

▶ [T8] 낚시하시계

201714170 조해성 201714168 유호원 201310507 정희찬
201613575 루카이 201712845 류한길

Table of Contents

- ▶ **Activity 2051. Implement Class & Methods Definition**
- ▶ **Activity 2052. Implements Windows**
- ▶ **Activity 2055. Write Unit Test Code**
- ▶ **Activity 2061. Unit Testing**
- ▶ **Activity 2063. System Testing**

Activity 2051

Type	Class
Name	SystemUI
Purpose	User가 Digitalwatch를 사용할 수 있게 해주는 클래스
Overview	N/A
Cross Reference	Functions : R0, R1, R2, R3, R4, R5, R6 Use Cases : "showTime", "adjustTime", "showTimer", "setTimer", "startTimer", "pauseTimer", "resetTimer", "buzzTimer", "showAlarm", "nextAlarm", "addAlarm", "deleteAlarm", "buzzAlarm", "stopAlarm", "showStopwatch", "startStopwatch", "recordStopwatch", "pauseStopwatch", "resetStopwatch", "showTide", "nextTide", "showMoon phase", "modeSelect", "nextMode"
Exceptional Course of Events	N/A

Activity 2051

Type	Class
Name	CreateMode
Purpose	System이 모드를 생성하는 클래스
Overview	N/A
Cross Reference	Functions :R6 Use Cases : "modeSelect", "nextMode"
Exceptional Course of Events	N/A

Type	Class
Name	DeleteMode
Purpose	System이 모드를 삭제하는 클래스
Overview	N/A
Cross Reference	Functions :R6 Use Cases : "modeSelect", "nextMode"
Exceptional Course of Events	N/A

Activity 2051

Type	Class
Name	ModeSelector
Purpose	User가 모드를 선택하도록 해주는 클래스
Overview	N/A
Cross Reference	Funtions :R6 Use Cases : "modeSelect", "nextMode"
Exceptional Course of Events	N/A

Type	Class
Name	TimeKeeping
Purpose	System이 현재시간을 보여주게 하는 클래스
Overview	N/A
Cross Reference	Funtions :R0 Use Cases : "showTime", "adjustTime"
Exceptional Course of Events	N/A

Activity 2051

Type	Class
Name	TimeDB
Purpose	System이 현재시간을 저장하게 만드는 클래스
Overview	N/A
Cross Reference	Functions : R0 Use Cases : "showTime", "adjustTime"
Exceptional Course of Events	N/A

Type	Class
Name	Alarm
Purpose	User가 Alarm을 사용할수 있게 해주는 클래스
Overview	N/A
Cross Reference	Functions : R2 Use Cases : "showAlarm", "nextAlarm", "addAlarm", "deleteAlarm", "buzzAlarm", "stopAlarm"
Exceptional Course of Events	N/A

Activity 2051

Type	Class
Name	Timer
Purpose	User가 Timer를 사용할 수있게 해주는 클래스
Overview	N/A
Cross Reference	Functions : R1 Use Cases : “showTimer”, “setTimer”, “startTimer”, “pauseTimer”, “resetTimer”, “buzzTimer”
Exceptional Course of Events	N/A

Type	Class
Name	Stopwatch
Purpose	User가 스탑워치를 사용할수 있게 해주는 클래스
Overview	N/A
Cross Reference	Functions :R3 Use Cases : “showStopwatch”, “startStopwatch”, “recordStopwatch”, “pauseStopwatch”, “resetStopwatch”
Exceptional Course of Events	N/A

Activity 2051

Type	Class
Name	Tide
Purpose	User가 Tide를 확인할수 있게 해주는 클래스
Overview	N/A
Cross Reference	Functions : R4 Use Cases : "showTide", "nextTide", "calculateTide"
Exceptional Course of Events	N/A

Type	Class
Name	Moonphase
Purpose	User가 Moonphase를 확인할수있게 해주는 클래스
Overview	N/A
Cross Reference	Functions : R5 Use Cases : "showMoonphase", "CalculateMoonphase"
Exceptional Course of Events	N/A

