

객체지향개발방법론

OOAD 3rd_iteration (Object-Oriented Design)

201212519 컴퓨터공학과 김선우
201510624 컴퓨터공학과 김용현
201614157 컴퓨터공학과 김도연
201614158 컴퓨터공학과 장다혜

Activity :
System Testing Refine

Ref.#	Test case ID	Test Scenario
R 1.1	T.timekeeping.2	2100 이상으로 증가함. 1970 밑으로 내려감. INPUT : year 2100이상에서 B INPUT : year 1970 month 1 day 1에서 day에 커서를 둔 후 D
R 2.1	T.stopwatch.4	59분 59초 99 이상으로 증가 INPUT : B -> 1시간 대기

R 4.3	T.timer.5	타이머 일시정지상태에서 모드 변환이 되지 않음
R 4.2	T.timer.7	INPUT : AA -> B -> AA -> B -> B -> C
R 6.2	T.brightness.2	종료 후 B버튼에 반응함 INPUT : D -> B -> B -> D -> B
R 1.3.2	T.buzzer.2	다른 버튼이 원래의 기능대로 동작한다. INPUT : 알람이 울리는 중 C

R 1.1 ←	T.timekeeping. 2 ←	연도가 지정 범위를 벗어나지 않음을 검증한다. ←	F ←	2100 이상으로 증가함 ← 1970 밑으로 내려감 ←
---------	-----------------------	--------------------------------	-----	--

```

case 2: // year
  if (button == 1) {
    time.add(Calendar.YEAR, amount: 1);

    currentDate.setYear(time.get(Calendar.YEAR));
  } else if (button == 3) {
    time.add(Calendar.YEAR, amount: -1);
    if (time.get(Calendar.YEAR) < 1970)
      time.set(Calendar.YEAR, 1970);
    currentDate.setYear(time.get(Calendar.YEAR));
  }
  break;

```



```

case 2: // year
  if (button == 1) {
    time.add(Calendar.YEAR, amount: 1);
    if (time.get(Calendar.YEAR) > 2100)
      time.set(Calendar.YEAR, 2100);
    currentDate.setYear(time.get(Calendar.YEAR));
  } else if (button == 3) {
    time.add(Calendar.YEAR, amount: -1);
    if (time.get(Calendar.YEAR) < 1970)
      time.set(Calendar.YEAR, 1970);
    currentDate.setYear(time.get(Calendar.YEAR));
  }
  break;

```

```

case 4: // day
  if (button == 1) {
    time.add(Calendar.DATE, amount: 1);
    currentDate.setDay(time.get(Calendar.DATE));
  } else if (button == 3) {
    time.add(Calendar.DATE, amount: -1);

    currentDate.setDay(time.get(Calendar.DATE));
  }
  break;

```



```

case 4: // day
  if (button == 1) {
    time.add(Calendar.DATE, amount: 1);
    currentDate.setDay(time.get(Calendar.DATE));
  } else if (button == 3) {
    time.add(Calendar.DATE, amount: -1);
    if (time.get(Calendar.YEAR) < 1970)
      time.set(Calendar.YEAR, 1970);
    currentDate.setDay(time.get(Calendar.DATE));
  }
  break;

```

R 2.1	T.stopwatch.4	스탑워치 값이 정상 범주 이상으로 측정하게 되면 스톱워치 동작이 정지됨을 검증합니다.	F	59분 59초 99 이상으로 증가
-------	---------------	---	---	--------------------

```

if (stopwatchTime.getSecond() >= 59 && stopwatchTime.getMinute() >= 59) {
    OnOff = false;
    stopwatchTime.setMinute(59);
    stopwatchTime.setSecond(59);
    stopwatchTime.setM_second(99);
} else if (stopwatchTime.getM_second() >= 99) {
    stopwatchTime.setM_second(0);
    stopwatchTime.setSecond(stopwatchTime.getSecond() + 1);
} else if (stopwatchTime.getSecond() >= 59) {
    stopwatchTime.setSecond(0);
    stopwatchTime.setMinute(stopwatchTime.getMinute() + 1);
}

```

stopwatchTime.getM_second
를 확인하는 조건을 if문에서
제거

R 6.2	T.brightness.2	밝기 조절 기능이 정상적으로 종료됨을 검증합니다.	F	종료 후 B버튼에 반응함
-------	----------------	-----------------------------	---	---------------

```
public void reqTurnOffBC() {  
    boolean check = BLC.checkBC();  
    if (check) {  
        BLC.daylong(TKM.getCurrentTime());  
        gui.watchBLC(BLC.getBrightLevel());  
    }  
}
```



```
public void reqTurnOffBC() {  
    boolean check = BLC.checkBC();  
    if (check) {  
        BLC.daylong(TKM.getCurrentTime());  
    }  
    gui.watchBLC(BLC.getBrightLevel());  
}
```

