

[T8] 낚시하시계

-3rd Cycle Demonstration-

201714170 조해성 201714168 유호원 201310507 정희찬
201613575 루카이 201712845 류한길

Table of Contents

1. SYSTEM TEST

- 1.1 SPECIFICATION REVIEW
- 1.2 CATEGORY-PARTITION TESTING REPORT
- 1.4 BRUTE FORCE TESTING REPORT

2. STATIC ANALYSIS TEST

- 2.1 BUGS
- 2.2 VULNERABILITY
- 2.3 CODE SMELLS
- 2.4 COVERAGE
- 2.5 DUPLICATION

3. OOPT REVIEW

1. System Test (Specification Review)

수정 및 요구사항 #160

[2nd] Stage 2040 Design * 뒤로 | 6/127 | 다음 >

admin Admin이(가) 약 12시간 전에 추가함.

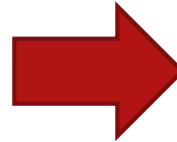
상태:	신규	시작시간:	2019/06/07
우선순위:	보통	완료기한:	2019/06/10
담당자:	-	진행도:	<div style="width: 0%;"></div> 0%
		추정시간:	

설명

Activity 2041. Define Real Use Cases.

1.(A) User가 mode 버튼을 4회 눌러 mode select로 진입한다.

=> 'reset' 버튼을 4회 눌러 '모 변경' 요청.



대응방안->

Typical Courses of Events	<p>(A): Actor / (S) : System</p> <ol style="list-style-type: none"> 1.(A) User가 reset버튼을 4번 눌러 mode select로 진입한다. 2.(S) mode select 설정 화면을 표시한다 3.(A) User가 start 버튼을 눌러 TimeKeeping 모드를 선택하거나 선택 해제한다. 4.(A) User가 mode 버튼을 눌러 Timer 모드로 화면을 넘긴다. 5.(A) User가 start 버튼을 눌러 Timer 모드를 선택하거나 선택 해제한다.
---------------------------	---

=> 잘못 명시된 버튼 이름을 수정하였다.

1. System Test (Category-partition Testing Report)

수정 및 요구사항 #161

[2nd] Category Partitional Testing report - fail result. ◀ 뒤로 | 5/127 | 다음 ▶

admin Admin이(가) 약 12시간 전에 추가함.

상태:	신규	시작시간:	2019/06/07
우선순위:	낮음	완료기한:	2019/06/10
담당자:	-	진척도:	0%
		추정시간:	

설명

1. timekeeping mode에서 유효한 알람 입력 후 reset button을 1회 눌러 변화가 없는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> reset button을 1회 눌러 변화가 있었음을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
2. timekeeping mode에서 유효한 알람 입력 후 reset button을 1회 눌러 변화가 없는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> reset button을 1회 눌러 변화가 있었음을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
3. timekeeping mode에서 유효한 알람 입력 후 reset button을 1회 눌러 변화가 없는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> reset button을 1회 눌러 변화가 있었음을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
4. timekeeping mode에서 유효한 알람 입력 후 mode button을 4회 이상 눌러 mode가 바뀌는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> mode button을 4회 이상 눌러 mode가 정상적으로 변함을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
5. timekeeping mode에서 유효한 알람 입력 후 mode button을 4회 이상 눌러 mode가 바뀌는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> mode button을 4회 이상 눌러 mode가 정상적으로 변함을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
6. timekeeping mode에서 유효한 알람 입력 후 mode button을 4회 이상 눌러 mode가 바뀌는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> mode button을 4회 이상 눌러 mode가 정상적으로 변함을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
7. timer mode에서 start button을 4회 이상 눌러 timer가 정상적으로 멈췄다 작동했다 하는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> start button을 4회 이상 눌러 timer가 정상적으로 멈췄다 다시 작동을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
8. timer mode에서 mode button을 4회 이상 눌러 mode가 바뀌는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> mode button을 4회 이상 눌러 mode가 정상적으로 변함을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
9. stopwatch mode에서 adjust button을 1회 눌러 시간이 기록되는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> adjust button을 1회 눌러 시간이 정상적으로 기록되었음을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
10. stopwatch mode에서 start button을 4회 이상 눌러 stopwatch가 정상적으로 작동하는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> start button을 4회 이상 눌러 stopwatch가 정상적으로 작동을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
11. stopwatch mode에서 mode button을 4회 이상 눌러 mode가 바뀌는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
-> mode button을 4회 이상 눌러 mode가 정상적으로 변함을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.

