

Gusio(거시오) Digital Watch

OOPT Stage 2050, 2060

Project Team

Team 2

Date

2019-05-27

Member

201511282 이재승

201511291 장유준

201511303 최원경

201511296 조현근

201511293 전상우

목 차

Activity 2055. Write Unit Test Code

Activity 2061. Unit Testing

Activity 2063. System Testing

Activity 2055. Write Unit Test Code

1. Alarm Test

```
public class AlarmTest {  
    @Test(timeout=1000)  
    public void getInform() {  
        Time t=new Time();  
        Alarm test=new Alarm(t);  
        int[] time = new int[3];  
        time[0] = 0;  
        time[1] = 0;  
        time[2] = 0;  
        int state = 10;  
        int []inform = new int[]{2, time[0], time[1], time[2], 0, 0, state};  
        test.getInform();  
    }  
    @Test  
    public void activate() {  
        Time t=new Time();  
        Alarm test=new Alarm(t);  
        int state = 10;  
        test.activate();  
        assertEquals(test.getState(), actual: 9);  
    }  
}
```

2. Dice Test

```
public class DiceTest {
    @Test
    public void setDice() {
        Dice test = new Dice();
        test.setDice(1);
        assertEquals(test.getDis()[3], actual: 2);
    }
    @Test(timeout=1000)
    public void testchangeDiceNum() {
        Dice test = new Dice();
        test.changeDiceNum();
    }
    @Test(timeout=1000)
    public void rollDice() {
        Dice test = new Dice();
        test.rollDice();
    }
}
```

```
@Test
public void getDis() {
    Dice test=new Dice();
    int disNum = 0;
    int isSet = 8;
    int [] dis = new int[] {4,0,0,disNum,0,0,isSet};
    assertEquals(dis,test.getDis());
}
```

3. RR Test

```
public class RRTest {  
  
    @Test  
    public void getBuzzerFlag() {  
        RR test=new RR();  
        assertNotNull(test.getBuzzerFlag());  
    }  
}
```

4. Timer Test

```
public class TimerTest {  
  
    @Test  
    public void getBuzzerFlag() {  
        Timer test=new Timer();  
        assertNotNull(test.getBuzzerFlag());  
    }  
}
```

5. Time Test

```
public class TimeTest {  
  
    @Test  
    public void getSystemClock() {  
        Time test=new Time();  
        assertNotNull(test.getSystemClock());  
    }  
  
}
```

6. GUI Test

```
public class GUITest {  
  
    @Test  
    public void GUI_btn() {  
        GUI test=new GUI();  
        assertNotNull(test.GUI_btn());  
    }  
  
}
```

7. Select_Mode Test

```
public class Select_ModeTest {  
  
    @Test  
    public void run() {  
        Select_Mode test=new Select_Mode();  
        assertEquals(test.run( input: 5), actual: 0);  
    }  
  
    @Test  
    public void checkMode() {  
        Select_Mode test=new Select_Mode();  
        assertNotNull(test.checkMode( n: 2));  
    }  
}
```

