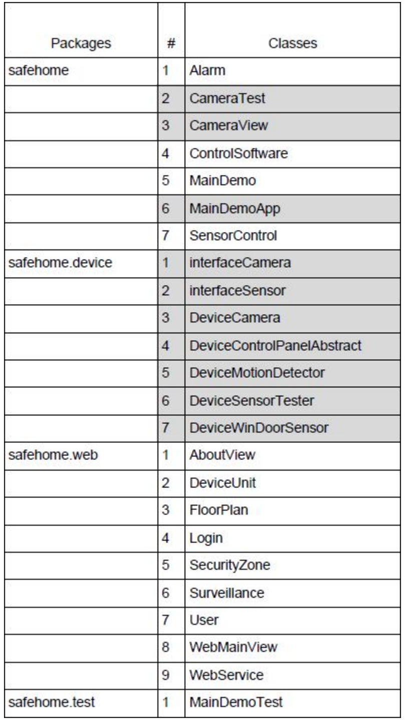
cyclometic complexities of methods: 368



Coverage

Session: MainDemoApp (2015. 6. 7 오전 3:11:21)

Counter	Coverage	Covered	Missed	Total
Instructions	98.5 %	6,524	96	6,620
Branches	93.2 %	437	32	469
Lines	97.8 %	1,505	34	1,539
Methods	95.9 %	163	7	170
Types	100.0 %	31	0	31
Complexity	90.4 %	368	39	407



Packages	#	Classes
safehome	1	Alarm
	2	CameraTest
	3	CameraView
	4	ControlSoftware
	5	MainDemo
	6	MainDemoApp
	7	SensorControl
safehome.device	1	interfaceCamera
	2	interfaceSensor
	3	DeviceCamera
	4	DeviceControlPanelAbstract
	5	DeviceMotionDetector
	6	DeviceSensorTester
	7	DeviceWinDoorSensor
safehome.web	1	AboutView
	2	DeviceUnit
	3	FloorPlan
	4	Login
	5	SecurityZone
	6	Surveillance
	7	User
	8	WebMainView
	9	WebService
safehome.test	1	MainDemoTest

4. Testing Method

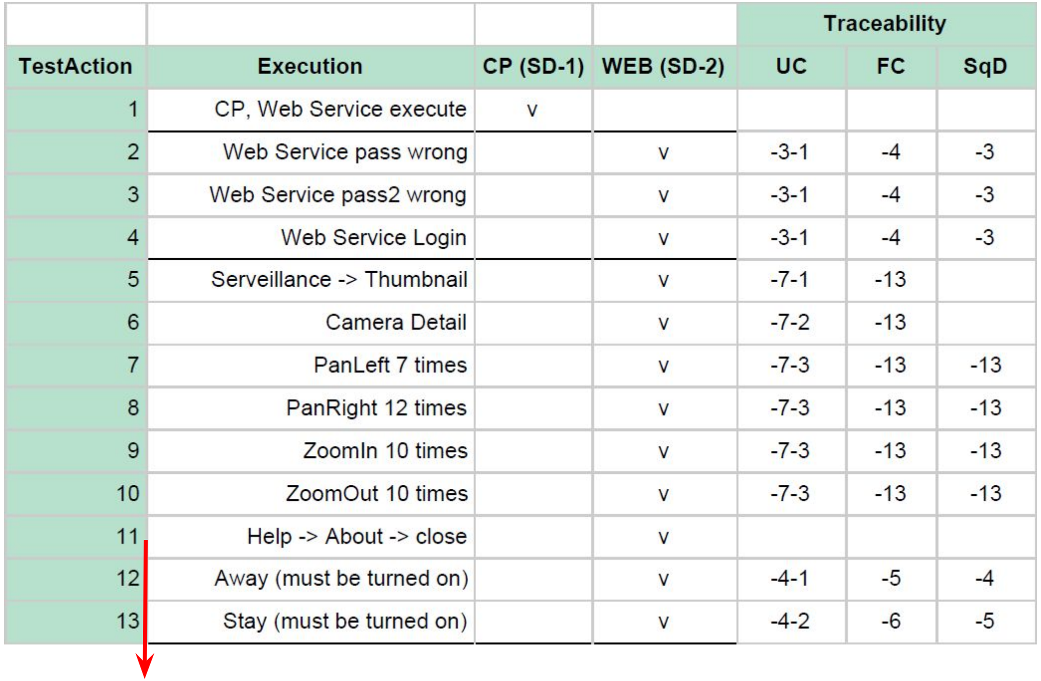
Sequence of execution:

