Software Verification

-2nd system test & Static Analysis-

Team 6
201210908 윤성일
201311265 김상원
201214150 정성철

KU 건국대학교 KONKUK UNIV.

Specification Review

➔키보드 &마우스 입력에 대한 명세와 ATM에 들어있는 현금에 대한 가정이 추가

- 시스템에 대한 컴퓨터 상에서의 시뮬레이션이므로 모든 입력은 키보드와 마우스를 통해 이루어진다.
- ATM에 들어있는 현금은 무한대라고 가정한다.

➔ print error와 Check error 가 지워지고 새로운 Check Validation이 추가

R6.1	Check Validation	Hidden
R6.2	Update Database	Hidden

김 상 원
정 성 철
윤 성 일

Specification Review

➔ Use case by Event 에서 중복되던 case 수정

Use cased by Actor

Deposit	Withdraw	Transfer	Check balance	Pay utility bill
Exchange	Insert	Loan	Deposit without bankbook	

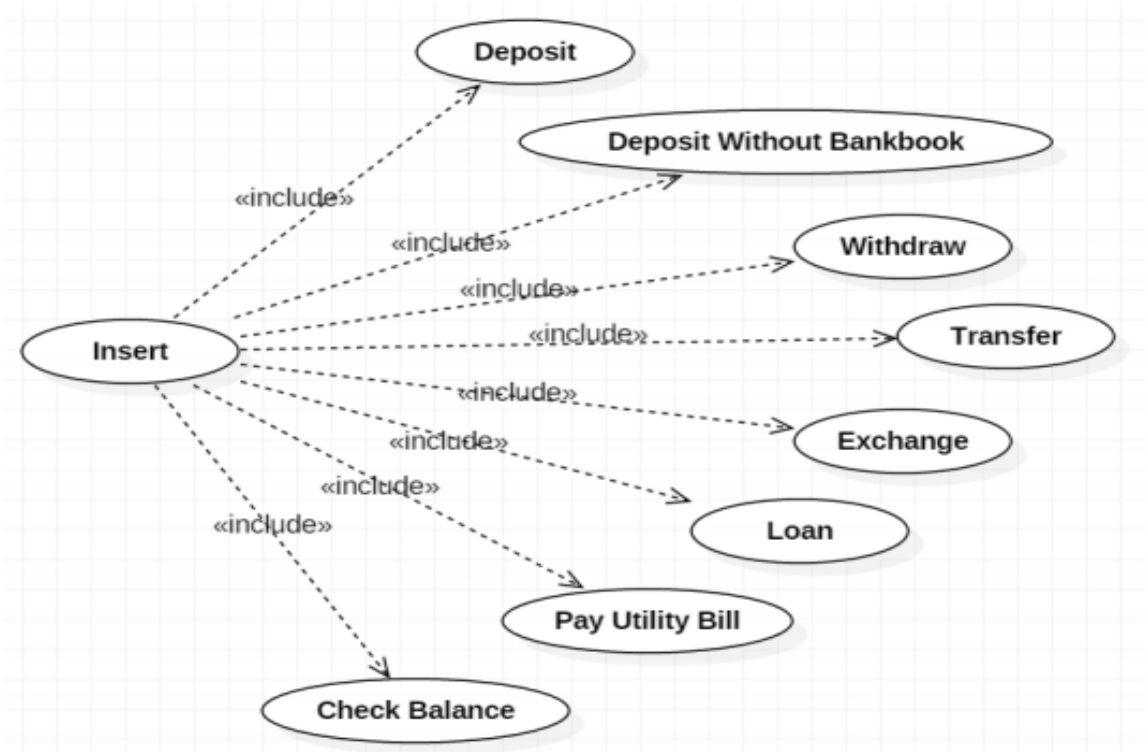
Use Cases by Event

Print Transaction Receipt	Print Error	Do Forced Termination
Take Charge	Check Password	Transaction Lock

김 상 원
정 성 철
윤 성 일

Specification Review

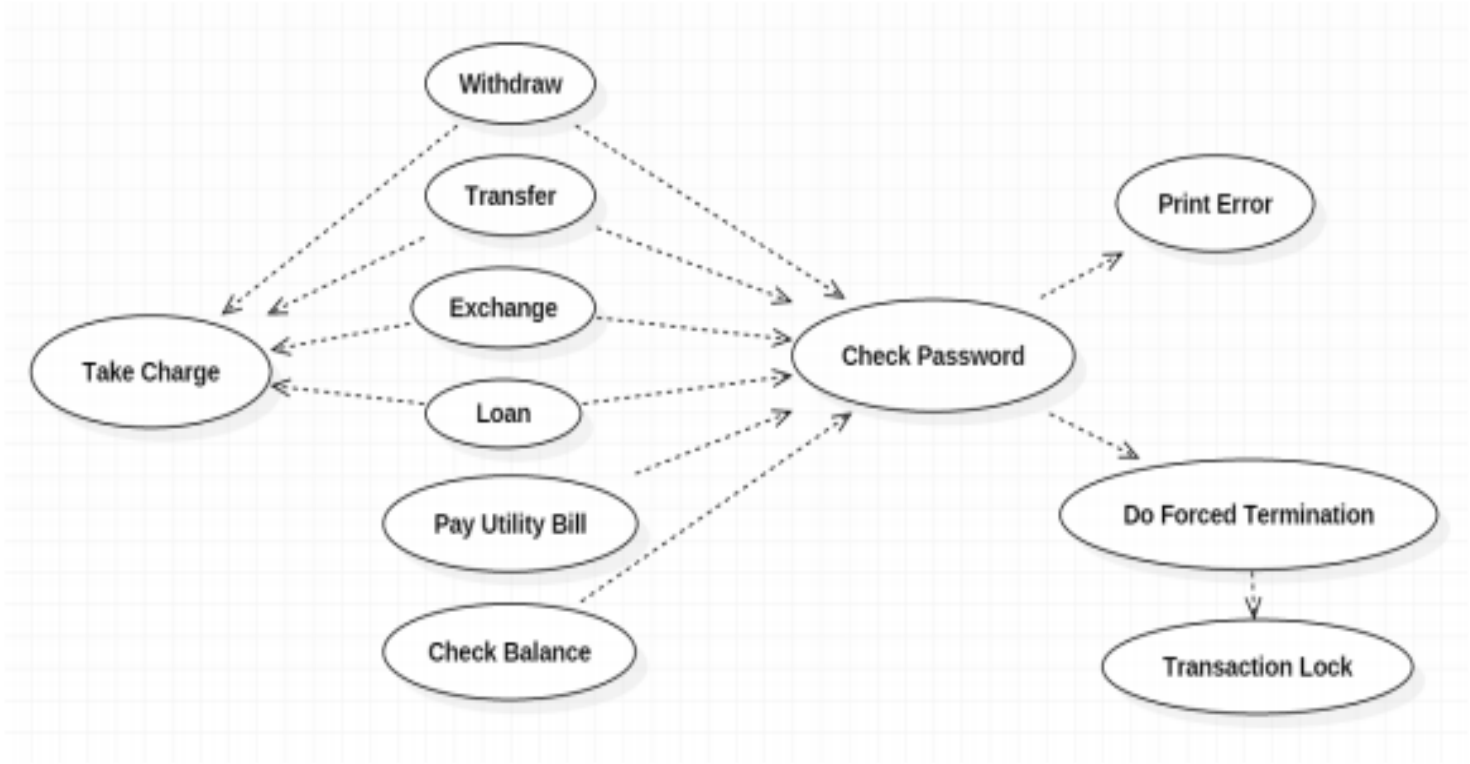
➔ Insert 의 포함 관계에 대해 상세하게 수정



김 상 원
정 성 철
윤 성 일

Specification Review

➔ exchange의 check Password에 대한 화살표가 추가



김 상 원
정 성 철
윤 성 일

Specification Review

➔ 시스템 보고서 v1 내용이 반영되지 않음

Use Case	2. Withdraw
Actors	Customer
Description	-Customer can withdraw cash from account by using card or bankbook. -Customer can only withdraw money in unit of 10000₩ and 50000₩ under balance. -Customer has withdrawing limit at a day. -Customer needs to choose the number of 50000₩. -Customer needs to know password of an account. -If customer successfully withdraws, ATM requests Offer to update server information.(Hidden)

김 상 원
정 성 철
윤 성 일

Specification Review

➔ 시스템 보고서 v1 내용이 반영되지 않음

Activity 2031. Define Essential Use Cases

Use Case	1. Deposit
Actor	Customer
Purpose	Deposit cash into account by bankbook or check card, or credit card
Overview	Customer inputs card or bankbook, and cash, to deposit cash into account or credit card.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R2.1, R2.3, R3.2, R4.1 Use case : Print Transaction Receipt, Print Error, Insert, Update Server Information
Pre-Requisites	N/A

김 상 원
정 성 철
윤 성 일

Specification Review

➔ 3회 이상 오류에서 3회 발생시로 Use case에 대한 명세가 보다 명확

Use Case	12. Do Forced Termination
Actor	(None)
Purpose	Immediately end transaction when error occurs 3 times
Overview	System ends transaction automatically and immediately when error occurs 3 times.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1. R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R2.1, R2.3, R2.4, R5.1, R6.2 Use Case : Deposit, Deposit Without Bankbook, Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Check Balance, Insert, Print Error, Transaction Lock, Update Database
Pre-Requisites	Use case "Print Error" occurred 3 times

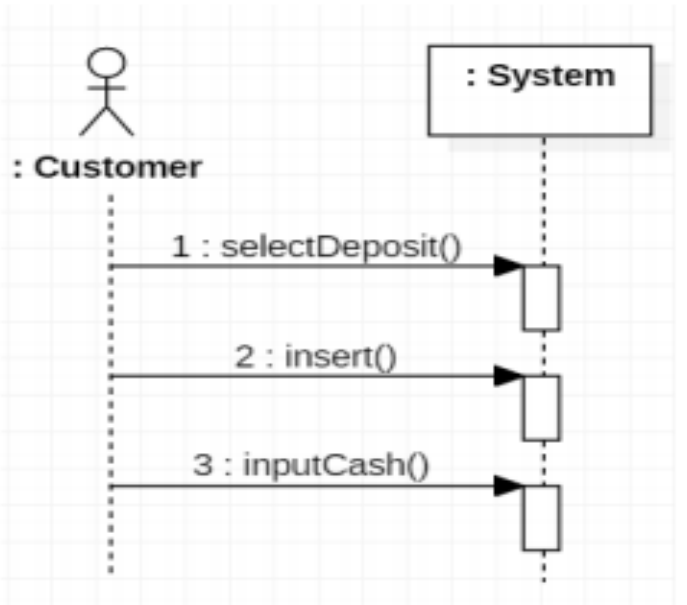
김 상 원
정 성 철
윤 성 일

Specification Review

➔ 시스템 보고서 v1 내용이 반영되지 않음

Activity 2035. Define System Sequence Diagrams

Use Case : Deposit
1. Customer selects deposit menu from the basic screen. 2. System prints out "Insert check card or bankbook or credit card". 3. Customer inserts credit card, check card or bankbook("Insert"). 4. Invoke "Check Validation". If inserted method is valid, ask customer to input cash. 5. Customer Inputs cash in unit of 10000₩ and 50000₩. 6. System checks amount of cash inputted. 7. If total amount of money is correct, invoke "Print transaction receipt". 8. If Transaction Receipt is successfully printed, invoke "Update Database".