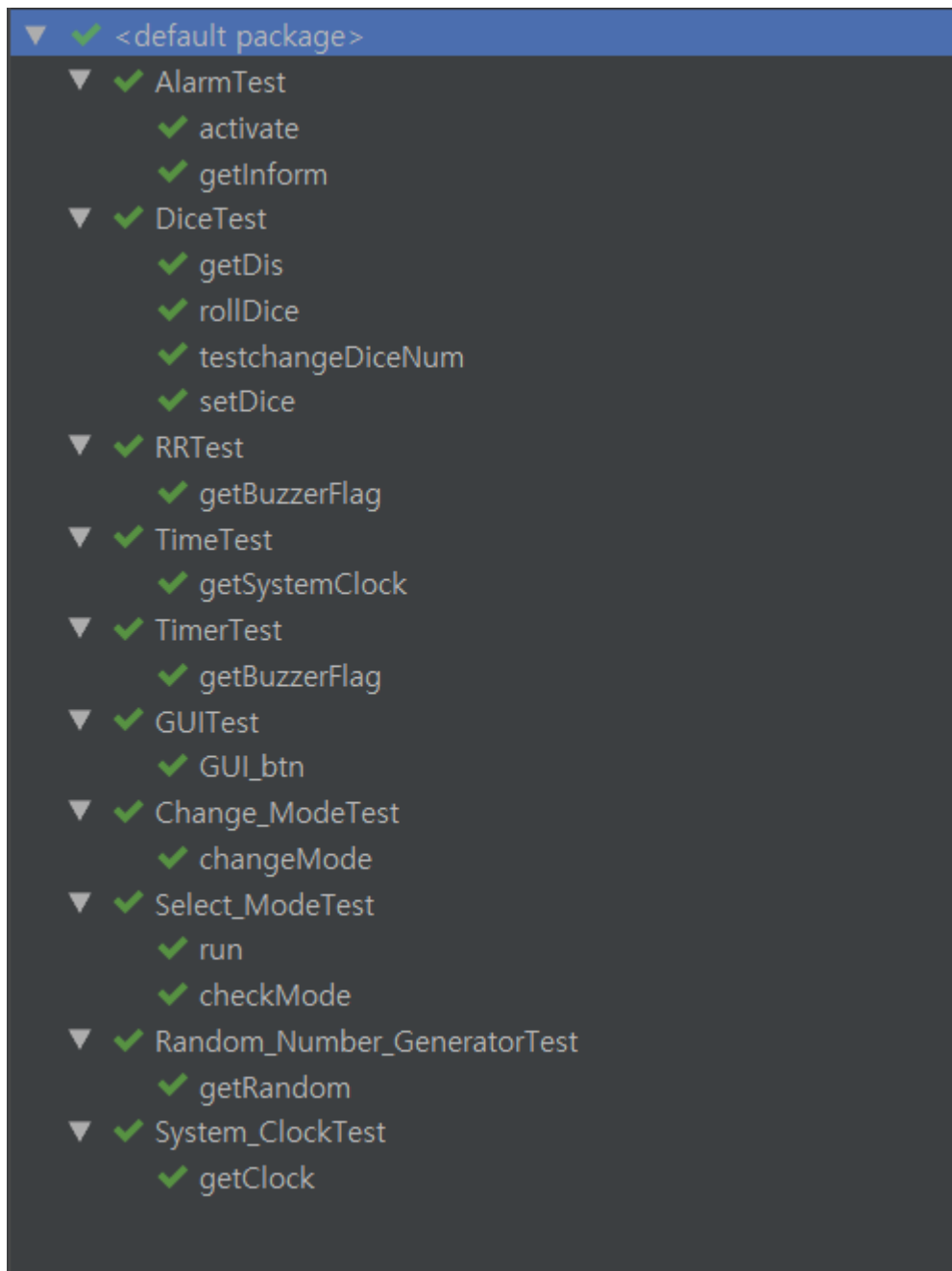
8. Random_Number_Generator Test

```
public class Random_Number_GeneratorTest {  
  
    @Test  
    public void getRandom() {  
        Random_Number_Generator test=new Random_Number_Generator();  
        assertNotNull(test.getRandom());  
    }  
}
```

9. System_Clock Test

```
public class System_ClockTest {  
  
    @Test  
    public void getClock() {  
        System_Clock test=new System_Clock( m: 0, d: 0, d2: 0, h: 0, m2: 0, s: 0);  
        assertNotNull(test.getClock());  
    }  
}
```


Activity 2061. Unit Testing



Activity 2063. System Testing

Test Num	Test 항목	Description	Use case	System Function
1.1	시간 표시 실험	시간이 1 초씩 증가하는지 TEST	1. showTime	R 1.1
1.2	시간 전환 실험	시간이 초->분->시로 알맞게 증가하는지 TEST	1. showTime	R 1.1
1.3	시간 설정 실험	시간을 설정할 수 있는지 TEST	2. setTime	R 1.2
2.1	스톱워치 표시 실험	스톱워치의 상태가 표시되는지 TEST	3. showSW	R 2.1
2.2	스톱워치 시작 실험	스톱워치가 시작하는지 TEST	4. startSW	R 2.2
2.3	스톱워치 시간 전환 실험	시간이 초->분->시로 알맞게 증가하는지 TEST	4. startSW	R 2.2
2.4	시간 저장 실험	스톱워치의 현재 시간이 저장되었는지 TEST	5. saveRecords	R 2.3
2.5	저장 시간 표시 실험	저장된 시간이 보여지는지 TEST	5. saveRecords	R 2.3
2.6	스톱워치 정지 실험	스톱워치가 정지하는지 TEST	6. stopSW	R 2.4
2.7	스톱워치 초기화 실험	스톱워치가 0 으로 초기화되는지 TEST	7. resetSW	R 2.5
2.8	저장 시간 초기화 실험	저장된 기록도 초기화되는지 TEST	7. resetSW	R 2.5
3.1	알람 표시 실험	알람의 현재 상태가 표시되는지 TEST	8. showAlarm	R 3.1
3.2	알람 선택 실험	저장된 알람을 보여주는지 TEST	9. selectAlarm	R 3.2
3.3	알람 기본값	저장된 기본값이	9. selectAlarm	R 3.2