R 1.3.2	T.buzzer.2	buzzer의 작동이 시작하는 시점에서 버튼 기능에 우선함을 검증합니다.	F	다른 버튼이 원래의 기능대로
------------	------------	--	---	-----------------------

```
public void reqStopRingin() {  
    ringTh.setRinging(false);  
    thread.interrupt();  
    ringTh.setOnOff(false);  
    ring.setOnOff(false);  
  
    controlAlarmIndicator();  
    if(currentState==23){  
        currentState=19;  
        TRM.getTimerTime().setM_second(0);  
    }  
}
```

```
public void reqStopRingin() {  
    ringTh.setRinging(false);  
    thread.stop();  
    ringTh.setOnOff(false);  
    ring.setOnOff(false);  
  
    controlAlarmIndicator();  
    if(currentState==23){  
        currentState=19;  
        TRM.getTimerTime().setM_second(0);  
    }  
}
```

Activity :
Refine Static Analysis

1. FindBugs

ii. High Priority←

1. DLS_DEAD_LOCAL_STORE←

- a. 내용 : 사용되지 않은 지역 변수가 존재합니다.←
- b. 위치 : Main.java:5←

```
DWS_controller dws_controller = new DWS_controller();  
dws_controller.start();
```

```
public void start(){  
    long delay = 0;  
    long interval = 100;  
    displayTimer.scheduleAtFixedRate(displayTask, delay, interval);  
}
```

DWS 사용할 메소드 추가

1. FindBugs

2. ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD

- a. 내용 : Static field에 쓰는 메서드 인스턴스가 존재합니다.
- b. 해결 방법 : Setter 메서드를 사용하기를 권장드립니다.
- c. 위치 :



static 변수에서 static 삭제

2. PMD

1. SystemPrintln

- a. 내용 : System.out.println이 사용됨
- b. 해결 방법 : 해당 구문 제거
- c. 위치 : DWS_controller.java:521



삭제 완료

3. PositionLiteralsFirstInComparisons

- a. 내용 : 문자열을 비교할 때는 상수 문자열을 앞쪽에 배치하세요.
- b. 해결 방법 : 문자열 상수의 equals()를 호출하세요.



```
.....  
if("black".equals(SW[currentIndex][2])){
```

```
if("TRM".equals(quitSW))
```

```
if("ALM".equals(selectedSW[2][1])){
```

2. PMD

1. AvoidDuplicateLiterals

- 내용 : 상수 리터럴이 반복적으로 나타나고 있습니다
- 해결 방법 : enum이나 static final로 선언하여 사용하세요
- 위치 :

```
private static final String serif="Serif";
```

```
private static final String[] world = {"KOREA", "AMERICA", "EUROPE", "CHINA", "RUSSIA", "SINGAPORE"};
```

3. UseEqualsToCompareStrings

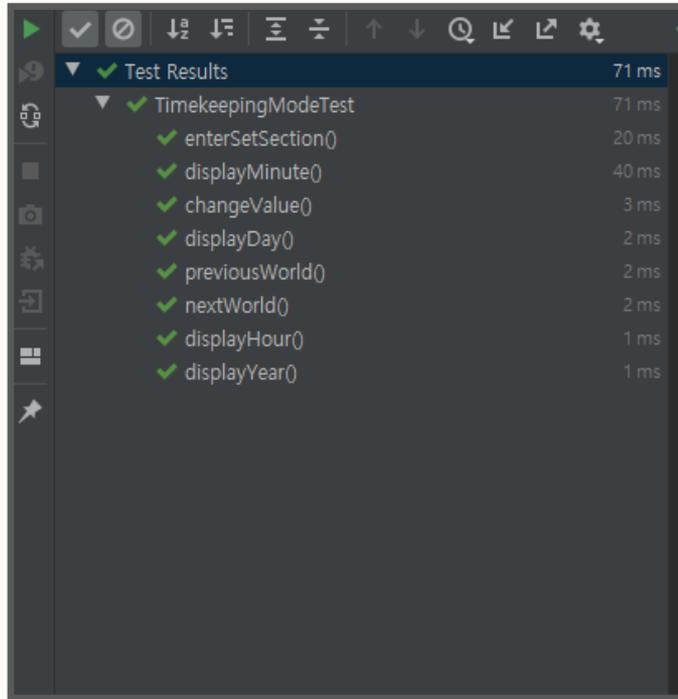
- 내용 : 문자열 비교에 ==를 쓰지 마십시오
- 해결 방법 : equals()를 호출하세요.
- 위치 : SWMode.java:46

```
46
```

```
if("black".equals(SW[currentIndex][2]))
```