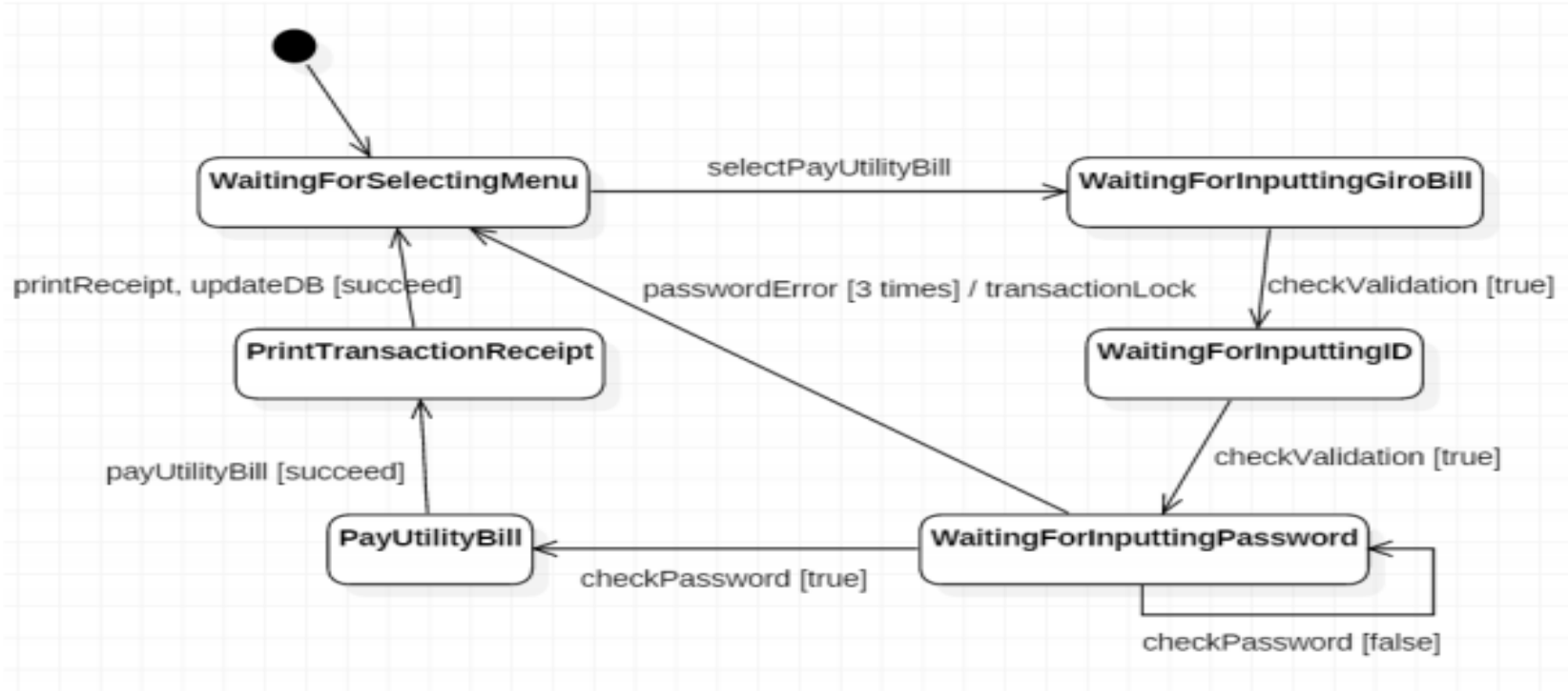


김 상 원
정 성 철
윤 성 일

Specification Review

➔ Password 3회 오류 발생시에 transctionLock에 대한 명시가 추가

- Pay Utility Bill



김 상 원
정 성 철
윤 성 일

Specification Review

➔ 모든 Function 과 Use case Operation의 관계가 정확하게 보이지않음

Activity 2039. Analyze (2030) Traceability Analysis

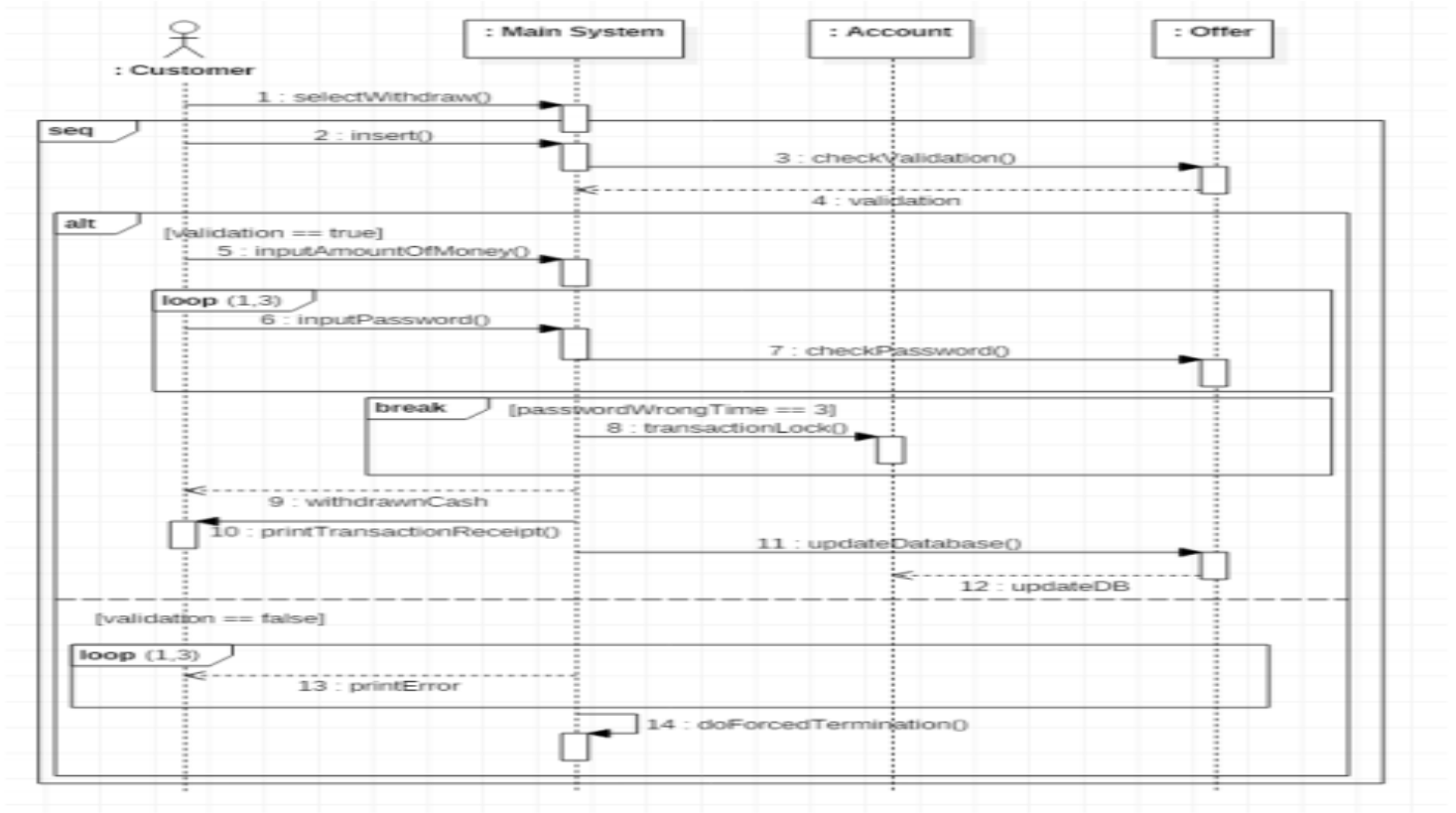
Ref.#	Function	Use Case	Operation
R1.1	Deposit	Deposit	1. selectDeposit
R1.2	Deposit Without Bankbook	Deposit Without Bankbook	2. selectDepositWithoutBankbook
R1.3	Withdraw	Withdraw	3. selectWithdraw
R1.4	Transfer	Transfer	4. selectTransfer
R1.5	Exchange	Exchange	5. selectExchange
R1.6	Loan	Loan	6. selectLoan
R1.7	Pay Utility Bill	Pay Utility Bill	7. selectPayUtilityBill
R1.8	Check Balance	Check Balance	8. selectCheckBalance
R2.1	Insert	Insert	9. selectForeignCurrency
R2.2	Print Transaction Receipt	Print Transaction Receipt	10. insert
R2.3	Print Error	Print Error	11. inputCash
R2.4	Do Forced Termination	Do Forced Termination	12. inputAccountNumber
R3.1	Take Charge	Take Charge	13. inputAmountOfMoney
R4.1	Check Password	Check Password	14. inputPassword
R5.1	Transaction Lock	Transaction Lock	15. Validation
R6.1	Check Validation	Check Validation	
R6.2	Update Database	Update Database	

김 상 원
정 성 철
윤 성 일

Specification Review

➔ Password checking에 대한 부분과 Password 오류 발생시에 대한 명시가 추가

3. Withdraw



김 상 원
정 성 철
윤 성 일

Specification Review

➔ CheckBalance 명시되어 있으나 실제 프로그램에선 거래내역조회 라는 이름으로 명시

Type	Method
Name	checkBalance()
Purpose	거래내역을 확인한다
Cross Reference	Usecase: 8 Functional: R1.8
Input(Method)	
Output(Method)	void
Abstract operation(Method)	최근 계좌의 거래내역을 출력한다.
Exceptional Courses of Events	-

김 상 원
정 성 철
윤 성 일

Specification Review

➔ CheckBalance 명시되어 있으나 실제 프로그램에선 거래내역조회 라는 이름으로 명시

Type	Method
Name	checkBalance()
Purpose	거래내역을 확인한다
Cross Reference	Usecase: 8 Functional: R1.8
Input(Method)	
Output(Method)	void
Abstract operation(Method)	최근 계좌의 거래내역을 출력한다.
Exceptional Courses of Events	-

김 상 원
정 성 철
윤 성 일

SOFTWARE
DEVELOP
ENVIRON

Brute Force Test

Brute Force test case

No.	테스트 설명
1	입금 페이지 안내 문구 확인
2	무통장 입금 해보기
3	거래 정지 기능 작동 확인
4	통장으로 대출 실행 시 안내 문구 유무 확인
5	키보드 입력 확인
6	대출 한도 기능 확인
7	아주 큰 금액을 입력
8	대출 상환 여부 확인
9	한도를 초과한 대출 실행 시 안내 문구 적절성 여부
10	대출 실행 후 GUI 상태 확인
11	자기 자신에게 송금 해보기
12	10000원 단위가 아닌 돈을 입금해보기
13	출금 단계 적절성 확인
14	지폐로 출금이 불가능한 금액 출금해보기
15	거래 정지된 계좌에서 출금해보기
16	5만원 이상 출금시 지폐 종류 개수 출력 확인
17	대출, 송금, 환전시 수수료 처리 여부 확인
18	신용 등급 조회
19	명세서 출력 여부 확인
20	환전에서 1만, 10만, 100만 버튼 작동 확인

김 상 원
정 성 철
윤 성 일

Brute Force Test

Brute Force test Result

No.	Expected output	P/F
1	입금 페이지 안내 문구 확인	P
2	무통장 입금 해보기	F
3	거래 정지 기능 작동 확인	P
4	통장으로 대출 실행 시 안내 문구 유무 확인	P
5	키보드 입력 확인	P
6	대출 한도 기능 확인	P
7	아주 큰 금액을 입력	P
8	대출 상환 여부 확인	P
9	한도를 초과한 대출 실행 시 안내 문구 적절성 여부	P
10	대출 실행 후 GUI 상태 확인	P
11	자기 자신에게 송금 해보기	P
12	10000원 단위가 아닌 돈을 입금해보기	P
13	출금 단계 적절성 확인	P
14	지폐로 출금이 불가능한 금액 출금해보기	F
15	거래 정지된 계좌에서 출금해보기	P
16	5만원 이상 출금시 지폐 종류 개수 출력 확인	P
17	대출, 송금, 환전시 수수료 처리 여부 확인	F
18	신용 등급 조회	F
19	명세서 출력 여부 확인	F
20	환전에서 1만, 10만, 100만 버튼 작동 확인	F

Pass : 14

Fail : 7

14/20 ➔ 70% Pass

1차 Brute Force Test 기준
70% 상승!

김 상 원
정 성 철
윤 성 일

Category Partition Test

Category Partition test case

➔키보드 입력이 추가 되어 새로운 Category Partition Test 진행

➔3 개의 Group & 11 개의 Category 로 진행

Generate Test Case

➔총 46080 개의 Test case 생성

➔Single Constraint 적용 후 기존 개의 test case 에서 123개의 Test case 생성

➔Error Constraint 적용 후 기존 123개의 test case 에서 68개의 test case 생성

➔If-Property Constraints 적용 후 기존 68개의 test case 에서 62개의 test case 생성

김 상 원
정 성 철
윤 성 일

Category Partition Test

Category Partition test case

Category	Representative classes of value	Ref #
Program mode	Deposit	101
	Deposit without bankbook	102
	Withdraw	103
	Loan	104
	Transfer	105
	Exchange	106
	Pay utility bill	107
	Check balance	108
Availability	Available	111
	Unavailable	112
Address length	Valid length	201
	Short address	202
	Long address	203
Bill range	Valid unit (10000, 50000)	211
	0 < 10000	212
	10000 < 50000	213
	50000 < 100000	214
	Over integer range	215
Address input type	Number	221
	Not number	222
Bill input type	Number	231
	Not number	232
Check sender address	Exist address	401
	Non-exist address	402
Check loan limit	Input < limit	411
	Input > limit	412

김 상 원
정 성 철
윤 성 일

Category Partition Test

Category Partition test case

Category	Representative classes of value	Ref #
Check password	Correct password	421
	Incorrect password	422
Check receiver address	Exist address	431
	Non-exist address	432
	Same with sender address	433
Check balance	Enough money	441
	Not enough money	442

김 상 원
정 성 철
윤 성 일

Category Partition Test

Single Constraint

Category	Representative classes of value
Address input type	Not number
Bill input type	Not number
Address length	Short address
	Long address
Bill range	Over integer range
Check sender address	Non-exist address
Check loan limit	Input > limit
Check password	Incorrect password
Check receiver address	Non-exist address
Check balance	Not enough money
Program mode	Check balance