	실험	0 시 0 분 0 초인지 TEST		
3.4	알람 설정 실험	알람을 설정할 수 있는지 TEST	10. setAlarm	R 3.3
3.5	알람 벨 실험	설정된 알람의 시간에 벨이 울리는지 TEST	11. ringAlarm	R 3.4
3.6	알람 벨 종료 실험	알람의 벨이 울리고 있을 때 종료할 수 있는지 TEST	12. stopAlarm	R 3.5
4.1	타이머 표시 실험	타이머의 현재 상태가 표시되는지 TEST	13. showTimer	R 4.1
4.2	타이머 설정 실험	타이머를 설정할 수 있는지 TEST	14. setTimer	R 4.2
4.3	타이머 시작 실험	타이머가 시작되는지 TEST	15. startTimer	R 4.3
4.4	타이머 시간 전환 실험	시간이 초->분->시로 알맞게 증가하는지 TEST	15. startTimer	R 4.3
4.5	타이머 정지 실험	타이머가 정지하는지 TEST	16. stopTimer	R 4.4
4.6	타이머 초기화 실험	타이머가 0 으로 초기화되는지 TEST	17. resetTimer	R 4.5
4.7	타이머 벨 실험	타이머가 0 이 되었을 때 벨이 울리는지 TEST	18. ringTimer	R 4.6
5.1	주사위 표시 실험	주사위 게임의 현재상태가 표시되는지 TEST	19. showDice	R 5.1
5.2	주사위 설정 실험	주사위의 개수를 설정할 수 있는지 TEST	20. setDice	R 5.2
5.3	주사위 개수 실험	주사위의 개수가 1 과 2 만 가능한지 TEST	20. setDice	R 5.2
5.4	주사위 게임 실험	주사위의 랜덤 값이 0~ 주사위개수*6 사이인지 TEST	21. rollDice	R 5.3
6.1	룰렛 게임 표시 실험	룰렛 게임의 현재 상태가 표시되는지 TEST	22. showRR	R 6.1

6.2	룰렛 설정 실험	룰렛 게임의 확률을 설정할 수 있는지 TEST	23. setRR	R 6.2
6.3	룰렛 확률 범위 실험	룰렛 게임의 확률이 2~10 분의 1 인지 TEST	23. setRR	R 6.2
6.4	룰렛 게임 실험	룰렛 게임의 확률이 1/인원수인지 TEST	24. playRR	R 6.3
6.5	룰렛 벨 실험	꽁이 나오면 벨이 울리는지 TEST	25. ringRR	R 6.4
7.1	메뉴 선택 실험	4 가지 메뉴를 고를 수 있는지 TEST	26. selectMenu	R 7.1
7.2	메뉴 값 저장실험	메뉴의 조합을 바꿔도 값이 저장되는지 TEST	26. selectMenu	R 7.1
8.1	모드 전환 실험	menu 버튼을 누를 때마다 모드가 바뀌는지 TEST	27. changeMode	R 8.1
8.2	기능 보존 실험	기능을 실행중에 menu 버튼을 눌러도 계속 진행중인지 TEST	27. changeMode	R 8.1

Activity 2066. Testing Traceability Analysis

operation in SSD	operation in interaction diagram	Method	class	Unit Test
1: enterSetMode - 1,20,21,27,35,40	-1 enterSetMode() -> :1	:1 enterSetMode() : void	Time	timesetMode()
2: exitSetMode - 3,23,24,29,37,42	-2 timesetMode(input) -> :2	:2 timesetMode(input : Integer) : void		updateTime()
3: timesetMode- 2	-3 exitSetMode()-> :3	:3 exitSetMode() : void		getSystemClock()
4: startSW- 5,6	-4 updateTime()-> :4	:4 updateTime() : void		

