

# SW Verification

201211387 하헌규

201211337 김재현

201211354 성현구

201211369 이병찬

# 목차

- Specification Review (2nd)
- System Test (2nd)
  - Brute Force Test
  - Category Partitioning Test
  - Pairwise Combination Test
  - Result & CTIP
- Static Analysis
  - JDepend
  - Code Scroll
  - Findbugs
  - SonarCloud

# Test Model

20180601043245-f9...  
f907472

20180601043245-f907472: testing start

Edit

 Nineclown released this 6 hours ago · 2 commits to master since this release

▲ 2018년 6월 1일 업로드된 파일을 기준으로 진행 (이후 SMA팀의 지속적인 변경 발생)

# Specification Review

- 전체적인 문서 완성도 우수
- 미비했던 형식 정의가 해결되고 명세됨
- 수수료에 대한 명세를 표기할 수 있는 문서를 찾지 못한 듯

# System Test - Brute Force Test

brute force test (43)

ID	Title	Test Run 5/27/2018	Test Run 6/1/2018		
C158	user에게 계좌 정보를 입력 받는 기능	Passed	Passed	C179	User에게 계좌정보를 입력 받는 기능
C159	입금 후 계좌의 잔액이 정상적으로 증가하였는지	Failed	Passed	C180	계좌의 잔액을 출력하는 기능
C160	User에게 계좌정보를 입력 받는 기능	Passed	Passed	C181	거래 종류와 은행에 따른 수수료를 계산하는 기능
C161	환율을 받아오는 기능	Failed	Passed	C182	개인식별번호를 입력 받아 사용자의 카드 목록을 조회하는 기능
C162	User가 입력한 금액을 환율에 맞게 계산하여 계좌에 입금하는 기능	Failed	Passed	C183	선택된 카드를 정지시키는 기능
C163	입금 후 계좌의 잔액이 정상적으로 증가하였는지	Failed	Passed	C184	분실된 카드의 재발급을 신청하는 기능
C164	User에게 계좌정보를 입력 받는 기능	Passed	Passed	C185	잘못된 개인식별번호를 입력할 경우 에러를 띄우는 기능
C165	출금 후 계좌의 잔액이 정상적으로 감소하였는지	Failed	Passed	C186	User가 입력한 로또 용지의 숫자를 읽는 기능
C166	User가 출금하려는 금액이 User의 계좌 잔액보다 클 경우 에러를 띄우는 기능	Passed	Passed	C187	해당하는 계좌의 당점번호를 받아오는 기능
C167	User에게 계좌정보를 입력 받는 기능	Passed	Passed	C188	로또 용지의 숫자와 당점번호를 비교한 후 당점여부를 판단하는 기능
C168	환율을 받아오는 기능	Failed	Passed	C189	User에게 계좌정보를 입력 받는 기능
C169	User가 출금하려는 금액을 환율에 맞게 계산하여 계좌에서 출금하는 기능	Failed	Passed	C190	당점금 입금 후 계좌의 잔액이 정상적으로 증가하였는지
C170	출금 후 계좌의 잔액이 정상적으로 감소하였는지	Failed	Passed	C191	User가 n번 인증을 실패한 경우 거래를 제한하는 기능
C171	User가 출금하려는 금액이 User의 계좌 잔액보다 클 경우 에러를 띄우는 기능	Passed	Passed	C192	인터페이스의 언어가 전환되는 기능
C172	User에게 계좌정보를 입력 받는 기능	Passed	Passed	C193	Admin의 입력에 따라 Admin 계정을 추가하는 기능
C173	출금된 계좌와 입금된 계좌의 잔액이 정상적으로 변경되었는지	Passed	Passed	C194	Admin 식별번호를 중복되지 않게 생성하는 기능
C174	User가 이체하려는 금액이 User의 계좌 잔액보다 클 경우 에러를 띄우는 기능	Passed	Passed	C195	Admin 계정을 제거하는 기능
C175	대표 User의 계좌정보, 금액, 인원수를 입력 받는 기능	Passed	Passed	C196	마지막 Admin 계정을 삭제하는 경우 에러를 띄우며 제거되지 않는 기능
C176	금액과 인원수를 이용해 이체 금액을 계산하는 기능	Passed	Passed	C197	ATM의 사용자 기능을 제한하는 기능
C177	User에게 계좌정보를 입력 받는 기능	Passed	Passed	C198	ATM이 사용자 기능 제한 상태인 경우, 제한을 해제하는 기능
C178	User가 선택한 조회 기간에 해당하는 거래 내역을 출력하는 기능	Failed	Passed	C199	현재 지폐 보유량을 변경하는 기능
				C200	ATM의 지폐 보유량을 조회하는 기능

▲ 동일한 테스트 결과 Pass의 비율이 65%에서 91%로 증가

# System Test - Category Partitioning Test

날짜 끝	형식에 맞는 날짜	30900
	시작 날짜보다 작은 날짜	30901
	시작 날짜와 같은 날짜	30902
	형식에 맞지않는 날짜	30903

관리자 아이디	존재하는 아이디	30500
	존재하지 않는 아이디	30501
	형식에 맞지않는 입력	30502

동결 후 기능	입금	30300
	외화 입금	30301
	출금	30302
	외화 출금	30303
	송금	30304
	분할 결제	30305
	거래내역조회	30306
	잔액조회	30307
	복권당첨금수령	30308



날짜 끝 <sup>Ⓢ</sup>	형식에 맞는 날짜 <sup>Ⓢ</sup>	30900 <sup>Ⓢ</sup>
	시작 날짜보다 작은 날짜 <sup>Ⓢ</sup>	30901 <sup>Ⓢ</sup>
	형식에 맞지않는 날짜 <sup>Ⓢ</sup>	30902 <sup>Ⓢ</sup>

관리자 아이디 <sup>Ⓢ</sup>	존재하는 아이디 <sup>Ⓢ</sup>	30500 <sup>Ⓢ</sup>
	존재하지 않는 아이디 <sup>Ⓢ</sup>	30501 <sup>Ⓢ</sup>

동결 후 기능 <sup>Ⓢ</sup>	입금 <sup>Ⓢ</sup>	30300 <sup>Ⓢ</sup>
	외화 입금 <sup>Ⓢ</sup>	30301 <sup>Ⓢ</sup>
	출금 <sup>Ⓢ</sup>	30302 <sup>Ⓢ</sup>
	외화 출금 <sup>Ⓢ</sup>	30303 <sup>Ⓢ</sup>
	송금 <sup>Ⓢ</sup>	30304 <sup>Ⓢ</sup>
	분할 결제 <sup>Ⓢ</sup>	30305 <sup>Ⓢ</sup>
	거래내역조회 <sup>Ⓢ</sup>	30306 <sup>Ⓢ</sup>
	잔액조회 <sup>Ⓢ</sup>	30307 <sup>Ⓢ</sup>
	카드 분실 신고 <sup>Ⓢ</sup>	30708 <sup>Ⓢ</sup>
	복권당첨금수령 <sup>Ⓢ</sup>	30309 <sup>Ⓢ</sup>