Error Constraint

Category	Representative classes of value
Check receiver address	Same with sender address

김 상 원
정 성 철
윤 성 일

Category Partition Test

If-Property Constraints

Category	Representative classes of value	Constraints
Program mode	Deposit	[property deposit]
	Deposit without bankbook	[property noBankbook]
	Withdraw	[property withdraw]
	Loan	[property loan]
	Transfer	[property transfer]
	Exchange	[property exchange]
	Pay utility bill	[property utility]
Availability	Available	[property AB]
	Unavailable	[property UAB]
Bill input type	Number	[if !utility]
	Not number	[if !utility]
Address length	Valid lenght	[property VA]
Bill range	Valid unit (10000, 50000)	[if !utility] [property VM]
	0 < 10000	[if !utility]
	10000 < 50000	[if !utility]
	50000 < 100000	[if !utility]
Check sender address	Exist address	[if VA && (transfer || exchange || utility || loan || withdraw)] [property ESA]
Check loan limit	Input < limit	[if (VM && ESA) && loan]
Check password	Correct password	[if ESA && !deposit && !noBankbook] [property CP]
Check receiver address	Exist address	[if VA && (transfer || deposit || noBankbook)] [property ERA]
Check balance	Enough money	[if CP && (withdraw || transfer || utility || exchange)]

김 상 원
정 성 철
윤 성 일

Category Partition Test

Category Partition Test Result

ref #	sequence	description	P/F
1	108	계좌 잔액을 조회한다.	P
2	222	숫자가 아닌 문자로 계좌번호에 입력한다.	P
3	232	숫자가 아닌 문자로 입금금액을 입력한다.	P
4	202	기준 길이보다 짧은 계좌번호를 입력한다.	P
5	203	기준 길이보다 긴 계좌번호를 입력한다.	P
6	215	금액 입력란에 int타입이 표현가능한 범위 이상의 값을 입력한다.	P
7	402	송신 계좌 번호에 존재 하지 않는 계좌 번호를 입력한다.	P
8	412	대출 금액에 대출 한도보다 큰 금액을 입력한다.	P
9	422	틀린 비밀번호를 입력한다.	P
10	432	수신 계좌 번호에 존재 하지 않는 계좌 번호를 입력한다.	F
11	433	송신 계좌 번호와 같은 계좌 번호를 수신 계좌 번호로 입력한다.	P
12	442	잔액이 충분하지 않은 계좌에 출금, 송금, 환전을 실행한다.	P
13	101, 111, 201, 211, 221, 231, 431	면서 거래 정지되지 않은 계좌에 적절한 계좌번호와 적절한 금액을 입력하고 존재하는 송신 계	F
14	101, 111, 201, 212, 221, 231, 431	정지되지 않은 계좌에 적절한 계좌번호와 0원에서 10000원 사이의 금액을 입력하고 존재하는	F
15	101, 111, 201, 213, 221, 231, 431	지되지 않은 계좌에 적절한 계좌번호와 10000원에서 50000원 사이의 금액을 입력하고 존재하	F
16	101, 111, 201, 214, 221, 231, 431	지되지 않은 계좌에 적절한 계좌번호와 50000원에서 100000원 사이의 금액을 입력하고 존재하	F
17	101, 112, 201, 211, 221, 231, 431	하면서 거래가 정지된 계좌에 적절한 계좌번호와 적절한 금액을 입력하고 존재하는 송신 계좌	F
18	101, 112, 201, 212, 221, 231, 431	래가 정지된 계좌에 적절한 계좌번호와 0원에서 10000원 사이의 금액을 입력하고 존재하는 송	F
19	101, 112, 201, 213, 221, 231, 431	가 정지된 계좌에 적절한 계좌번호와 10000원에서 50000원 사이의 금액을 입력하고 존재하는	F
20	101, 112, 201, 214, 221, 231, 431	가 정지된 계좌에 적절한 계좌번호와 50000원에서 100000원 사이의 금액을 입력하고 존재하는	F
21	102, 111, 201, 211, 221, 231, 431	행하면서 거래 정지되지 않은 계좌에 적절한 계좌번호와 적절한 금액을 입력하고 존재하는 송신	F

김 상 원
정 성 철
윤 성 일

Category Partition Test

Category Partition Test Result

ref #	sequence	description	P/F
22	102, 111, 201, 212, 221, 231, 431	거래 정지되지 않은 계좌에 적절한 계좌번호와 0원에서 10000원 사이의 금액을 입력하고 존재	F
23	102, 111, 201, 213 221, 231,, 431	래 정지되지 않은 계좌에 적절한 계좌번호와 10000원에서 50000원 사이의 금액을 입력하고 존	F
24	102, 111, 201, 214, 221, 231, 431	래 정지되지 않은 계좌에 적절한 계좌번호와 50000원에서 100000원 사이의 금액을 입력하고 존	F
25	102, 112, 201, 211, 221, 231, 431	진행하면서 거래가 정지된 계좌에 적절한 계좌번호와 적절한 금액을 입력하고 존재하는 송신 계	F
26	102, 112, 201, 212, 221, 231, 431	서 거래가 정지된 계좌에 적절한 계좌번호와 0원에서 10000원 사이의 금액을 입력하고 존재하	F
27	102, 112, 201, 213, 221, 231, 431	거래가 정지된 계좌에 적절한 계좌번호와 10000원에서 50000원 사이의 금액을 입력하고 존재	F
28	102, 112, 201, 214, 221, 231, 431	거래가 정지된 계좌에 적절한 계좌번호와 50000원에서 100000원 사이의 금액을 입력하고 존재	F
29	103, 111, 201, 211, 221, 231, 401,	지 않은 계좌에 적절한 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서	P
30	103, 111, 201, 212, 221, 231, 401,	계좌에 적절한 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서 0원에서	P
31	103, 111, 201, 213, 221, 231, 401,	좌에 적절한 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서 10000원에	P
32	103, 111, 201, 214, 221, 231, 401,	좌에 적절한 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서 50000원에	P
33	103, 112, 201, 211, 221, 231, 401,	지된 계좌에 적절한 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서 적	P
34	103, 112, 201, 212, 221, 231, 401,	좌에 적절한 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서 0원에서 1	P
35	103, 112, 201, 213, 221, 231, 401,	에 적절한 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서 10000원에서	P
36	103, 112, 201, 214, 221, 231, 401,	적절한 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서 50000원에서 1	P
37	104, 111, 201, 211, 221, 231, 401,	를 진행하면서 거래 정지되지 않은 계좌에 적절한 비밀번호와 적절한 금액을 입력하고 대출을	P
38	104, 111, 201, 212, 221, 231, 401,	하면서 거래 정지되지 않은 계좌에 적절한 비밀번호와 0원에서 10000원 사이의 금액을 입력하	F
39	104, 111, 201, 213, 221, 231, 401,	서 거래 정지되지 않은 계좌에 적절한 비밀번호와 10000원에서 50000원 사이의 금액을 입력	F
40	104, 111, 201, 214, 221, 231, 401,	서 거래 정지되지 않은 계좌에 적절한 비밀번호와 50000원에서 100000원 사이의 금액을 입력	F
41	104, 112, 201, 211, 221, 231, 401,	거래를 진행하면서 거래 정지된 계좌에 적절한 비밀번호와 적절한 금액을 입력하고 대출을 받	F
42	104, 112, 201, 212, 221, 231, 401,	행하면서 거래 정지된 계좌에 적절한 비밀번호와 0원에서 10000원 사이의 금액을 입력하고 대	F

김 상 원
정 성 철
윤 성 일

Category Partition Test

Category Partition Test Result

ref #	sequence	description	P/F
43	104, 112, 201, 213, 221, 231, 401,	하면서 거래 정지된 계좌에 적절한 비밀번호와 10000원에서 50000원 사이의 금액을 입력하고	F
44	104, 112, 201, 214, 221, 231, 401,	하면서 거래 정지된 계좌에 적절한 비밀번호와 50000원에서 100000원 사이의 금액을 입력하고	F
45	105, 111, 201, 211, 221, 231, 401,	한 내 계좌번호와 정확한 비밀번호를 입력하고 정확한 상대방 계좌번호를 입력하고 충분한 잔	F
46	105, 111, 201, 212, 221, 231, 401,	좌번호와 정확한 비밀번호를 입력하고 정확한 상대방 계좌번호를 입력하고 충분한 잔액이 있	F
47	105, 111, 201, 213, 221, 231, 401,	번호와 정확한 비밀번호를 입력하고 정확한 상대방 계좌번호를 입력하고 충분한 잔액이 있는상	F
48	105, 111, 201, 214, 221, 231, 401,	번호와 정확한 비밀번호를 입력하고 정확한 상대방 계좌번호를 입력하고 충분한 잔액이 있는상	F
49	105, 112, 201, 211, 221, 231, 401,	내 계좌번호와 정확한 비밀번호를 입력하고 정확한 상대방 계좌번호를 입력하고 충분한 잔액이	F
50	105, 112, 201, 212, 221, 231, 401,	호와 정확한 비밀번호를 입력하고 정확한 상대방 계좌번호를 입력하고 충분한 잔액이 있는 상	F
51	105, 112, 201, 213, 221, 231, 401,	와 정확한 비밀번호를 입력하고 정확한 상대방 계좌번호를 입력하고 충분한 잔액이 있는 상태	F
52	105, 112, 201, 214, 221, 231, 401,	와 정확한 비밀번호를 입력하고 정확한 상대방 계좌번호를 입력하고 충분한 잔액이 있는상태에	F
53	106, 111, 201, 211, 221, 231, 401,	거래 정지되지 않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 적절한 금액을	F
54	106, 111, 201, 212, 221, 231, 401,	되지 않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 0원에서 10000원사이의	F
55	106, 111, 201, 213, 221, 231, 401,	지 않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 10000원에서 50000원사이	F
56	106, 111, 201, 214, 221, 231, 401,	지 않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 50000원에서 100000원사이	F
57	106, 112, 201, 211, 221, 231, 401,	거래 정지된 않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 적절한 금액을	F
58	106, 112, 201, 212, 221, 231, 401,	지된 않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 0원에서 10000원사이의	F
59	106, 112, 201, 213, 221, 231, 401,	된 않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 10000원에서 50000원사이	F
60	106, 112, 201, 214, 221, 231, 401,	않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 50000원에서 100000원사이	F
61	107, 111, 201, 221, 401, 421, 441	지 않은 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태	P
62	107, 112, 201, 221, 401, 421, 441	지된 계좌에 정확한 내 계좌번호와 정확한 비밀번호를 입력하고 충분한 잔액이 있는 상태에서	P

김 상 원
정 성 철
윤 성 일

Category Partition Test

Category Partition test Result

ref #	P/F	ref #	P/F	ref #	P/F
1	P	21	F	41	F
2	P	22	F	42	F
3	P	23	F	43	F
4	P	24	F	44	F
5	P	25	F	45	F
6	P	26	F	46	F
7	P	27	F	47	F
8	P	28	F	48	F
9	P	29	P	49	F
10	F	30	P	50	F
11	P	31	P	51	F
12	P	32	P	52	F
13	F	33	P	53	F
14	F	34	P	54	F
15	F	35	P	55	F
16	F	36	P	56	F
17	F	37	P	57	F
18	F	38	F	58	F
19	F	39	F	59	F
20	F	40	F	60	F
				61	P
				62	P

Pass : 20

Fail : 42

20/62➔32% Pass

김 상 원
정 성 철
윤 성 일

Pairwise Test

Pairwise test table unit

동작모드: deposit, no_bank, withdraw, loan, transfer, exchange, utility, check_bal
계좌상태: avail, n/a
계좌형식: num, non-num
금액형식: num, non-num
계좌길이: vaild, short, long
금액범위: valid, 0~10K, 10~50K, 500~100K
송금계좌: exist, non-exi
대출한계: in<lim, in>lim
비밀번호: correct, incorrect
받는계좌: exist, non-exi
계좌잔고: enough, not_eno
if [동작모드] in {"deposit", "no_bank", "check_bal"} then [송금계좌] = "non-exi";
if [동작모드] in {"withdraw", "loan", "exchange", "utility"} then [받는계좌] = "non-exi";
if [동작모드] not in {"loan"} then [대출한계] = "in<lim";
if [동작모드] in {"utility"} then [금액형식] = "";

