언제 할지 알려주시계

OOPT 3rd Cycle

Project Team
T5

Date
2019-06-03

Team Information

201511243 김동언

201511262 박우진

201511284 이종빈

201511295 조범석

1. Specification Review

1.1. Stage 2030 Analysis

Issue	[2031] Use Case 16
Accept/Reject	Accept
Description	다른 Stage 2030들과는 다르게, Stage 2040부터 나타나는 'D 버튼'과 같은 특정 버튼이 언급되어 있어 일관성을 해침.
Co-respond	마지막 알람일 때, 버튼을 누르면 첫번째 알람을 보여준다로 수정.

Issue	[2031] Use Case 17
Accept/Reject	Accept
Description	Typical Courses of Events와 비교해봤을 때, Alternative Courses of Events와의 숫자들이 일치 하지 않음.
Co-respond	해당하는 Typical Courses of Events에 맞게 번호 수정완료.

1.2. Stage 2040 Design

Issue	[2041] Use Case 17
Accept/Reject	Accept
Description	(9)에서 B버튼을 누르는 것이 아닌 A버튼을 눌러야 함. 치명적인 오류는 아니나, (2)~(18)에서 'C버튼을 누르면 (20)으로 이동한다.'를 언급하였기에, (11)~(18)에서 '2번' 내용을 다시 제시할 필요가 없다고 판단됨.
Co-respond	(9)의 경우 A버튼을 누르는 것으로 수정하고, 반복되는 내용을 제거함.

2. Brute Force Testing

2.1. Testing Result

Testing Num	Result	Description	Test Output
1-1	Pass	'년도'1의 값을 99보다 크게 설정한다	연도1 뿐만 아니라 모든 값이 이상한 값으로 바뀌고 원하는 시간 변경이 불가능해진다. (이후 선택된 값이 아닌 다른 값이 같이 바뀜)
1-2	Fail	모드 설정에서 9999년 12월 31일로 설정 후, 하루가 지나면 각 숫자가 값들이 한 칸씩 밀린다.	시계의 각 숫자 값들이 한 칸씩 밀려나 정상적인 시간 값을 보여주지 못함.
2-1	Pass	타이머 설정 후 모드 변경하여 타이머를 제외하면 알람이 울리는지 확인한다.	타이머 30초 설정 후 모드 변경하여 타이머를 제외해도 30초 후에 알람이 울림
2-2	Pass	타이머가 0이 되어 알람이 울리고 멈추게 한 뒤 모든 값(시분초)이 0으로 초기화 되는지 확인한다.	타이머가 0이 되어 알람이 울리고, 알람을 멈추면 설정한 타이머 시간으로 다시 채워진다는 내용이 스펙에 나와 있지 않음.
3-1	Pass	수정모드에서 D버튼을 누르면 enabled된 알람이 disabled 상태로 변경되는지 확인한다.	수정모드에서 D버튼을 눌러도 enabled된 알람이 disabled 상태로 바뀌지 않는다. (00시00분00초 상태로 En이 활성화됨)
3-2	Pass	수정모드 접근 후 D버튼을 눌러 0으로 리셋하고 C버튼으로 저장하면 알람이 리셋 되는지 확인한다.	C버튼을 눌러 수정모드 접근 후 D버튼을 누르면 0으로 리셋 된다. (스펙과 다름)
4-1	Pass	'년도 1'값을 99이상으로 설정한다.	연도1 뿐만아니라 모든 값이 이상한 값으로 바뀌고 원하는 D-day 변경이 불가능해진다. (이후 선택된 값이 아닌 다른 값이 같이 바뀜)
4-2	Pass	종료일을 시작일보다 이전으로 설정한다.	종료일이 시작일 보다 이전으로 선택 가능하며 음수 값으로 표시된다.
4-3	Pass	(종료일 - 시작일)이 9999보다 크도록 설정한다.	4자리 이상(9999이상)의 d-day는 표현이 안된다. (앞의 자리가 제외되는듯함)
4-4	Fail	시작 날짜에서 종료 날짜로 변경 시 커서가 년도1로 위치한다.	커서가 현재 상태에 머물러있음.
4-5	Pass	시작일과 종료일이 같고, 날짜를 오늘보다 이후로 설정한 경우 d-day와 % 표기가 잘 되는지 확인한다.	d-day에서는 잔여일이 잘 나타나지만 % 에서는 'done'으로 표시된다.
4-6	Pass	종료날짜만 있으면 Remain-day 방식으로만 보여주는지 확인한다.	종료날짜만 설정 불가능 하다.
5-1	Pass	타이머 설정 후 모드변경 하여 타이머를 제외하면 알람이 울리는지 확인한다.	타이머 30초 설정 후 모드변경하여 타이머를 제외해도 30초 후에 알람이 울린다.

