

Software Modeling & Analysis

낚시하시계

(Fishing Digital Watch)

3rd Cycle 대응보고서

Team 8

201714170 조해성

201714168 유호원

201310507 정희찬

201613575 루카이

201712845 류한길

Index

1. System Test

1.1 Specification Review

1.1.1 Stage 2040 Design

1.2 Category-partition Testing Report

1.2.1 Testing Case

1.2.2 Testing Result

1.3 Pairwise Testing Report

1.3.1 Testing Case

1.3.2 Testing Result

1.4 Brute Force Testing Report

1.4.1 Testing Result

2. Static Analysis Test

2.1 Bugs

2.2 Vulnerability

2.3 Code Smells

2.4 Coverage

2.5 Duplications

3. OOPT Review

1. System Test

1.1 Specification Review

1.1.1 Stage 2040 Design

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:	-	진척도:	<div style="width: 0%; border: 1px solid gray; display: inline-block;"></div> 0%
		추정시간:	

설명

Activity 2041. Define Real Use Cases.
1.(A) User가 mode 버튼을 4회 눌러 mode select로 진입한다.

=> 'reset 버튼을 4회 눌러'로 변경 요망.

대응방안->

Typical Courses of Events	(A): Actor / (S) : System 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.

1.2 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:	<div style="width: 0%;"></div> 0%
		추정시간:	

설명

- timekeeping mode에서 유효한 일을 입력 후 reset button을 1회 눌러 변화가 없는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
=> reset button을 1회 눌러 변화가 없었음을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
- timekeeping mode에서 유효한 달을 입력 후 reset button을 1회 눌러 변화가 없는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
=> reset button을 1회 눌러 변화가 없었음을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
- timekeeping mode에서 유효한 년을 입력 후 reset button을 1회 눌러 변화가 없는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
=> reset button을 1회 눌러 변화가 없었음을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
- timekeeping mode에서 유효한 일을 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 변환을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
- timekeeping mode에서 유효한 달을 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 변환을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
- timekeeping mode에서 유효한 년을 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 변환을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
- 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.
- 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 변환을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.
- 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.
- 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.
- 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 변환을 확인하였으나, sleep mode에서 1분간 대기 시 실제 1분보다 적은 시간이 흐름.

대응방안-> Fail이 뜬 부분은 모두 number 2001 (PC 절전모드)와 관련된 사항으로서 디지털 시계의 Requirement상 절전모드는 전혀 고려하지 않았고, 프로젝트 발주자(a.k.a. 교수님)도 절전모드는 필요 없을거라 하셨기에 대응하지 않는다.

1.3 Pairwise Testing Report

100% PASS

1.4 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%; height: 10px; background-color: #ccc;"></div> 0%
		추정시간:	

설명

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

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

대응방안->

대응방안->

2031. Define Essential Use Case

Use case	25.modeSelect
Actor	User
Purpose	User가 6개의 기능 중 4개를 선택해 버튼으로 모드 이동이 가능하게 한다.
Overview	User가 버튼을 눌러Select 화면으로 들어간 후, 시계의 버튼들을 이용해 4가지의 기능을 선택적으로 골라 이용할 수 있도록 한다.
Type	Evident
Cross Reference	Functional Requirements : R.6.0
Pre-Requisites	Alarm모드가 아니어야 한다.

2041. Define Real Use Cases

Use case	25. <u>modeSelect</u>
Actor	User
Purpose	User가 6개의 기능 중 4개를 선택해 버튼으로 모드 이동이 가능하게 한다.
Overview	User가 버튼을 눌러Select 화면으로 들어간 후, 시계의 버튼들을 이용해 4가지의 기능을 선택적으로 골라 이용할 수 있도록 한다.
Type	Evident
Cross Reference	Functional Requirements : R.6.0
Pre-Requisites	Alarm 모드가 아니어야 한다.

=> 현재 mode selector로 가는 방법이 reset버튼을 4번누르는건데 reset버튼을 누르면 알람이 초기화된다. 다른 Timer나 Stopwatch는 reset버튼을 통해 모드로 가는 것이 별로 큰 문제가 되지 않을 것이라 판단했지만, Alarm에서는 꽤 중요한 문제라고 판단하여 Alarm에서는 mode selector로 가지 못하게 막아두었다.

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

레드마인에 있는 정보들을 참고하여 issue 수정하였음.