김 상 원
정 성 철
윤 성 일

Pairwise Test

Pairwise test Result

동작모드	계좌상태	계좌형식	금액형식	계좌길이	금액범위	송금계좌	대출한계	비밀번호	받는 계좌	계좌잔고	P/F
deposit	n/a	num	non-num	vaild	0~10K	non-exi	in<lim	correct	non-exi	enough	P
withdraw	avail	non-num	num	short	0~10K	exist	in<lim	incorrect	non-exi	not_eno	F
transfer	avail	num	non-num	long	500~100K	exist	in<lim	correct	exist	not_eno	F
loan	n/a	non-num	num	long	valid	non-exi	in>lim	incorrect	non-exi	enough	F
deposit	avail	num	num	vaild	valid	non-exi	in<lim	incorrect	exist	not_eno	F
loan	n/a	num	non-num	short	10~50K	exist	in>lim	correct	non-exi	not_eno	P
exchange	avail	non-num	non-num	vaild	10~50K	exist	in<lim	incorrect	non-exi	enough	P
no_bank	n/a	non-num	num	short	10~50K	non-exi	in<lim	correct	exist	enough	P
check_bal	avail	non-num	non-num	short	valid	non-exi	in<lim	correct	exist	enough	F
exchange	n/a	non-num	num	short	500~100K	non-exi	in<lim	incorrect	non-exi	not_eno	F
exchange	avail	num	num	long	valid	exist	in<lim	correct	non-exi	enough	F
loan	avail	non-num	num	vaild	0~10K	non-exi	in>lim	correct	non-exi	enough	P
no_bank	avail	num	non-num	vaild	500~100K	non-exi	in<lim	incorrect	non-exi	enough	F
transfer	n/a	non-num	num	short	valid	non-exi	in<lim	incorrect	non-exi	enough	F
deposit	n/a	non-num	non-num	long	500~100K	non-exi	in<lim	correct	non-exi	not_eno	F
withdraw	n/a	num	non-num	long	10~50K	non-exi	in<lim	correct	non-exi	enough	F
transfer	n/a	non-num	num	vaild	10~50K	exist	in<lim	correct	exist	enough	P
withdraw	avail	num	non-num	vaild	valid	exist	in<lim	incorrect	non-exi	not_eno	P
check_bal	n/a	num	num	vaild	10~50K	non-exi	in<lim	incorrect	non-exi	not_eno	P
check_bal	n/a	num	non-num	long	500~100K	non-exi	in<lim	correct	non-exi	not_eno	F
no_bank	avail	non-num	non-num	long	0~10K	non-exi	in<lim	correct	exist	not_eno	F
transfer	avail	non-num	num	long	0~10K	non-exi	in<lim	incorrect	exist	not_eno	F
withdraw	n/a	non-num	num	short	500~100K	exist	in<lim	correct	non-exi	not_eno	F
deposit	n/a	non-num	non-num	short	10~50K	non-exi	in<lim	correct	exist	enough	F
loan	n/a	num	num	long	500~100K	non-exi	in<lim	incorrect	non-exi	enough	F
no_bank	avail	non-num	non-num	long	valid	non-exi	in<lim	correct	exist	not_eno	F
check_bal	avail	num	num	vaild	0~10K	non-exi	in<lim	correct	non-exi	not_eno	P
loan	avail	non-num	num	vaild	500~100K	exist	in>lim	correct	non-exi	enough	P
exchange	n/a	num	num	vaild	0~10K	exist	in<lim	incorrect	non-exi	not_eno	P

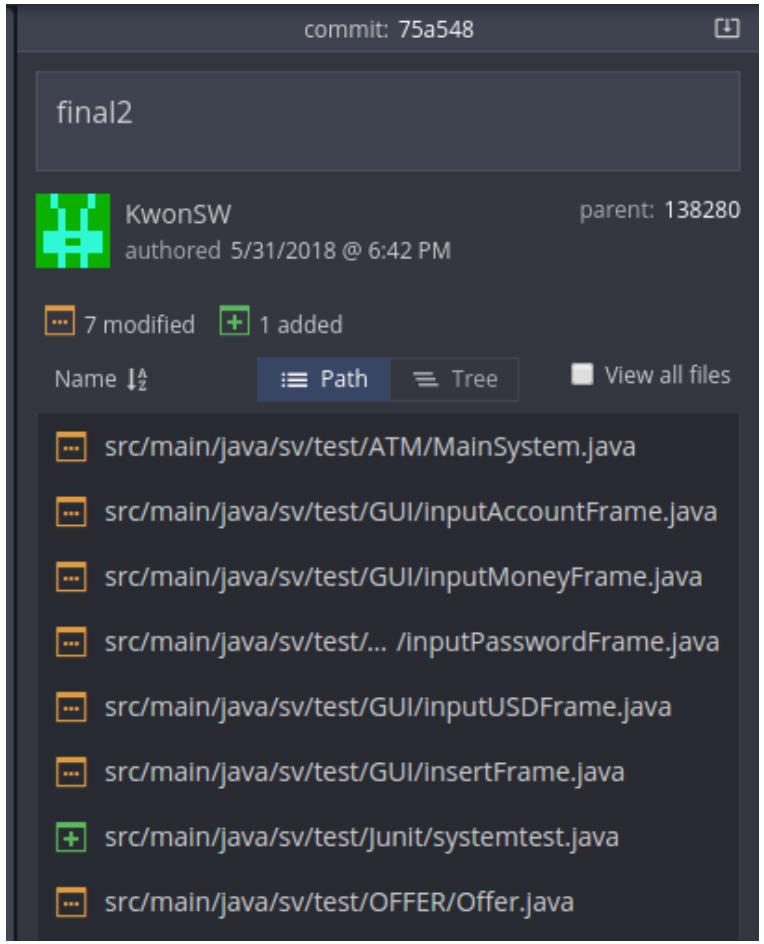
Pass : 11

Fail : 18

11/29 ➔ 37% Pass

김 상 원
정 성 철
윤 성 일

정적 분석



5월 31일 오후 6시 44분 업로드 #52

Sonarqube

Findbugs / Checkstyle / PMD

Codescroll Inspector for JAVA

김 상 원
정 성 철
윤 성 일

Result - Sonar way

총 이슈 258개 발생

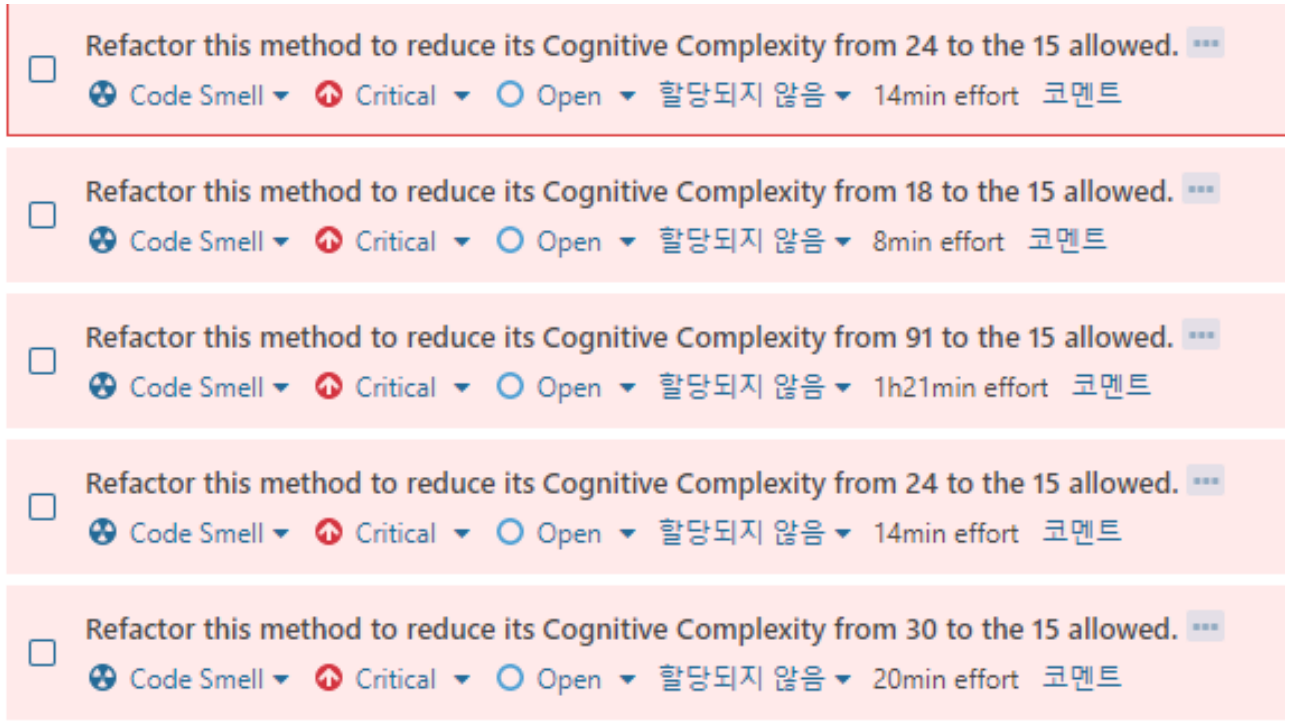
보안성 E등급

유지보수성 B등급

김 상 원
정 성 철
윤 성 일

Result - Sonar way

주요 이슈



너무 많은 중첩된 if-else문과 for문들로 인해 Complexity가 너무 높아짐.

김 상 원
정 성 철
윤 성 일

Result - Sonar way

주요 이슈

```
else if (e.getSource() == b1) {
	if (money.getText().length() > 1) {
		int num = 0;
		for (int k = money.getText().length() - 2; money.getText().charAt(k) == '0'; k--) {
			num++;
		}
		if (num > 4) {
			money.setText(money.getText().substring(0, money.getText().length() - num + 3) + "��");
		} else {
			for (num = 4 - num; num > 0; num--)
				money.setText(money.getText().substring(0, money.getText().length() - 1) + "0��");
		}
	} else
		money.setText("10000��");
else if (e.getSource() == b10) {
	if (money.getText().length() > 1) {
		int num = 0;
		for (int k = money.getText().length() - 2; money.getText().charAt(k) == '0'; k--) {
			num++;
		}
		if (num > 5) {
			money.setText(money.getText().substring(0, money.getText().length() - num + 4) + "��");
		} else {
			for (num = 5 - num; num > 0; num--)
				money.setText(money.getText().substring(0, money.getText().length() - 1) + "0��");
		}
	} else
```

```
	} else {
		switch (menu) {
		case 0: // ϐU϶
			if (money.getText().length() > 5 && money.getText()
					.substring(money.getText().length() - 5, money.getText().length() - 1).equals("0000")) {
				if ((error = main.deposit(
						Integer.parseInt(money.getText().substring(0, money.getText().length() - 1)))) == 0) {
					new ReceiptFrame(main);
				} else {
					new errorPrintFrame(error);
				}
				this.dispose();
			} else {
				state.setText("10000϶϶ ϐϐϐϐϐθ϶ ϐUϡϐϐϐϐUθ϶.");
				state.setOpaque(true);
				state.setBackground(Color.LIGHT_GRAY);
			}
			break;
		case 1: // ϐϐϐϐϐϐϐU϶
			if (money.getText().length() > 5 && (money.getText()
					.substring(money.getText().length() - 5, money.getText().length() - 1).equals("0000"))) {
				if ((error = main.depositWithoutBank(
						Integer.parseInt(money.getText().substring(0, money.getText().length() - 1)))) == 0) {
					new ReceiptFrame(main);
				} else {
					new errorPrintFrame(error);
				}
				this.dispose();
			} else {
				state.setText("10000϶϶ ϐϐϐϐϐθ϶ ϐUϡϐϐϐϐUθ϶.");
				state.setOpaque(true);
				state.setBackground(Color.LIGHT_GRAY);
			}
			break;
		case 2: // ϐϐü
			main.bank = Integer.parseInt(money.getText().substring(0, money.getText().length() - 1));
			new inputPasswordFrame(4, main, account);
			this.dispose();
			break;
		case 3: // ϐϐϐ
```

김 상 원
정 성 철
윤 성 일

Result - Sonar way

주요 이슈

Password 라는 이름의 변수를 포착하여 이에 대한 보안 처리를 요구함.

