SOFTWARE VERIFICATION

2nd Test - System Testing & Static Analysis
- T3 -

송지연
윤상혁
장서연

PART 1 Specification Review

PART 2 Brute Force Test

PART 3 Category Partition Test

PART 4 Pairwise Combination Test

PART 5 Static Analysis

PART 6 Conclusion

PART 1

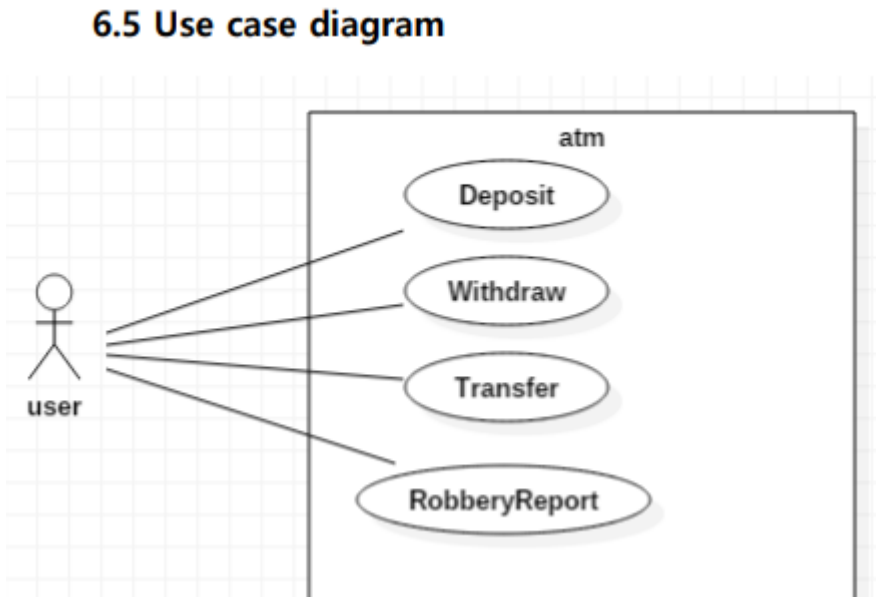
Specification Review

PART 1

Specification Review

Stage 1000

Function	Description
Deposit	User가 사용할 카드나 통장에서 요청한 금액만큼 돈을 받은 뒤, 입금을 해준다
Withdraw	User가 사용할 카드나 통장에서 요청한 금액만큼 돈을 출금해준다.
Transfer	User가 입력한 카드나 통장에서 목적 계좌에 돈을 이체한다.
Language	화면의 인터페이스를 쓰고 싶은 언어로 선택한다.
Check Balance	User가 원하는 카드나 통장의 잔고를 확인한다.
Add Cash	관리자가 요청한 금액만큼 ATM의 현금을 추가한다.
Check Cash	ATM의 현금 잔액을 확인한다.



기능 통일

사라진 기능이 남아있고, 있는 기능이 기술되지 않은 문제

PART 1

Specification Review

Stage 1004

4. Activity 1004. Record Terms in Glossary

Glossary	Description
Admin	관리자 모드

용어 정리

사용하는 용어를 모두 정리하지 않았다.

PART 1

Specification Review

Stage 2033

6. Activity 2034. Refine Glossary

Term	Category	Description
Admin	Class	관리자의 아이디, 패스워드, 현금을 관리하는 클래스
ATM	Class	전반적인 ATM시스템 기능들 구현
User_Data	Class	User의 정보들(이름, 은행, 계좌번호, 비밀번호, 잔액)을 관리하는 클래스
admin_cash	Attributes	관리자 모드의 현금 잔액
Admin_id	Attributes	관리자 모드의 관리자 아이디