2019/06/07

- 14:50 수정 및 요구사항 #165 (신규): [2nd] Static Analysis - Code Smell.
[Sonarqube - Code Smell]
<http://ec2-15-164-99-14.ap-northeast-2.compute.amazonaws.com:9000/project/issues?id=SMA08...>
admin Admin
- 14:04 수정 및 요구사항 #164 (신규): [2nd] Static Analysis - Vulnerability.
[Sonarqube - Vulnerability]
<http://ec2-15-164-99-14.ap-northeast-2.compute.amazonaws.com:9000/project/issues?id=SM...>
admin Admin
- 13:56 수정 및 요구사항 #163 (신규): [2nd] Static Analysis - Bug.
[Sonarqube - Bug]
<http://ec2-15-164-99-14.ap-northeast-2.compute.amazonaws.com:9000/project/issues?id=SMA08&resolv...>
admin Admin
- 13:37 수정 및 요구사항 #162 (신규): [2nd] Brute force testing report - fail result.
1. alarm mode에서 reset button을 4회 눌러 mode select로 진입할 수 있는지 확인한다.
=> alarm mode에서 reset button을 4회 눌러도 mode sel...
admin Admin
- 13:35 수정 및 요구사항 #161 (신규): [2nd] Category Partitional Testing report - fail result.
1. timekeeping mode에서 유효한 일을 입력 후 reset button을 1회 눌러 변화가 없는지 확인하고, sleep mode에서 1분간 대기 후 시간이 제대로 흘렀는지 확인한다.
=...
admin Admin
- 13:13 수정 및 요구사항 #160 (신규): [2nd] Stage 2040 Design
Activity 2041. Define Real Use Cases.
1. (A) User가 mode 버튼을 4회 눌러 mode select로 진입한다.
=> 'reset' 버튼을 4회 눌러'로 변경 요망.
admin Admin

2.1 Bugs

[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.

다음과 같이 검증팀에서 권고한 issue들을 위주로 코드를 수정하였다

CreateMode

```
public class CreateMode {
// ..... private String[] createList;

Class CreateMode defines fields that are used only as locals ... 19일 전 ▾ L5 🔗
Bug Major Open 할당되지 않음 correctness

Unread field: CreateMode.createList ... 19일 전 ▾ L5 🔗
Bug Major Open 할당되지 않음 30min effort performance

private ArrayList<String> createList;

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

1. Unread field: CreateMode.createList - This field is never read. Consider removing it from the class.

대응방안 ->

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

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

쓰이지 않는 필드를 제거하라는 메시지이므로 쓰이지 않는 필드인 private String[] createList; 를 제거했다

2. Class CreateMode defines fields that are used only as locals - This class defines fields that are used in a local only fashion, specifically private fields or protected fields in final classes that are accessed first in each method with a store vs. a load. This field could be replaced by one or more local variables.

대응방안 ->

```
public class CreateMode {

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

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

DeleteMode

Class DeleteMode defines fields that are used only as locals ... 19일 전 ▾ L5 🔗

🐛 Bug 🚨 Major 🔓 Open 할당되지 않음 correctness

Unread field: DeleteMode.deleteList ... 19일 전 ▾ L5 🔗

🐛 Bug 🚨 Major 🔓 Open 할당되지 않음 30min effort performance

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

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

1. Unread field: DeleteMode.deleteList - This field is never read. Consider removing it from the class.

대응방안 ->

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

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

쓰이지 않는 필드인 private String[] deleteList; 를 제거하라는 메세지이므로 제거했다

2. Class DeleteMode defines fields that are used only as locals - This class defines fields that are used in a local only fashion, specifically private fields or protected fields in final classes that are accessed first in each method with a store vs. a load. This field could be replaced by one or more local variables.

대응방안 ->


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

private으로 설정된 필드가 하나의 메서드에서만 사용되어 문제가 있다고 판단하여 필드를 사용하는 대신 deleteModeByList에 deleteList를 직접전달할 수 있게 바꿨다

ModeSelector

```
//현재 선택된 모드정보들을 갖고 있을
public class ModeSelector {
    // private String[] settingModeList;
    // private String[] createList;
    // private String[] deleteList;
    //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
}
```

1. Unused field: ModeSelector.createList, Unused field: ModeSelector.deleteList - This field is never used. Consider removing it from the class.

대응방안 ->

```
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
}
```

쓰이지 않는 필드인 private String[] settingModeList;, private String[] createList;, private String[] deleteList를 제거했다

2. Class ModeSelector defines fields that are used only as locals 두개 - This class defines fields that are used in a local only fashion, specifically private fields or protected fields in final classes that are accessed first in each method with a store vs. a load. This field could be replaced by one or more local variables.