# System Test - Category Partitioning Test

1,813,985,280,000



7,776,025 [Error]



43,241 [Single][Error]



89 [Single][Error][Property]

# System Test - Category Partitioning Test

Test Cases (91)		Test Run 5/27/2018	Test Run 6/1/2018				
C1	20200	Failed	Failed	C34	31290	Passed	Passed
C2	30101	Passed	Passed	C35	31292	Passed	Passed
C3	30102	Passed	Passed	C36	31293	Passed	Passed
C4	30201	Passed	Failed	C37	31294	Passed	Passed
C5	30202	Passed	Passed	C38	31295	Passed	Passed
C6	30300	Passed	Passed	C39	31300	Passed	Passed
C7	30301	Passed	Passed	C40	31302	Passed	Passed
C8	30302	Passed	Passed	C41	31303	Failed	Passed
C9	30303	Passed	Passed	C42	31304	Passed	Passed
C10	30304	Passed	Passed	C43	31401	Passed	Passed
C11	30305	Passed	Passed	C44	31402	Passed	Passed
C12	30306	Passed	Passed	C45	31500	Passed	Passed
C13	30307	Passed	Passed	C46	31501	Passed	Passed
C14	30308	Passed	Passed	C47	31601	Failed	Failed
C15	30501	Passed	Failed	C48	31602	Failed	Failed
C16	30601	Passed	Failed	C49	31701	Passed	Failed
C17	30602	Passed	Failed	C50	31702	Failed	Failed
C18	30603	Passed	Failed	C41	10000.10100.20000.30000.30100.30200.31000.31100.31201.40000	Passed	Passed
C19	30701	Passed	Passed	C42	10000.10100.20000.30000.30100.30200.31000.31100.31201.40000	Passed	Passed
C20	30801	Passed	Passed	C43	10000.10100.20000.30000.30100.30200.31000.31100.31201.40000	Passed	Passed
C21	30901	Passed	Passed	C44	10000.10100.20001.30000.30100.30200.31000.31101.31201.40000	Passed	Passed
C22	30902	Passed	Passed	C45	10000.10100.20001.30000.30100.30200.31001.31101.31201.40000	Passed	Passed
C23	31001	Passed	Passed	C46	10000.10100.20001.30000.30100.30200.31002.31101.31201.40000	Passed	Passed
C48	10000.10101.20000.30000.30100.30200.31001.31100.31201.40000	Failed	Failed	C47	10000.10101.20000.30000.30100.30200.31000.31100.31201.40000	Failed	Passed
C49	10000.10101.20000.30000.30100.30200.31002.31100.31201.40000	Failed	Failed	C73	10000.10104.20006.30000.30100.30200.30800.30900.31002.40002	Passed	Passed
C50	10000.10101.20001.30000.30100.30200.31000.31101.31201.40000	Failed	Passed	C77	10000.10104.20007.30000.30100.30200.31000.40001	Passed	Passed
C51	10000.10101.20001.30000.30100.30200.31001.31101.31201.40000	Failed	Passed	C78	10000.10104.20007.30000.30100.30200.31001.40001	Passed	Passed
C52	10000.10101.20001.30000.30100.30200.31002.31101.31201.40000	Failed	Passed	C79	10000.10104.20007.30000.30100.30200.31002.31002.40001	Passed	Passed
C53	10000.10102.20002.30000.30100.30200.31000.31100.31201.40000	Passed	Passed	C80	10000.10104.20008.31400.31502	Passed	Passed
C54	10000.10102.20002.30000.30100.30200.31001.31100.31201.40000	Passed	Passed	C81	10000.10104.20009.30000.30100.30200.31000.31600.31700	Passed	Passed
C55	10000.10102.20002.30000.30100.30200.31002.31100.31201.40000	Passed	Passed	C82	10000.10104.20009.30000.30100.30200.31001.31600.31700	Passed	Passed
C56	10000.10102.20003.30000.30100.30200.31000.31101.31201.40000	Passed	Passed	C83	10000.10104.20009.30000.30100.30200.31002.31600.31700	Passed	Passed
C57	10000.10102.20003.30000.30100.30200.31001.31101.31201.40000	Passed	Passed	C84	10001.20100.30500.30600	Passed	Passed
C58	10000.10102.20003.30000.30100.30200.31002.31101.31201.40000	Passed	Passed	C85	10001.20101.30600.30700	Failed	Passed
C59	10000.10103.20002.30000.30100.30200.31000.31100.31201.40000	Failed	Passed	C86	10001.20102	Passed	Passed
C60	10000.10103.20002.30000.30100.30200.31001.31100.31201.40000	Failed	Passed	C87	10001.20103	Passed	Passed
C61	10000.10103.20002.30000.30100.30200.31002.31100.31201.40000	Failed	Passed	C88	10001.20104.31100.31201	Passed	Passed
C62	10000.10103.20003.30000.30100.30200.31000.31101.31201.40000	Failed	Passed	C89	10001.20104.31101.31201	Passed	Passed
C63	10000.10103.20003.30000.30100.30200.31001.31101.31201.40000	Failed	Passed	C90	10001.20105.30400	Passed	Passed
C64	10000.10103.20003.30000.30100.30200.31002.31101.31201.40000	Failed	Passed	C91	10001.20105.30401	Passed	Passed
C65	10000.10104.20004.30001.30100.30200.31000.31100.31201.40000	Passed	Passed	C92	10200	Passed	Passed
C66	10000.10104.20004.30001.30100.30200.31001.31100.31201.40000	Passed	Passed	C301	30300	Passed	Passed
C67	10000.10104.20004.30001.30100.30200.31002.31100.31201.40000	Passed	Passed				
C68	10000.10104.20005.30001.30100.30200.31000.31100.31201.40000	Passed	Passed				
C69	10000.10104.20005.30001.30100.30200.31001.31100.31201.40000	Passed	Passed				
C70	10000.10104.20005.30001.30100.30200.31002.31100.31201.40000	Passed	Passed				
C71	10000.10104.20006.30000.30100.30200.30800.30900.31000.40002	Passed	Passed				
C72	10000.10104.20006.30000.30100.30200.30800.30900.31001.40002	Passed	Passed				