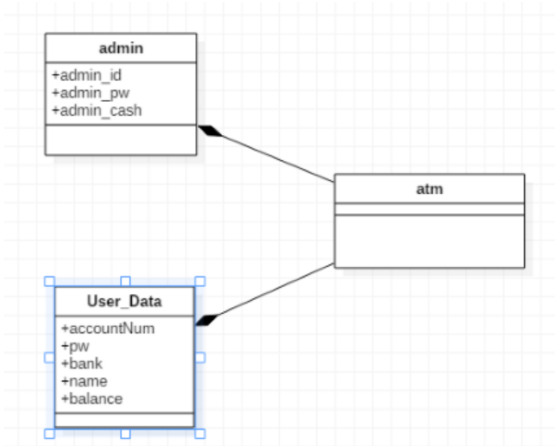
용어 정리

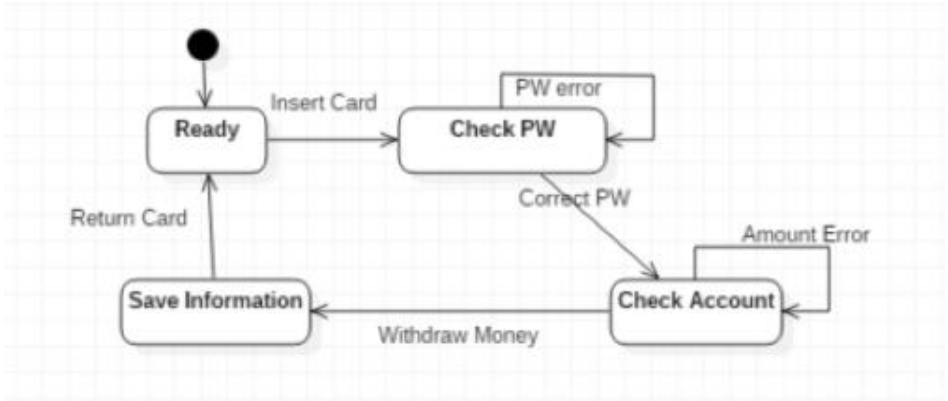
사용하는 용어를 모두 정리하지 않았다.

PART 1

Specification Review

Stage 2030-2037





다이어그램

실제 구현과 맞지 않는 다이어그램

PART 1

Specification Review

General Problems

GUI 미구현

GUI가 구현되지 않았다.

용어 정리

사용하는 용어에 대한 정확한 설명이 없다.
입력에 대한 정의가 생략되었다.

문서 불완전

생략된 기능, 구현되지 않았는데 삭제되지 않은 기능이 있다.
다이어그램에 대한 수정이 이루어지지 않았다.

PART 2

Brute Force Test

PART 2

Brute Force Test

1	메뉴 선택 화면에서 잘못된 입력(number)	Success
2	메뉴 선택 화면에서 잘못된 입력(malformed)	Success
3	Database의 계좌 보유액을 조정(1,000,000,000,000,000) 후, 900,000,000,000,000 출금	Failed
4	Database의 계좌 보유액을 조정(1,000,000,000,000,000) 후, 900,000,000,000,000 송금	Failed
5	자기 자신에게 올바른 금액 이체	Success
6	반복 입금으로 ATM 보유액이 long int 범위 이상	Success
7	프로그램 종료, 재시작 후 Database 보존 여부 확인	Success
8	관리자 모드에서 addcash 선택 후, long int 범위 이상의 금액 입금	Failed
9	Deposit 금액 입력 시 long int 범위 이상의 금액 입력	Failed
10	Withdraw 금액 입력 시 long int 범위 이상의 금액 입력	Failed
11	Transfer 금액 입력 시 long int 범위 이상의 금액 입력	Failed

Success 6/11
Failed 5/11

PART 2

Brute Force Test

CONCLUSION

- long int 범위 이상의 int 값이 들어올 경우 에러가 발생한다.
  ->예외처리가 필요하다.
- Specification 명세가 아직 부족하다.

PART 3

Category Partition Test

PART 3

Category Partition Test

Deposit	101	잘못된 계좌번호(number) 입력	Success
	102	잘못된 계좌번호(malformed) 입력	Success
	103	올바른 계좌번호 입력했으나, 잘못된 금액(< 0) 입력	Success
	104	올바른 계좌번호 입력했으나, 잘못된 금액(malformed) 입력	Success
	105	올바른 계좌번호 입력 후, 올바른 금액 입력	Success
Withdraw	201	잘못된 계좌번호(number) 입력	Success
	202	잘못된 계좌번호(malformed) 입력	Success
	203	올바른 계좌번호 입력 후, 잘못된 비밀번호(number) 입력	Success
	204	올바른 계좌번호 입력 후, 잘못된 비밀번호(malformed) 입력	Success
	205	올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(< 0) 입력	Success
	206	올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(> AccountAmount) 입력	Success
	207	올바른 계좌번호와 비밀번호 입력 후, 잘못된 금액(> ATMamount) 입력	Success
	208	올바른 계좌번호와 비밀번호 입력 후, 올바른 금액 입력	Success
Transfer	301	잘못된 출금 계좌번호(number) 입력	Success
	302	잘못된 출금 계좌번호(malformed) 입력	Success
	303	올바른 출금 계좌 입력 후, 잘못된 비밀번호(number) 입력	Success
	304	올바른 출금 계좌 입력 후, 잘못된 비밀번호(malformed) 입력	Success
	305	올바른 출금 계좌와 비밀번호 입력 후, 잘못된 입금 계좌(number) 입력	Success
	306	올바른 출금 계좌와 비밀번호 입력 후, 잘못된 입금 계좌(malformed) 입력	Success
	307	올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(< 0) 입력	Success
	308	올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(< 출금 계좌 보유액) 입력	Success
	309	올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 잘못된 금액(malformed) 입력	Success
	310	올바른 출금 계좌, 비밀번호, 입금 계좌 입력 후, 올바른 금액 입력	Success