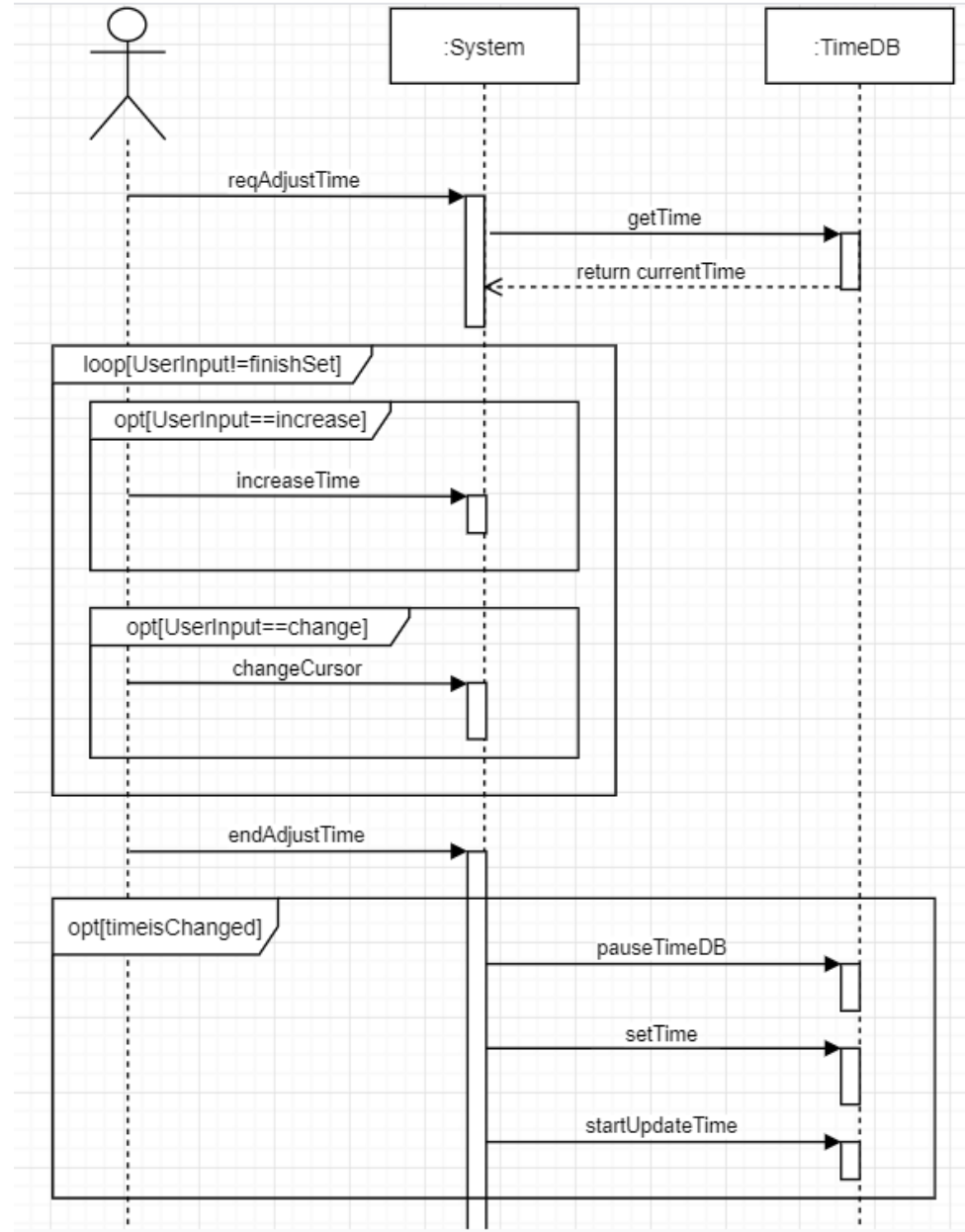
Activity 2052

Name	adjustTime
Responsibilities	TimeKeeping모드의 “adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	adjustTime으로 진입한다
Pre-conditions	TimeKeeping모드여야 한다
Post-Conditions	N/A

Name	increaseTime
Responsibilities	adjustTime에서 “adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	사용자가 원하는 만큼 시간을 증가시킨다
Pre-conditions	adjustTime을 진입해야 한다
Post-Conditions	ChangeCurser로 진행할수 있다

Name	ChangeCurser
Responsibilities	adjustTime에서 “mode”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	다음 커서로 바꾼다
Pre-conditions	adjustTime을 진입해야 한다
Post-Conditions	현재 시간을 증가시킬수 있다

