

Static Analysis Report

T3

201414184 송지연

201611275 윤상혁

201511290 장서연

2018.06.04

Contents

1. [Code Scroll3](#)
 - 1.1 [JAVA_644](#)
 - 1.2 [JAVA_664](#)
 - 1.3 [SUN_10](#)
 - 1.4 [SUN_11](#)
 - 1.5 [SUN_26](#)
2. [Find Bug](#)
3. [Check Style](#)
4. [PMD](#)

1. Code scroll

1.1 JAVA_64 – 반복문 안에서 string concatenation assignment 연산자의 사용 금지

반복문 안에서 String concatenation 연산자를 사용함.	JAVA_64	높음	ATM.java	421	ATM.deleteInfo(User_Data)	ATM
반복문 안에서 String concatenation 연산자를 사용함.	JAVA_64	높음	ATM.java	425	ATM.deleteInfo(User_Data)	ATM
반복문 안에서 String concatenation 연산자를 사용함.	JAVA_64	높음	ATM.java	426	ATM.deleteInfo(User_Data)	ATM
반복문 안에서 String concatenation 연산자를 사용함.	JAVA_64	높음	ATM.java	457	ATM.saveInfo(User_Data)	ATM
반복문 안에서 String concatenation 연산자를 사용함.	JAVA_64	높음	ATM.java	466	ATM.saveInfo(User_Data)	ATM
반복문 안에서 String concatenation 연산자를 사용함.	JAVA_64	높음	ATM.java	470	ATM.saveInfo(User_Data)	ATM
반복문 안에서 String concatenation 연산자를 사용함.	JAVA_64	높음	ATM.java	471	ATM.saveInfo(User_Data)	ATM

```
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
461
462     for(i = 0; i < len; i++)
463     {
464         if(i == len-1)
465         {
466             line += l[i];
467         }
468         else
469         {
470             line += l[i];
471             line += "\r\n";
472         }
473     }
```

1.2 JAVA_66 – 조건문에 선언 연산자 사용 금지

condition 에 직접적인 assignment operator 를 사용함.	JAVA_66	매우 높음	ATM.java	485	ATM.calculate(User_Data, int)	ATM
---	---------	-------	----------	-----	-------------------------------	-----

```

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 }

```

1.3 Sun_10 – 지역변수 선언 시 초기화 검사

지역 변수(line) 선언 시 초기화를 하지 않음.	Sun_10	낮음	ATM.java	394	ATM.deletelInfo(User_Data)	ATM
지역 변수(account) 선언 시 초기화를 하지 않음.	Sun_10	낮음	ATM.java	395	ATM.deletelInfo(User_Data)	ATM
지역 변수(line) 선언 시 초기화를 하지 않음.	Sun_10	낮음	ATM.java	439	ATM.savelInfo(User_Data)	ATM
지역 변수(account) 선언 시 초기화를 하지 않음.	Sun_10	낮음	ATM.java	440	ATM.savelInfo(User_Data)	ATM
지역 변수(line) 선언 시 초기화를 하지 않음.	Sun_10	낮음	ATM.java	502	ATM.find(String)	ATM

```

String line;
String account;

```

1.4 Sun_11 - if 문에서 중괄호 사용 검사

if 문에서 중괄호가 사용되지 않음.	Sun_11	매우 낮음	ATM.java	66	ATM.checkAmountError(int)	ATM
if 문에서 중괄호가 사용되지 않음.	Sun_11	매우 낮음	ATM.java	301	ATM.checkBalance()	ATM
if 문에서 중괄호가 사용되지 않음.	Sun_11	매우 낮음	ATM.java	371	ATM.robberyreport()	ATM

```

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

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

```

1.5 Sun_26 – 괄호 안의 수식에 연산자 혼용 금지

괄호를 사용하지 않고 여러 연산자를 사용하여 수식을 혼용함.	Sun_26	낮음	ATM.java	44	ATM.getAmount()	ATM
괄호를 사용하지 않고 여러 연산자를 사용하여 수식을 혼용함.	Sun_26	낮음	ATM.java	66	ATM.checkAmountError(int)	ATM