대응방안->

절전모드는 디지털 시계에 불필요한 Requirement라 생각하여 대응하지 않는다.

1. System Test (Brute Force Testing Report)

수정 및 요구사항 #162

[2nd] Brute force testing report - fail result. < 뒤로 | 4/127 | 다음 >

admin Admin이(가) 약 12시간 전에 추가함.

상태:	신규	시작시간:	2019/06/07
우선순위:	낮음	완료기한:	2019/06/10
담당자:	-	진척도:	<div style="width: 0%;"></div> 0%
		추정시간:	

설명

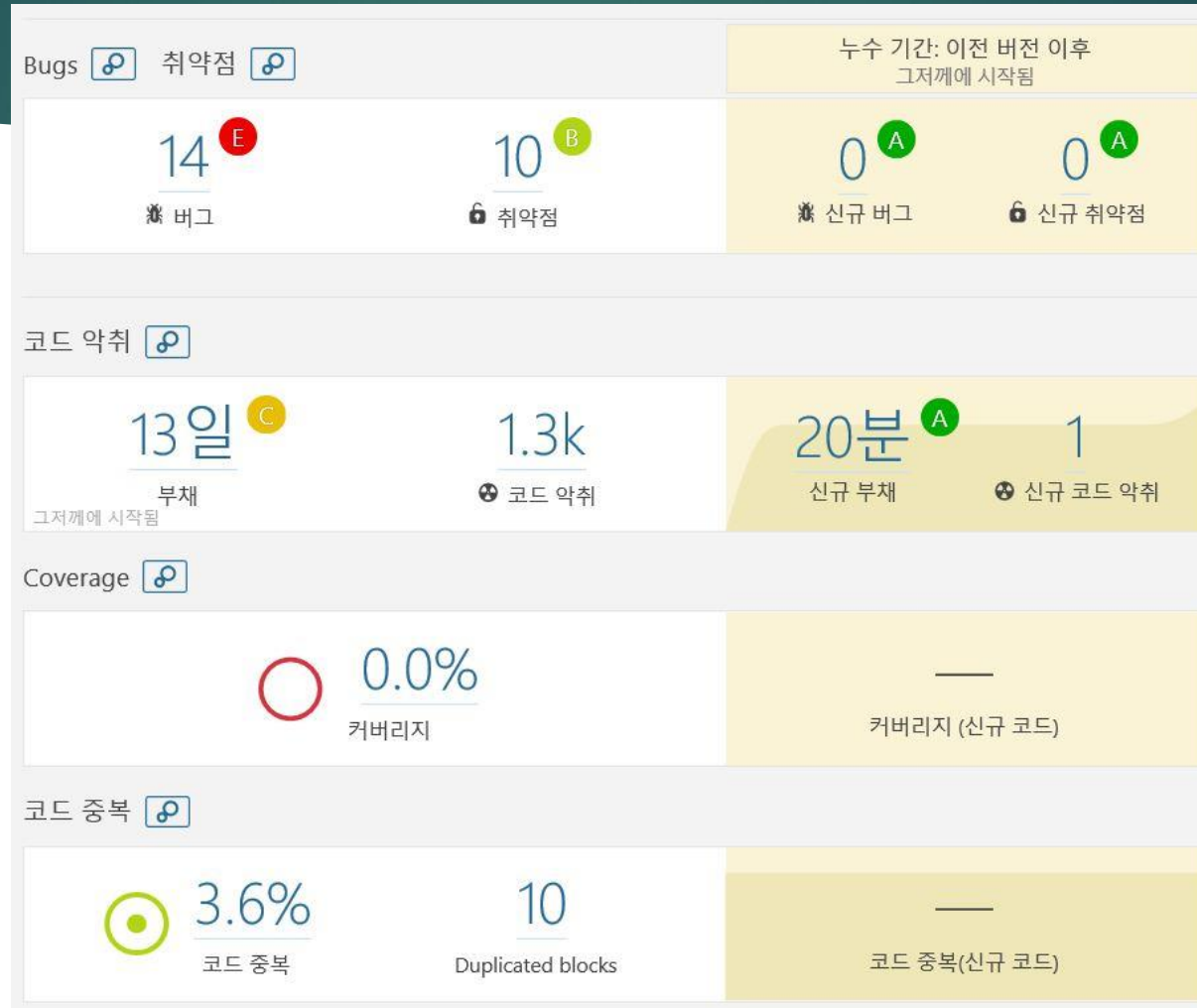
1. alarm mode에서 reset button을 4회 눌러 mode selector로 진입할 수 있는지 확인한다.