Activity 2052



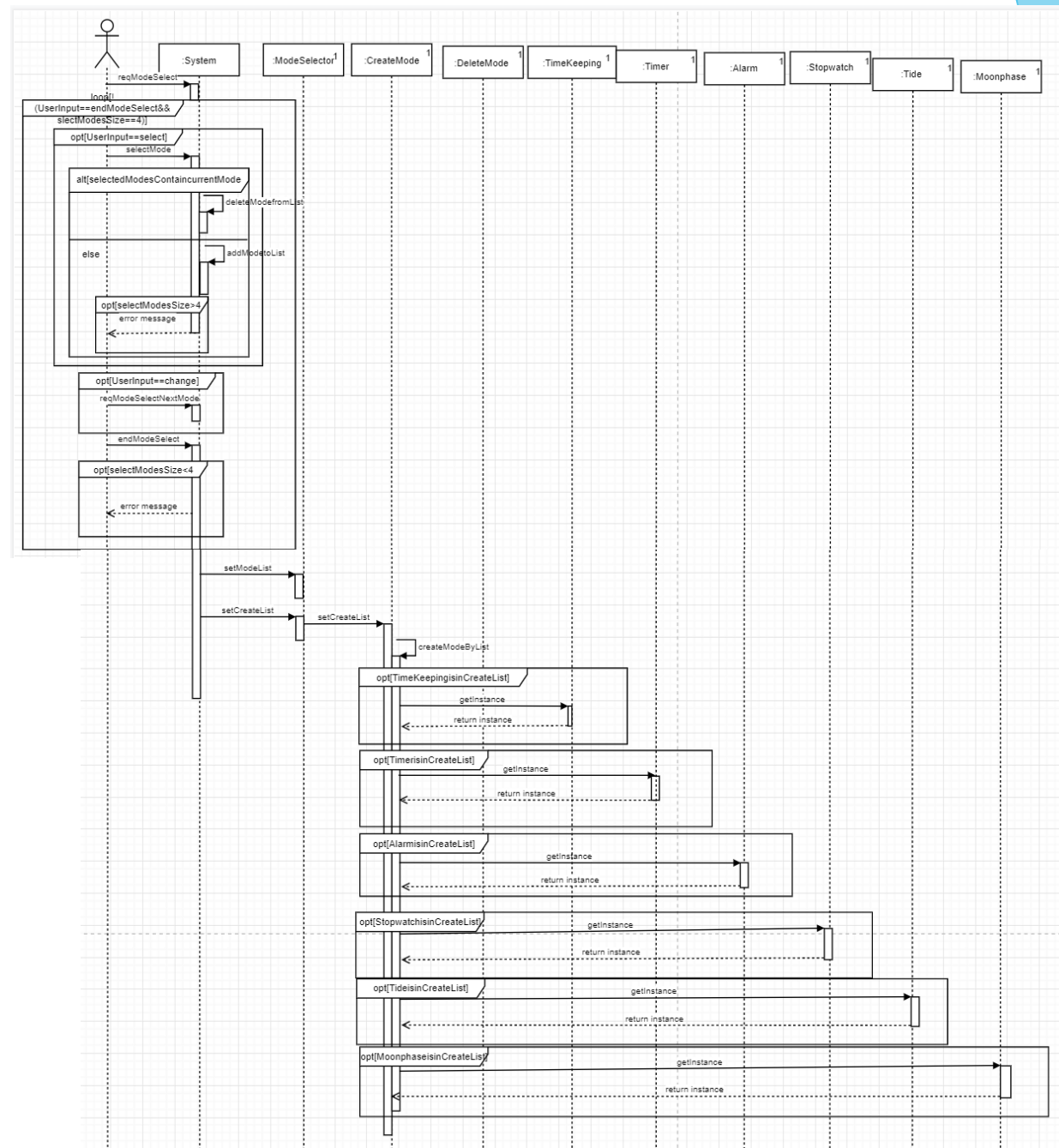
Activity 2052

Name	reqModeSelect
Responsibilities	어떤 모드에서든지 “Reset”버튼을 4회 누른다
Type	GUI
Cross References	Funtional Requirements : R.6.0
Notes	모드선택으로 접근한다
Pre-conditions	Timer모드여야 한다 TimerState가 0 이어야 한다
Post-Conditions	N/A

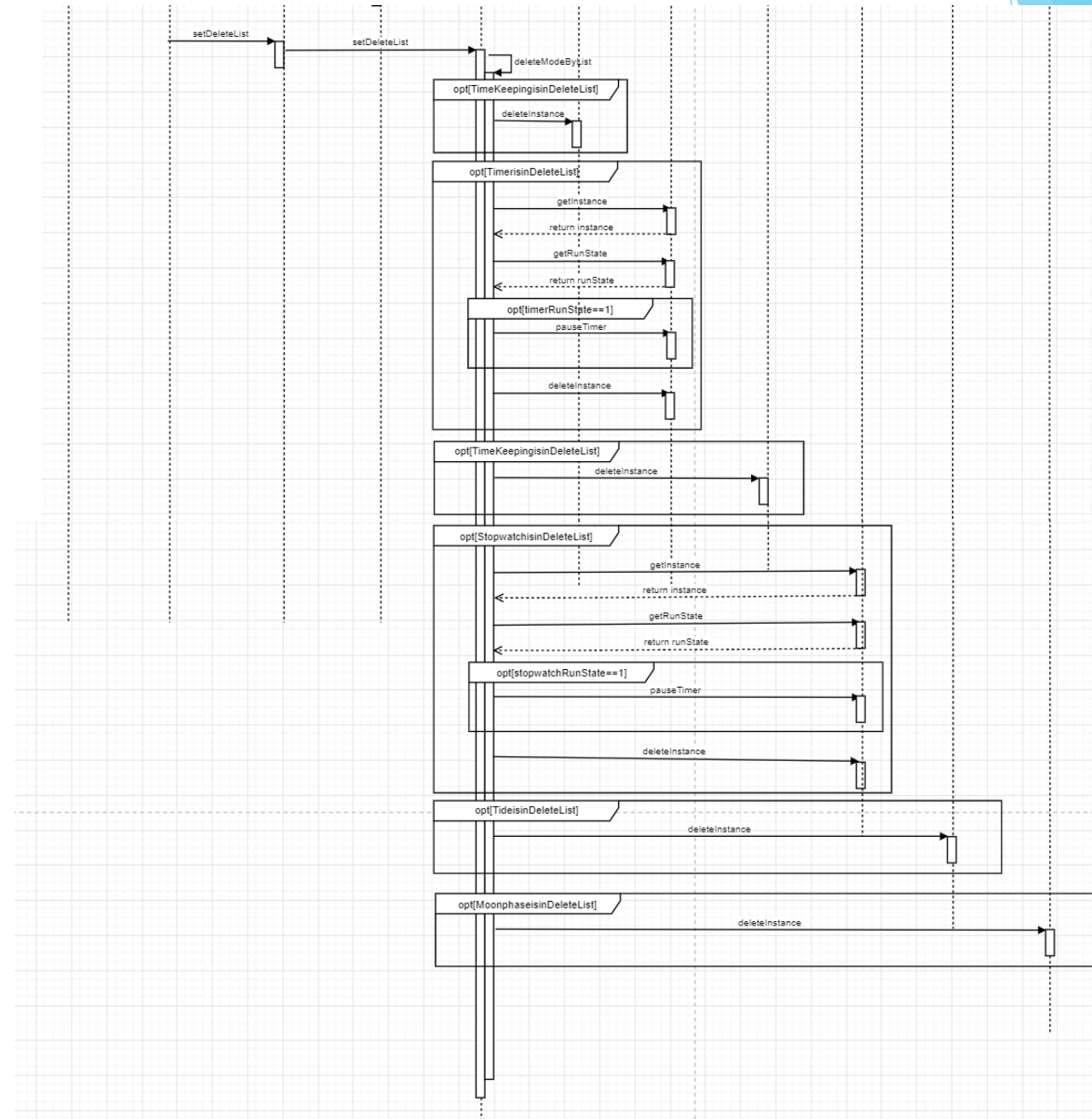
Name	SelectMode
Responsibilities	Stopwatch모드의 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.6.0
Notes	사용하고 싶은 모드를 선택한다
Pre-conditions	
Post-Conditions	4가지 모드 초과시 에러메세지를 출력한다