▲ 테스트 결과 Pass의 비율이 80%에서 91%로 증가

# System Test - Pairwise Test

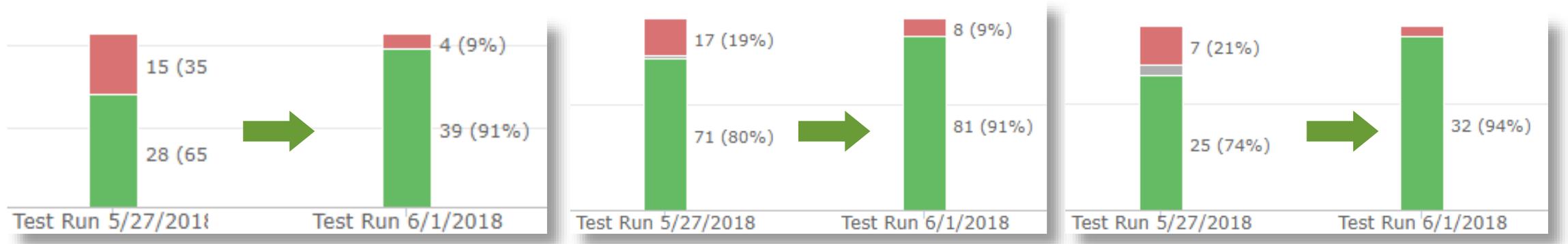
pair_wise (34)			
ID	Title	Test Run 5/27/2018	Test Run 6/1/2018
C93	10000.10100.20002.20200.30001.30101	Passed	Passed
C94	10000.10100.20003.20200.30000.30102	Passed	Passed
C95	10000.10100.20003.20200.30000.30101	Passed	Passed
C96	10000.10100.20006.20200.30001.30102	Passed	Passed
C97	10000.10101.20001.20200.30001.30101	Passed	Passed
C98	10000.10101.20007.20200.30001.30100.30200.30400.30800.30900.31002.31100.31	Passed	Passed
C99	10000.10101.20004.20200.30001.30100.30201	Passed	Passed
C100	10000.10102.20004.20200.30001.30102	Passed	Passed
C101	10000.10102.20009.20200.30001.30102	Passed	Passed
C102	10000.10102.20006.20200.30001.30101	Passed	Passed
C103	10000.10103.20003.20200.30001.30101	Passed	Passed
C104	10000.10103.20007.20200.30000.30101	Passed	Passed
C105	10000.10103.20008.20200.30000.30101	Failed	Passed
C106	10000.10103.20009.20200.30000.30100.30202	Failed	Passed
C107	10000.10104.20004.20200.30001.30100.30202	Passed	Passed
C108	10000.10104.20006.20200.30000.30101	Passed	Passed
C109	10000.10104.20007.20200.30001.30100.30201	Passed	Passed
C110	10000.10104.20008.20200.30000.30100.30202	Passed	Passed
C111	10000.10104.20008.20200.30001.30101	Failed	Passed
C112	10001.20100.20200.30401.30501	Passed	Passed
C113	10001.20101.20200.30401.30501	Failed	Passed
C114	10001.20101.20200.30400.30501	Passed	Passed
C115	10001.20101.20200.30401.30500.30602	Failed	Passed
C116	10001.20102.20200.30401.30502	Passed	Passed
C117	10001.20102.20200.30401.30500.30601	Passed	Passed
C118	10001.20103.20200.30400.30501	Passed	Passed
C119	10001.20103.20200.30401.30500.30600.30700.30110.31300	Passed	Passed
C120	10001.20103.20200.30401.30500.30601	Failed	Passed
C121	10001.20103.20200.30401.30502	Failed	Passed
C122	10001.20104.20200.30400.30502	Passed	Passed
C123	10001.20104.20200.30400.30501	Passed	Passed
C124	10001.20105.20200.30401.30500.30602	Passed	Passed
C202	31601	Untested	Failed
C203	31701	Untested	Failed

▲ 테스트 결과 Pass의 비율이 78%에서 94%로 증가

# System Test - 오류

- 복권 당첨 : 복권의 주차 형식을 0 이상인 정수로 정의하였으나 0 및 현재 존재하지 않는 주차를 정상적인 형식으로 인식하고, 복권 번호의 형식도 1~45까지로 정의되었으나 0을 정상적인 형식으로 인식한다.
- 날짜 : 경계값을 무시한다 (ex. 각 날짜의 00시 00분을 무시한다)
- 메시지 : Invalid와 Incorrect에 따른 행동이 일관적이지 않다.
- 언어 변경 : 언어 변경시 Admin이 로그아웃된다.

# System Test - Result



## ▲ System Test 결과

# System Test - Result

- Brute Force Test : 65%(28/43) → **91%**(39/43)
- Category Partitioning Test : 80%(71/88) → **91%**(81/89)
- Pairwise Combination Test : 74% (25/32) → **94%**(32/34)

# System Test - in CTIP

<input type="checkbox"/>	①	메시지에 대한 명세에 관한 질문입니다. <b>Conversation</b>	
<small>#55 opened 3 hours ago by gkgjsrb</small>			
<input type="checkbox"/>	①	중복되어 보이는 test case result가 등록된 이유 <b>Conversation</b>	
<small>#53 opened 4 days ago by Nineclown</small>			
<input type="checkbox"/>	①	Definition: 관리자 비밀번호 및 연락처 <b>Failed Test</b>	1
<small>#52 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	거래 내역 조회: 같은 날짜 입력 오류 발생 <b>Conversation Failed Test</b>	1
<small>#49 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	복권 : 로또번호 0 에러. <b>Failed Test</b>	2
<small>#36 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	복권 : 주차 0 입력 에러. <b>Failed Test</b>	2
<small>#35 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	즉시 언어변경 문제. <b>Conversation Failed Test</b>	3
<small>#33 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	Failed test: Admin의 입력에 따라 Admin 계정을 추가하는 기능 <b>Failed Test</b>	2
<small>#31 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	Failed test: 인터페이스의 언어가 전환되는 기능 <b>Conversation Failed Test</b>	2
<small>#30 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	Failed test: 해당하는 회차의 당첨번호를 받아오는 기능 <b>Failed Test</b>	1
<small>#29 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	Failed test: User가 입력한 로또 용지의 숫자를 읽는 기능 <b>Failed Test</b>	1
<small>#28 opened 5 days ago by jaehyun739</small>			
<input type="checkbox"/>	①	Definition: User가 선택한 조회 기간에 해당하는 거래 내역을 출력하는 기능 <b>Conversation Failed Test</b>	6
<small>#26 opened 5 days ago by jaehyun739</small>			

# Static Analysis - JDepend

## Summary

[\[summary\]](#) [\[packages\]](#) [\[cycles\]](#) [\[explanations\]](#)