김 상 원
정 성 철
윤 성 일

Result - Sonar way

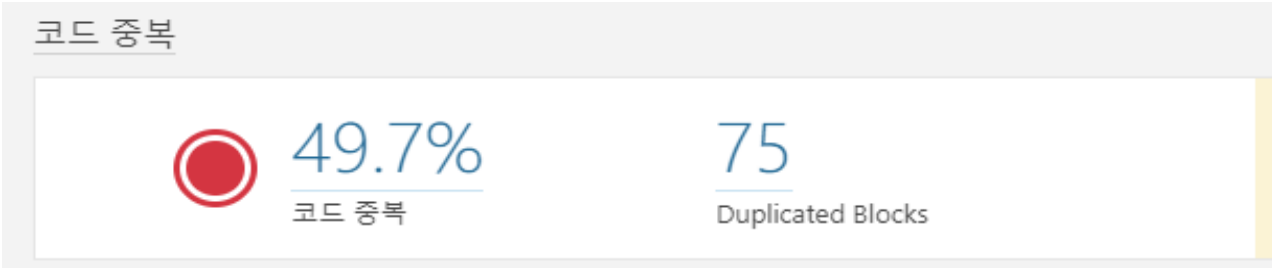
```
 } else if (e.getSource() == b[2]) {
        if (count == 0) {
                password = "2";
        } else {
                password = password + "2";
        }
        l[count++].setText("*");
        checkPassword();
 } else if (e.getSource() == b[3]) {
        if (count == 0) {
                password = "3";
        } else {
                password = password + "3";
        }
        l[count++].setText("*");
        checkPassword();
 } else if (e.getSource() == b[4]) {
        if (count == 0) {
                password = "4";
        } else {
                password = password + "4";
        }
        l[count++].setText("*");
        checkPassword();
 } else if (e.getSource() == b[5]) {
        if (count == 0) {
                password = "5";
        l[count++].setText("*");
        checkPassword();
 } else if (e.getSource() == b[6]) {
        if (count == 0) {
                password = "6";
        } else {
                password = password + "6";
        }
        l[count++].setText("*");
        checkPassword();
 } else if (e.getSource() == b[7]) {
        if (count == 0) {
                password = "7";
        } else {
                password = password + "7";
        }
        l[count++].setText("*");
        checkPassword();
 } else if (e.getSource() == b[8]) {
        if (count == 0) {
                password = "8";
        } else {
                password = password + "8";
        }
        l[count++].setText("*");
        checkPassword();
 } else if (e.getSource() == b[9]) {
        if (count == 0) {
                password = "9";
```

김 상 원
정 성 철
윤 성 일

SOFTWARE
DEVELOP
ENVIRON

코드 중복도 분석

	Duplicated lines (%)	Duplicated lines
src/main/java/sv/test/GUI/inputMoneyFrame.java	82.3%	399
src/main/java/sv/test/GUI/inputUSDFrame.java	75.7%	215
src/main/java/sv/test/GUI/insertFrame.java	73.5%	303
src/main/java/sv/test/GUI/inputAccountFrame.java	69.2%	173
src/main/java/sv/test/GUI/inputPasswordFrame.java	65.7%	287
src/main/java/sv/test/GUI/errorPrintFrame.java	54.2%	39
src/main/java/sv/test/GUI/ReceiptFrame.java	46.4%	39
src/main/java/sv/test/GUI/mainFrame.java	27.0%	41
src/main/java/sv/test/GUI/selectCountry.java	26.2%	27
src/main/java/sv/test/GUI/printLogFrame.java	21.3%	17
src/main/java/sv/test/Junit/systemtest.java	0.0%	0
src/main/java/sv/test/OFFER/Offer.java	0.0%	0
src/main/java/sv/test/ATM/MainSystem.java	0.0%	0
src/main/java/sv/test/Junit/Filetest.java	0.0%	0
src/main/java/sv/test/ATM/Account.java	0.0%	0



전체 프로그램 코드 중복도 49.7%

단일 파일 최대 중복도 82.3%

김 상 원
정 성 철
윤 성 일

코드 중복도 분석

```
add(b[0], 1, 6, 1, 1);
add(b[1], 0, 3, 1, 1);
add(b[2], 1, 3, 1, 1);
add(b[3], 2, 3, 1, 1);
add(b[4], 0, 4, 1, 1);
add(b[5], 1, 4, 1, 1);
add(b[6], 2, 4, 1, 1);
add(b[7], 0, 5, 1, 1);
add(b[8], 1, 5, 1, 1);
add(b[9], 2, 5, 1, 1);
```

inputMoneyFrame.java

동일 파일내 존재 코드

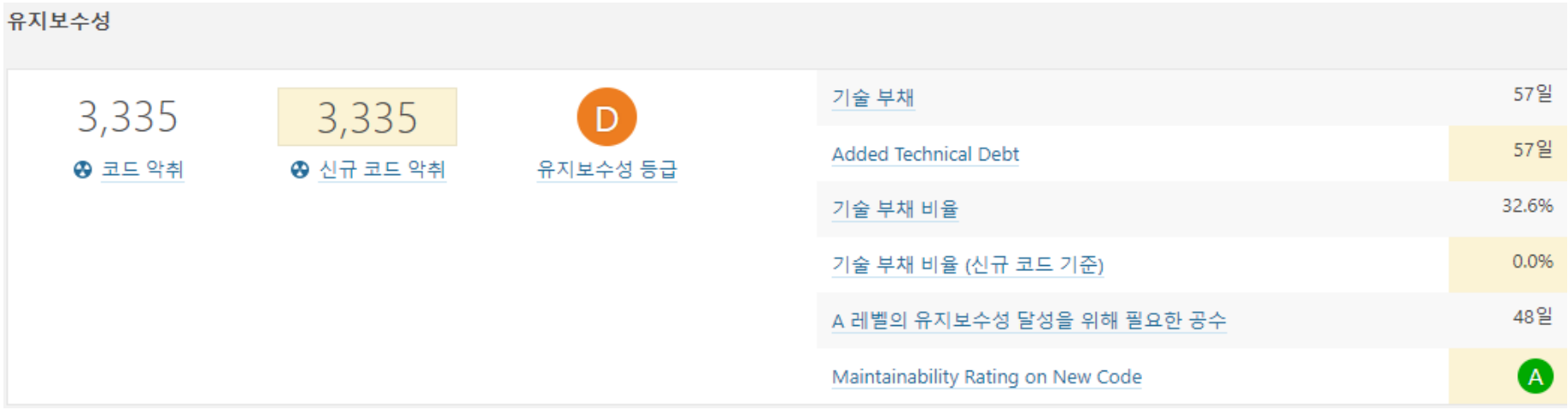
생성자 오버로딩 부분

```
add(b[0], 1, 6, 1, 1);
add(b[1], 0, 3, 1, 1);
add(b[2], 1, 3, 1, 1);
add(b[3], 2, 3, 1, 1);
add(b[4], 0, 4, 1, 1);
add(b[5], 1, 4, 1, 1);
add(b[6], 2, 4, 1, 1);
add(b[7], 0, 5, 1, 1);
add(b[8], 1, 5, 1, 1);
add(b[9], 2, 5, 1, 1);
```

대부분의 코드 중복이 GUI관련 파일에서 나타나는 것으로 나타남.

김 상 원
정 성 철
윤 성 일

Result – CheckStyle



유지보수성 D등급 판정

총 3335개의 코드 악취가 발견됨.

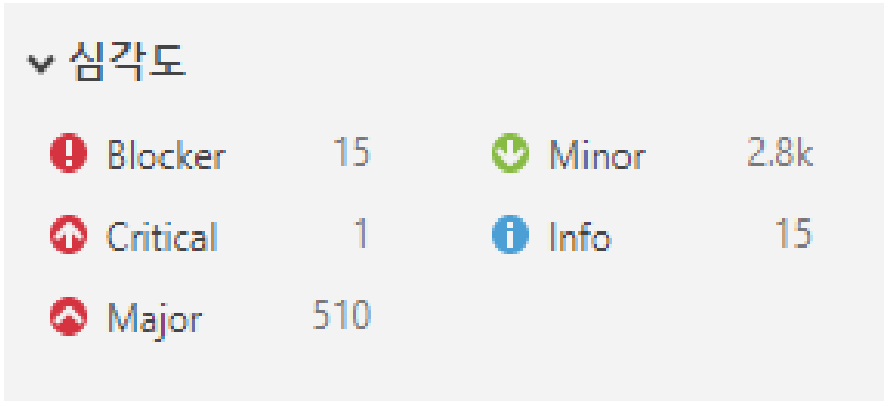
김 상 원
정 성 철
윤 성 일

Result – CheckStyle

Indentation (심각도 Minor)가

2500개로 가장 많이 발견

Major 이슈 510개 발견



김 상 원
정 성 철
윤 성 일

Result – CheckStyle

주요 이슈

String Literal Equality

Code Smell Critical 태그 없음 다음 기간 이후 적용됨 2018년 5월 4일 Checkstyle (Java) 상수/이슈: 5min

Checks that string literals are not used with == or !=.

문자열을 단순히 '!=' 으로만 비교함. Equals를 사용해야 함.

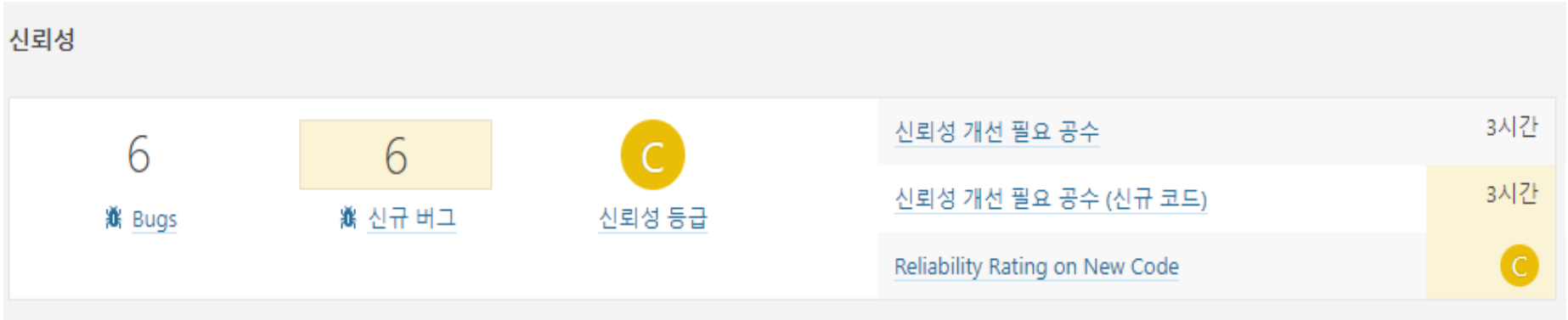
```
if(account.getBank() != "국민은행")
```

Literal Strings should be compared using equals(), not '!='. 12분 전 L173

Code Smell Critical Open 할당되지 않음 5min effort 코멘트 No tags

김 상 원
정 성 철
윤 성 일

Result – Findbugs

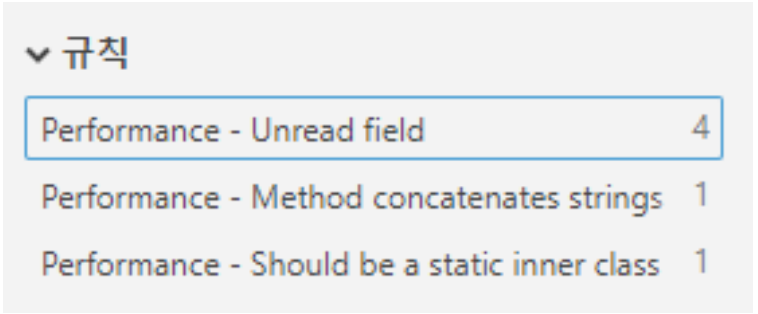


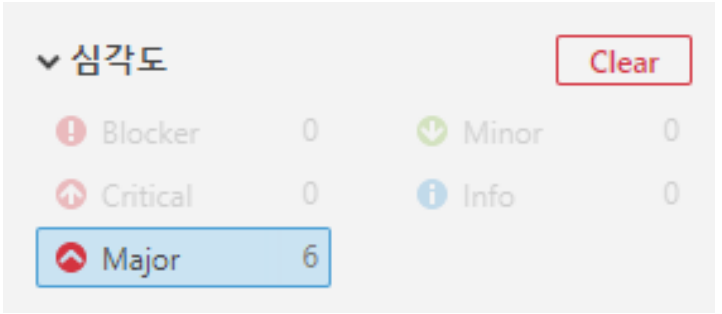
신뢰성 C등급 판정

총 6개의 버그가 발견됨.

김 상 원
정 성 철
윤 성 일

Result – Findbugs



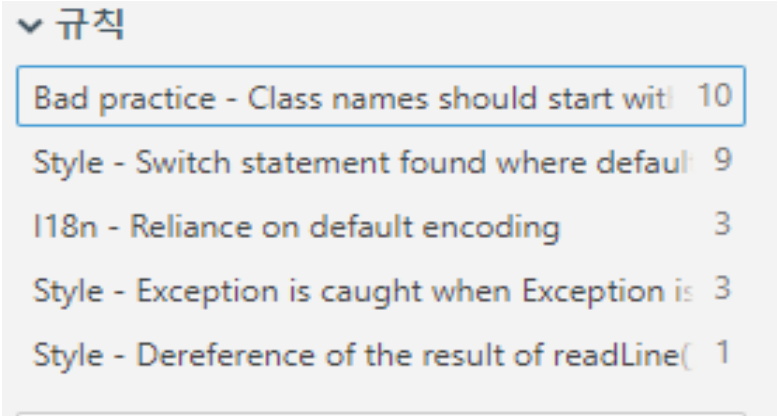


발견된 모든 이슈가 성능과 관련된 이슈임.

발견된 6개 이슈 모두 Major등급으로 판정되었음.