=> alarm mode에서 reset button을 4회 눌러도 mode selector로 진입할 수 없음.

대응방안->

지난 대응보고서에 Alarm모드에서 mode selector로 못가게 막아놓은 이유를 설명하고, 보고서를 수정하였기에 대응하지 않음.

2. Static Analysis Test



2. Static Analysis Test (Bugs)

설명

[Sonarqube - Bug]

☞ <http://ec2-15-164-99-14.ap-northeast-2.compute.amazonaws.com:9000/project/issues?id=SMA08&resolved=false&types=BUG>

[digital-watch/src/main/java/Alarm.java]

- 1) Either re-interrupt this method or rethrow the "InterruptedException" Bug 해결 요망.
→ thread에게 interrupt를 발생시켜 주어야 함.

[digital-watch/src/main/java/Timer.java]

- 1) Either re-interrupt this method or rethrow the "InterruptedException" Bug 해결 요망.
→ thread에게 interrupt를 발생시켜 주어야 함.

[digital-watch/src/main/java/SystemUI.java]

- 1) Either re-interrupt this method or rethrow the "InterruptedException" Bug 해결 요망.
→ thread에게 interrupt를 발생시켜 주어야 함.
- 2) Add an end condition to this loop. Bug 해결 요망.
→ true를 이용한 무한 loop문으로 이해하였는데 end condition을 추가하는 것이 가능하면 해결 바람.

cf)

- 1) Class CreateMode defines fields that are used only as locals. Bug.
→ private로 사용되는 변수/객체에 대하여 해당 Bug issue가 발생.
- 2) Unread field: oooooo Bug.
→ 해당 code block이 never read로 표현됨.
- 3) Method oooooooooooooo accesses list or array with constant index. Bug.
→ 변수를 통한 index 접근을 하는데 constant index를 사용한다고 함.
∴ false alarm으로 판단하여 해결을 권고하지 않겠음.

cf2)

- 중요도 minor에 해당하는 Bug.
→ 우선순위가 상대적으로 낮으므로 여유가 될 시 해결을 권고함.

cf3)

- 각 issue에 대해 세부 내용을 보는 방법은 첨부한 사진을 참고할 것.

📎 1.png (74.6 KB) 🗑️ admin Admin, 2019/06/07 13:55

2. Static Analysis Test (Bugs)

```
public class CreateMode {
    private ArrayList<String> createList;

    //현재 생성해야하는 모드들이 createList에 저장
    public void setCreateList(ArrayList<String> createList) {
        this.createList = createList;
        createModeByList();
    }
}
```

```
public class DeleteMode {
    private ArrayList<String> deleteList;

    //현재 삭제해야 하는 모드들을 저장해놓는 메서드
    public void setDeleteList(ArrayList<String> deleteList) {
        this.deleteList = deleteList;
        deleteModeByList();
    }
}
```

```
public class ModeSelector {
    //String[]에서 ArrayList로 수정
    private ArrayList<String> settingModeList = new ArrayList<>(); //현재 선택된 모드들의 list(최대 4개)
    private ArrayList<String> createList = new ArrayList<>(); //생성해야하는 리스트들
    private ArrayList<String> deleteList = new ArrayList<>(); //삭제해야 하는 리스트들
}
```

Unread field: OOOO

대응방안->

쓰이지 않는 필드를 제거하라는 메시지가
므로 쓰이지 않는 필드를 제거했다

2. Static Analysis Test (Bugs)

```
public class CreateMode {  
  
    //현재 생성해야하는 모드들이 createList에 저장  
    public void setCreateList(ArrayList<String> createList) {  
        createModeByList(createList);  
    }  
}
```

```
public class DeleteMode {  
  
    //현재 삭제해야 하는 모드들을 저장해놓는 메서드  
    public void setDeleteList(ArrayList<String> deleteList) {  
        deleteModeByList(deleteList);  
    }  
}
```

```
public class ModeSelector {  
    //String[]에서 ArrayList로 수정  
    private ArrayList<String> settingModeList= new ArrayList<>(); //현재 선택된 모드들의 list(최대 4개)
```

```
    public void setCreateList(ArrayList<String> createList){  
        createMode.setCreateList(createList);  
    }  
  
    public void setDeleteList(ArrayList<String> deleteList){  
        deleteMode.setDeleteList(deleteList);  
    }  
}
```