Package	Total Classes	Abstract Classes	Concrete Classes	Afferent Couplings	Efferent Couplings	Abstractness	Instability	Distance
<a href="#">com.swad.cppatm</a>	1	0	1	1	5	0	0.83	0.17
<a href="#">com.swad.cppatm.application</a>	10	0	10	2	8	0	0.8	0.2
<a href="#">com.swad.cppatm.enums</a>	6	0	6	2	1	0	0.33	0.67
<a href="#">com.swad.cppatm.exceptions</a>	15	0	15	2	1	0	0.33	0.67
<a href="#">com.swad.cppatm.ui</a>	94	0	94	1	13	0	0.93	0.07
<a href="#">com.swad.cppatm.ui.components</a>	1	0	1	1	3	0	0.75	0.25

▲ JDepend Summary

# Static Analysis - JDepend

- CC :: Concrete Class
  - 인터페이스나 추상클래스가 아닌 Concrete Class의 수를 표현
- AC :: Abstract Class
  - 추상 클래스나 인터페이스의 수를 나타내며 확장성의 척도
- Ca :: Afferent Couplings
  - 현재 패키지의 클래스에 의존하고 있는 패키지의 수를 나타내며 책임의 척도
- Ce :: Efferent Couplings
  - 현재 패키지의 클래스들이 의존하고 있는 패키지의 수를 나타내며 독립성의 척도
- A :: Abstractness (  $A = AC/CC+AC$  )
  - 추상화 정도를 나타내며, 0은 구체적인 패키지를, 1은 추상적인 패키지를 표현.
- I :: Instability (  $I = Ce/(Ce+Ca)$  )
  - 변화에 대한 안정성을 나타내며 0부터 1사이의 값을 가짐
  - 0은 외부 변화에도 끄떡없는 패키지이며 1은 작은 변화에도 쉽게 흔들릴 수 있는 패키지를 표현
- D :: Distance to Main Sequence
  - Main Sequence로 부터의 거리를 나타내며, 0은 Main Sequence와 완전 가깝고 1은 완전 먼 상태
  - Main Sequence란 이상적인 패키지로 완전 추상적이면서 안정적인거나 완전 구체적이면서 불안정한 패키지를 표현
- Cycle :: Package dependency cycles
  - 패키지들 상호 간에 의존성을 가지고 있을 때 발생
  - 안 좋은 상황이기 때문에 경고 아이콘으로 표현

# Static Analysis - JDepend

## 추상화된 클래스

안정성이 높은 것이 권장된다.

= Ce가 적고 Ca는 많은 편이 좋다.

= 다른 클래스의 영향을 적게 받는 것이 좋다.

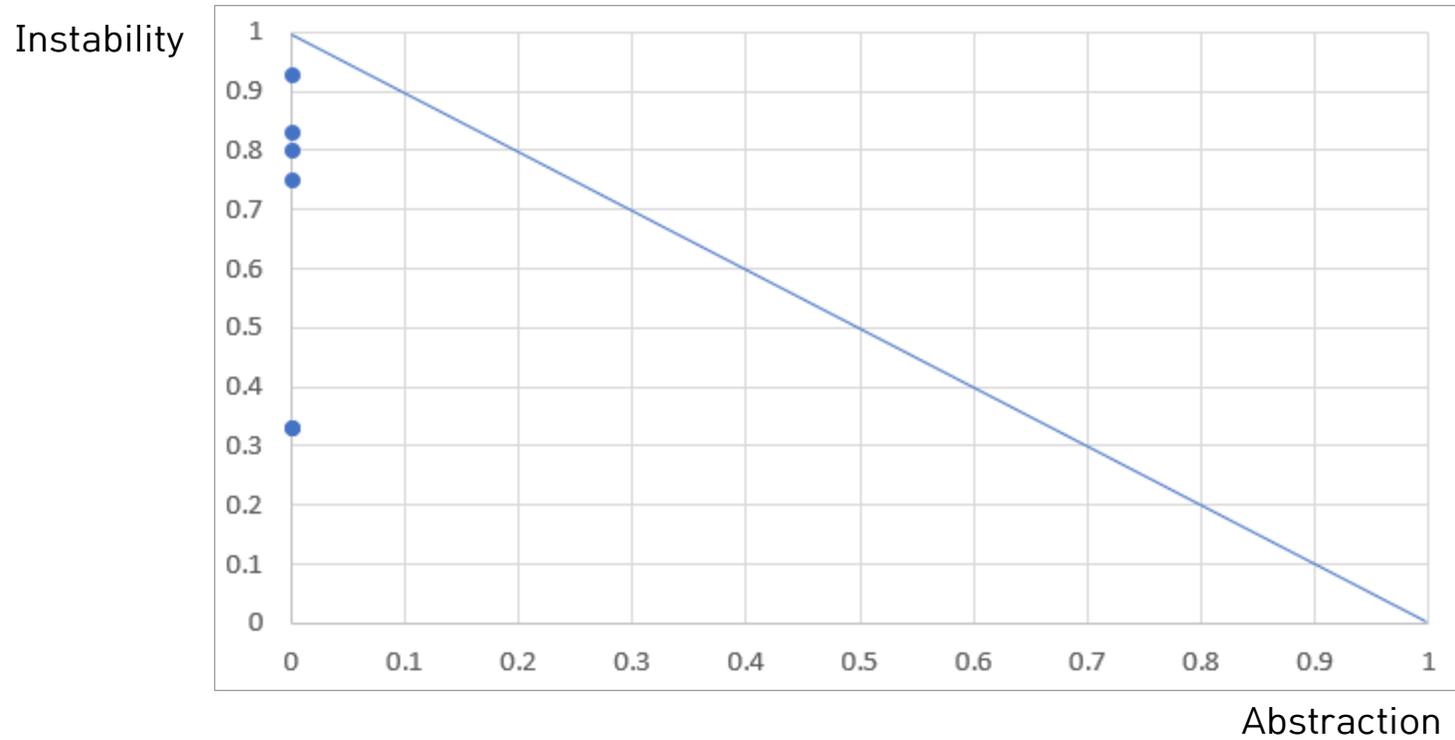
## 구체적인 클래스

안정성이 낮은 것이 권장된다.

= Ce가 많고 Ca는 적은 편이 좋다.

= 다른 클래스에 영향을 적게 미치는 것이 좋다.