Name	ChangeCurser
Responsibilities	adjustTime에서 “mode”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	다음 커서로 바꾼다
Pre-conditions	adjustTime을 진입해야 한다
Post-Conditions	현재 시간을 증가시킬수 있다

Activity 2052



Activity 2052



Activity 2055

```
1  import org.junit.Test;
2
3  import javax.swing.*;
4
5  import static org.junit.Assert.*;
6
7  public class SystemUITest {
8
9      @Test
10     public void main() {
11         SystemUI systemUI = new SystemUI();
12
13         assertNotNull(systemUI);
14     }
15
16     @Test
17     public void run() {
18     }
19 }
```

```
1  import org.junit.Test;
2
3  import java.util.ArrayList;
4
5  import static org.junit.Assert.*;
6
7  public class CreateModeTest {
8
9      @Test
10     public void setCreateList() {
11
12         CreateMode createMode = new CreateMode();
13         ArrayList<String> arrayList=new ArrayList<>();
14         arrayList.add("Stopwatch");
15         arrayList.add("Alarm");
16
17         Stopwatch stopwatch=Stopwatch.getInstance();
18         Alarm alarm=Alarm.getInstance();
19         Stopwatch.deleteInstance();
20         Alarm.deleteInstance();
21
22         createMode.setCreateList(arrayList);
23
24         Stopwatch stopwatch1=Stopwatch.getInstance();
25         Alarm alarm1= Alarm.getInstance();
26
27         assertEquals(stopwatch, stopwatch1);
28         assertEquals(alarm, alarm1);
29     }
30 }
```

Activity 2055

```
1 import org.junit.Test;
2
3 import java.util.ArrayList;
4
5 import static org.junit.Assert.*;
6
7 public class DeleteModeTest {
8
9     @Test
10    public void setDeleteList() {
11
12        DeleteMode deleteMode = new DeleteMode();
13        ArrayList<String> arrayList=new ArrayList<>();
14        arrayList.add("Stopwatch");
15        arrayList.add("Alarm");
16
17        Stopwatch stopwatch=Stopwatch.getInstance();
18        Alarm alarm=Alarm.getInstance();
19
20        deleteMode.setDeleteList(arrayList);
21
22        Stopwatch stopwatch1 =Stopwatch.getInstance();
23        Alarm alarm1=Alarm.getInstance();
24
25        assertEquals(stopwatch, stopwatch1);
26        assertEquals(alarm, alarm1);
27    }
28 }
```

Activity 2055

```
29 @Test
30 public void getNextMode() {
31     ModeSelector modeSelector = new ModeSelector("TimeKeeping", "Timer", "Alarm", "Stopwatch");
32
33     String mode=modeSelector.getNextMode("TimeKeeping");
34     assertEquals(mode, "Timer");
35     mode=modeSelector.getNextMode("Timer");
36     assertEquals(mode, "Alarm");
37     mode=modeSelector.getNextMode("Alarm");
38     assertEquals(mode, "Stopwatch");
39     mode=modeSelector.getNextMode("Stopwatch");
40     assertEquals(mode, "TimeKeeping");
41 }
42
43 @Test
44 public void getDefaultNextMode() {
45     ModeSelector modeSelector=new ModeSelector();
46
47     String nextMode= modeSelector.getDefaultNextMode("Timer");
48     assertEquals(nextMode, "Alarm");
49     nextMode=modeSelector.getDefaultNextMode("Alarm");
50     assertEquals(nextMode, "Stopwatch");
51     nextMode=modeSelector.getDefaultNextMode("Stopwatch");
52     assertEquals(nextMode, "Tide");
53     nextMode=modeSelector.getDefaultNextMode("Tide");
54     assertEquals(nextMode, "Moonphase");
55     nextMode=modeSelector.getDefaultNextMode("Moonphase");
56     assertEquals(nextMode, "TimeKeeping");
57     nextMode=modeSelector.getDefaultNextMode("TimeKeeping");
58     assertEquals(nextMode, "Timer");
59 }
60
```

```
1 import org.junit.Test;
2
3 import java.lang.reflect.Array;
4 import java.util.ArrayList;
5
6 import static org.junit.Assert.*;
7
8 public class ModeSelectorTest {
9
10
11     @Test
12     public void setSettingModeList() {
13         ModeSelector modeSelector= new ModeSelector();
14
15         ArrayList<String> settingModeList= new ArrayList<String>();
16         settingModeList.add("TimeKeeping");
17         settingModeList.add("Timer");
18         settingModeList.add("Alarm");
19         settingModeList.add("Stopwatch");
20
21         modeSelector.setSettingModeList(settingModeList);
22         assertEquals(modeSelector.getModelist(), settingModeList);
23     }
24
25     @Test
26     public void getModelist() {
27     }
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62 @Test
63 public void setCreateList() {
64 }
65
66
67 @Test
68 public void setDeleteList() {
69 }
70
```