대응방안 ->

```
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);
}
```

private으로 설정된 필드가 하나의 메서드에서만 사용되어 문제가 있다고 판단하여 기존에 있던 필드를 삭제하고 createMode.setCreateList()와 deleteMode.setDeleteList()에 인자를 바로 전달해 주는 방식으로 바꿨다

Timer

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

Either re-interrupt this method or rethrow the "InterruptedException". ...

14일 전 L27

🐛 Bug 🚨 Major ✅ Confirmed 할당되지 않음 15min effort

🔍 cwe, error-handling, multi-threading

1. Either re-interrupt this method or rethrow the "InterruptedException".-

InterruptedExceptions should never be ignored in the code, and simply logging the exception counts in this case as "ignoring". The throwing of the InterruptedExceptions clears the interrupted state of the Thread, so if the exception is not handled properly the fact that the thread was interrupted will be lost. Instead, InterruptedExceptions should either be rethrown - immediately or after cleaning up the method's state - or the thread should be re-interrupted by calling Thread.interrupt() even if this is supposed to be a single-threaded application. Any other course of action risks delaying thread shutdown and loses the information that the thread was interrupted - probably without finishing its task.

대응방안->

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

InterruptedException 발생 시 interrupt를 다시 발생시키도록 수정하였다.

Alarm

```
try {
    buzzAlarm();
    //System.out.println(alarmState);
    if(alarmState == true) {
        toolkit.bEEP();
    }
    Thread.sleep(500);
} catch (InterruptedException e) {break;}
```

Either re-interrupt this method or rethrow the "InterruptedException". ...

13일 전 ▾ L127 🔗

🐛 Bug 🚫 Major ✅ Confirmed 할당되지 않음 15min effort

🔍 cwe, error-handling, multi-threading

대응방안->

```
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();
    }
}
```

InterruptedException 발생 시 interrupt를 다시 발생시키도록 수정하였다. 이에 while문을 탈출하는 것도 interrupt를 확인하고 나가는 것으로 수정하였다.

SystemUI

Add an end condition to this loop. ...

13일 전 ▾ L1443 🔗 🔍

🐛 Bug 🚫 Blocker ✅ Confirmed 할당되지 않음 15min effort

🔍 cert

Either re-interrupt this method or rethrow the "InterruptedException". ...

15일 전 ▾ L1440 🔗 🔍

🐛 Bug 🚫 Major ✅ Confirmed 할당되지 않음 15min effort

🔍 cwe, error-handling, multi-threading

대응방안->

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

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

InterruptedException 발생 시 interrupt를 다시 발생시키도록 수정하고, while문을 탈출하는 경우를 추가해주었다.

2.2 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.
```

다음과 같이 검증팀이 권고한 issue들을 위주로 코드 수정을 진행하였다

CreateMode

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

    createModeByList();
}
```

Store a copy of "createList". ... 20일 전 ▾ L10 🔗
🔒 Vulnerability 🟢 Minor 🔵 Confirmed 할당되지 않음 5min effort 🔗 cert, cwe, unpredictable

1. Store a copy of "createList". - Mutable objects are those whose state can be changed. For instance, an array is mutable, but a String is not. Mutable class members should never be returned to a caller or accepted and stored directly. Doing so leaves you vulnerable to unexpected changes in your class state.

대응방안 ->

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

현재 issue가 나온 변수는 버그수정단계에서 삭제했다

DeleteMode

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

    deleteModeByList();
}
```

Store a copy of "deleteList". ... 20일 전 ▾ L12 🔗
🔒 Vulnerability 🟢 Minor 🔵 Confirmed 할당되지 않음 5min effort 🔗 cert, cwe, unpredictable

1. Store a copy of "createList". - Mutable objects are those whose state can be changed. For instance, an array is mutable, but a String is not. Mutable class members should never be returned to a caller or accepted and stored directly. Doing so leaves you vulnerable to unexpected changes in your class state.

대응방안 ->

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

현재 issue가 나온 변수는 버그수정단계에서 삭제했다

ModeSelector

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

Store a copy of "settingModeList". ... 20일 전 ▾ L45 🔗
🔒 Vulnerability 🟢 Minor 🟡 Confirmed 할당되지 않음 5min effort 🏷️ cert, cwe, unpredictable

//      createMode.setCreatelist(createList);
//      deleteMode.setDeletelist(deleteList);
}
```

1. Store a copy of "createList". - Mutable objects are those whose state can be changed. For instance, an array is mutable, but a String is not. Mutable class members should never be returned to a caller or accepted and stored directly. Doing so leaves you vulnerable to unexpected changes in your class state.