# Static Analysis - JDepend



▲ Abstraction - Instability Metrics

# Static Analysis - JDepend

## com.swad.cppatm.exceptions

Afferent Couplings: 2

Efferent Couplings: 1

Abstractness: 0

Instability: 0.33

Distance: 0.67

$$1/(2+1) = 0.33\dots$$

$$D=1-(A+I)=0.67\dots$$

## com.swad.cppatm

Afferent Couplings: 1

Efferent Couplings: 5

Abstractness: 0

Instability: 0.83

Distance: 0.17

$$5/(1+5) = 0.83\dots$$

$$D=1-(A+I)=0.17\dots$$

com.swad.cppatm 패키지가 더 이상적  
com.swad.cppatm.exception 패키지는 추상화하거나 Instability를 높여야 함

# Static Analysis - JDepend

## Cycles

### **com.swad.cppatm**

com.swad.cppatm.ui  
com.swad.cppatm

### **com.swad.cppatm.ui**

com.swad.cppatm  
com.swad.cppatm.ui

- 상호 의존적인 관계
- 재사용성을 저해함
- 코드 변경시 도미노 효과 발생 가능
- 메모리 누수 발생 가능

# Static Analysis - Code Scroll

이름	제목	
> <input type="checkbox"/>  Android	Android spec(api level 16)을 기반으로 올바른 코딩...	
> <input checked="" type="checkbox"/>  CERT_Secure_Coding	Java Platform standard edition 6,7에 대한 시큐어 프...	
> <input checked="" type="checkbox"/>  CODESCROLL_JAVA	일반적인 자바프로그램에서 지켜야 할 규칙과 권고들	
> <input checked="" type="checkbox"/>  EGOV_Inspection_St	전자정부 표준 Inspection 규칙모음	
> <input type="checkbox"/>  IEC_60880	Nuclear Power Plants의 안전성을 위한 표준	
> <input checked="" type="checkbox"/>  Sun_Code_Conventi	파일이름, 파일 구성, 들여쓰기, 주석, 선언, 문장, 이...	

→ 3874개 규칙위반 발견

# Static Analysis - Code Scroll

EGOV	JAVA	SUN	CERT_SECURE_CODING
EGOV_01_EMPTYCATCHBLOCK	JAVA_44	SUN_13	ERROR
EGOV_12_SWITCHSTMTSSHOUHDAVEDEFAULT	JAVA_65	SUN_10	ERROR00-J
EGOV_14_FINALFIELDCOULDBESTATIC	JAVA_61	SUN_26	ERROR02-J
EGOV_25_SYSTEMPRINTLN	JAVA_63	SUN_27	ERROR08-J
EGOV_26_VARIABLENAMINGCONVENTIONS	JAVA_64	SUN_24	NUM
EGOV_28_AVOIDARRAYLOOPS	JAVA_71	SUN_22	NUM02-J
EGOV_33_STRINGTOSTRING	JAVA_72	SUN_20	NUM06-J
EGOV_37_UNUSEDPRIVATEFIELD	JAVA_68	SUN_19	VNA
EGOV_39_UNUSEDFORMALPARAMTER		SUN_17	VNA04-J
		SUN_14	OBJ
		SUN_03	OBJ05-J
		SUN_05	OBJ01-J
		SUN_06	OBJ03-J
		SUN_07	OBJ04-J
		SUN_08	META
		SUN_11	META04-J
			META05-J

	매우 높음
	높음
	낮음
	매우 낮음

# Static Analysis - Findbugs

Warning Type	Number
<a href="#">Bad practice Warnings</a>	11
<a href="#">Correctness Warnings</a>	4
<a href="#">Internationalization Warnings</a>	5
<a href="#">Malicious code vulnerability Warnings</a>	4
<a href="#">Multithreaded correctness Warnings</a>	1
<a href="#">Performance Warnings</a>	25
<a href="#">Dodgy code Warnings</a>	17
<b>Total</b>	<b>67</b>

▲ Summary

# Static Analysis - Findbugs

번호	탐지 유형	사례 및 설명
1	Bad practice	클래스 명명규칙, Null 처리 실수 등 개발자의 나쁜 습관을 탐지
2	Correctness	잘못된 상수, 무의미한 메소드 호출 등 문제의 소지가 있는 코드를 탐지
3	Dodgy code	int의 곱셈결과를 long으로 변환하는 등 부정확하거나 오류를 발생시킬 수 있는 코드를 탐지
4	Experimental	메소드에서 생성된 stream이나 리소스가 해제하지 못한 코드를 탐지
5	Internatinalization	Default 인코딩을 지정하지 않은 경우 등 지역특성을 고려하지 않은 코드 탐지
6	Maclicious codevulnerability	보안 코드에 취약한 가변적인 배열이나 컬렉션, Hashtable 탐지
7	Multithreaded correctness	멀티쓰레드에 안전하지 않은 객체 사용 등을 탐지
8	Performance	미사용 필드, 비효율적 객체생성 등 성능에 영향을 주는 코드를 탐지
9	Security	CSS, DB 패드워드 누락 등 보안에 취약한 코드를 탐지