Activity 2055

```
1 import org.junit.Test;
2
3 import static org.junit.Assert.*;
4
5 public class TimeKeepingTest {
6
7     @Test
8     public void getInstance() {
9         TimeKeeping timeKeeping = TimeKeeping.getInstance();
10
11         TimeKeeping.deleteInstance();
12
13         TimeKeeping timeKeeping1 = TimeKeeping.getInstance();
14
15         assertEquals(timeKeeping, timeKeeping1);
16     }
17
18     @Test
19     public void deleteInstance() {
20         TimeKeeping timeKeeping = TimeKeeping.getInstance();
21
22         TimeKeeping.deleteInstance();
23
24         TimeKeeping timeKeeping1 = TimeKeeping.getInstance();
25
26         assertEquals(timeKeeping, timeKeeping1);
27     }
28
29     @Test
30     public void setTime() {
31         TimeKeeping timeKeeping = TimeKeeping.getInstance();
32
33         timeKeeping.setTime("2015 3 30 9 15 20");
34         String tmp = timeKeeping.getTime();
```

```
35
36         assertEquals(tmp, "2015 3 30 9 15 0");
37     }
38
39     @Test
40     public void getTime() {
41         TimeKeeping timeKeeping = TimeKeeping.getInstance();
42
43         timeKeeping.setTime("2015 3 30 9 15 20");
44         String tmp = timeKeeping.getTime();
45
46         assertEquals(tmp, "2015 3 30 9 15 0");
47     }
48 }
```

Activity 2055

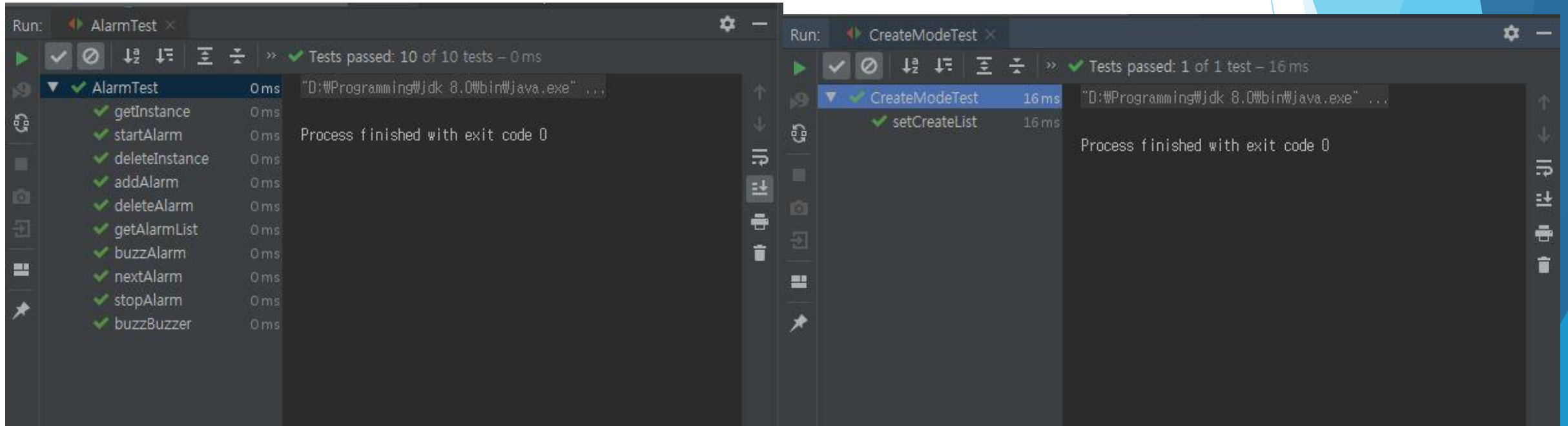
```
1 import org.junit.Test;
2
3 import java.util.HashMap;
4
5 import static org.junit.Assert.*;
6
7 public class TimeDBTest {
8
9     @Test
10    public void getInstance() {
11        TimeDB timeDB = TimeDB.getInstance();
12
13        assertNotNull(timeDB);
14    }
15
16    @Test
17    public void setTime() {
18        TimeDB timeDB = TimeDB.getInstance();
19
20        String time = "2015 11 20 2 15";
21        timeDB.setTime(time);
22        String tmp=timeDB.getTime();
23
24        time= time+ " 0";
25        assertEquals(tmp, time);
26    }
27 }
```

```
28
29 @Test
30 public void setMonthMap() {
31     TimeDB timeDB = TimeDB.getInstance();
32
33     timeDB.setMonthMap(2019);
34
35     HashMap<Integer, Integer> tmpMap = new HashMap<>();
36
37     tmpMap=timeDB.getMonthMap();
38
39     tmpMap.get(2);
40     assertEquals(28+"", tmpMap.get(2)+"");
41 }
42
43 @Test
44 public void getTime() {
45     TimeDB timeDB= TimeDB.getInstance();
46
47     String time="2015 3 30 9 30";
48     timeDB.setTime("2015 3 30 09 30");
49
50     time=time+" 0";
51     assertEquals(time, timeDB.getTime());
52 }
53
54 @Test
55 public void getMonthMap() {
56     TimeDB timeDB=TimeDB.getInstance();
57
58     timeDB.setMonthMap(2019);
59     HashMap<Integer, Integer> tmpMap = new HashMap<>();
60
61     tmpMap=timeDB.getMonthMap();
62     assertEquals(tmpMap.get(2)+"", 28+"");
63 }
```