Activity :
Refine Unit Testing

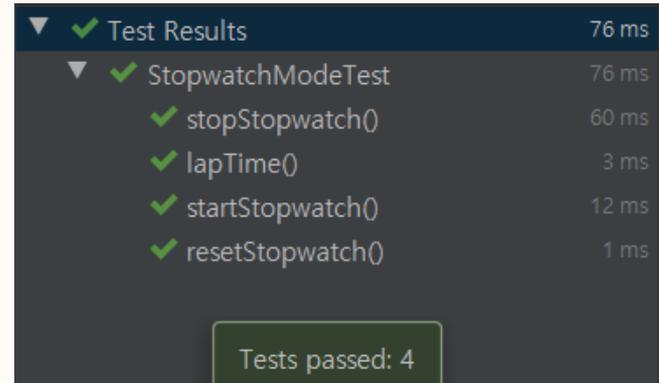
1. Timekeeping Mode



A screenshot of a test runner interface showing the results for 'Timekeeping Mode'. The interface includes a toolbar at the top with icons for play, check, refresh, and other controls. The test results are displayed in a tree view with a dark background and light text. The 'Test Results' folder is expanded, showing a total time of 71 ms. Underneath, the 'TimekeepingModeTest' is expanded, showing a total time of 71 ms and a list of seven methods, each with a green checkmark and its execution time.

Test Results	71 ms
TimekeepingModeTest	71 ms
enterSetSection()	20 ms
displayMinute()	40 ms
changeValue()	3 ms
displayDay()	2 ms
previousWorld()	2 ms
nextWorld()	2 ms
displayHour()	1 ms
displayYear()	1 ms

2. Stopwatch Mode



A screenshot of a test runner interface showing the results for 'Stopwatch Mode'. The interface includes a toolbar at the top with icons for play, check, refresh, and other controls. The test results are displayed in a tree view with a dark background and light text. The 'Test Results' folder is expanded, showing a total time of 76 ms. Underneath, the 'StopwatchModeTest' is expanded, showing a total time of 76 ms and a list of four methods, each with a green checkmark and its execution time. A green box at the bottom of the screenshot displays the text 'Tests passed: 4'.

Test Results	76 ms
StopwatchModeTest	76 ms
stopStopwatch()	60 ms
lapTime()	3 ms
startStopwatch()	12 ms
resetStopwatch()	1 ms

Tests passed: 4

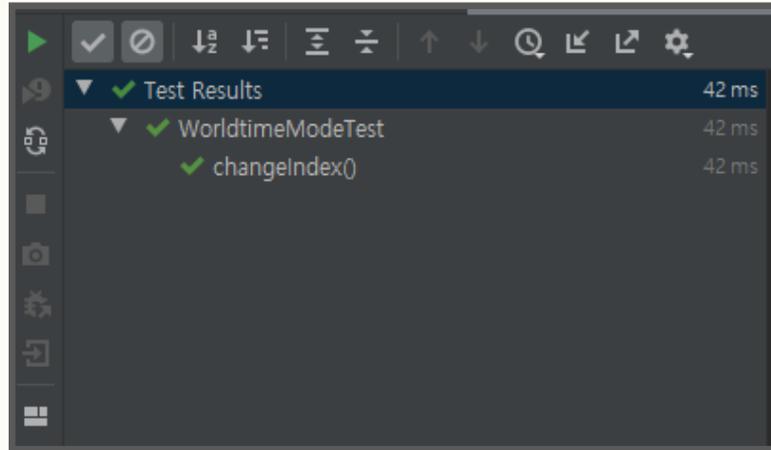
3. Timer Mode

✓  Test Results	5 s 48 ms
✓  TimerModeTest	5 s 48 ms
 resetTimer()	13 ms
 changeValue()	1 ms
 startTimer()	4 s 33 ms
 stopTimer()	1 s 1 ms

4. Alarm Mode

✓  Test Results	229 ms
✓  AlarmModeTest	229 ms
 changeValue()	12 ms
 getAlarmTime()	
 removeAlarmNum()	
 isAlarmIndicator()	2 ms
 displayTime()	215 ms