5-2	Pass	수정모드 접근 후 D버튼을 눌러 0으로 리셋하고 C버튼으로 저장하면 알람이 리셋 되는지 확인한다.	C버튼을 눌러 수정모드 접근 후 D버튼을 누르면 0으로 리셋 된다. (스펙과 다름)
6-1	Pass	모드 변경 시 Beep 소리가 나는지 확인한다.	Beep 소리가 나지 않는다.

2.2. Countermeasure

Testing Num	A/R	Description
1-2	Accepted	시간이 흘러 10000년이 되면 1970년 1월 1일로 날짜를 변경함
4-4	Accepted	D-day page를 변경하면 커서가 “년도1”로 위치하도록 변경됨.

3. Static Analysis

3.1. Testing Result (Bug)

Issue & num	A/R	Title	Description
Bug 01	Accept	Bug 01 : Unread field: GUI.DigitalWatch$Bell.isBeeping	Boolean -> boolean으로 수정
Bug 02	Accept	Bug 02 : Call the method Thread.start() to execute the content of the run() method in a dedicated thread.	run() -> start() 으로 수정.
Bug 03	Accept	Bug 03 : GUI.DigitalWatch$Bell.play() explicitly invokes run on a thread (did you mean to start it instead?)	run() -> start() 으로 수정.
Bug 04	Accept	Bug 04 : Unread field: Logic.AlarmTime.m_timer	Timer m_timer 삭제
Bug 05	Accept	Bug 05 : Unread field: Logic.Dday.m_timer	Timer m_timer 삭제
Bug 06	Accept	Bug 06 : Unread field: Logic.IntervalTimer.m_timer	Timer m_timer 삭제
Bug 07	Accept	Bug 07 : Unread field: Logic.StopWatch.m_timer	Timer m_timer 삭제
Bug 08	Accept	Bug 08 : Unread field: Logic.TimeKeeping.m_timer	Timer m_timer 삭제
Bug 09	Accept	Bug 09 : Unread field: Logic.WatchSystem.m_timer	Timer m_timer 삭제
Bug 10	Reject	Bug 10 : Logic.WatchSystem.digitIdeal(Object) concatenates strings using + in a loop	루프가 최대 7번번 밖에 안하기 때문에 굳이 stringbuffer를 사용할 이유가 없음 -> 성능에 크게 영향을 미치지 않음

Bug 11	Accept	Bug 11 : Equality tests should not be made with floating point values.	calDday == 0 -> ==0f 로 수정
Bug 12	Accept	Bug 12 : Self assignment of field WatchTimer.m_timer in new Logic.WatchTimer()	Timer m_timer 삭제
Bug 13	Accept	Bug 13 : Uninitialized read of m_timer in new Logic.WatchTimer()	Timer m_timer 삭제

3.2. Testing Result (Critical)

Issue & num	A/R	Title	Description
Critical 01	Reject	Critical 01 : Refactor this method to reduce its Cognitive Complexity from 17 to the 15 allowed.	GUI Layer부분이 Thick하므로 복잡성이 높을 수 밖에 없음
Critical 02	Reject	Critical 02 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 03	Reject	Critical 03 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 04	Accept	Critical 04 : Avoid empty if statements	System.out.println("none"); 으로 작업 없음을 대략 명시함
Critical 05	Reject	Critical 05 : Refactor this method to reduce its Cognitive Complexity from 53 to the 15 allowed.	해당가능에 필요한 메소드이므로 복잡도를 낮추기엔 어렵다.
Critical 06	Reject	Critical 06 : The Cyclomatic Complexity of this method "buttonB" is 13 which is greater than 10 authorized.	GUI와 관련된 메서드이기 때문에 두꺼운 메서드가 되었음
Critical 07	Reject	Critical 07 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 08	Reject	Critical 08 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에

			필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 09	Reject	Critical 09 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 10	Reject	Critical 10 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 11	Reject	Critical 11 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 12	Reject	Critical 12 : Refactor this method to reduce its Cognitive Complexity from 29 to the 15 allowed.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 13	Reject	Critical 13 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 14	Reject	Critical 14 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 15	Reject	Critical 15 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 16	Reject	Critical 16 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 17	Reject	Critical 17 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다.