김 상 원
정 성 철
윤 성 일

Result – Findbugs



유지보수성 부분에서도 26개의 이슈 발생

10개의 Major이슈 발생

김 상 원
정 성 철
윤 성 일

Result – Findbugs

주요 이슈

Performance - Unread field

Bug Major performance 다음 기간 이후 적용됨 2018년 5월 4일 FindBugs (Java) 상수/이슈: 30min

This field is never read. Consider removing it from the class.

4개의 위반사항이 발견됨

절대 실행되지 않는 코드가 있다고 판단됨.

김 상 원
정 성 철
윤 성 일

Result – Findbugs

주요 이슈

Performance - Method concatenates strings using + in a loop

Bug Major performance 다음 기간 이후 적용됨 2018년 5월 4일 FindBugs (Java) 상수/이슈: 30min

The method seems to be building a String using concatenation in a loop. In each iteration, the String is converted to a StringBuffer/StringBuilder, appended to, and converted back to a String. This can lead to a cost quadratic in the number of iterations, as the growing string is recopied in each iteration.

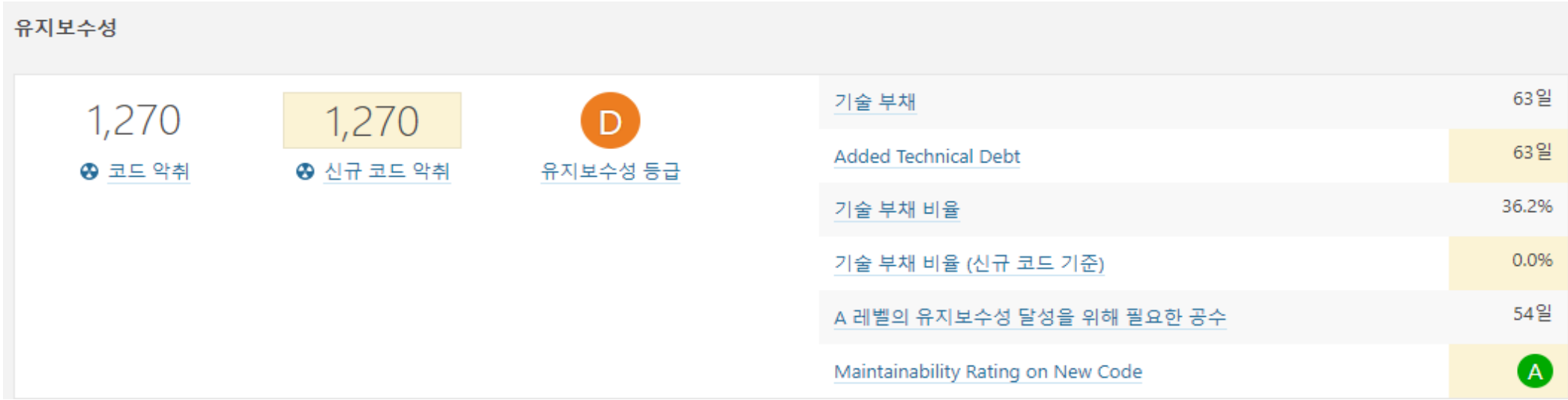
Better performance can be obtained by using a StringBuffer (or StringBuilder in Java 1.5) explicitly.

1개의 위반사항 발견

문자열을 이어 붙이는데 반복문 내에서 '+'를 사용하였음...

김 상 원
정 성 철
윤 성 일

Result – PMD

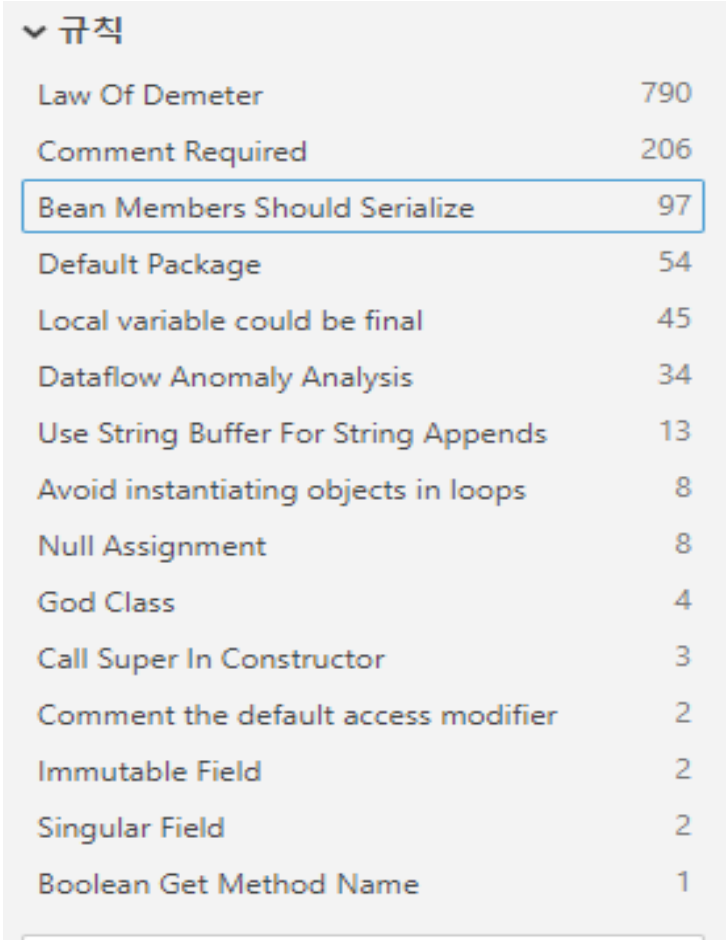


유지보수성 D등급 판정

총 1270개의 코드 악취가 발견됨.

김 상 원
정 성 철
윤 성 일

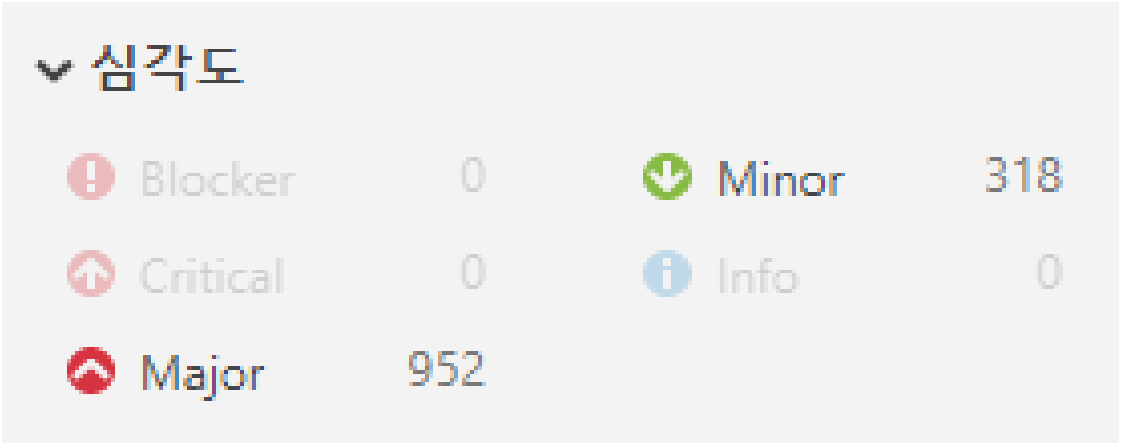
Result – PMD



Law Of Demeter (심각도 Major)가

790개로 가장 많이 발견

Major 이슈 952개 발견



김 상 원
정 성 철
윤 성 일

Result – PMD

주요 이슈

Law Of Demeter

pmd:LawOfDemeter

Code Smell Major 태그 없음 다음 기간 이후 적용됨 2018년 5월 4일 PMD (Java) 상수/이슈: 30min

The Law of Demeter is a simple rule, that says "only talk to friends". It helps to reduce coupling between classes or objects. See also the references: Andrew Hunt, David Thomas, and Ward Cunningham. The Pragmatic Programmer. From Journeyman to Master. Addison-Wesley Longman, Amsterdam, October 1999.; K.J. Lieberherr and I.M. Holland. Assuring good style for object-oriented programs. Software, IEEE, 6(5):38–48, 1989.; http://www.ccs.neu.edu/home/lieber/LoD.html; http://en.wikipedia.org/wiki/Law_of_Demeter

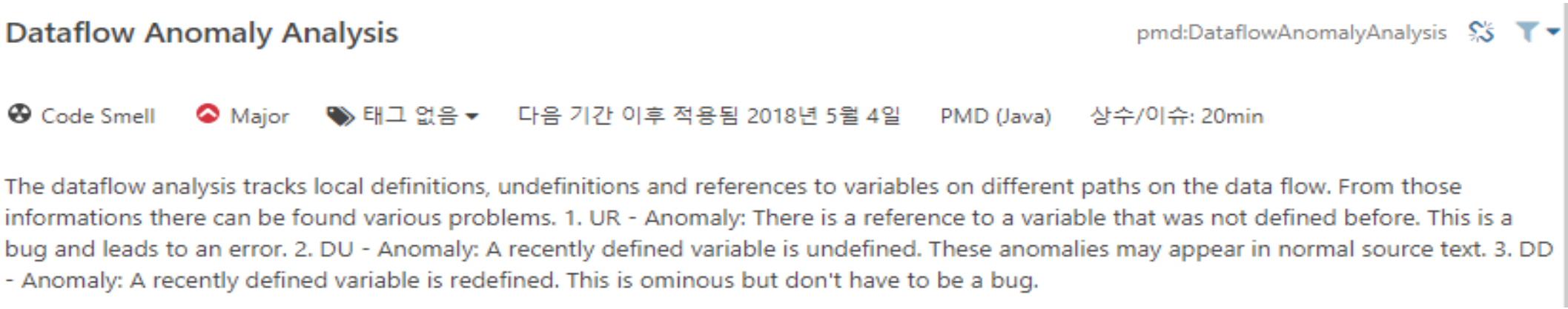
790개의 위반사항 발견

'Don't talk to Stranger' -> 'Low Coupling', 'High Cohesion'

김 상 원
정 성 철
윤 성 일

Result – PMD

주요 이슈



34개의 위반사항 발견

변수의 선언과 사용(def-use pair) 문제

변수의 중복된 선언, 선언 후 사용 X 등의 문제

김 상 원
정 성 철
윤 성 일

Unit test Coverage

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
DSLAB2018_T6-master	10.6 %	1,310	10,996	12,306
src/main/java	10.6 %	1,310	10,996	12,306
GUI	0.0 %	0	9,571	9,571
ATM	42.9 %	801	1,064	1,865
OFFER	32.6 %	163	337	500
Junit	93.5 %	346	24	370

전체 10.6%의 Unit test Coverage를 보임.

김 상 원
정 성 철
윤 성 일

Sonarqube 종합 결과

신뢰성

6 Bugs	6 신규 버그	C 신뢰성 등급	신뢰성 개선 필요 공수	3시간
			신뢰성 개선 필요 공수 (신규 코드)	3시간
			Reliability Rating on New Code	C

보안성

25 취약점	25 신규 취약점	E 보안성 등급	보안성 개선 필요 공수	1일 2시간
			보안성 개선 필요 공수 (신규 코드)	1일 2시간
			Security Rating on New Code	E

유지보수성

4,864 코드 악취	4,864 신규 코드 악취	E 유지보수성 등급	기술 부채	134일
			Added Technical Debt	134일
			기술 부채 비율	76.5%
			기술 부채 비율 (신규 코드 기준)	0.0%
			A 레벨의 유지보수성 달성을 위해 필요한 공수	126일
			Maintainability Rating on New Code	A

김 상 원
정 성 철
윤 성 일

Sonarqube 종합 결과

코드 중복

49.7% Duplicated lines (%)	Duplicated blocks	75
	Duplicated lines	1,540
	Duplicated files	10

복잡도

526 Complexity	Complexity /function	4.3
	Complexity /file	35.1
	Complexity /class	29.2
	Cognitive Complexity	709

이슈

4,895 Issues	4,895 New issues	Open issues	4,895
		Reopened issues	0
		Confirmed issues	0
		False positive issues	0
		미대응 이슈	0