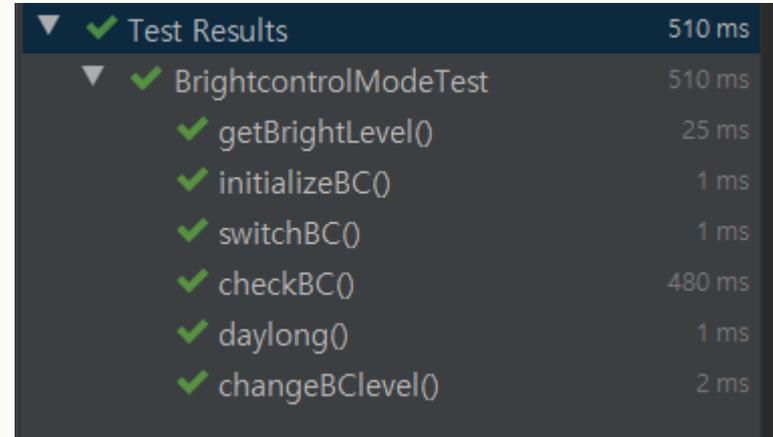
5. WorldTime Mode



A screenshot of a test runner interface. The interface has a dark background with a toolbar at the top containing various icons for navigation and settings. The test results are displayed in a tree view. The root node is 'Test Results' with a green checkmark and a duration of 42 ms. It is expanded to show a sub-node 'WorldtimeModeTest' with a green checkmark and a duration of 42 ms. This sub-node is further expanded to show a single test case 'changeIndex()' with a green checkmark and a duration of 42 ms.

✓ Test Results	42 ms
✓ WorldtimeModeTest	42 ms
✓ changeIndex()	42 ms

6. Brightcontrol Mode



A screenshot of a test runner interface showing test results for Brightcontrol Mode. The interface has a dark background. The test results are displayed in a tree view. The root node is 'Test Results' with a green checkmark and a duration of 510 ms. It is expanded to show a sub-node 'BrightcontrolModeTest' with a green checkmark and a duration of 510 ms. This sub-node is further expanded to show several test cases: 'getBrightLevel()' (25 ms), 'initializeBC()' (1 ms), 'switchBC()' (1 ms), 'checkBC()' (480 ms), 'daylong()' (1 ms), and 'changeBClevel()' (2 ms).

✓ Test Results	510 ms
✓ BrightcontrolModeTest	510 ms
✓ getBrightLevel()	25 ms
✓ initializeBC()	1 ms
✓ switchBC()	1 ms
✓ checkBC()	480 ms
✓ daylong()	1 ms
✓ changeBClevel()	2 ms

7. Software Select Mode

✓ Test Results	12 ms
✓ SWModeTest	12 ms
✓ selectSW()	12 ms
✓ changeSW()	

8. Ring test

✓ Test Results	557 ms
✓ RingTest	557 ms
✓ checkAlarm()	55 ms
✓ checkTimer()	2 ms
✓ ringing()	500 ms

Activity : 소감

김선우 : 주먹구구식으로 구현하던 때와 다르게 분석과 설계를 체계적으로 하고 나니 구현하기 편하다는 것을 느꼈습니다. 정해진 기간 동안 구현, document, presentation를 제작해보면서 개발자로 일하는 과정을 미리 경험해 본 것 같아서 좋았습니다. 팀 내 뿐만 아니라 SQA와도 충분히 communication하고, 회의 시간을 조율하고 정하는 것이 쉽지 않았지만, 이 수업이 아니었다면 대학생 때 경험하지 못할 값진 경험이었습니다.

김용현 : SQA와 협력을 통해 모델링에 대한 피드백을 받고 수정을 할 때 SQA의 꼼꼼한 덕분에 자세한 피드백을 받을 수 있어 좋았고 팀원들과 분업을 통해 코드를 작성하고 연결하는 것이 편리해서 좋았으며, 자동으로 코드를 검사하고 버그를 잡아주는 CTIP환경을 처음 접했는데 신선한 경험을 해서 좋았습니다.

김도연 : 분석, 설계 및 구현 순으로 진행되는 것과 그 과정 안에서 팀원들간의 소통의 중요성을 알게 되었습니다. 또한 SQA와의 협력을 통해 저희 팀이 미처 발견하지 못한 것과 구현이 미흡한 부분에 대한 피드백을 받아 수정할 수 있어 좋았고 새로운 tool들을 경험할 수 있어 좋았습니다. 3rd iteration을 마치고 보니 저희 팀원들과 SQA 팀은 혼자서 수행하고 끝내는게 아닌 소통하고 협력하여 하나의 목표를 이뤄내는 관계임을 알 수 있었습니다. 감사합니다.

장다혜 : 단계적 개발과정에 있어 지속적인 소통의 중요성과 팀원들간의 협업이 중요하다고 느꼈고, 새로운 도구들을 접할 수 있어 좋았습니다. 코로나로 인해 빡빡한 스케줄 내에서 개발하러니 체력적으로도 매우 지치고, 시간에 쫓긴 것 같아 아쉬웠지만, 좋은 학우분들과 함께여서 많은 것들을 배웠던 값진 시간이었습니다. ;)