대응방안 ->

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

값이 임의로 변경될 수 있으므로 .clone()과 static캐스팅을 통해 this.settingModeList에 값을 넣어줄 수 있게 바꿨다

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

Return a copy of "settingModeList". ... 20일 전 ▾ L54 🔗
🔒 Vulnerability 🟢 Minor 🟡 Confirmed 할당되지 않음 5min effort 🏷️ cert, cwe, unpredictable

}
```

2. Return a copy of "settingModeList". - Mutable objects are those whose state can be changed. For instance, an array is mutable, but a String is not. Mutable class members should never be returned to a caller or accepted and stored directly. Doing so leaves you vulnerable to unexpected changes in your class state.

대응방안 ->

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

리턴할때 값이 임의로 변경될 수 있으므로 .clone()과 static캐스팅을 통해 settingModeList의 값을 리턴시킬 수 있게 했다

```

public void setCreateList(ArrayList<String> createList){
    this.createList=createList;
}

createMode.setCreateList(this.createList);
}

public void setDeleteList(ArrayList<String> deleteList){
    this.deleteList=deleteList;
}

deleteMode.setDeleteList(this.deleteList);
}

```

3. Store a copy of “createList.” / store a copy of “deleteList” - Mutable objects are those whose state can be changed. For instance, an array is mutable, but a String is not. Mutable class members should never be returned to a caller or accepted and stored directly. Doing so leaves you vulnerable to unexpected changes in your class state.

대응방안 ->

```

public void setCreateList(ArrayList<String> createList){
    createMode.setCreateList(createList);
}

public void setDeleteList(ArrayList<String> deleteList){
    deleteMode.setDeleteList(deleteList);
}

```

현재 issue가 나온 변수들은 버그수정단계에서 삭제되었다

Alarm

```

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

Explicitly declare the visibility for "alarmList". ... 15일 전 L6 No tags
Vulnerability Minor Open 할당되지 않음 5min effort

Explicitly declare the visibility for "time_array". ... 13일 전 L9 No tags
Vulnerability Minor Open 할당되지 않음 5min effort

```

대응방안->

```

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

```

alarmList와 time_array의 접근 지정자를 private로 설정해주었다.

Timer

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

Explicitly declare the visibility for "toolkit". ...

15일 전 L21

Vulnerability Minor Open 할당되지 않음 5min effort

No tags

대응방안->

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

toolkit의 접근 지정자를 private로 지정해 주었다.

Tide

digital-watch/src/main/java/Tide.java

Explicitly declare the visibility for "time_array". ...

13일 전 L9

Vulnerability Minor Open 할당되지 않음 5min effort

No tags

대응방안->

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

time_array의 접근 지정자를 private로 지정해 주었다.

2.3 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 항목. 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.

다음과 같이 검증팀에서 권고한 issue들 위주로 코드 수정을 진행하였다

CreateMode

```
public class CreateMode {
```

Add a constructor to the class, or provide default values. ...

20일 전 L4

Code Smell Major Open 할당되지 않음 5min effort

pitfall

```
Uninitialized field String[] createList;  
1 private ArrayList<String> createList;
```

1. Add a Constructor to the class, or provide default values.

대응방안 ->

```

public class CreateMode {

    public CreateMode(){

    }

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

```

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

DeleteMode

```

Add a constructor to the class, or provide default values. ... 20일 전 ▾ L6 🔗
🚩 Code Smell 🚩 Major 🔵 Open 할당되지 않음 5min effort 🗑️ pitfall

// private String[] deleteList;
1 private ArrayList<String> deleteList;

```

1. Add a Constructor to the class, or provide default values.

대응방안 ->

```

public class DeleteMode {

    public DeleteMode(){

    }

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

```

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

Stopwatch

```

public void pauseStopwatch() {
    service.shutdown();
    this.runState = 0;
    if (getTime().equals("0 0 0 0"))
        zeroState = 1;
}

```

```

Missing curly brace. ... 18일 전 ▾ L100 🔗
🚩 Code Smell 🚩 Critical 🔵 Open 할당되지 않음 2min effort 🗑️ based-on-misra, cert, pitfall

```

1. Missing curly brace.

대응방안->

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

{}를 추가해주었다.

```
public String getTime() {  
    String time = Integer.toString(this.hours) + " " + Integer.toString(this.minutes) +
```

Change this instance-reference to a static reference. ... 17일 전 L68
Code Smell Major Open 할당되지 않음 5min effort pitfall

Split this 160 characters long line (which is greater than 120 authorized). ... 17일 전 L68
Code Smell Major Open 할당되지 않음 1min effort convention

2. Change this instance-reference to a static reference.

대응방안-> static변수에 접근하는 것이라서 "this."를 삭제했다. 이후 같은 이슈는 모두 this. 를 삭제하는 방식으로 수정하였다.

3. split this 160 characters long line

대응방안-> 불필요한 부분을 삭제했다.