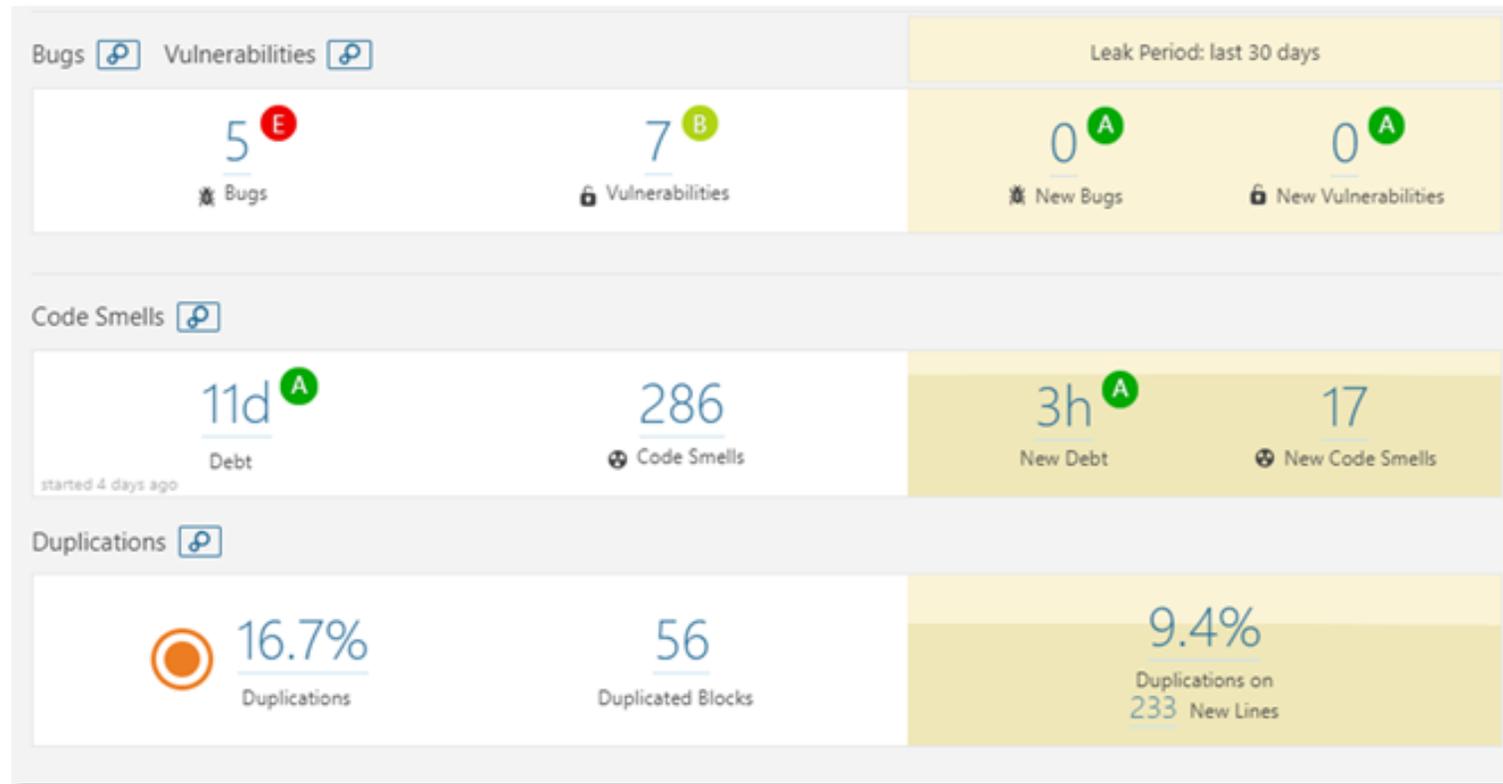
# Static Analysis - Findbugs

DE	DE_MIGHT_IGNORE: Method might ignore exception	예외처리를 하지 않음. 빈 catch절이 존재.
OS	OS_OPEN_STREAM: Method may fail to close stream	stream close하지 않음.
	OS_OPEN_STREAM_EXCEPTION_PATH: Method may fail to close stream on exception	stream 예외 처리를 하지 않음
RV	RV_RETURN_VALUE_IGNORED_BAD_PRACTICE: Method ignores exceptional return value	확인되지 않은 값을 리턴함.
SW	SW_SWING_METHODS_INVOKED_IN_SWING_THREAD: Certain swing methods needs to be invoked in Swing thread	Swing thread를 호출해야 함.
RpC	RpC_REPEATED_CONDITIONAL_TEST: Repeated conditional tests	조건부 테스트가 중복 수행됨.
Dm	DM_STRING_TOSTRING: Method invokes toString() method on a String	String이 toString()을 사용할 필요가 없음.
	DM_DEFAULT_ENCODING: Reliance on default encoding	byte to string or string to byte변환을 수행할 때 Charset 객체를 명시적으로 지정해야 함.
EI	EI_EXPOSE_REP: May expose internal representation by returning reference to mutable object	객체의 필드 중 하나에 저장된 변경 가능한 객체 값에 대한 참조를 반환하면 객체의 내부 표현이 노출됨.
MS	MS_SHOULD_BE_FINAL: Field isn't final but should be	static 변수가 public이라서 위험. Final을 명시하여야 함.

# Static Analysis - Findbugs

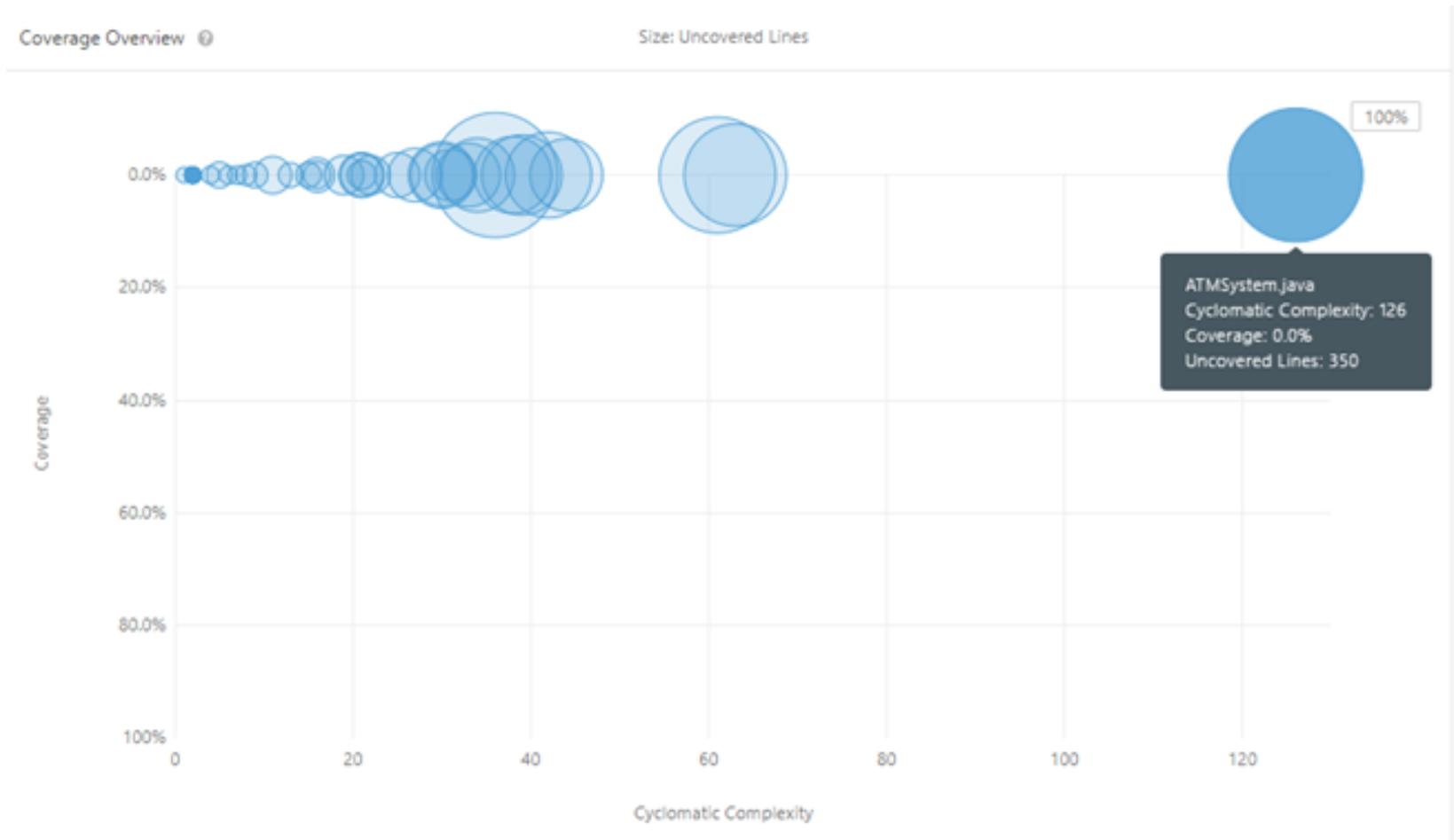
LI	LI_LAZY_INIT_STATIC: Incorrect lazy initialization of static field	multi-thread환경에서 초기화를 보장하지 못함.
SIC	SIC_INNER_SHOULD_BE_STATIC_ANON: Could be refactored into a named static inner class	내부 클래스이지만, 이를 만든 객체에 대한 내부 참조를 사용하지 않음.
DB	DB_DUPLICATE_SWITCH_CLAUSES: Method uses the same code for two switch clauses	코드가 중복 사용됨.
PZLA	PZLA_PREFER_ZERO_LENGTH_ARRAYS: Consider returning a zero length array rather than null	null을 사용하는 것이 적절한지 검토해봐야함. 0을 사용하는게 나은 경우가 있다.
REC	REC_CATCH_EXCEPTION: Exception is caught when Exception is not thrown	하나의 catch로 여러 예외를 처리하고 있음. Throw 하지 않은 Exception까지 catch 하려함.
SF	SF_SWITCH_NO_DEFAULT: Switch statement found where default case is missing	switch 문에 default case가 없음.
	SF_SWITCH_FALLTHROUGH: Switch statement found where one case falls through to the next case	switch 문에 특정 case가 수행 후, 다음 case로 fall through증임.
UwF	UwF_FIELD_NOT_INITIALIZED_IN_CONSTRUCTOR: Field not initialized in constructor but dereferenced without null check	생성 시 초기화 되지 않은 값이 null check없이 참조 해제 됨.