			또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 18	Accept	Critical 18 : Avoid empty if statements	System.out.println("none"); 으로 작업 없음을 대략 명시함
Critical 19	Reject	Critical 19 : Refactor this method to reduce its Cognitive Complexity from 33 to the 15 allowed.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 20	Accept	Critical 20 : Avoid empty if statements	System.out.println("none"); 으로 작업 없음을 대략 명시함
Critical 21	Reject	Critical 21 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 22	Reject	Critical 22 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 23	Reject	Critical 23 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 24	Reject	Critical 24 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 25	Reject	Critical 25 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 26	Reject	Critical 26 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick 해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 27	Accept	Critical 27 : Avoid empty catch blocks	e.printStackTrace(); 추가
Critical 28	Accept	Critical 28 : Avoid empty catch blocks	e.printStackTrace(); 추가

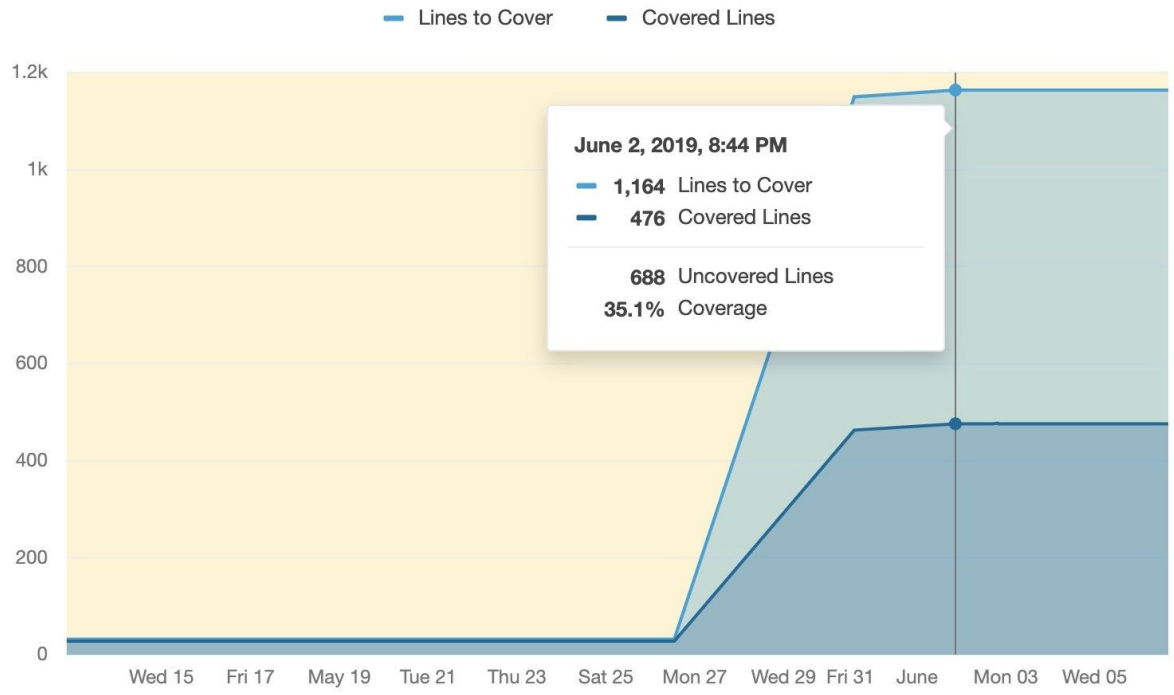
Critical 29	Accept	Critical 29 : Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.	System.out.println("") action 구체화
Critical 30	Accept	Critical 30 : Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.	System.out.println("") action 구체화
Critical 31	Accept	Critical 31 : Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.	System.out.println("") action 구체화
Critical 32	Accept	Critical 32 : "if ... else if" constructs should end with "else" clauses.	else 부분을 추가함.
Critical 33	Reject	Critical 33 : Define a constant instead of duplicating this literal "000000" 4 times.	000000이 4번밖에 나오지 않는데, final 이용해서 상수로 만들어 주는 것이 코드상 직관성이 더 떨어지는 것 같음
Critical 34	Accept	Critical 34 : "if ... else if" constructs should end with "else" clauses.	else 부분을 추가함.
Critical 35	Reject	Critical 35 : Refactor this method to reduce its Cognitive Complexity from 51 to the 15 allowed.	현재 모드와 관련된 메서드이기 때문에 두꺼운 메서드가 되었음
Critical 36	Reject	Critical 36 : The Cyclomatic Complexity of this method "increaseData" is 24 which is greater than 10 authorized.	현재 모드와 관련된 메서드이기 때문에 두꺼운 메서드가 되었음
Critical 37	Accept	Critical 37 : "if ... else if" constructs should end with "else" clauses.	else 부분을 추가함.
Critical 38	Accept	Critical 38 : "if ... else if" constructs should end with "else" clauses.	else 부분을 추가함.
Critical 39	Reject	Critical 39 : Refactor this method to reduce its Cognitive Complexity from 68 to the 15 allowed.	GUI와 관련된 메서드기 때문에 두꺼운 메서드가 되었음
Critical 40	Reject	Critical 40 : The Cyclomatic Complexity of this method "digitIdeal" is 27 which is greater than 10 authorized.	GUI와 관련된 메서드기 때문에 두꺼운 메서드가 되었음
Critical 41	Reject	Critical 41 : Define a constant instead of duplicating this literal "HHmmss" 4 times.	HHmmss이 4번밖에 나오지 않는데, final 이용해서 상수로 만들어 주는 것이 코드상 직관성이 더 떨어지는 것 같음
Critical 42	Reject	Critical 42 : Define a constant instead of duplicating this literal "zzzzzzzz" 4 times.	zzzzzz이 4번밖에 나오지 않는데, final 이용해서 상수로 만들어 주는 것이 코드상 직관성이 더 떨어지는 것 같음