PART 3

Category Partition Test

Check Balance	401	잘못된 계좌번호 입력	Success
	403	올바른 계좌번호 입력 후, 잘못된 비밀번호 입력	Success
	405	올바른 계좌번호 입력 후, 올바른 비밀번호 입력	Success
Robbery Report	501	잘못된 계좌번호 입력	Success
	503	올바른 계좌번호 입력 후, 잘못된 비밀번호 입력	Success
	505	올바른 계좌번호 입력 후, 올바른 비밀번호 입력	Success
AddCash	601	잘못된 관리자 아이디 입력	Success
	602	올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(number) 입력	Success
	603	올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(malformed) 입력	Success
	604	올바른 관리자 아이디 비밀번호를 입력 했으나, 잘못된 금액(< 0) 입력	Success
	605	올바른 관리자 아이디 비밀번호를 입력 했으나, 잘못된 금액(malformed) 입력	Success
	606	올바른 관리자 아이디, 비밀번호, 금액 입력	Success
Check Cash	701	잘못된 관리자 아이디 입력	Success
	702	올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(number) 입력	Success
	703	올바른 관리자 아이디를 입력 했으나, 잘못된 관리자 비밀번호(malformed) 입력	Success
	704	올바른 관리자 아이디, 비밀번호 입력	Success

Success **39/39**

Failed **0/39**

PART 3

Category Partition Test

CONCLUSION

- 이전 발표에 대한 피드백이 잘 수행되어 예외처리가 대체적으로 잘 구현되었다.

PART 4

Pairwise Combination Test

PART 4

Pairwise Combination Test

105-704	입금 후 ATM 잔액 확인	Success
105-405	입금 후 계좌 잔액 확인	Success
208-704	출금 후 ATM 잔액 확인	Success
208-405	출금 후 계좌 잔액 확인	Success
310-405	송금 후 출금 계좌 잔액 확인	Success
310-405	송금 후 입금 계좌 잔액 확인	Success
505-505	도난 신고 후 도난신고	Success
505-208	도난 신고 후 출금	Success
505-105	도난 신고 후 입금	Success
505-310	도난 신고 후 송금	Success
505-405	도난 신고 후 계좌 잔액 확인	Success

Success
11/11
Failed
0/11

PART 4

Pairwise Combination Test

CONCLUSION

- 도난신고 후 계좌를 삭제하는 기능인 것으로 확인

PART 5

Static Analysis

PART 5-1

Code Inspector

PART 5-1

Code Inspector

JAVA_66 - 반복문 안에서 string concatenation assignment 연산자의 사용 금지

```
452         for(i =0; i< len; i++)
453         {
454
455             if((l[i].substring(0, 6)).equals(account))
456             {
457                 l[i] = d.getAccountNum() +" " + d.getPw()+" " + d.getBank()+" " + d.getName()+" " + d.getBalance();
458
459             }
460         }
```

PART 5-1

Code Inspector

JAVA_64 - 반복문 안에서 string concatenation assignment 연산자의 사용 금지

```
417         for(i = 0; i < len; i++)
418         {
419             if(i == len-1)
420             {
421                 line += l[i];
422             }
423             else
424             {
425                 line += l[i];
426                 line += "\r\n";
427             }
428         }
```

PART 5-1

Code Inspector

JAVA_66 - 조건문에 선언 연산자 사용 금지

```
481     public static boolean calculate(User_Data d, int money)
482     {
483         int balance = Integer.parseInt(d.getBalance());
484 
485         if((balance += money) < 0)
486         {
487             return false;
488         }
489         else
490         {
491             d.setBalance(Integer.toString(balance));
492             return true;
493         }
494 
495     }
```

PART 5-1

Code Inspector

Sun_10 - 지역변수 선언 시 초기화 검사

```
String line;
String account;
```

PART 5-1

Code Inspector

Sun_11 - if문에서 중괄호 사용 검사

```
if(amount <= 1000000 && amount > 0)
    return true;
```

```
if(d == null)
    System.out.println("cannot find the account");
```