Activity 2055

```
61     assertEquals(tmpMap.get(2)+"", 28+"");
62 }
63
64 @Test
65 public void updateTime() {
66     TimeDB timeDB=TimeDB.getInstance();
67
68     String time = "2015 3 30 9 30 10";
69     timeDB.setTime(time);
70
71     timeDB.updateTime();
72     String tmp=timeDB.getTime();
73
74     String upTime="2015 3 30 9 30 0";
75     assertEquals(upTime, tmp);
76 }
77
78 @Test
79 public void startUpdateTime() {
80 }
81
82 @Test
83 public void pauseTimeDB() {
84 }
85
86 @Test
87 public void run() {
88 }
89
```

Activity 2061

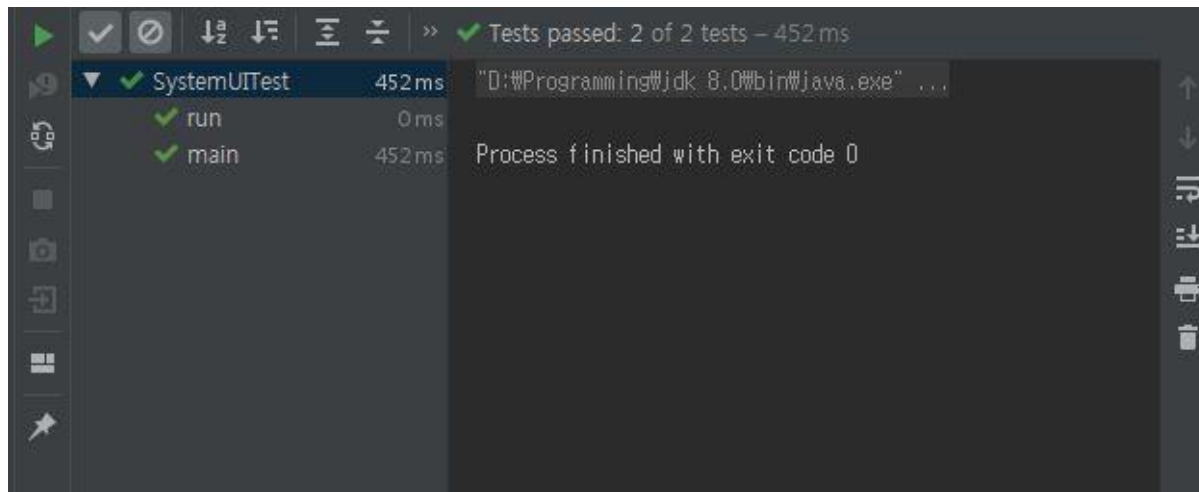


Activity 2061

The image displays four screenshots of an IDE's test runner, showing the execution of different test classes. Each screenshot includes a toolbar with icons for running, pausing, and refreshing tests, along with a summary of test results and a list of individual test methods.

- DeleteModeTest:** Shows 1 of 1 test passed in 0 ms. The test method `setDeleteList` passed in 0 ms. The process finished with exit code 0.
- MoonphaseTest:** Shows 3 of 3 tests passed in 0 ms. The test methods `getInstance`, `deleteInstance`, and `showMoonphase` all passed in 0 ms. The process finished with exit code 0.
- ModeSelectorTest:** Shows 6 of 6 tests passed in 0 ms. The test methods `setDeleteList`, `getDefaultNextMod`, `setSettingModeList`, `getNextMode`, `getModeList`, and `setCreateList` all passed in 0 ms. The process finished with exit code 0.
- StopwatchTest:** Shows 11 of 11 tests passed in 16 ms. The test methods `getInstance`, `getZeroSate`, `deleteInstance`, `getRunState`, `setStopwatch`, `pauseStopwatch`, `updateTime`, `getTime`, `recordStopwatch`, `startStopwatch`, and `resetStopwatch` all passed. `resetStopwatch` took 16 ms. The process finished with exit code 0.