김 상 원
정 성 철
윤 성 일

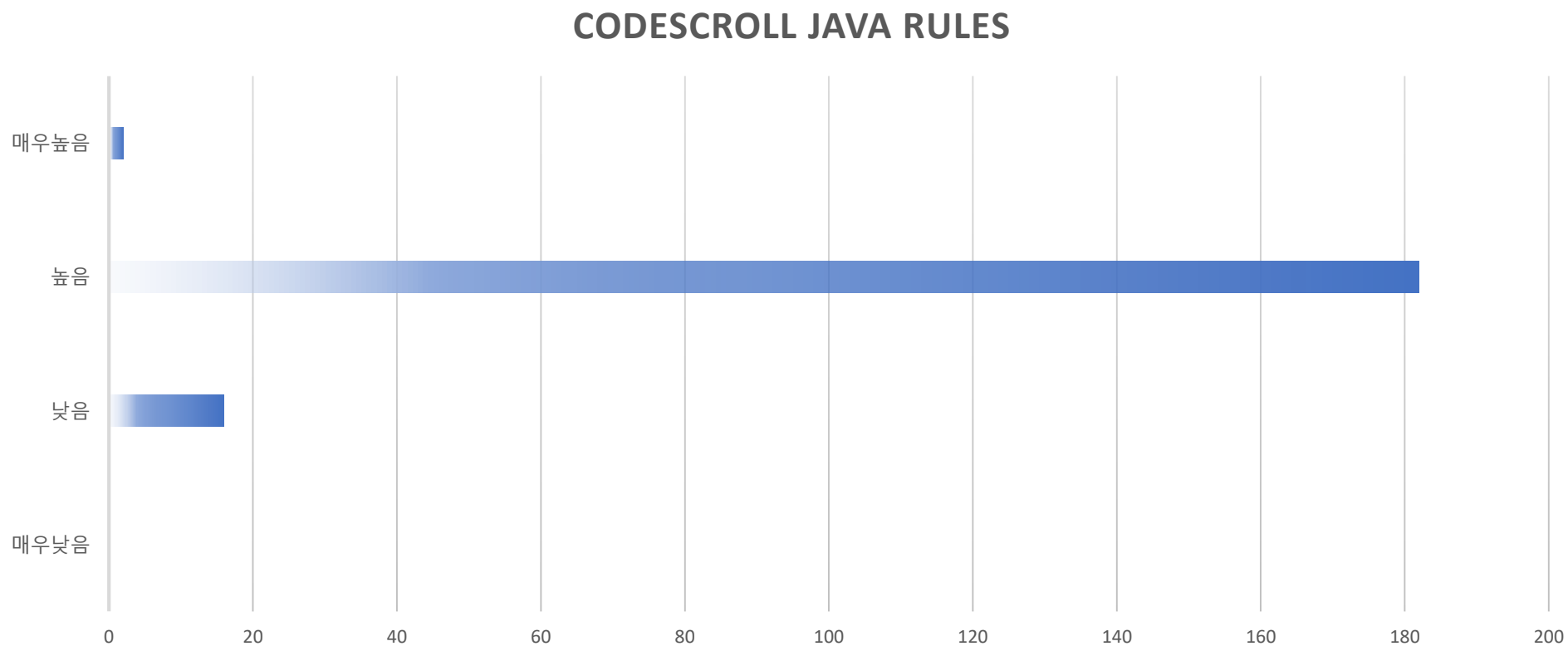
Codescroll Inspector for JAVA

진단 메시지	규칙	심각도	소속	개수
체인형 메서드 호출이 올바른 형태가 아님	VNA04-J	낮음	inputUSDFrame.actionPerformed(ActionEvent)	92
괄호를 사용하지 않고 여러 연산자를 사용하여 수식을혼용함.	Sun_26	낮음	insertFrame.actionPerformed(ActionEvent)	6
숫자 상수를 사용함(for loop의 counter로 -1이나 0, 1을 사용하는 경우 제외)	Sun_24	매우 낮음	selectCountry.keyTyped(KeyEvent)	626
non-static 필드가 public으로 선언.	Sun_22	매우 낮음	MainSystem	4
클래스의 이름(selectCountry)이 규칙(^[A-Z]([a-z]|[A-Z]|[0-9])*)에 맞지 않음.	Sun_18	매우 낮음	GUI	13
지역 변수 선언문과 실행 문장 사이에 빈 줄이 없음.	Sun_17	매우 낮음	selectCountry.selectCountry(MainSystem)	20
Switch 문에 default case가 존재하지 않음.	Sun_13	매우 낮음	insertFrame.keyTyped(KeyEvent)	11
if 문에서 중괄호가 사용되지 않음.	Sun_11	매우 낮음	MainSystem.transfer(int, Account)	16
지역 변수(error) 선언 시 초기화를 하지 않음.	Sun_10	낮음	insertFrame.actionPerformed(ActionEvent)	20
identifier(bank)가 이미 필드로 선언되어 있어 hide가 발생함	Sun_09	낮음	MainSystem.payUtilityBill(Account)	10
블록의 시작 부분이 아닌 곳에 선언함	Sun_08	매우 낮음	selectCountry.selectCountry(MainSystem)	32
한 줄에 여러 식별자를 선언함	Sun_07	매우 낮음	insertFrame	10
라인의 길이가 (80)을 넘음	Sun_06	매우 낮음	Offer.Offer(int)	161
해당 소스 파일이 C 스타일 주석으로 시작하지 않음	Sun_03	매우 낮음	GUI	15
mutable 클래스가 신뢰할 수 없는 코드에 객체를 안전하게 전달하기 위한 복사 기능을 제공	OBJ04-J	낮음	GUI	18
non-static, non-final인 필드가 private으로 선언되지 않음.	OBJ01-J	높음	selectCountry	58
생성자 내에서 final이나 private이 아닌 메서드를 호출함.	MET05-J	높음	inputPasswordFrame.inputPasswordFrame(int, MainSystem)	221
메서드의 설명 주석이 없거나, 설명 주석 내용에 모든 태그가 포함되어있지 않음	JAVA_72	높음	Offer	122
클래스의 설명 주석이 없거나, 설명 주석 내용에 모든 태그가 포함되어있지 않음	JAVA_71	높음	OFFER	15
identifier(bank)가 이미 필드로 선언되어 있어 hide가 발생함	JAVA_70	높음	MainSystem.payUtilityBill(Account)	10
*를 이용한 import문을 사용함.	JAVA_68	낮음	GUI	16
condition에 직접적인 assignment operator를 사용함.	JAVA_66	매우 높음	MainSystem.exchange(int, String)	1
반복문 안에서 String concatenation 연산자를 사용함.	JAVA_64	높음	Offer.readDatabase(Account)	13
반복문 안에서 synchronized 메서드(addKeyListener)를 호출함.	JAVA_62	높음	inputPasswordFrame.inputPasswordFrame(int, MainSystem)	8
Switch 문에 default case가 존재하지 않음.	JAVA_61	높음	inputPasswordFrame.selectmenu()	11
static, local, anonymous가 아닌 내부 클래스가 사용됨.	JAVA_49	높음	inputAccountFrame	3
String 비교에 "==" 또는 "!=" 을 사용함. equals 메서드를 사용하는것을 권장한다.	JAVA_07	매우 높음	MainSystem.transfer(int, Account)	1
(Exception)을 catch함.	ERR08-J	높음	Offer.checkValid(Account)	3
적절한 로그를 사용하지 않음.	ERR02-J	높음	Offer.checkValid(Account)	2

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

Codescroll_java_rules 위배 현황



김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈

```
public void selectmenu() {
    int error;
    switch (menu) {
    case 0: // withdraw
        if ((error = main.withdraw(main.bank)) == 0) {
            new ReceiptFrame(main);
        } else {
            new errorPrintFrame(error);
        }
        this.dispose();
        break;
    case 1: // exchange
        if ((error = main.exchange(main.bank, main.card)) == 0) {
            new ReceiptFrame(main);
            this.dispose();
        } else {
            new errorPrintFrame(error);
```

심각도 : 높음

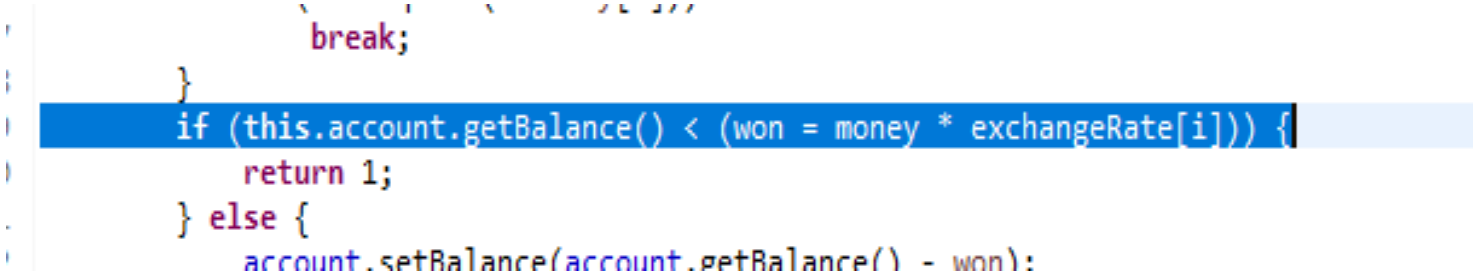
규칙 : JAVA_61

Switch문에 default가 없음.

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈



심각도 : 매우 높음

규칙 : JAVA_66

조건문에서 직접적인 assignment operator를 사용함.

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈

```
public int payUtilityBill(Ac
    String[] list_bank = { "
    int bank;
    for(bank = 0; bank < list
        if(list_bank[bank].ec
            break;
    }
```

심각도 : 높음

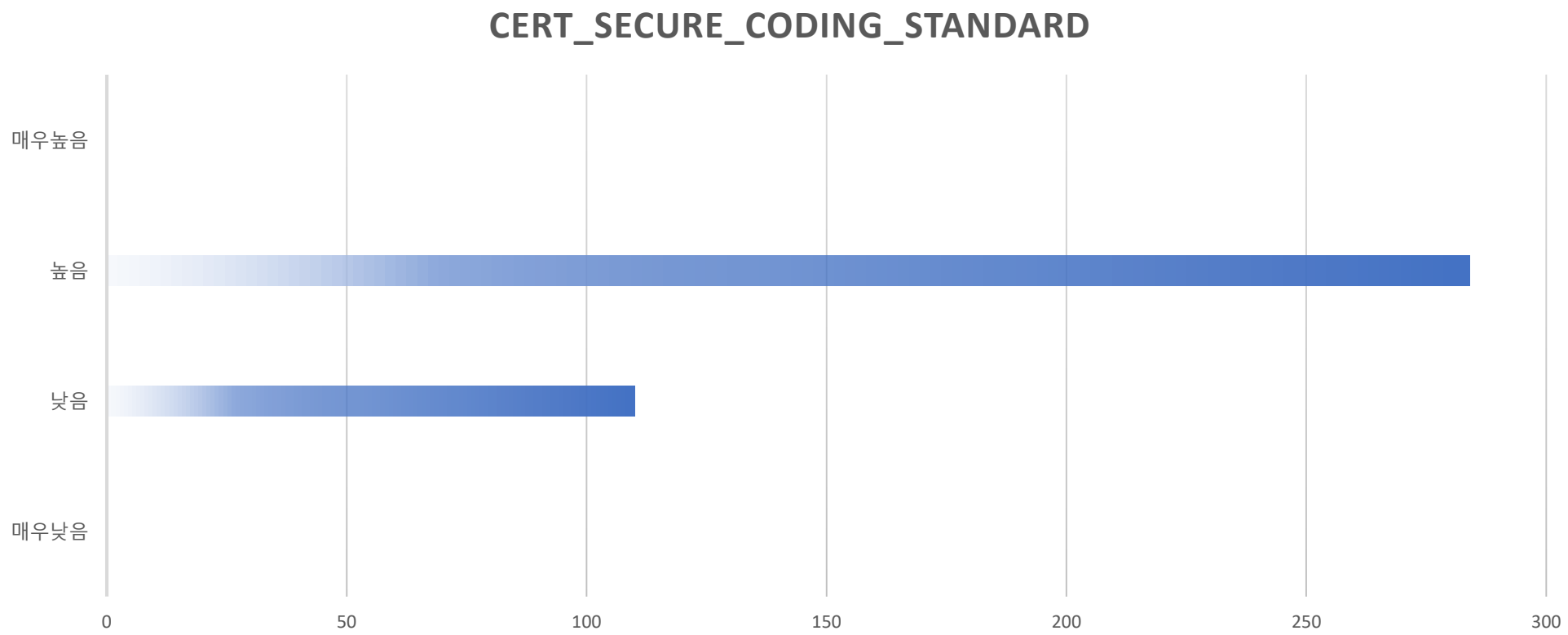
규칙 : JAVA_70

identifier(bank)가 이미 필드로 선언되어 있어 hide가 발생함

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

Cert_secure_coding_standard 위배 현황



김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈

```
public class selectCountry extends JFrame implements ActionListener, KeyListener {
    MainSystem main;
    JButton[] country;
    GridBagLayout gbl;
    GridBagConstraints gbc;
```

심각도 : 높음

규칙 : OBJ01-J

non-static, non-final인 필드가 private으로 선언되지 않음.

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈

```
        add(DD, 0, 0, 1, 1);
        add(reset, 2, 6, 1, 1);
        this.setSize(500, 500);
        this.setVisible(true);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
```

심각도 : 높음

규칙 : MET05-J

생성자 내에서 final이나 private이 아닌 메서드를 호출함.