```
String time = hours + " " + minutes + " " + seconds + " " + times;
```

```
public void setStopwatch(String time) {  
    String[] times = time.split("\\s");
```

Rename "times" which hides the field declared at line 15. ... 18일 전 L74
Code Smell Major Open 할당되지 않음 5min effort based-on-misra, cert, pitfall, suspicious

4. 중복되는 변수 이름

대응방안->

```
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 int times; 100분의 1초를 세기 위한 변수와 이름이 겹치므로 해당부분을 times에서 timeS로 수정해주었다.

Remove this call from a constructor to the overridable "setStopwatch" method. ...

18일 전 L44

Code Smell Critical Open 할당되지 않음 10min effort

cert, pitfall

5. 생성자에 override될 수 있는 메서드가 사용되어 이후 초기화시 문제가 발생할 수 있음

대응방안->

```
private Stopwatch(String time) {
    String[] timeS = time.split(" ");

    hours = Integer.parseInt(timeS[0]);
    minutes = Integer.parseInt(timeS[1]);
    seconds = Integer.parseInt(timeS[2]);
    times = Integer.parseInt(timeS[3]);
}
```

생성자에서 해당 메서드를 사용하지 않도록 수정하였다.

SystemUI

```
265 //자바 자체적으로 LongClickListener가 없음
266 btnMode.addMouseListener(new MouseAdapter() {
267     @Override
268     public void mousePressed(MouseEvent e) {
269
270     }
271
272     @Override
273     public void mouseReleased(MouseEvent e) {
274         super.mouseReleased(e);
275     }
276 });
277
278 btnReset.addActionListener(new ActionListener() {
```

Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation. ...

17일 전 L268

Code Smell Critical Open 할당되지 않음 5min effort

suspicious

1. Add a nested comment explaining why this method is empty. throw an UnsupportedOperationException or complete this implementation.

대응방안 ->

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

사용하지 않는 리스너이므로 삭제했다

```

467     if(str[1].equals("1")) str[1]="JANUARY";
Missing curly brace. ... 14일 전 L467
Code Smell Critical Open 할당되지 않음 2min effort based-on-misra, cert, pitfall

At most one statement is allowed per line, but 2 statements were found on this line. ... 14일 전 L467
Code Smell Major Open 할당되지 않음 1min effort style

468     else if(str[1].equals("2")) str[1]="FEBRUARY";
Missing curly brace. ... 14일 전 L468
Code Smell Critical Open 할당되지 않음 2min effort based-on-misra, cert, pitfall

At most one statement is allowed per line, but 2 statements were found on this line. ... 14일 전 L468
Code Smell Major Open 할당되지 않음 1min effort style

469     else if(str[1].equals("3")) str[1]="MARCH";
Missing curly brace. ... 14일 전 L469
Code Smell Critical Open 할당되지 않음 2min effort based-on-misra, cert, pitfall

At most one statement is allowed per line, but 2 statements were found on this line. ... 14일 전 L469
Code Smell Major Open 할당되지 않음 1min effort style

470     else if(str[1].equals("4")) str[1]="APRIL";

```

위의 issue는 467~495줄까지 반복

```

else
Missing curly brace. ... 14일 전 L623
Code Smell Critical Open 할당되지 않음 2min effort based-on-misra, cert, pitfall

    strr=String.format("%02d:%02d:%02d",Integer.parseInt(str[0]),Integer.parseInt(str[1]),
        Integer.parseInt(str[2]));
    lblTime.setText(strr);

1073     if(ur1!=null)
Missing curly brace. ... 7일 전 L1073
Code Smell Critical Open 할당되지 않음 2min effort based-on-misra, cert, pitfall

1074     img=new ImageIcon(ur1); // 물 모양 표시
1075     else
Missing curly brace. ... 7일 전 L1075
Code Smell Critical Open 할당되지 않음 2min effort based-on-misra, cert, pitfall

1367     if(ur1!=null)
Missing curly brace. ... 7일 전 L1367
Code Smell Critical Open 할당되지 않음 2min effort based-on-misra, cert, pitfall

1368     img=new ImageIcon(ur1); // 조수 모양 표시
1369     else
Missing curly brace. ... 7일 전 L1369
Code Smell Critical Open 할당되지 않음 2min effort based-on-misra, cert, pitfall

```

2. Missing curly brace / At most one statement is allowed per line, but 2 statements were found on this line. - While not technically incorrect, the omission of curly braces can be misleading, and may lead to the introduction of errors during maintenance / For better readability, do not put more than one statement on a single line.