PART 5-1

Code Inspector

Sun_26 - 괄호 안의 수식에 연산자 혼용 금지

```
if(amount.charAt(i) >= 48 && amount.charAt(i) <= 57)
```

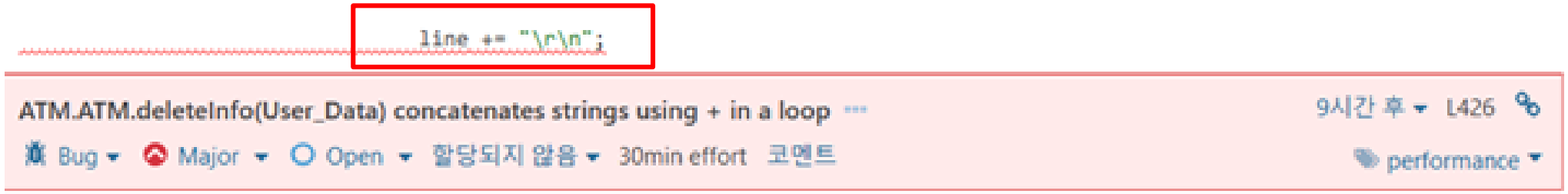
```
if(amount <= 1000000 && amount > 0)
```

PART 5-2

Find Bugs

PART 5-2

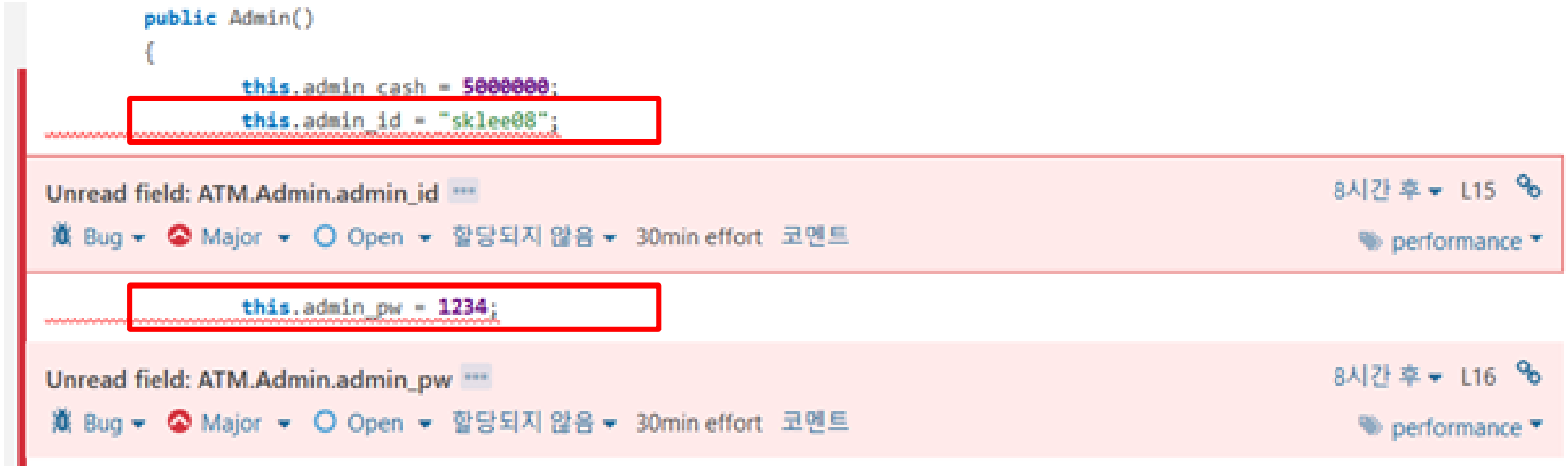
Find Bugs



String concatenating을 할 경우 + 보다는 append를 추천.

PART 5-2

Find Bugs



Admin_id, admin_pw를 선언하고 사용하지 않고 constant string 비교를 사용함.