Class OOOO defines fields that are used only as local

대응방안->

private으로 설정된 필드가 하나의 메서드에서만 사용되어 문제가 있다고 판단하여 필드를 사용하는 대신 직접 전달해줄 수 있게 바꿨다

2. Static Analysis Test (Bugs)

```
try {
    Thread.sleep( millis: 1000);
} catch (InterruptedException e) {
    Thread.currentThread().interrupt();
}
```

```
@Override
public void run() {
    int ↓=0;
    // TODO Auto-generated method stub
    while(!Thread.currentThread().isInterrupted()) {
```

Either re-interrupt this method or rethrow the “InterruptedException”

```
while(!Thread.currentThread().isInterrupted()){

    try {
        buzzAlarm();
        //System.out.println(alarmState);
        if(alarmState == true) {
            toolkit.beep();
        }
        Thread.sleep( millis: 500);
    } catch (InterruptedException e){
        Thread.currentThread().interrupt();
    }
}
```

```
} catch (InterruptedException e) {
    e.printStackTrace();
    Thread.currentThread().interrupt();
}
```

대응방안->

InterruptedException 발생 시 interrupt를 다시 발생시키도록 수정하고 필요한 경우 while문을 탈출하는 것도 수정하였다

2. Static Analysis Test (Vulnerability)

[digital-watch/src/main/java/CreateMode.java]

- 1) [Store a copy of "oooooooo".] Vulnerability 해결 요망.
→ 값이 임의로 변경될 수 있으므로 *.clone()을 사용할 것을 권고함.

[digital-watch/src/main/java/DeleteMode.java]

- 1) [Store a copy of "oooooooo".] Vulnerability 해결 요망.
→ 값이 임의로 변경될 수 있으므로 *.clone()을 사용할 것을 권고함.

[digital-watch/src/main/java/ModeSelector.java]

- 1) [Store a copy of "oooooooo".] Vulnerability 해결 요망.
→ 값이 임의로 변경될 수 있으므로 *.clone()을 사용할 것을 권고함.
- 2) [Return a copy of "oooooooo".] Vulnerability 해결 요망.
→ 마찬가지로 값이 변경될 수 있으므로 *.clone()을 사용할 것을 권고함.

2. Static Analysis Test (Vulnerability)

```
//현재 생성해야하는 모드들이 createList에 저장  
public void setCreateList(ArrayList<String> createList) {  
    createModeByList(createList);  
}
```

```
//현재 삭제해야 하는 모드들을 저장해놓는 메서드  
public void setDeleteList(ArrayList<String> deleteList) {  
    deleteModeByList(deleteList);  
}
```

```
public void setSettingModeList(ArrayList<String> settingModeList){  
    this.settingModeList= (ArrayList<String>) settingModeList.clone();  
}
```

```
public ArrayList<String> getModelList(){  
    return (ArrayList<String>) this.settingModeList.clone();  
}
```

```
public void setCreateList(ArrayList<String> createList){  
    createMode.setCreateList(createList);  
}  
  
public void setDeleteList(ArrayList<String> deleteList){  
    deleteMode.setDeleteList(deleteList);  
}
```

Store a copy of "OOOO"
Return a copy of "OOOO"

대응방안->

몇몇 issue가 나온 변수는 버그수정단계에서 삭제하거나 값이 임의로 변경될 수 있으므로 .clone()과 static캐스팅을 통해 값을 넣어줄수 있게 바꿨다

2. Static Analysis Test (Vulnerability)

```
private List<String> alarmList = new ArrayList<>();  
private TimeDB timeDB;  
private String alarm; //알람 울리는지 상태 확인  
private String[] time_array;
```

```
private static class Buzzer extends Thread{  
    private Toolkit toolkit = Toolkit.getDefaultToolkit();  
    public void run() {
```

```
private TimeDB timeDB;  
private int j,k;  
private String[] time_array;  
private int year, month, date;  
private int lunarYear, lunarMonth, lunarDate;
```

Explicitly declare the visibility
for“OOOO”

대응방안->

몇몇 issue가 나온 접근 지정자를 private로
지정해 주었다.

