

# Object-Oriented Analysis and Design

## Team 5 : RDM Watch(Random Decision Making) 2050. Object-Oriented Implementation

컴퓨터공학부 201411300 이정우  
컴퓨터공학부 201714150 김동진  
컴퓨터공학부 201714155 윤태성  
수학과 201410334 고길재

**2051.**

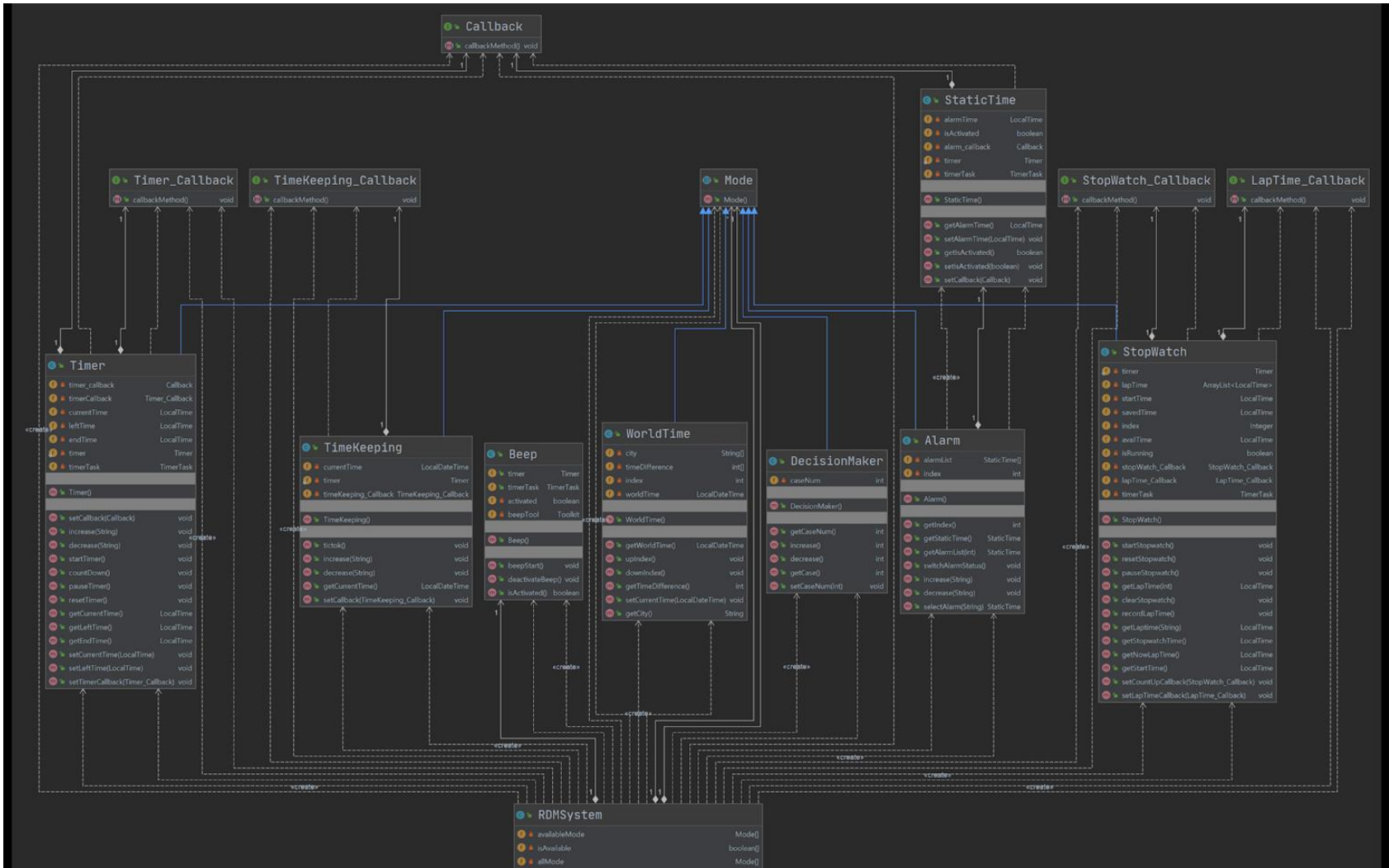
**Source Code +**

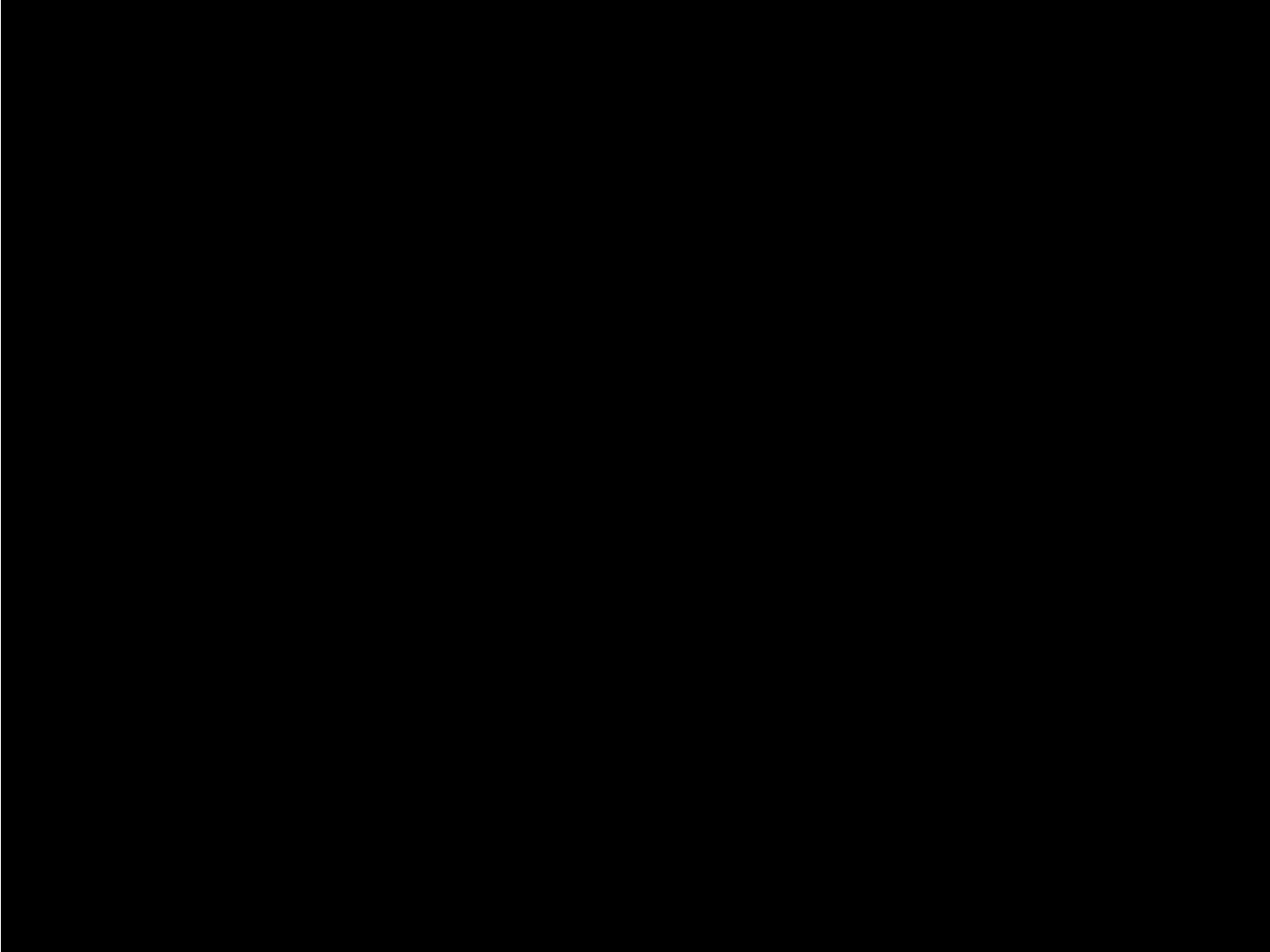
**Class Diagram + Demo Video**

## Source Code

**Github address :**

**<https://github.com/feuziq47/softwarev-v>**





**2052.**

**Testing**

# 01. Set Current Time

```
public class TimeKeepingTest {  
    /**  
     *기존의 시간에서 설정한 시간으로 현재 시간이 제대로 변경되는지 Test한다.  
     */  
    @Test  
    public void setCurrentTime(){  
        RDMSystem rdm = new RDMSystem();  
        LocalDateTime testLDT;  
        int temp = 0;  
        boolean tmp1 = false, tmp2 = false, tmp3 = false, tmp4 = false, tmp5 = false, tmp6 = false, tmp7 = false, tmp8 = false, tmp9 = false, tmp10 = false, tmp11 = fa  
        rdm.decodeButtonInput("LONG_MO");  
        tmp1 = rdm.isSettingMode();  
  
        rdm.decodeButtonInput("RE");  
        testLDT = LocalDateTime.now();  
        temp = rdm.getTimeKeepingCurrentTime().getYear() - testLDT.getYear();  