...78 test action

3.b. Integration Testing

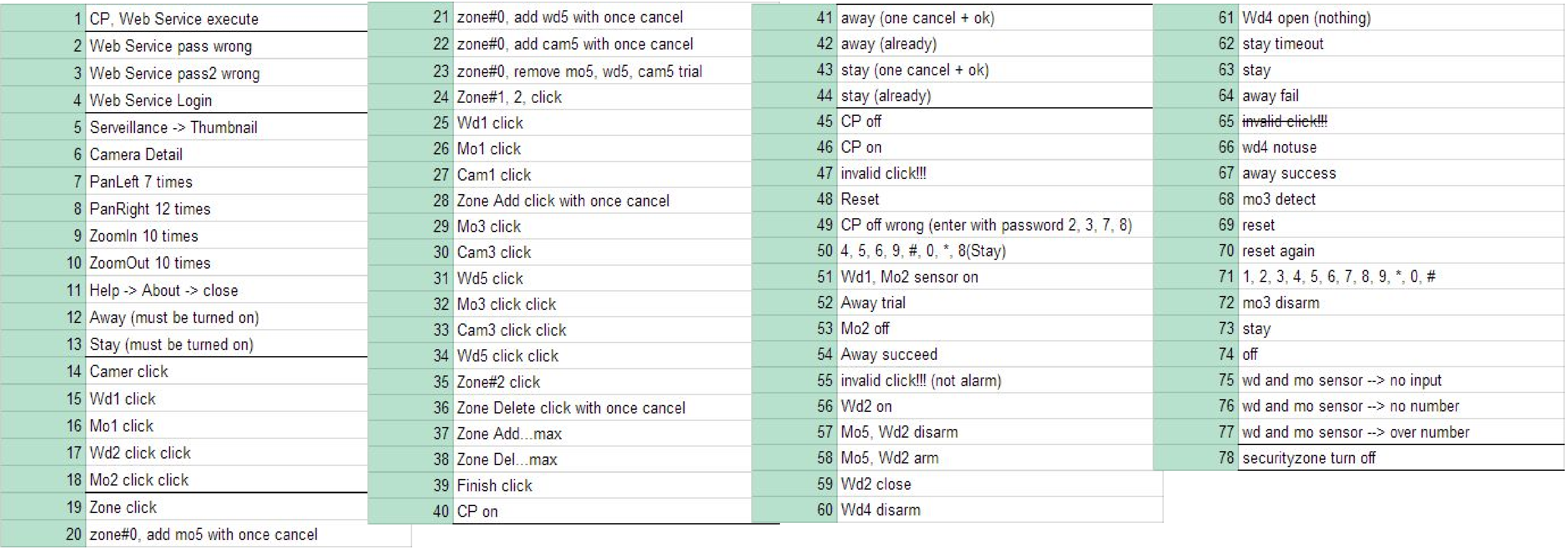
follows these test action

(Cyclometic Complexity (90.4%))



				Traceability		
TestAction	Execution	CP (SD-1)	WEB (SD-2)	UC	FC	SqD
1	CP, Web Service execute	v				
2	Web Service pass wrong		v	-3-1	-4	-3
3	Web Service pass2 wrong		v	-3-1	-4	-3
4	Web Service Login		v	-3-1	-4	-3
5	Serveillance -> Thumbnail		v	-7-1	-13	
6	Camera Detail		v	-7-2	-13	
7	PanLeft 7 times		v	-7-3	-13	-13
8	PanRight 12 times		v	-7-3	-13	-13
9	ZoomIn 10 times		v	-7-3	-13	-13
10	ZoomOut 10 times		v	-7-3	-13	-13
11	Help -> About -> close		v			
12	Away (must be turned on)		v	-4-1	-5	-4
13	Stay (must be turned on)		v	-4-2	-6	-5

