

Static Analysis Report for Team B5

Project Team

Team 4

Latest update on:

2020-06-17

Team Information

201410546 김태형

201611251 공민정

201611276 이규은

201611309 최지현

Table of Contents

- 1. Static Analysis
 - 1.1 Overview
 - 1.2 Bugs
 - 1.3 Vulnerability
 - 1.4 Code Smells
 - 1.5 Coverage

1.1 Overview

22 🐛 New Bugs

Reliability **E**

29 🔒 New Vulnerabilities

Security **B**

0 🛡️ New Security Hotspots

— Reviewed

Security Review **A**

69d Added Debt

3.3k 🐛 New Code Smells

Maintainability **E**

▼ Type **BUG** Clear

🐛 Bug	22
🔒 Vulnerability	29
🐛 Code Smell	3.3k

Ctrl + click to add to selection

▼ Severity

🔴 Blocker	7	🟢 Minor	2
🔴 Critical	0	🔵 Info	0
🔴 Major	13		

▼ Type **VULNERABILITY** Clear

🐛 Bug	22
🔒 Vulnerability	29
🐛 Code Smell	3.3k

Ctrl + click to add to selection

▼ Severity

🔴 Blocker	0	🟢 Minor	29
🔴 Critical	0	🔵 Info	0
🔴 Major	0		

▼ Type **CODE SMELL** Clear

🐛 Bug	22
🔒 Vulnerability	29
🐛 Code Smell	3.3k

Ctrl + click to add to selection

▼ Severity

🔴 Blocker	29	🟢 Minor	2.1k
🔴 Critical	56	🔵 Info	41
🔴 Major	1k		



해당 보고서 [링크](#)¹

¹ 로그인 방법은 slack 에 업데이트 하겠습니다.

1.2 Bugs

- Blocker(7)

Add an end condition to this loop. (False Alarms)

src/AlarmMode.java

□ Add an end condition to this loop. Why is this an issue? 13 days ago ▾ L49 🔗

🔊 Bug ▾ 🚫 Blocker ▾ 🔵 Open ▾ Not assigned ▾ 15min effort Comment 🔑 cert ▾

- 모두 spinning 하는 스레드에 대한 False Alarm 임.

분석

```
alarmChecker = new Thread(new Runnable() {
    @Override
    public void run() {
        while(true) {
            try {
                Thread.sleep(6000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            if(checkRingAlarmExist()){
                try {
                    Thread.sleep(60000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }
});
alarmChecker.start();
}
```

Add an end condition to this loop. Why is this an issue? 13 days ago ▾ L49 🔗

🔊 Bug ▾ 🚫 Blocker ▾ 🔵 Open ▾ Not assigned ▾ 15min effort Comment 🔑 cert ▾

위의 무한 루프문은 새 스레드에서 수행되는 코드로 무한히 루프하는 것이 스레드의 목적에 적절하므로 버그라고 볼 수 없다. (무시해도 좋습니다.)

- Miner(2)

Use "Thread.currentThread().getContextClassLoader()" instead. Why is this an issue? 13 days ago ▾ L25 🔗

🔊 Bug ▾ 🟢 Minor ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment 🔑 jee ▾

- this.getClass().getClassLoader() 대신
Thread.currentThread().getContextClassLoader() 사용을 권장하는 내용.

- 원인 : getClassLoader() 는 JEE 문맥에 따라 적절한 class 를 반환하지 않을 가능성이 있기 때문.

- Major(13)

Either re-interrupt this method or rethrow the "InterruptedException". (11)

- 원인 : 스레드에서 발생한 exception 은 단순 logging 처리로 끝나면 안되고, exception 을 throw 하거나 re-interrupt 해야 한다는 경고문.

분석

```

@Test public void buzzerMultifulRingTest()
{
    Buzzer buzzer = new Buzzer();
    buzzer.OnBuzzer();
    try {
        Thread.sleep(15000);
    } catch (InterruptedException e) {

        Either re-interrupt this method or rethrow the "InterruptedException". Why is this an issue? 4 hours ago L80
        Bug Major Open Not assigned 15min effort Comment cwe, error-handling, multi-threading

        e.printStackTrace();
    }
    buzzer.OnBuzzer();
}

```

테스트문의 일부에서 스레드에서 발생한 예외처리를 단순 logging 으로 끝내고 있음. Thread.currentThread().interrupt(); 문을 추가하여 re-interrupting 을 권장함.

Unused field: Alarm.ringTime (1)

Alarm Class 의 ring Time 필드는 사용되지 않았으므로 삭제.

new Buzzer() invokes Thread.start() (1)

생성자에서 스레드를 실행하는 것은 Buzzer 클래스를 상속받는 클래스의 생성자가 있을 경우, 해당클래스의 생성자가 실행되기 전에 스레드가 먼저 실행될 수 있고 오작동의 원인이 될 수 있음.

1.2 Vulnerability

- Minor (29)

Use a logger to log this exception.

- Throwable.printStackTrace 는 민감한 정보를 출력할 수 있으므로 개발자가 정의한 Logger 를 사용하기를 권장하는 내용.