Activity 2061

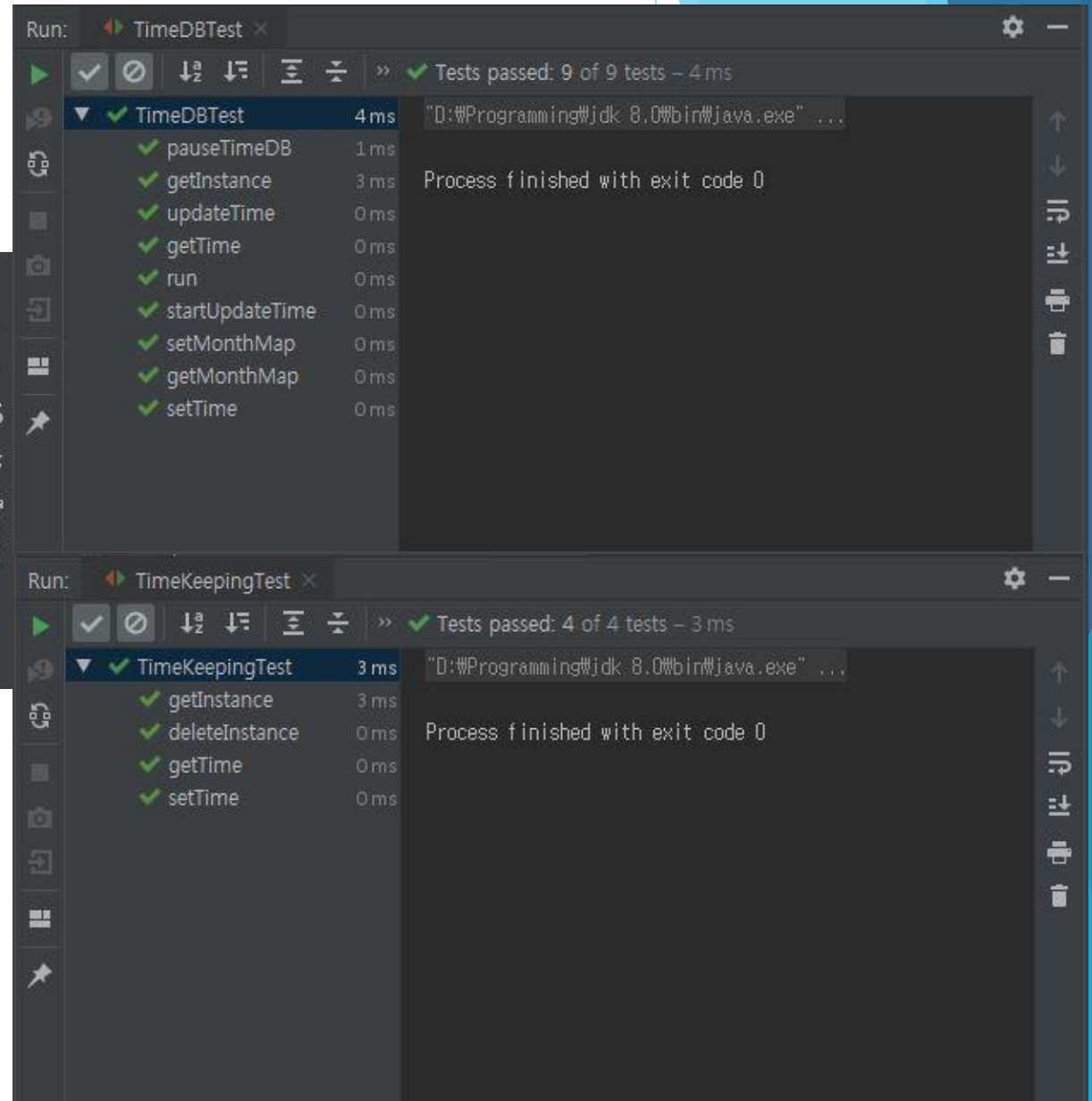


Run: SystemUITest x

Tests passed: 2 of 2 tests – 452 ms

- SystemUITest 452 ms
 - run 0 ms
 - main 452 ms

Process finished with exit code 0



Run: TimeDBTest x

Tests passed: 9 of 9 tests – 4 ms

- TimeDBTest 4 ms
 - pauseTimeDB 1 ms
 - getInstance 3 ms
 - updateTime 0 ms
 - getTime 0 ms
 - run 0 ms
 - startUpdateTime 0 ms
 - setMonthMap 0 ms
 - getMonthMap 0 ms
 - setTime 0 ms

Process finished with exit code 0

Run: TimeKeepingTest x

Tests passed: 4 of 4 tests – 3 ms

- TimeKeepingTest 3 ms
 - getInstance 3 ms
 - deleteInstance 0 ms
 - getTime 0 ms
 - setTime 0 ms