4. Testing Method

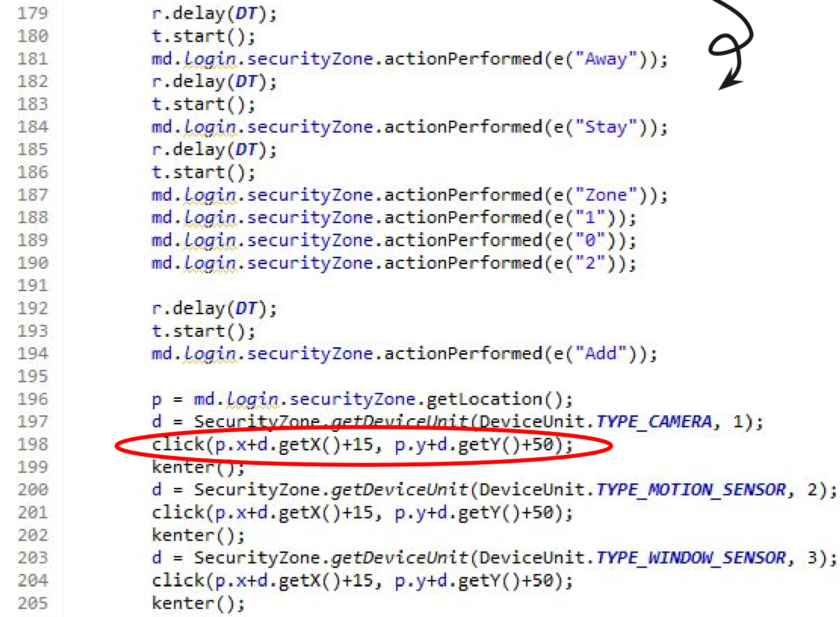


1	CP, Web Service execute
2	Web Service pass wrong
3	Web Service pass2 wrong
4	Web Service Login
5	Serveillance -> Thumbnail
6	Camera Detail
7	PanLeft 7 times
8	PanRight 12 times
9	ZoomIn 10 times
10	ZoomOut 10 times
11	Help -> About -> close
12	Away (must be turned on)
13	Stay (must be turned on)
14	Camer click
15	Wd1 click
16	Mo1 click
17	Wd2 click click
18	Mo2 click click
19	Zone click
20	zone#0, add mo5 with once cancel
21	zone#0, add wd5 with once cancel
22	zone#0, add cam5 with once cancel
23	zone#0, remove mo5, wd5, cam5 trial
24	Zone#1, 2, click
25	Wd1 click
26	Mo1 click
27	Cam1 click
28	Zone Add click with once cancel
29	Mo3 click
30	Cam3 click
31	Wd5 click
32	Mo3 click click
33	Cam3 click click
34	Wd5 click click
35	Zone#2 click
36	Zone Delete click with once cancel
37	Zone Add...max
38	Zone Del...max
39	Finish click
40	CP on
41	away (one cancel + ok)
42	away (already)
43	stay (one cancel + ok)
44	stay (already)
45	CP off
46	CP on
47	invalid click!!!
48	Reset
49	CP off wrong (enter with password 2, 3, 7, 8)
50	4, 5, 6, 9, #, 0, *, 8(Stay)
51	Wd1, Mo2 sensor on
52	Away trial
53	Mo2 off
54	Away succeed
55	invalid click!!! (not alarm)
56	Wd2 on
57	Mo5, Wd2 disarm
58	Mo5, Wd2 arm
59	Wd2 close
60	Wd4 disarm
61	Wd4 open (nothing)
62	stay timeout
63	stay
64	away fail
65	~~invalid click!!!~~
66	wd4 notuse
67	away success
68	mo3 detect
69	reset
70	reset again
71	1, 2, 3, 4, 5, 6, 7, 8, 9, *, 0, #
72	mo3 disarm
73	stay
74	off
75	wd and mo sensor --> no input
76	wd and mo sensor --> no number
77	wd and mo sensor --> over number
78	securityzone turn off

4.a. JUnit Testing

*GUI Testing: actionPerfomed, click, key enter ...