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

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

2. Find Bug

ATM.ATM.deleteInfo(User_Data) concatenates strings using + in a loop 9시간 후 L426

🌊 Bug 🔴 Major 🔵 Open 👉 할당되지 않음 ⌚ 30min effort 🗨️ 코멘트 🔍 performance

ATM.ATM.saveInfo(User_Data) concatenates strings using + in a loop 9시간 후 L471

🌊 Bug 🔴 Major 🔵 Open 👉 할당되지 않음 ⌚ 30min effort 🗨️ 코멘트 🔍 performance

src/main/java/ATM/Admin.java

Unread field: ATM.Admin.admin_id 9시간 후 L15

🌊 Bug 🔴 Major 🔵 Open 👉 할당되지 않음 ⌚ 30min effort 🗨️ 코멘트 🔍 performance

Unread field: ATM.Admin.admin_pw 9시간 후 L16

🌊 Bug 🔴 Major 🔵 Open 👉 할당되지 않음 ⌚ 30min effort 🗨️ 코멘트 🔍 performance

```

.....
line += "\r\n";
.....

```

ATM.ATM.deleteInfo(User_Data) concatenates strings using + in a loop 9시간 후 L426

🌊 Bug 🔴 Major 🔵 Open 👉 할당되지 않음 ⌚ 30min effort 🗨️ 코멘트 🔍 performance

String concatenating 을 할 경우 + 보다는 append 를 추천.

```

public Admin()
{
    this.admin_cash = 5000000;
    this.admin_id = "sklee08";
}

```

Unread field: ATM.Admin.admin_id ... 8시간 후 L15

Bug Major Open 할당되지 않음 30min effort 코멘트 performance

```

    this.admin_pw = 1234;
}

```

Unread field: ATM.Admin.admin_pw ... 8시간 후 L16

Bug Major Open 할당되지 않음 30min effort 코멘트 performance

Admin_id, admin_pw 를 선언한 후 사용하지 않고 constant string 비교를 사용함.

```

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;
}

```

3. Check Style



```

public static void main(String[] args) throws IOException {
}

```

Remove this throws clause. ... 9시간 후 L8

Code Smell Blocker Open 할당되지 않음 15min effort error-handling

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

```
static String location = "./src/database.txt";
```

Explicitly declare the visibility for "location". ...

9시간 후 L9

Vulnerability Minor Open 할당되지 않음 5min effort

No tags

누구나 볼 수 있는 멤버변수 보다는 private 을 사용할 것.

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

```
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201

    1 if(d == null)
    {
        System.out.println("cannot find the account");
    }
    else
    {
        String pw = getPw();
        2 if(d.checkAccountPw(account, pw))
        {
            int amount = getAmount();
            3 if(!(checkAmountError(amount)))
            {
                System.out.println("Ãß,øµÈ ±Ý%× Æ·Ã!");
            }
        }
    }

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343

    1 if(admin.CheckAdminID(id))
    {
        String pw = getAdminPw();
        2 if(admin.CheckAdminPw(pw))
        {
            int index = getAdminMenu();
            3 if(index == 1)
            1 if(from == null)
            {
                System.out.println("cannot find the account!");
            }
            else
            {
                String pw = getPw();
                2 if(from.checkAccountPw(account, pw))
                {
                    String to_account = getTransferAccount();
                    3 if(to_account.equals(account))
```

중첩된 if 문을 줄일 것

Refactor this method to reduce its Cognitive Complexity from 35 to the 15 allowed.

Code Smell Critical +14

- 1 +1
- 2 +1
- 3 +2 (incl 1 for nesting)
- 4 +3 (incl 2 for nesting)
- 5 +1
- 6 +4 (incl 3 for nesting)
- 7 +1
- 8 +5 (incl 4 for nesting)
- 9 +1
- 10 +6 (incl 5 for nesting)
- 11 +1
- 12 +7 (incl 6 for nesting)
- 13 +1
- 14 +1

너무 복잡한 함수

4. PMD

19일 D 부채 2시간 후에 시작됨	797 코드 약취	5일 A 신규 부채	366 신규 코드 약취
--	--------------	----------------------------	-----------------

중요도가 높은 알림의 경우 상당수가 CheckStyle 과 겹침.