대응방안 ->

```
if(str[1].equals("1")){
    str[1]="JANUARY";
}
else if(str[1].equals("2")){
    str[1]="FEBRUARY";
}
else if(str[1].equals("3")){
    str[1]="MARCH";
}
else if(str[1].equals("4")){
    str[1]="APRIL";
}
else if(str[1].equals("5")){
    str[1]="MAY";
}
else if(str[1].equals("6")){
    str[1]="JUNE";
}
else if(str[1].equals("7")){
    str[1]="JULY";
}
else if(str[1].equals("8")){
    str[1]="AUGUST";
}
else if(str[1].equals("9")){
    str[1]="SEPTEMBER";
}
else if(str[1].equals("10")){
    str[1]="OCTOBER";
}
}
```

```
else if(str[1].equals("11")){
    str[1]="NOVEMBER";
}
else if(str[1].equals("12")){
    str[1]="DECEMBER";
}
else{
    str[1]="이걸 보고있다면 버그임";
}

if(timekeepingAdjustState){
    strr=String.format("%02d:%02d:%02d",hour,minute, second);
    lblTime.setText(strr);
    if(month==1){
        str[1]="JANUARY";
    }
    else if(month==2){
        str[1]="FEBRUARY";
    }
    else if(month==3){
        str[1]="MARCH";
    }
    else if(month==4){
        str[1]="APRIL";
    }
    else if(month==5){
        str[1]="MAY";
    }
}
```

```

else if(month==5){
    str[1]="MAY";
}
else if(month==6){
    str[1]="JUNE";
}
else if(month==7){
    str[1]="JULY";
}
else if(month==8){
    str[1]="AUGUST";
}
else if(month==9){
    str[1]="SEPTEMBER";
}
else if(month==10){
    str[1]="OCTOBER";
}
else if(month==11){
    str[1]="NOVEMBER";
}
else if(month==12){
    str[1]="DECEMBER";
}

```

```

else{
    strr=String.format("%02d:%02d:%02d", Integer.parseInt(str[0]), Integer.parseInt(str[1]),
        Integer.parseInt(str[2]));
}
lblTime.setText(strr);

```

```

if(url!=null) {
    img = new ImageIcon(url); // 달 모양 표시
}
else {
    img = new ImageIcon(moonphase.getMoonphase());
}

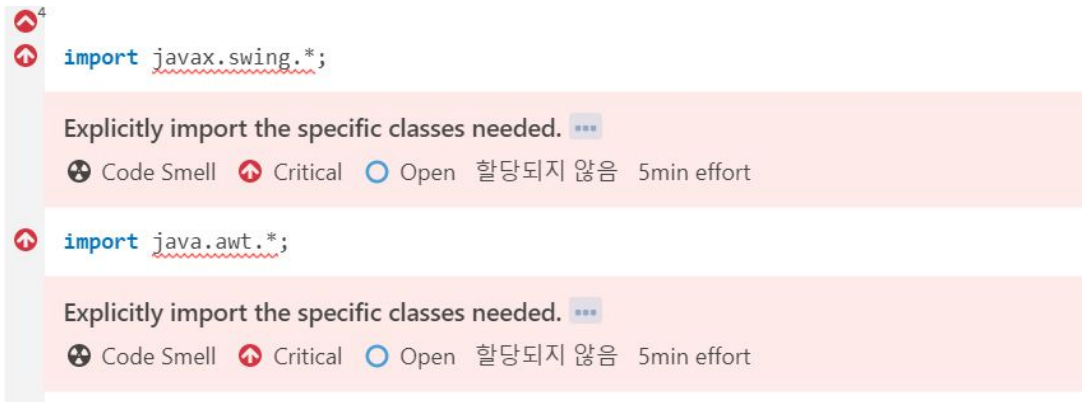
```

```

if(url!=null) {
    img = new ImageIcon(url); // 조수 모양 표시
}
else {
    img = new ImageIcon(tide.getTide());
}

```

해당 issue들이 두개씩 코드의 467~495줄까지 반복되고 SystemUI의 다른 줄에도 나타나 전체적으로 중괄호{}와 한 라인에 한 개의 코드를 넣어줌으로써 수정했다