Ex1: away, stay, zone button, edit device to zone

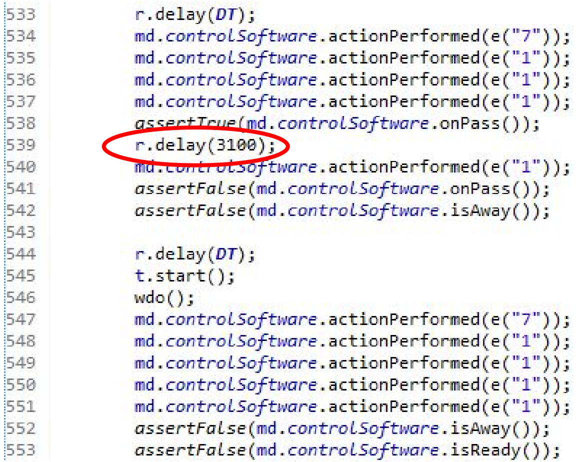


```
r.delay(DT);
t.start();
md.login.securityZone.actionPerformed(e("Away"));
r.delay(DT);
t.start();
md.login.securityZone.actionPerformed(e("Stay"));
r.delay(DT);
t.start();
md.login.securityZone.actionPerformed(e("Zone"));
md.login.securityZone.actionPerformed(e("1"));
md.login.securityZone.actionPerformed(e("0"));
md.login.securityZone.actionPerformed(e("2"));

r.delay(DT);
t.start();
md.login.securityZone.actionPerformed(e("Add"));

p = md.login.securityZone.getLocation();
d = SecurityZone.getDeviceUnit(DeviceUnit.TYPE_CAMERA, 1);
click(p.x+d.getX()+15, p.y+d.getY()+50);
kenter();
d = SecurityZone.getDeviceUnit(DeviceUnit.TYPE_MOTION_SENSOR, 2);
click(p.x+d.getX()+15, p.y+d.getY()+50);
kenter();
d = SecurityZone.getDeviceUnit(DeviceUnit.TYPE_WINDOW_SENSOR, 3);
click(p.x+d.getX()+15, p.y+d.getY()+50);
kenter();
```

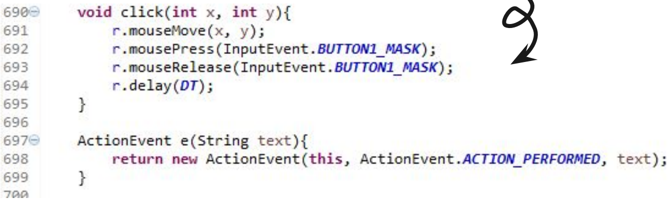
Ex2: away, stay, delay, windoor sensor open, ...



```
r.delay(DT);
md.controlSoftware.actionPerformed(e("7"));
md.controlSoftware.actionPerformed(e("1"));
md.controlSoftware.actionPerformed(e("1"));
md.controlSoftware.actionPerformed(e("1"));
assertTrue(md.controlSoftware.onPass());
r.delay(3100);
md.controlSoftware.actionPerformed(e("1"));
assertFalse(md.controlSoftware.onPass());
assertFalse(md.controlSoftware.isAway());

r.delay(DT);
t.start();
wdo();
md.controlSoftware.actionPerformed(e("7"));
md.controlSoftware.actionPerformed(e("1"));
md.controlSoftware.actionPerformed(e("1"));
md.controlSoftware.actionPerformed(e("1"));
md.controlSoftware.actionPerformed(e("1"));
assertFalse(md.controlSoftware.isAway());
assertFalse(md.controlSoftware.isReady());
```