Critical 43	Reject	Critical 43 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	단계별로 예외처리가 필요하므로 코드 수정 불가능
Critical 44	Reject	Critical 44 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	해당 기능 구현을 위해 연속적인 Branch가 생성될 수 밖에 없었다.
Critical 45	Accept	Critical 45 : "if ... else if" constructs should end with "else" clauses.	else 부분을 추가함.
Critical 46	Accept	Critical 46 : "if ... else if" constructs should end with "else" clauses.	else 부분을 추가함.
Critical 47	Reject	Critical 47 : Refactor this method to reduce its Cognitive Complexity from 24 to the 15 allowed.	해당가능에 필요한 메소드이므로 복잡도를 낮추기엔 어렵다
Critical 48	Reject	Critical 48 : The Cyclomatic Complexity of this method "iconIdeal" is 18 which is greater than 10 authorized.	GUI와 관련된 메서드이기 때문에 두꺼운 메서드가 되었음
Critical 49	Accept	Critical 49 : "if ... else if" constructs should end with "else" clauses.	else 부분을 추가함.
Critical 50	Reject	Critical 50 : Define a constant instead of duplicating this literal "000000" 3 times.	000000이 3번밖에 나오지 않는데, final 이용해서 상수로 만들어 주는 것이 코드상 직관성이 더 떨어지는 것 같음
Critical 51	Accept	Critical 51 : Remove or correct this assertion.	assertEquals(alarm.getAlarmTime(0).getEnabled(), true) -> assertTrue(alarm.getAlarmTime(0).getEnabled()) 로 변경
Critical 52	Accept	Critical 52 : Remove or correct this assertion.	assertEquals(alarm.getAlarmTime(0).getEnabled(), false) -> assertFalse(alarm.getAlarmTime(0).getEnabled()) 로 변경
Critical 53	Accept	Critical 53 : Remove or correct this assertion.	assertEquals(it.getIsEnabled(), false) -> assertFalse(it.getIsEnabled()) 로 변경
Critical 54	Accept	Critical 54 : Remove or correct this assertion.	assertEquals(sw.getActivated(), false) -> assertFalse() 로 변경
Critical 55	Accept	Critical 55 : Remove or correct this assertion.	assertEquals(tk.getDisplayFormat(), false) -> assertFalse() 로 변경
Critical 56	Accept	Critical 56 : Remove or correct this assertion.	assertEquals(wt.getActived(), false) -> assertFalse() 로 변경
Critical 57	Accept	Critical 57 : Remove or correct this assertion.	assertEquals(wt.getActived(), false) -> assertFalse(wt.getActived()) 로 변경
Critical 58	Accept	Critical 58 : Remove or correct this assertion.	assertEquals(it.getInEnabled(), false) -> assertFalse(it.getInEnabled()) 로 변경
Critical 59	Accept	Critical 59 : Remove or correct this assertion.	assertEquals(alarm.getAlarmTime(0).getEnabled(), true) ->

			assertTrue(alarm.getAlarmTime(0).getEnabled()) 로 변경
Critical 60	Accept	Critical 60 : Remove or correct this assertion.	assertEquals(alarm.getAlarmTime(0).getEnabled(), false) -> assertFalse(alarm.getAlarmTime(0).getEnabled()) 로 변경
Critical 61	Accept	Critical 61 : Remove or correct this assertion.	assertEquals(d.getDisplayType(), true) -> assertTrue(d.getDisplayType())으로 변경
Critical 62	Accept	Critical 62 : Remove or correct this assertion.	assertEquals(sw,getActivated(), false) -> assertFalse(sw,getActivated()) 로 변경
Critical 63	Accept	Critical 63 : Remove or correct this assertion.	assertEquals(wt,getActived(), false) -> assertFalse(wt.getActived()) 로 변경

3.3. Increase Code Coverage

Before



After