        tmp2 = (temp == 1);  
  
        rdm.decodeButtonInput("ST");  
        testLDT = LocalDateTime.now();  
        temp = rdm.getTimeKeepingCurrentTime().getYear() - testLDT.getYear();  
        tmp3 = (temp == 0);  
  
        rdm.decodeButtonInput("AD");  
        rdm.decodeButtonInput("RE");  
        testLDT = LocalDateTime.now();  
        temp = rdm.getTimeKeepingCurrentTime().getMonth().getValue() - testLDT.getMonth().getValue();  
        tmp4 = (temp == 1);  
    }  
}
```

## 02. Set Timer

```
/**
 * setCurrentTime 이 설정된 값으로 초기화 되는지 확인
 * /
@Test
void setTimer() {
    Timer timer = new Timer();
    LocalDateTime standard = LocalDateTime.of(0,0,3);
    timer.setCurrentTime(standard);

    assertTrue(timer.getCurrentTime().getHour() == standard.getHour() &&
        timer.getCurrentTime().getMinute() == standard.getMinute() &&
        timer.getCurrentTime().getSecond() == standard.getSecond());
}
```



### 03. Start Timer

```
/**
 * LeftTime 이 countDown을 호출한뒤, 시간이 감소하는지 확인
 */
@Test
void StartTimer() throws InterruptedException {
    Timer timer = new Timer();
    LocalDateTime standard = LocalDateTime.of(0,0,10);
    timer.setLeftTime(standard);