Process finished with exit code 0

Activity 2063

Test Number	Test 항목	Description	Use Case	System Function	P/F
1	getInstance	인스턴스를 통해 객체를 가져오는지 확인	getInstance	R6.0	p
2	deleteInstance	인스턴스를 통해 가져온 객체를 삭제하는지 확인	deleteInstance	R6.0	p
3	getTime Test	시간이 정상적으로 1초마다 업데이트 되는지 확인한다.	getTime	R0.0	p
4	setTime Test	adjustTime으로 새롭게 시간을 설정한 후 설정한 시간이 잘 업데이트되는지 확인한다.	setTime	R0.1	p
5	showAlarm Test	화면에 알람목록의 첫번째 알람이 정상적으로 출력되는지 확인한다	showAlarm	R2.1	p
6	nextAlarm Test	버튼 입력에 따라 현재 알람에서 다음알람으로 제대로 넘어가는지 확인한다	nextAlarm	R2.2	p
7	addAlarm Test	알람을 설정 가능할경우에만 알람을 새롭게 설정하고 알람목록에 알람이 정상적으로 갱신되는지 확인한다	addAlarm	R2.3	p
8	deleteAlarm Test	현재 지우려고 하는 알람이 정상적으로 지워지는지 확인하고 알람이 4개 설정되어 있을때 알람이 삭제하는 것만 가능한지 확인한다	deleteAlarm	R2.4	p
9	buzzAlarm Test	알람이 설정된 시간이 되었을 때에 맞춰 잘 울리는지 확인하고 바뀐 알람상태를 잘 반환해주는지 확인한다	buzzAlarm	R2.5	p
10	StopAlarm Test	알람이 울릴 때 어떤 모드에서든지 버튼 입력이 있으면 울리던 알람이 즉각적으로 잘 멈추는지 확인한다.	stopAlarm	R2.6	p
11	getZerostate	기능이 실행되면 해당 기능의 runstate가 0으로 바뀌게 만드는지 확인한다	pauseTimer buzzAlarm pauseStopwatch	R1.3 R2.4 R3.3	p
12	getRunstate	기능이 실행되면 해당기능의 runstate가 으로 바뀌게 만드는지 확인한다	startTimer'stopAlarm startStopwatch	R1.2 R2.5 R3.1	p
13	getTime	스탑위치로 보낸 시간이 똑같이 출력되는지 확인한다	showStopwatch	R3.0	p
14	updateTime	갱신된 시간이 스톱위치에도 잘 적용되는지 확인한다	showStopwatch	R3.0	p
15	set Stopwatch Test	스톱위치의 현재 시간이 정상적으로 보이는지 확인한다	set Stopwatch	R3.1	p
16	start Stopwatch Test	버튼 입력에 따라 스톱위치가 정상적으로 시간을 갱신하는지 확인한다	start Stopwatch	R3.2	p
17	record Stopwatch Test	스톱위치가 동작 중일 때 버튼을 입력하면 정확하게 버튼을 입력한 때에 나타났던 시간이 잘 기록되는지 확인한다.	record Stopwatch	R3.3	p
18	pause Stopwatch Test	스톱위치가 동작 중일 때 버튼을 입력하면 정확하게 버튼을 입력한 때에 잘 멈추는지 확인한다.	pause Stopwatch	R3.4	p
19	reset Stopwatch Test	스톱위치 모드에서 스톱위치의 동작 중 여부에 관계없이 버튼을 입력하면 스톱위치가 정상적으로 초기화되는지 확인한다.	reset Stopsatch	R3.5	p
20	show Moonphase Test	날짜에 맞게 달의 위상이 정확하게 나타나는지 확인한다.	show Moonphase	R5.1	p

Activity 2063

21	Moonphase Test	현재 날짜를 음력으로 바꾸고 음력날짜에 맞는 달 모양의 그래픽으로 바꿔주는지 확인한다	calculate Moonpahse	R5.2	p
22	modeSelect Test	모드들을 선택할때 정상적으로 체크/체크해제가 되는지 확인한다 모드들을 정상적으로 4개를 선택해야 modeSelect를 빠져나갈 수 있는지 확인한다	modeSelect	R6.1	p
23	setDeleteList	설정된 모드가 담긴 리스트가 삭제되는지 확인한다	modeSelect	R6.0	p
24	getDefaultNextMod	ModeSelect화면에 모든 6개의 모드가 나오는지 확인한다	modeSelect	R6.0	p
25	setSettingModelist	ModeDelect에서 설정한 모드와 ModeList와 같은지 확인	modeSelect	R6.0	p
26	getNextMode	ModeList의 다음모드가 잘 적용되는지 확인	modeSelect	R6.0	p
27	getModelist	ModeList가 정상적으로 불러와지는지 확인한다	modeSelect	R6.0	p
28	setCreateList	ModeList가 정상적으로 설정되었는지 확인한다	modeSelect	R6.0	p
29	pauseTimeDB	시간을 조정하는 동안 시간 갱신을 잠시 멈추는 기능이 정상적으로 작동하는지 확인한다	adjustTime	R0.1	p
30	getTime	TimeDB에 설정한 시간이 설정하려던 시간과 일치하는지 확인	adjustTime	R0.1	p
31	updateTime	사용자가 설정한 시간이 잘 갱신이 되었는지 확인한다	adjustTime	R0.1	p
32	startUpdateTime	시간갱신이 잘 시작는지 확인한다	showTime	R0.0	p
33	setMonthMap	Monthmap이 잘 구성이 되는지 확인한다	showTime	R0.0	p
34	getMonthMap	Monthmap이 잘 받아와 지는지확인한다	showTime	R0.0	p
35	setTime	설정된 날짜를 연월일 나누어 정확히 저장하는지 확인한다	adjustTime	R0.1	p

Q&A