5: Record- 10	-5 startSW()-> : 6	:5 getSystemClock(): int			
6: showRecords- 8,9	-6 run()-> : 43	:6 startSW() : void	Stop Watch	Record()	
7: pauseSW- 11,12	-7 updateSW()-> : 10	:7 showRecords() : void		getRecord()	
8: resetSW- 13	-8 showRecords()-> : 7	:8 getRecord() : void		updateSW()	
9: selectAlarm- 14,15,16	-9 getRecord()-> : 8	:9 Record() : void		showRecords()	
10: activate- 17,18,19	-10 Record()-> : 9	:10 updataSW() : void			
11: changeAlarm- 22	-11 pauseSW()-> : 11	:11 pauseSW() : void			
12: stopAlarm- 25,26	-12 updateSW()-> : 10	:12 resetSW() : void			
13: changeTimer- 28	-13 resetSW() -> : 12	:13 selectAlarm() : void		Alarm Manager	selectAlarm()
14: startTimer- 30,31,32	-14 selectAlarm(input)-> : 13	:14 enterSetMode() : void			
15: pauseTimer- 32	-15 selectAlarm(input)-> : 13	:15 exitSetMode() : void			
16: resetTimer- 33	-16 getInform()-> : 31	:16 enterSetMode() : void	Timer	resetTimer()	
17: changeDice- 35	-17 activate()-> : 32	:17 changeTimer(input : Integer) : void		updateTimer()	
18: rollDice- 37,38	-18 getSystemClcok()-> : 5	:18 exitSetMode() : void		changeTimer()	
19: changeRR- 40	-19 getGapTime(t1,t2)-> : 37	:19 startTimer() : void			
20: playRR- 42,43,44	-20 enterSetMode()-> : 14	:20 pauseTimer() : void			
21: selectMode- 45,46	-21 enterSetMode()-> : 33	:21 resetTimer() : void			
22: checkMode- 47	-22 changeAlarm(Button)-> : 34	:22 updateTimer() : void			
23: changeMode- 48,49	-23 exitSetMode()-> : 15	:23 enterSetMode() : void	Dice	setDice()	
	-24 exitSetMode()-> : 35	:24 changeDice(Button : Integer) : void		changeDice()	
	-25 stopAlarm()-> : 36	:25 exitSetMode() : void			
	-26 stopBuzzer()-> : 46	:26 rollDice() :void			
	-27 enterSetMode()-> : 16	:27 enterSetMode() : void	Russian Roulette	playRR()	
	-28 changeTimer(Button) -> : 17	:28 changeRR(Button : Integer) : void		setRR()	
	-29 exitSetMode()-> : 18	:29 exitSetMode() : void			

	-30 startTimer()-> :19	:30 playRR() : void		
	-31 updateTimer()-> :22	:31 getInform() : void	Alarm	stopAlarm()
	-32 pauseTimer()-> :20	:32 activate() : void		changeAlarm()
	-33 resetTimer()-> :21	:33 enterSetMode() : void		Activate()
	-34 enterSetMode()-> :23	:34 changeAlarm(Button : Integer) : void		getGapTime()
	-35 changeDice(Button) -> :24	:35 exitSetMode() : void		
	-36 exitSetMode()-> :25	:36 stopAlarm() : void		
	-37 rollDice()-> :26	:37 getGanTime(t1,t2) : int		
	-38 getRandom(DiceNum)-> :47	:38 changeMode() : void	change Mode	changeMode()
	-39 enterSetMode()-> :27	:39 nextMode() : void		
	-40 changeRR(Button) -> :28	:40 selectMode() : void	selectMenu	selectMode()
	-41 exitSetMode()-> :29	:41 nextMode() : void		nextMode()
	-42 playRR()-> :30	:42 checkMode() : void		checkMode()
	-43 setModeNum(ratio)-> :48	:43 getClock() : void	System Clock	
	-44 getRandom(ratio) -> :47	:44 run() : void		
	-45 selectMode()-> :40	:45 getClock() : void	System Clock2	
	-46 nextMode()-> :41	:46 stopBuzzer() : void	Buzzer	stopBuzzer()
	-47 checkMode()-> :42	:47 getRandom() : int	Random Number Generator	setModeNum()
	-48 changeMode()-> :38	:48 setModeNum(mNum) : void		
	-49 nextMode()-> :39			