    timer.countDown();
    TimeUnit.SECONDS.sleep(2);
    assertTrue(timer.getLeftTime().isBefore(standard));
}
```

## 04. Pause Timer

```
/**
 * countDown 한 후, pauseTimer을 했을때, LeftTime이 정지 상태로 유지하는지 확인
 */
@Test
void PauseTimer() throws InterruptedException {
    Timer timer = new Timer();
    LocalDateTime standard = LocalDateTime.of(0,0,10);
    timer.setLeftTime(standard);
    timer.countDown();
    TimeUnit.SECONDS.sleep(3);
    timer.pauseTimer();
    TimeUnit.SECONDS.sleep(3);
    assertEquals(timer.getLeftTime().getSecond(),
        LocalDateTime.of(0, 0, 6).getSecond());
}
```

## 05. Reset Timer

```
/**
 * resetTimer을 했을때, LeftTime값과 CurrentTime으로 초기화되는지 확인
 */
@Test
void ClearTimer() throws InterruptedException {
    Timer timer = new Timer();
    LocalDateTime standard = LocalDateTime.of(0,0,10);
    timer.setCurrentTime(standard);
    timer.setLeftTime(LocalDateTime.of(0,0,5));
    timer.resetTimer();
    assertTrue(timer.getCurrentTime().getHour() == timer.getLeftTime().getHour() &&
        timer.getCurrentTime().getMinute() == timer.getLeftTime().getMinute() &&
        timer.getCurrentTime().getSecond() == timer.getLeftTime().getSecond());
}
```