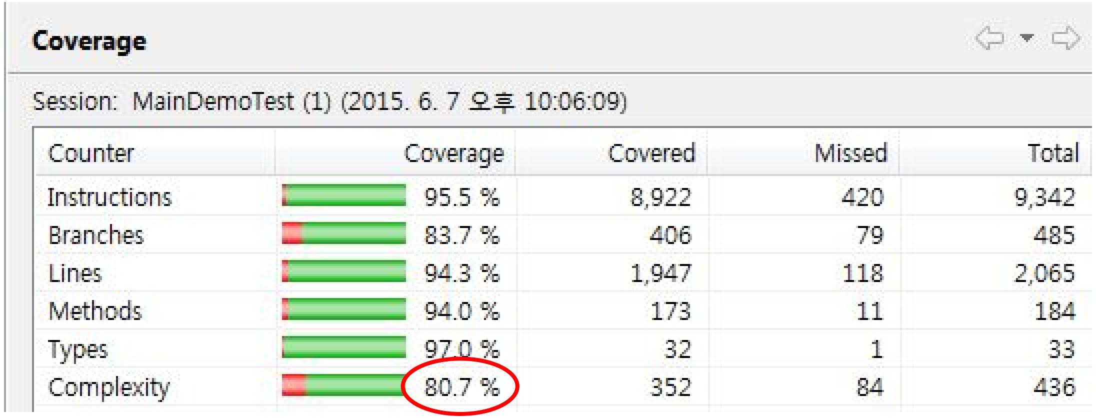
Examples of GUI testing methods: click, e (ActionEvent)



```
void click(int x, int y){
    r.mouseMove(x, y);
    r.mousePress(InputEvent.BUTTON1_MASK);
    r.mouseRelease(InputEvent.BUTTON1_MASK);
    r.delay(DT);
}

ActionEvent e(String text){
    return new ActionEvent(this, ActionEvent.ACTION_PERFORMED, text);
}
```

5.a. JUnit- Result

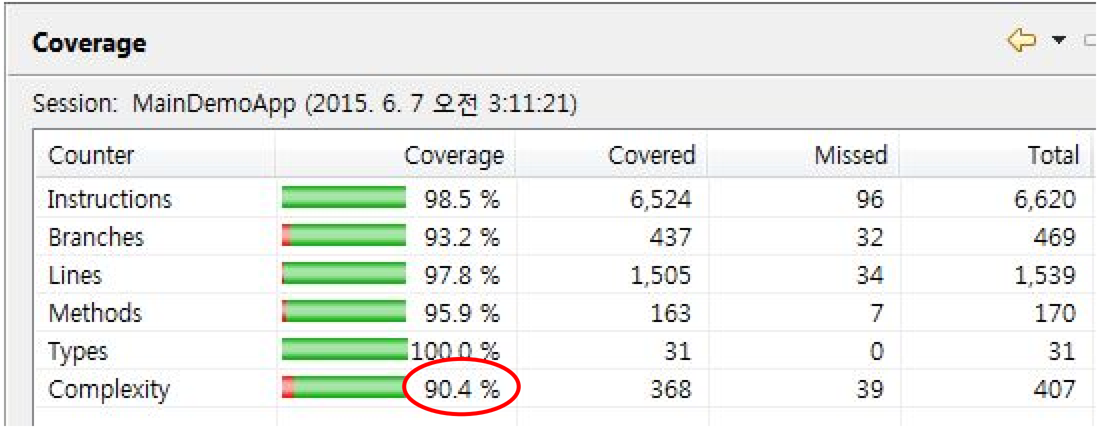


Coverage

Session: MainDemoTest (1) (2015. 6. 7 오후 10:06:09)

Counter	Coverage	Covered	Missed	Total
Instructions	95.5 %	8,922	420	9,342
Branches	83.7 %	406	79	485
Lines	94.3 %	1,947	118	2,065
Methods	94.0 %	173	11	184
Types	97.0 %	32	1	33
Complexity	80.7 %	352	84	436

5.b. Integration- Result



Coverage

Session: MainDemoApp (2015. 6. 7 오전 3:11:21)

Counter	Coverage	Covered	Missed	Total
Instructions	98.5 %	6,524	96	6,620
Branches	93.2 %	437	32	469
Lines	97.8 %	1,505	34	1,539
Methods	95.9 %	163	7	170
Types	100.0 %	31	0	31
Complexity	90.4 %	368	39	407