PART 5-2

Find Bugs

```
public boolean CheckAdminID(String id)
{
        if(id.equals("sklee08"))
        {
                return true;
        }
        else
                return false;
}

public boolean CheckAdminPW(String pw)
{
        if(pw.equals("1234"))
        {
                return true;
        }
        else
                return false;
}
```

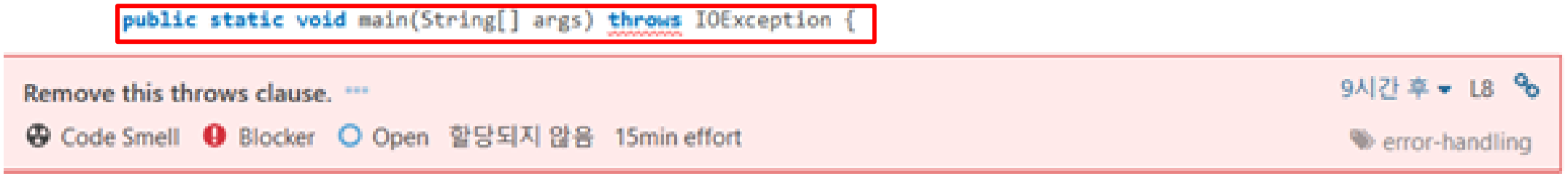
Admin_id, admin_pw를 선언하고 사용하지 않고 constant string 비교를 사용함.

PART 5-3

Check Style

PART 6-3

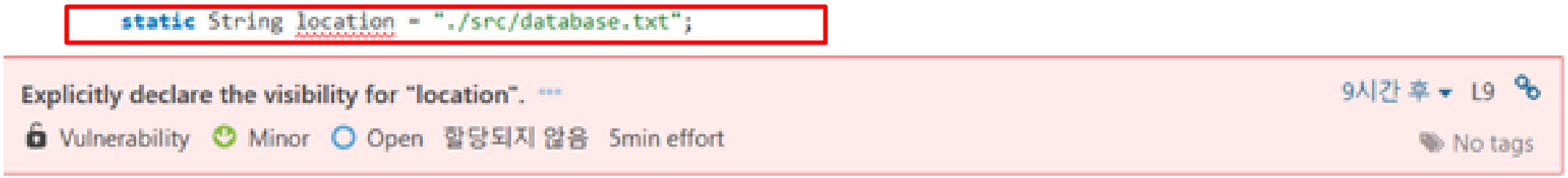
Check Style



무엇이든 throw할 수 있는 것보단 해당 케이스 마다 try - catch 를 사용할 것.

PART 6-3

Check Style



누구나 볼 수 있는 멤버변수 보다는 private을 사용할 것.

PART 6-3

Check Style

Refactor this code to not nest more than 3 if/for/while/switch/try statements.

Code Smell Critical +3

1 Nesting + 1

2 Nesting + 1

3 Nesting + 1

```
324             if(admin.CheckAdminID(id))
325             {
326                 String pw = getAdminPw();
327                 if(admin.CheckAdminPW(pw))
328                 {
329                     int index = getAdminMenu();
330                     if(index == 1)
```

중첩된 if문을 줄일것

PART 6-3

Check Style

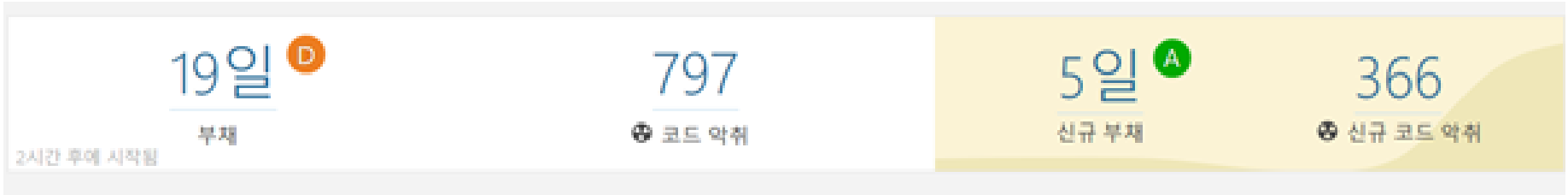
너무 복잡한 조건문

PART 5-4

PMD

PART 5-4

PMD



PMD

An open curly brace should be located at	114
Standard outputs should not be used direc	50
Close curly brace and the next "else", "catc	32
An open curly brace should be located at t	20
Magic numbers should not be used	19
Classes and methods that rely on the defau	13
Local variables should not be declared and	12
Strings literals should be placed on the left	10
Source code should be indented consistentl	7
Comments should not be located at the end	7
Close curly brace and the next "else", "catch	6
Control structures should use curly braces	6
Strings should not be concatenated using '+	6
Source files should have a sufficient density	5
Lines should have sufficient coverage by test	5

이전 도구의 분석 결과와 중복

PART 6

Conclusion

PART 6

Conclusion

예외 처리

구현 완성

Specification 재작성

END