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈

```
public boolean checkValid(Account account) {
    try {
        String direc = "./db❤❤/" + account.getBank() + "/" +account
        System.out.println(direc);
        this.input = new BufferedReader(new FileReader(direc));
        this.input.close();
        System.out.println("valid input");
    } catch(Exception e) {
        System.out.println("invalid input");
        return false;
    }
    return true;
```

심각도 : 높음

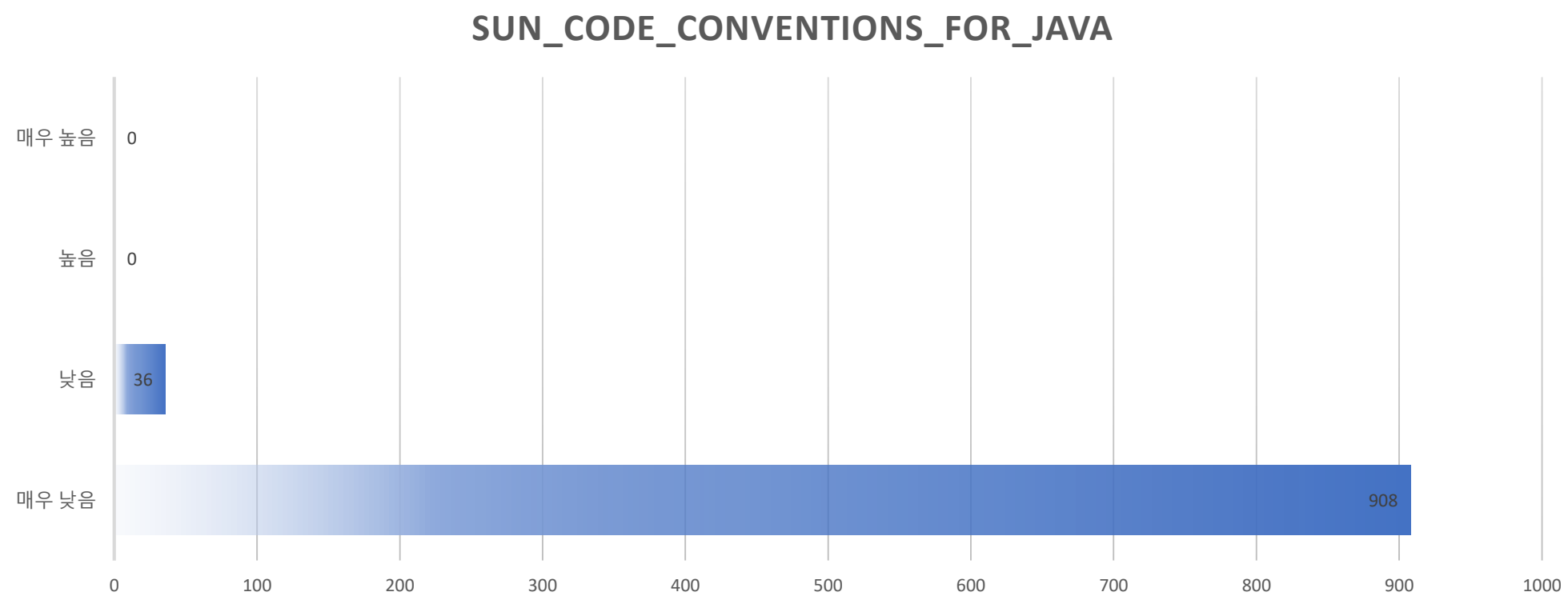
규칙 : ERR08-J

(Exception)을 catch함.

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

Sun_code_conventions_for_java위배 현황



김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈

```
                    break;
                case 4: // ****
                    if (money.getText().length() > 5 && (money.getText()
                            .substring(money.getText().length() - 5, money.getText().length() - 1).equals("0000"))) {
                        main.bank = Integer.parseInt(money.getText().substring(0, money.getText().length() - 1));
                        new inputPasswordFrame(2, main);
                        this.dispose();
                    } else {
```

심각도 : 매우 낮음

규칙 : Sun_06

라인의 길이가 80을 넘음

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈

```
public class insertFrame extends JFrame implements ActionListener, KeyListener {
    JTextField tf;
    JButton ok, b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, bb, reset;
    JLabel state;
    JComboBox<String> combo;
    MainSystem main, temp;
    int menu;
    GridBagLayout gbl;
    GridBagConstraints gbc;

    public insertFrame(int menu) {
        super("insert");
```

심각도 : 매우 낮음

규칙 : Sun_07

한 줄에 여러 식별자를 선언함.

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

주요 이슈

```
            } else {
                new errorPrintFrame(3);
                this.dispose();
            }
        } else if (e.getSource() == bb && tf.getText().length() > 0) {
            tf.setText(tf.getText().substring(0, tf.getText().length() - 1));
        } else if (e.getSource() == reset) {
            tf.setText("");
        }
    }
```

심각도 : 매우 낮음

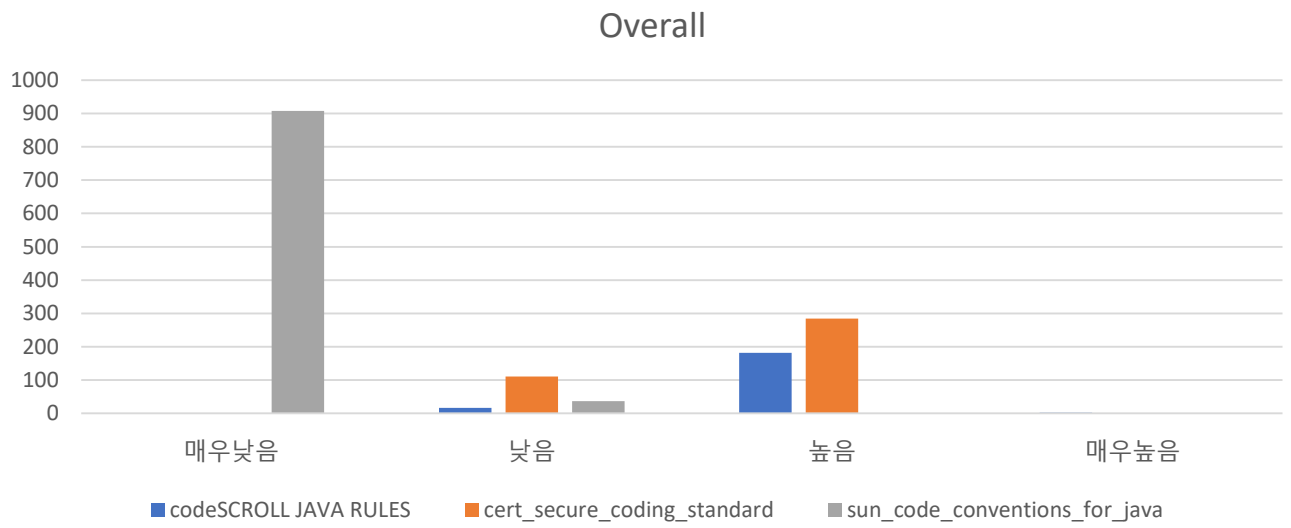
규칙 : Sun_26

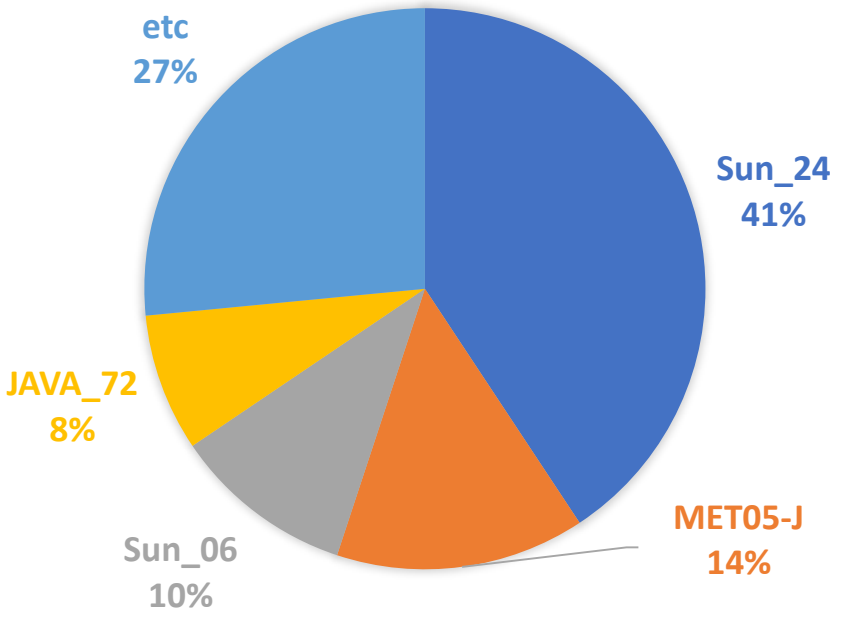
괄호를 사용하지 않고 여러 연산자를 사용하여 수식을 혼용함.

김 상 원
정 성 철
윤 성 일

Codescroll Inspector for JAVA

결과 종합



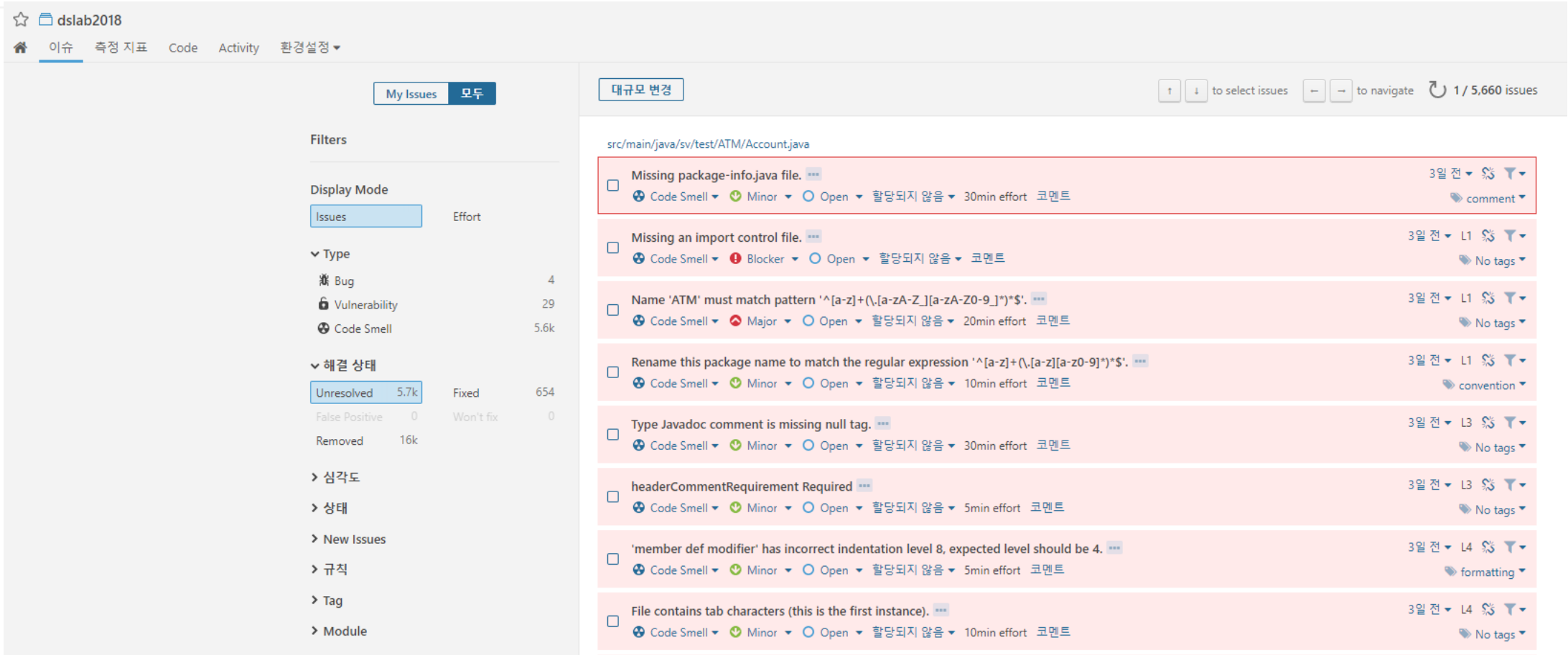


'sun_code_conventions_for_java'의 규칙 위배가 가장 많지만 심각도는 대부분 매우 낮음이나 낮음.

심각도가 높은 위배사항은 대부분 'codescroll_java_rules'와 'cert_secure_coding_standard'에서 나옴.

김 상 원
정 성 철
윤 성 일

이슈 확인



'Sonarqube'에 접속하여 상세 이슈 확인 가능 [http://52.231.190.169:9000/dashboard?id=dslab2018]

김 상 원
정 성 철
윤 성 일