Explicitly declare the visibility for "hour"

- 일부 명시적으로 visibility (public, private ...)가 정의되지 않은 field 에 대해 명시적인 visibility 사용을 권장하는 에러임.

Make this "public static instance" field final

```
public class TimeManager {  
    public static TimeManager instance = null;  
}
```

Make this "public static instance" field final Why is this an issue?

5 hours ago ▾ L6 🔗

🔒 Vulnerability ▾ 🟢 Minor ▾ 🔵 Open ▾ Not assigned ▾ 20min effort Comment

🔗 cert, cwe ▾

어느 객체에서도 instance 의 값을 변경할 수 있으므로 위험함을 경고하는 에러.
해당 문제를 해결하는 싱글턴 패턴은 [링크](#)를 참조하면 좋을 것으로 보입니다.

1.4 Code smell

- Blocker (29)

Add or update the header of this file.

- 파일 헤더부분에 라이선스나 소유권에 대한 상세를 적기를 권장하는 에러입니다.
무시해도 좋습니다. (추후 false positive 로 변경할 예정임.)

Move this "start" call to another method.

위의 new Buzzer() invokes Thread.start() 와 동일한 에러임.

- Critical(56)

Missing curly brace.

- if 문에서 {} 를 생략하는 것은 문법적으로 틀리지 않지만 버그의 원인이 될 수 있음을 지적함. 무시해도 좋습니다.

이외에는 짜잘한 코딩컨벤션에 대한 내용이 대부분입니다.

1.5 Code Coverage

1. Coverage

Element	Class, %	Method, %	Line, %
Alarm	100% (1/1)	37% (3/8)	50% (8/16)
AlarmMode	62% (5/8)	47% (17/36)	55% (62/112)
ButtonActionCallback			
Buzzer	100% (4/4)	90% (9/10)	79% (38/48)
CurrentTimeMode	100% (2/2)	72% (8/11)	79% (23/29)
MethodCallback			
Mode			
ModeManager	100% (3/3)	91% (22/24)	87% (95/108)
ModeManagerInteracter	100% (1/1)	100% (7/7)	100% (17/17)
Segment	100% (1/1)	80% (8/10)	92% (25/27)
StandardCallback			
StopWatchMode	100% (14/14)	65% (29/44)	75% (88/117)
System	0% (0/1)	0% (0/10)	0% (0/18)
Theme	100% (1/1)	100% (4/4)	100% (8/8)
ThemeMode	100% (1/1)	53% (7/13)	73% (30/41)
Time	100% (1/1)	47% (11/23)	59% (62/104)
TimeManager	100% (1/1)	100% (5/5)	100% (9/9)
TimerMode	100% (14/14)	63% (30/47)	66% (95/143)
WatchGUI	0% (0/7)	0% (0/14)	0% (0/105)
World	100% (1/1)	100% (1/1)	100% (4/4)
WorldTimeMode	100% (2/2)	52% (9/17)	61% (47/76)

- GUI, Callback 클래스를 제외한 대부분의 Class 에 대해 50% 이상의 Method Coverage 를 보였다.
- Time, Alarm Class 의 Cover 되지 않은 일부 Method 를 위한 UnitTest 작성이 필요해보임.

2. TestPass rate

▼ ✓ <default package>	50 s 35 ms
▼ ✓ AlarmTest	34 ms
✓ increaseOneHour()	30 ms
✓ increasefiveMinute()	1 ms
✓ deleteAlarm()	1 ms
✓ decideAlarm()	
✓ alarminitTime()	1 ms
✓ nextAlarm()	1 ms
▼ ✓ StopwatchTest	2 s 971 ms
✓ resetTimer()	4 ms
✓ isResetRun()	1 ms
✓ pauseStopwatch()	2 ms
✓ maxStopWatch()	1 s 51 ms
✓ startStopWatch()	1 s 912 ms
✓ continueStopwatch()	1 ms
▼ ✓ SystemTest	46 s 14 ms
✓ backToMainTest()	4 ms
✓ indexIncreaseTest()	1 ms
✓ buzzerMultifulRingTest()	46 s 3 ms
✓ modeConfigTest()	2 ms
✓ indexDecreaseTest()	1 ms
✓ useCurrentModeTest()	
✓ segmentLowerTest()	
✓ initWatchTest()	2 ms
✓ segmentUpperTest()	1 ms
▼ ✓ ThemeTest	
✓ prevTheme()	
✓ nextTheme()	
✓ initThemeMode()	

▼	✓	TimerTest	1 s 16 ms
	✓	continueTimer()	1 ms
	✓	startTimer()	1 s 13 ms
	✓	cancelTimer()	
	✓	maxIncreaseMinute()	1 ms
	✓	increaseTimerMinute()	
	✓	pauseTimer()	1 ms
	✓	maxIncreaseSeconds()	
	✓	increaseTimerSeconds()	
	✓	checkZeroNotStart()	
▼	✓	WorldTimeTest	
	✓	decreaseWorldTimeIndex()	
	✓	increaseWorldTimeIndex()	
	✓	holdCurrentWorldTime()	
	✓	releaseCurrentWorldTimeRock()	
	✓	initWorldTimeMode()	

- Test Pass Rate : 100.0%