## o6. Notify the end

```
* Timer가 종료되면 제대로 Beep음이 울리는 지 Test한다.  
*/  
@Test  
void NotifyTheEndOfTimer(){  
    Timer timer = new Timer();  
    RDMSystem rdm = new RDMSystem();  
    if(timer.getLeftTime().equals(timer.getEndTime())){  
        assertTrue(rdm.getBeep().isActivated());  
    }  
}
```

## 07. Start Stopwatch

```
* Stopwatch가 0초부터 시작되는지 Test한다.  
* Stopwatch의 시간이 시간의 흐름에 따라 제대로 증가하는 지 Test한다.  
*  
*/  
@Test  
public void startStopwatch() throws InterruptedException {  
    RDMSystem rdms = new RDMSystem();  
    rdms.decodeButtonInput("M0");  
    rdms.decodeButtonInput("ST");  
  
    Stopwatch sw = (StopWatch)rdms.getAllMode()[1];  
  
    TimeUnit.SECONDS.sleep(3);  
    assertTrue(sw.getStartTime().isAfter(LocalTime.of(0,0,0)));  
}
```

## 08. Pause Stopwatch

```
* Stopwatch가 정지 버튼에 해당하는 신호가 주어졌을때  
* Stopwatch에 표시된 현재 시간에서 정지하는지 Test한다.  
*/  
@Test  
public void pauseStopwatch() throws InterruptedException {  
    RDMSystem rdms = new RDMSystem();  
    Stopwatch sw = (StopWatch)rdms.getAllMode()[1];  
    rdms.decodeButtonInput("MO");  
    rdms.decodeButtonInput("ST");  
  
    LocalTime start = sw.getStartTime();  
  
    rdms.decodeButtonInput("ST");  
  
    LocalTime stop = sw.getStartTime();  
  
    TimeUnit.SECONDS.sleep(3);  
  
    assertTrue(start.getHour() == stop.getHour() &&  
        start.getMinute() == stop.getMinute() &&  
        start.getSecond() == stop.getSecond());  
}
```

## 09. Record Lap Time

```
* Stopwatch에서 Lap Time에 해당하는 신호가 주어졌을때  
* Stopwatch에 표시된 현재 시간이 기록되는지 test한다.  
*/  
@Test  
public void recordLapTime() throws InterruptedException {  
    RDMSystem rdms = new RDMSystem();  
    Stopwatch sw = (StopWatch)rdms.getAllMode()[1];  
    rdms.decodeButtonInput("MO");  
    rdms.decodeButtonInput("ST");  
  
    TimeUnit.SECONDS.sleep(3);  
  
    // 랩 타임  
    rdms.decodeButtonInput("AD");  
    //정지  
    rdms.decodeButtonInput("ST");  
  
    LocalTime lap = sw.getLapTime(0);  
    LocalTime cur = sw.getStartTime();  
    assertTrue(lap.getHour() == cur.getHour() &&  
        lap.getMinute() == cur.getMinute() &&  
        lap.getSecond() == cur.getSecond());  
}
```

## 10. Clear Stopwatch

```
* Stopwatch에서 Lap Time에 해당하는 신호가 주어졌을때  
* Stopwatch에 표시된 현재 시간이 기록되는지 test한다.  
*/  
@Test  
public void clearStopwatch() throws InterruptedException {  
    RDMSystem rdms = new RDMSystem();  
    Stopwatch sw = (StopWatch)rdms.getAllMode()[1];  
    rdms.decodeButtonInput("M0");  
    rdms.decodeButtonInput("ST");  
  
    TimeUnit.SECONDS.sleep(3);  
    rdms.decodeButtonInput("ST");  
    assertTrue(sw.getStartime().isAfter(LocalTime.of(0,0,0)));  
    rdms.decodeButtonInput("RE");  
  
    LocalTime temp = sw.getStartime();  
  
    assertTrue(temp.getHour() == 0 &&  
        temp.getMinute() == 0 &&  
        temp.getSecond() == 0);  
}
```



## 11. Activate alarm

\* 활성화 된 Alarm 시간이 되면 Beep음이 울리는 지 Test한다.