# Static Analysis - SonarCloud



▲ Summary

# Static Analysis - Cyclomatic Complexity



◀ java파일 별 Cyclomatic Complexity

# Static Analysis

```
catch (DataStoreError | InvalidBillException ex) {  
    ex.printStackTrace();  
    parentFrame.setContentPane(new SelectFunction(parentFrame, system).getPanel());  
}
```

JAVA\_44 : printStackTrace 메서드가 사용됨.



시스템 내부 데이터나 디버깅 관련 정보가 공개되면 공격의 빌미가 될 수 있다.



printStackTrace 메서드를 사용하지 말아야 한다.

# Static Analysis

```
public static Main getInstance() {  
    if ( application == null ) {  
        application = new Main();  
    }  
  
    return application;  
}
```

OBJ05-J : PUBLIC 메서드에서 PRIVATE 필드 리턴 시 레퍼런스를 그대로 리턴한다.



객체 내부의 상태를 외부에서 수정하도록 해서는 안되므로 MUTABLE 멤버에 대한 레퍼런스를 외부에 노출해서는 안된다.



CLONE을 리턴한다

# Static Analysis

```
try {  
    DataStore dataStore = new DataStore();  
    dataStore.saveAdminData(admins);  
} catch (DataStoreError er) {  
  
}
```

EGOV\_01\_EMPTYCATCHBLOCK / JAVA\_65 / ERROR00-J / SonarCloud / Findbugs DE  
: 빈 CATCH 절이 존재한다. 의도한 것이라면 COMMENT로 표시하는 것을 권장한다.



CATCH BLOCK에 반드시 예외를 다루는 코드를 작성한다.

# Static Analysis

```
switch (e.getKeyCode()) {  
    case KeyEvent.VK_ENTER:  
        next(parentFrame, system);  
}
```

EGOV\_12\_SWITCHSTMTSSHOUOLDHAVEDEFAULT / JAVA\_61 / SUN\_13 / SonarCloud / Findbugs SF  
: DEFAULT 없는 SWITCH 구문 사용을 금지한다.



SWITCH 구문에 반드시 DEFAULT LABEL이 있어야 한다.

# Static Analysis

```
private final int value;
```

EGOV\_14\_FINALFIELDCOULDBESTATIC / JAVA\_63 / SonarCloud / Findbugs MS

: 클래스의 FINAL 멤버 필드가 STATIC이 아니다.



FINAL FIELD를 STATIC으로 전환해서 OVERHEAD를 줄인다.

# Static Analysis

```
catch (AdminAlarmException e) {  
    System.out.println("Alter to admin");  
    //Alarm to Admin;
```

EGOV\_25\_SYSTEMPRINTLN / SonarCloud  
: PRINTLN 메서드가 사용되었다.



전용 로거를 사용할 것을 권장한다.

# Static Analysis

```
if (!card[i].equals("")) {  
    cardFields[i].setText(card[i].toString());  
}
```

EGOV\_33\_STRINGTOSTRING / SonarCloud / Findbugs DM  
: TOSTRING메서드가 사용되었다.

▼  
불필요한 TOSTRING 메서드 사용을 금지한다.

▼  
STRING 객체에서 TOSTRING 함수를 사용하는 것은 불필요하므로 해당 코드를 제거한다.

# Static Analysis

```
BufferedWriter bw = new BufferedWriter(output);
```

SonarCloud / Findbugs OS  
: Buffer를 Close하지 않았다.



Buffer를 Close해야 한다.

# Static Analysis

```
case QueryBalance: |
    if (!state.available()) {
        throw new NoneOfFunctionSelected();
    }
    break;
case QueryTransactionList:
    if (!state.available()) {
        throw new NoneOfFunctionSelected();
    }
    break;
```

SonarCloud / Findbugs RpC  
: 조건부 테스트가 중복 수행되었다.

▼  
조건부를 하나로 합친다.

# Static Analysis

```
} catch (Exception e) {  
}
```

ERROR08-J / SonarCloud / Findbugs REC  
: 하나의 catch로 여러 개의 예외를 처리하고 있다.



여러 가지 예외를 명시해야 한다.