3. Explicitly import the specific classes needed.

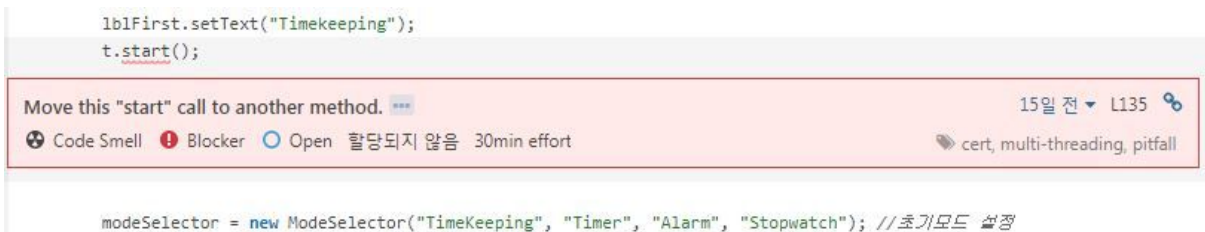
대응방안->

```

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;

```

불필요한 import를 제거하고 클래스를 분리시켜주었다.



4. Move this “start” call to another method.

대응방안-> SystemUI가 생성자로 생성됨과 동시에 스레드가 시작해야 하므로 수정하지 않았다.

Tide

```
341 youhw...      int[] dt = new int[221];
342               int td1, td2, k11, td, td0, t1, t2, jcount, m2, m1, m0, w, t11, tj1;
343               for (int i = 0; i < matchTable.length; i++) {

Rename "i" which hides the field declared at line 8. ... 14일 전 L343
☛ Code Smell 🚨 Major 🔓 Open 할당되지 않음 5min effort based-on-misra, cert, pitfall, suspicious

344               dt[i] = 0;
345               for (int j = 0; j < 12; j++) {

Rename "j" which hides the field declared at line 8. ... 14일 전 L345
☛ Code Smell 🚨 Major 🔓 Open 할당되지 않음 5min effort based-on-misra, cert, pitfall, suspicious
```

1. Rename "i" which hides the field declared at line 8

Rename "j" which hides the field declared at line 8

대응방안->

```
public class Tide {
    //private String currtime="2019 05 25 00 00 00"; //현재시간
    private int[] tideList={0,0,0};
    private int estide;
    private int wtide;
    private String sea;
    private TimeDB timeDB;
    private int k;
    String[] time_array;
    private int year, month, date;
    private int lunarYear, lunarMonth, lunarDate;
    private boolean leap;
}
```

위에서 변수가 선언되었지만 한번도 사용되지 않았기에 변수 선언 삭제

TimeDB

```
65               public void setMonthMap(int year) {
66                   //그레고리력 규칙 참고
67                   if(year%4==0) {
68                       if(year%100!=0)

Missing curly brace. ... 17일 전 L68
☛ Code Smell 🚨 Critical 🔓 Open 할당되지 않음 2min effort based-on-misra, cert, pitfall
```

1. Missing curly brace

```
if(year%4==0) {
    if(year%100!=0) {
        monthMap.put(2, 29);
    }
    else {
        if(year%400==0) {
            monthMap.put(2, 29);
        }
        else {monthMap.put(2, 28);}
    }
}
else {monthMap.put(2, 28);}
```

가독성을 위해 중괄호를 추가했다

```
91 public String getTime() {
92     String time= year+" "+month+" "+day+" "+hour+" "+minute+" "+second;
    }
}

Rename "time" which hides the field declared at line 18. ...
13일 전 L92
Code Smell Major Open 할당되지 않음 5min effort based-on-misra, cert, pitfall, suspicious
```

2. Rename "time" which hides the field declared at line 18.

```
14 private Thread thread;
15 private TimeDB runnable;
16 private ScheduledExecutorService service;
17
18 private int times =0;
```

먼저 선언된 time의 이름을 바꾸었다 이에 따라 발생하는 중복된 변수이름 문제도 Stopwatch와 같이 수정하였다.

3. 생성자에 override될 수 있는 메서드가 사용되어 이후 초기화시 문제가 발생할 수 있다. 대응방안->

```
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: \"###s\");

    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);
}
```

생성자에서 해당 메서드를 사용하지 않도록 변경하였다. setMonthMap은 이미 정해진 달과 일을 매핑하는 것으로 변경 가능성이 없기 때문에 같은 이슈가 발생하더라도 문제가 발생하지 않을 것이라고 생각해 변경하지 않았다.

Timer

```
private class Buzzer extends Thread{
```

Make this a "static" inner class. ...

🚫 Code Smell 🚩 Major 🔵 Open 할당되지 않음 15min effort

1. Make this a "static" inner class

대응방안->

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

outer class에 대한 reference는 없으므로 static으로 선언해주었다.

```
        Thread.sleep(1000);  
    }catch (InterruptedException e) {}
```

Either remove or fill this block of code. ...