\*/

@Test

```
public void activateAlarm() {
```

```
    RDMSystem rdm = new RDMSystem();
```

```
    Alarm al = new Alarm();
```

```
    if(al.getStaticTime().getIsActivated() && al.getStaticTime().getAlarmTime().isBefore(LocalTime.now())){  
        assertTrue(rdm.getBeep().isActivated());  
    }
```

```
}
```

## 12. Deactivate Alarm

- \* 비활성화된 Alarm 시간이 됐을 때, Beep이 울리지 않는지를 Test한다.
- \* 선택된 4가지 Mode에서 제외 될 경우 비활성화 되는지 test한다.

```
*/  
@Test  
public void deactivateAlarm() {  
    RDMSystem rdm = new RDMSystem();  
    Alarm al = new Alarm();  
    if(!al.getStaticTime().getIsActivated() && al.getStaticTime().getAlarmTime().isBefore(LocalTime.now())){  
        assertFalse(rdm.getBeep().isActivated());  
    }  
}
```

## 13. Set Alarm

```
* Alarm시간이 설정된 시간으로 변경되는지 Test한다.  
*  
* v2 -> 두 단계로 쪼개겠습니다. increase decrease  
*/  
@Test  
public void increase() {  
    RDMSystem rdm = new RDMSystem();  
    if((rdm.getCurrentMode() instanceof Alarm) && rdm.getIsSelectMode()){  
        LocalDateTime beforeTime = ((Alarm)rdm.getAllMode()[3]).getStaticTime().getAlarmTime();  
        rdm.decodeButtonInput("RE");  
        assertEquals(((Alarm)rdm.getAllMode()[3]).getStaticTime().getAlarmTime(), beforeTime.plusHours(1));  
    }  
}  
  
@Test  
public void decrease() {  
    RDMSystem rdm = new RDMSystem();  
    if((rdm.getCurrentMode() instanceof Alarm) && rdm.getIsSelectMode()){  
        LocalDateTime beforeTime = ((Alarm)rdm.getAllMode()[3]).getStaticTime().getAlarmTime();  
        rdm.decodeButtonInput("ST");  
        assertEquals(((Alarm)rdm.getAllMode()[3]).getStaticTime().getAlarmTime(), beforeTime.minusHours(1));  
    }  
}  
}
```

## 14. Switch Alarm

```
74 /**  
75  * 알람이 다음 알람으로 제대로 바뀌는 지 확인한다.  
76  */  
77  
78 @Test  
79 public void switchAlarm() {  
80     RDMSystem rdm = new RDMSystem();  
81     Alarm a1 = new Alarm();  
82     if(rdm.getCurrentMode() instanceof Alarm && !rdm.getIsSelectMode()){  
83         int beforeIndex = ((Alarm)rdm.getAllMode()[3]).getIndex();  
84         rdm.decodeButtonInput("ST");  
85         assertEquals(((Alarm)rdm.getAllMode()[3]).getIndex(), actual: beforeIndex+1);  
86  
87     }  
88 }  
89
```