# Static Analysis

```
case JOptionPane.YES_OPTION:
    try {
        system.askRenewCard(true);
    } catch (DataStoreError ex) {
    }

    JOptionPane.showMessageDialog(parentFrame,
        setLocalizedString(system, card[i].toString() + " 카드를 재발급 요청하셨습니다",
            "Info",
            JOptionPane.INFORMATION_MESSAGE);
case JOptionPane.NO_OPTION:
    parentFrame.setContentPane(new SelectFunction(parentFrame, system).getPanel());
    parentFrame.invalidate();
    parentFrame.validate();
    break;
```

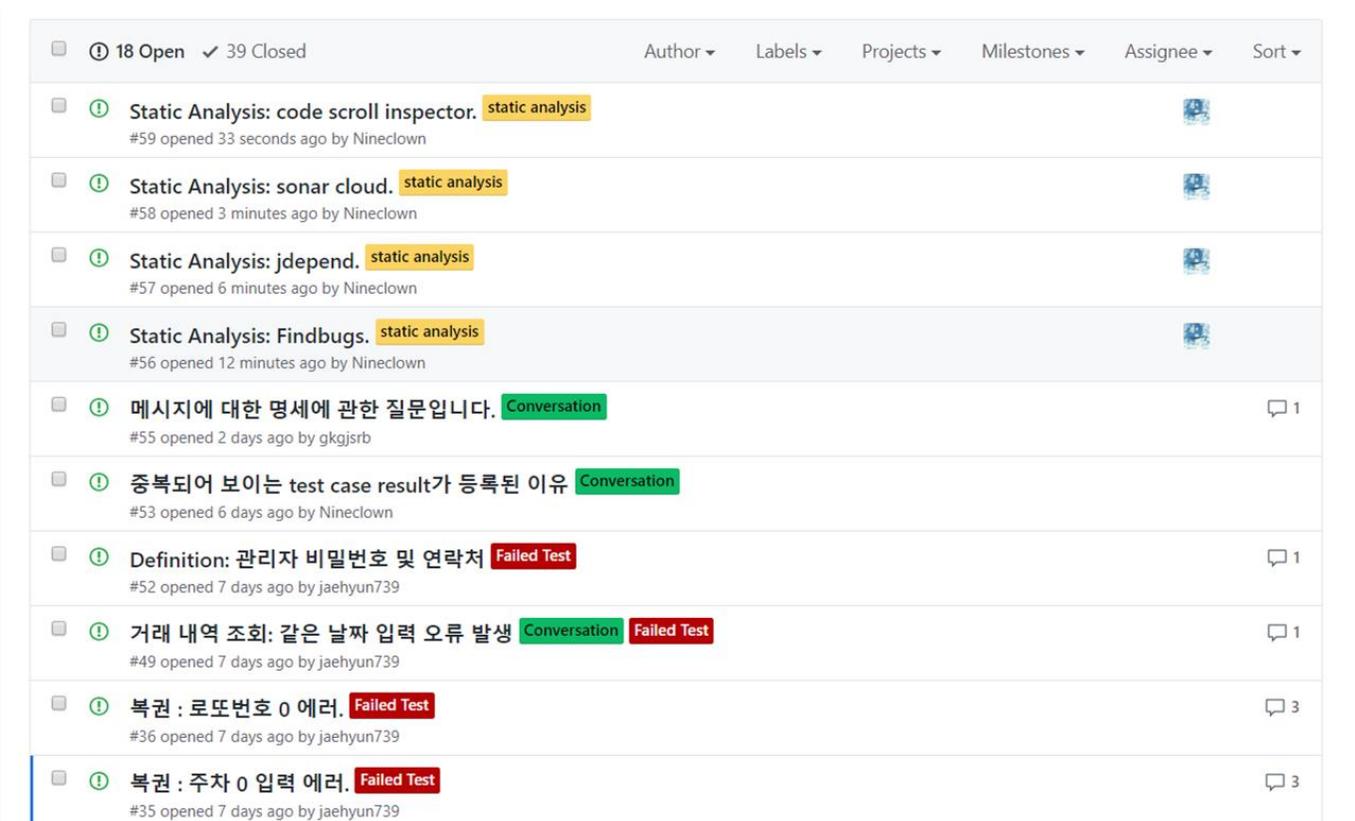
SonarCloud / Findbugs SF

: case문에 break;가 없다.



break;를 추가한다.

# Static Analysis - in CTIP



<input type="checkbox"/>	🕒 18 Open ✓ 39 Closed	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	🕒 1 Static Analysis: code scroll inspector. <b>static analysis</b>						
	#59 opened 33 seconds ago by Nineclown						
<input type="checkbox"/>	🕒 1 Static Analysis: sonar cloud. <b>static analysis</b>						
	#58 opened 3 minutes ago by Nineclown						
<input type="checkbox"/>	🕒 1 Static Analysis: jdepend. <b>static analysis</b>						
	#57 opened 6 minutes ago by Nineclown						
<input type="checkbox"/>	🕒 1 Static Analysis: Findbugs. <b>static analysis</b>						
	#56 opened 12 minutes ago by Nineclown						
<input type="checkbox"/>	🕒 1 메시지에 대한 명세에 관한 질문입니다. <b>Conversation</b>						1
	#55 opened 2 days ago by gkgjsrb						
<input type="checkbox"/>	🕒 1 중복되어 보이는 test case result가 등록된 이유 <b>Conversation</b>						
	#53 opened 6 days ago by Nineclown						
<input type="checkbox"/>	🕒 1 Definition: 관리자 비밀번호 및 연락처 <b>Failed Test</b>						1
	#52 opened 7 days ago by jaehyun739						
<input type="checkbox"/>	🕒 1 거래 내역 조회: 같은 날짜 입력 오류 발생 <b>Conversation</b> <b>Failed Test</b>						1
	#49 opened 7 days ago by jaehyun739						
<input type="checkbox"/>	🕒 1 복권 : 로또번호 0 에러. <b>Failed Test</b>						3
	#36 opened 7 days ago by jaehyun739						
<input type="checkbox"/>	🕒 1 복권 : 주차 0 입력 에러. <b>Failed Test</b>						3
	#35 opened 7 days ago by jaehyun739						

▲ CTIP 환경에 Issue 등록

# Static Analysis - in CTIP

The screenshot shows a Kanban board for the project 'The-Convenient-ATM'. The board is organized into four columns representing different stages of the project:

- To do (7 items):** Includes items like 'Static Analysis: code scroll inspector', 'Static Analysis: sonar cloud', 'Static Analysis: jdepend', and 'Static Analysis: Findbugs'. All are marked as 'static analysis'.
- In progress (6 items):** Includes items like 'Failed test: Admin의 입력에 따라 Admin 계정을 추가하는 기능', '즉시 언어변경 문제.', '복권 : 로또번호 0 에러.', and 'Failed test: 인터페이스의 언어가 전환 되는 기능'. Status tags include 'Failed Test' and 'Conversation'.
- Test result (4 items):** Includes items like 'Definition: 관리자 비밀번호 및 연락처', 'Failed test: 해당하는 회차의 당첨번호를 받아오는 기능', '거래 내역 조회: 같은 날짜 입력 오류 발생', and 'Definition: User가 선택한 조회 기간에 해당하는 거래 내역을 출력하는 기능'. Status tags include 'Failed Test' and 'Conversation'.
- Done (32 items):** Includes items like '지폐 보유량 변경', 'jar 파일.', '취소 버튼', '언어 변경', and 'Failed test: ATM 지폐 보유량 한도를 초과하는 금액을 입금한다.'. Status tags include 'bug' and 'help wanted'.

## ▲ Project Management

감사합니다 :)