🚫 Code Smell 🚩 Major 🔵 Open 할당되지 않음 5min effort

2. Either remove or fill this block of code

대응방안->

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

위의 bugs의 InterruptedException 문제를 해결하며 해결된 사항이다.

```
public String getTime() {  
    String time= Integer.toString(hours)+" "+Integer.toString(
```

Rename "time" which hides the field declared at line 18. ...

🚫 Code Smell 🚩 Major 🔵 Open 할당되지 않음 5min effort

👉 based-on-misr

3. 중복되는 변수 이름 문제

```
private int times =0;
```

대응방안->앞서 선언된 중복되는 변수 time을 times로 이름을 바꿔주었다. 이에 따라 발생하는 또다른 중복 이름 문제가 생기지 않도록 다른 코드도 수정해주었다.

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

Missing curly brace. ...

🚫 Code Smell 🚨 Critical 🔵 Open 할당되지 않음 2min effort

4. Missing curly brace.

대응방안->

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

{}를 추가해주었다.

```
private Timer(){
    setTimer("0 0 0");
}
```

Remove this call from a constructor to the overridable "setTimer" method. ... 18일 전 L52 🔗

🚫 Code Smell 🚨 Critical 🔵 Open 할당되지 않음 10min effort cert, pitfall

```
private Timer(String time){
    setTimer(time);
}
```

Remove this call from a constructor to the overridable "setTimer" method. ... 18일 전 L56 🔗

🚫 Code Smell 🚨 Critical 🔵 Open 할당되지 않음 10min effort cert, pitfall

5. 생성자에 override될 수 있는 메서드가 사용되어 이후 초기화시 문제가 발생할 수 있다.

해결방안->

```
private Timer(){
    hours = 0;
    minutes=0;
    seconds=0;

    zeroState=1;
}

private Timer(String time){
    String[] timeS = time.split("regex: \"\\W\\Ws\"");

    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;}
}
```

생성자에서 해당 메서드를 사용하지 않도록 변경하였다.

2.4 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
```

대응방안->




```
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
```

CalculateTide 메소드를 private에서 public으로 선언하였다.

2.5 Duplications

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

조해성-팀프로젝트의 진수를 맞본 느낌, 여태까지와의 팀플과는 차원이 다른 단계별로 하나의 프로젝트를 완성시키는 나름 재밌는 경험이었다. 소프트웨어공학이라는 내가 생각했을때 추상적이었던 학문이 좀 더 구체적으로 다가온 느낌

류한길-소프트웨어 모델링에 대한 프로세스를 구체적으로 잘 알게 되었음. 프로젝트를 만들면서 정밀하게 구성된 언어로 팀원과 소통하는 것이 재밌는 경험이었음. 더불어 팀원과의 소통이 많이 중요함을 알게 되었음. 팀플은 언제나 팀의 그림과 자신의 그림을 일치시키는 것에 신경써야 함을 잘 알게 되었음. 전체적으로 고생하기도 했지만 재밌었음.

정희찬 - git을 사용하는데 상당히 애를 먹었다. UI가 매우 두껍다보니 모든 팀원들이 UI를 건드리게 되어 버전이 종종 꼬였었다. 8학기를 다니면서 탑3안에 드는 뽕센 팀플이 아니었을까 싶은 생각이 든다. 그러나 대체로 막히는 부분 없이 구현되어, 모델링이 탄탄하게 된다면 설계상 미스가 났을법한 상황도 해결할 수 있음을 알게 되었다.

유호원- 소프트웨어 공학에서 배우던 내용들을 가지고 실제로 실습을 해보니 이론수업으로는 느낄수 없었던 여러가지를 느끼고 처음 소프트웨어 설계가 얼마나 중요한지 뼈저리게 느꼈다.

루카이 - 어려웠지만 잘 된 것 같고 좋았다.

정적분석이랑 시스템테스팅이랑 다르게 하면 뭐가 좋을까?

- 일단 시스템 테스트를 통해 전체적으로 현재 진행하고 있던 프로젝트의 문서의 일관성이라던가 버그의 현황을 파악하고 그 다음단계로 정적분석을 통해 좀 더 디테일한 코드의 문제를 수정하는게 여태까지 모델링수업을 통해 해왔던 추상화 -> 코드를 통한 구체화에 적합해 보인다
- 정적분석단계에서 코드의 많은 수정을 하지 않기 위해 시스템 테스트단계까지 확실하게 issue들을 잡고 가는게 좋아 보인다