2. Static Analysis Test (Code Smells)

[digital-watch/src/main/java/Alarm.java]
1) Add the "@Override" annotation above this method signature
→ @Override annotation 추가 요망.

[digital-watch/src/main/java/CreateMode.java]
1) Add a constructor to the class, or provide default values.
→ 생성자를 추가해주거나 혹은 default value를 할당해줄 것. 해결 요망.

[digital-watch/src/main/java/DeleteMode.java]
1) Add a constructor to the class, or provide default values.
→ 생성자를 추가해주거나 혹은 default value를 할당해줄 것. 해결 요망.

[digital-watch/src/main/java/Stopwatch.java]
1) Rename "times" which hides the field declared at line 15.
→ 같은 이름을 가진 변수가 존재함. 해결 요망.
2) Split this 160 characters long line (which is greater than 120 authorized).
→ code의 길이가 너무 길어 가독성을 해칠 수 있음. 해결 요망.
3) Missing curly brace.
→ 가급적이면 {}를 사용하여 가독성을 보장해줄 것. 해결 요망.

[digital-watch/src/main/java/SystemUI.java]
1) Move this "start" call to another method.
→ 생성자에 들어가 있는 경우 의도치 않게 상속받은 자식 클래스에서 thread를 실행시킬 수 있음. 해결 요망.
하지만 main에서 호출되는 생성자만큼 강력하게 권고하지는 않음.
2) Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.
→ Bug의 interrupt와 연관되는 함축. 해결 요망.
3) Missing curly brace.
→ 가급적이면 {}를 사용하여 가독성을 보장해줄 것. 해결 요망.
4) At most one statement is allowed per line, but 2 statements were found on this line.
→ 가급적이면 하나의 line에는 하나의 statement만 있도록 해줄 것. 해결 요망.
5) Avoid catching NullPointerException.
→ 객체가 null인 경우를 if문을 사용해서 표현할 것. 해결 요망.
이를 해결 시 자동으로 Either log or rethrow this exception. issue가 해결될 것으로 기대함.

[digital-watch/src/main/java/Tide.java]
1) Rename "i" which hides the field declared at line 8.
Rename "j" which hides the field declared at line 8.
Rename "i" which hides the field declared at line 8.
Rename "j" which hides the field declared at line 8.
→ 위에서 이미 i, j 변수가 선언되었음. 해결 요망.

[digital-watch/src/main/java/TimeDB.java]
1) Missing curly brace.
→ 가급적이면 {}를 사용해서 가독성을 보장해줄 것. 해결 요망.
2) Rename "time" which hides the field declared at line 18.
→ time이라는 변수가 이미 위에서 사용되었음. 해결 요망.

[digital-watch/src/main/java/Timer.java]
1) Make this a "static" inner class.
→ 내부 클래스의 경우 static으로 선언해줄 것. 해결 요망.
2) Either remove or fill this block of code.
→ 해당 code block을 비우거나 혹은 채울 것. bug의 interrupt와 연결되어 있으므로 해결 요망.
3) Rename "time" which hides the field declared at line 18.
→ time이라는 이름의 변수가 이미 사용되었음. 해결 요망.
4) Missing curly brace.
→ 가급적이면 {}를 사용해서 가독성을 높여줄 것. 해결 요망.

2. Static Analysis Test (Code Smells)

```
public class CreateMode {  
  
    public CreateMode(){  
  
    }  
  
    //현재 생성해야하는 모드들이 createList에 저장  
    public void setCreateList(ArrayList<String> createList) {  
        createModeByList(createList);  
    }  
}
```

```
public class DeleteMode {  
  
    public DeleteMode(){  
  
    }  
  
    //현재 삭제해야 하는 모드들을 저장해놓는 메서드  
    public void setDeleteList(ArrayList<String> deleteList) {  
        deleteModeByList(deleteList);  
    }  
}
```