## 15. Notify Alarm

```
/**  
 * 특정시간으로 알람을 설정한 다음,  
 * 해당 시간이 되면, 지정된 알람의 isActivated 가 True인지 확인  
 */
```

```
@Test  
public void notifyAlarm() throws ParseException, InterruptedException {  
    RDMSSystem rdms = new RDMSSystem();  
    Alarm alarm = (Alarm)rdms.getAllMode()[3];  
    StaticTime st = alarm.getAlarmList(index: 0);  
    st.setIsActivated(true);  
    st.setAlarmTime(LocalTime.of( hour: 22, minute: 49, second: 30));  
    Beep beep = rdms.getBeep();  
    assertFalse(beep.isActivated());  
    DateFormat dateFormatter = new SimpleDateFormat( pattern: "HH:mm:ss");  
    Date date = dateFormatter.parse( source: "22:49:50");  
  
    java.util.Timer timer = new java.util.Timer();  
  
    TimerTask timerTask = new TimerTask() {  
        public void run() {  
            assertTrue(beep.isActivated());  
            timer.cancel();  
        }  
    };  
  
    timer.schedule(timerTask , date);  
}
```

```
long currentMillis = System.currentTimeMillis();  
long givenDateMillis = LocalDateTime.of( year: 2020, month: 6, dayOfMonth: 8, hour: 22, minute: 50, second: 0)  
    .atZone(ZoneId.systemDefault()) ZonedDateTime  
    .toInstant() Instant  
    .toEpochMilli();  
  
Thread.sleep( givenDateMillis - currentMillis);
```

## 16. Set Case Number

```
/**
 * 입력한 값으로 Case Number가 설정 되는지 Test한다.
 */
@Test
public void setCaseNumber() throws InterruptedException {
    DecisionMaker dm = new DecisionMaker();
    dm.setCaseNum(10);
    assertEquals(dm.getCaseNum(), actual: 10);
}
```

## 17. Get Case

```
/**
 *입력한 Case Number내의 Case만 나오는지 Test한다.
 * Random한 값으로 Case가 나오는지 Test한다.
 */
@Test
public void getCase() throws InterruptedException {
    DecisionMaker dm = new DecisionMaker();
    dm.setCaseNum(10);
    for (int i = 0; i < 100; i++) {
        int caseNum = dm.getCase();
        assertTrue( condition: caseNum <= 10 && caseNum >= 0 );
    }
}
```

## 18. Switch city

```
/**
 * 완성된 파일 없으므로 지금은 생략
 * 선택된 City로 설정 되는지 Test한다.
 */
@Test
public void chooseCity() {
    WorldTime wt = new WorldTime();
    wt.upIndex();
    assertTrue(wt.getCity().equals("SYD"));
    wt.downIndex();
    assertTrue(wt.getCity().equals("NYC"));
}
```



## 19. Switch Current Mode

```
/**
 * 완성된 파일 없으므로 지금은 생략
 * 지정된 버튼을 눌렀을 때, 다음 Mode로 변경되는지 Test한다.
 */
@Test
public void changeCurrentMode() throws InterruptedException{
    RDMSystem rdms = new RDMSystem();
    rdms.changeCurrentMode();
    assertTrue(rdms.getCurrentMode() instanceof Stopwatch);
}
```

## 20. Select Modes

```
/**
 * 완성된 파일 없으므로 지금은 생략
 * 지정된 버튼을 눌렀을 때, Mode 선택창으로 넘어가는지 Test한다.
 * Mode 선택창에서 선택한 4가지의 Mode가 Activate 하게 설정 되는지 Test한다.
 */
@Test
public void selectModes() {
    RDMSystem rdms = new RDMSystem();
    rdms.selectMode( buttonInput: "ST");
    assertTrue( condition: rdms.getIsAvailable()[0] == false);
}
```

## 21. Activate Beep

```
/**
 * 완성된 파일 없으므로 지금은 생략
 * Beep음을 활성화 시켰을때, Beep음이 올라야 하는 모든 상황에서 Beep음이 올리는지 Test 한다.
 */
@Test
public void activateBeep() {
    Beep beep = new Beep();
    beep.beepStart();
    assertTrue(beep.isActivated());
    beep.deactivateBeep();
}
```

## 22. Deactivate Beep

\* 지정된 버튼을 눌렀을 때, Beep음이 종료되는지 *test* 한다.

\*/

@Test

```
public void deactivateBeep() {  
    Beep beep = new Beep();  
    boolean tmp = false;  
    beep.beepStart();  
    tmp = beep.isActivated();  
    beep.deactivateBeep();  
    assertTrue( condition: tmp && !beep.isActivated() );  
}
```

**2053.**

**2050 Traceability Analysis**