Add a constructor to the class, or provide default values

대응방안->

생성자를 생성해주었다 그리고 Uninitialized field라고 나오는 해당 필드는 버그 단계에서 수정을 하며 삭제했다

2. Static Analysis Test (Code Smells)

```
private Stopwatch(String time) {  
    String[] timeS = time.split( regex: "###");  
  
    hours = Integer.parseInt(timeS[0]);  
    minutes = Integer.parseInt(timeS[1]);  
    seconds = Integer.parseInt(timeS[2]);  
    times = Integer.parseInt(timeS[3]);  
}
```

```
private Timer(){  
    hours = 0;  
    minutes=0;  
    seconds=0;  
  
    zeroState=1;  
}  
  
private Timer(String time){  
    String[] timeS = time.split( regex: "###");  
  
    hours=Integer.parseInt(timeS[0]);  
    minutes=Integer.parseInt(timeS[1]);  
    seconds=Integer.parseInt(timeS[2]);  
  
    if(time.equals("0 0 0")) {  
        zeroState=1;  
    }  
    else {zeroState=0;}  
}
```

```
private TimeDB() {  
  
    year=2010;  
    month=1;  
    day=1;  
    hour=0;  
    minute=0;  
    second=0;  
  
    setMonthMap(year);  
}  
  
private TimeDB(String time) {  
    String[] timeS = time.split( regex: "###");  
  
    year=Integer.parseInt(timeS[0]);  
    month=Integer.parseInt(timeS[1]);  
    day=Integer.parseInt(timeS[2]);  
    hour=Integer.parseInt(timeS[3]);  
    minute=Integer.parseInt(timeS[4]);  
    second=0;  
  
    setMonthMap(year);  
}
```

Remove this call from a constructor to the overridavble "OOOO" method

대응방안->

생성자에서 해당 메서드를 사용하지 않도록 수정하였다.

2. Static Analysis Test (Code Smells)

```
if (getTime().equals("0 0 0 0")) {  
    zeroState = 1;  
}
```

```
265 //자바 자체적으로 longClickListener가 없음  
266  
267 btnReset.addActionListener(new ActionListener() {
```

```
String[] timeS = time.split( regex: "##s");  
  
hours = Integer.parseInt(timeS[0]);  
minutes = Integer.parseInt(timeS[1]);  
seconds = Integer.parseInt(timeS[2]);  
times = Integer.parseInt(timeS[3]);
```

```
private static class Buzzer extends Thread{  
    private Toolkit toolkit = Toolkit.getDefaultToolkit();
```

```
1 import javax.swing.ImageIcon;  
2 import javax.swing.JButton;  
3 import javax.swing.JFrame;  
4 import javax.swing.JLabel;  
5 import javax.swing.JPanel;  
6 import java.awt.event.ActionEvent;  
7 import java.awt.event.ActionListener;  
8 import java.awt.event.MouseAdapter;  
9 import java.awt.event.MouseEvent;
```

대응방안-> 기타 여러 code smells 수정

2. Static Analysis Test (Coverage)

실패한 테스트케이스는 MoonphaseTest에서 검출되었는데, 실패한 원인은 다음과 같다.

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_201.jdk/Contents/Home/bin/java ...  
----- IntelliJ IDEA coverage runner -----  
sampling ...  
include patterns:  
exclude patterns:  
java.lang.Exception: Method calculateMoonphase() should be public
```






```
public void calculateMoonphase() {  
    try { //String type을 Date type으로 캐스팅할때 생기는 예외의 예외처리  
        timeDB=TimeDB.getInstance();  
        String currtime = timeDB.getTime();  
        SimpleDateFormat format = new SimpleDateFormat( pattern: "yyyy MM dd kk mm ss"); //String  
        Date CurrentDate = format.parse(currtime); //현재시간을 parse()를 통해 Date형으로 변환  
        Date StandardDate = format.parse(stdtime); //기준시간을 parse()를 통해 Date형으로 변환
```

대응방안-> CalculateTide 메소드를 private에서 public으로 선언하였다.

2. Static Analysis Test (Duplication)

Duplicated lines (%) 3.6% 

누수 기간: 이전 버전 이후

	Duplicated lines (%)	Duplicated lines
 digital-watch/src/main/java/ModeSelector.java	15.1%	18
 digital-watch/src/main/java/Tide.java	8.9%	45
 digital-watch/src/main/java/SystemUI.java	2.9%	42

대응방안-> format이 비슷한 code들을 없애려고 하여 제외함.

3. OOPT Review





Q&A