“Priority” Category Partitioning Testing Tool

OOPT stage 2nd Cycle

Static Analysis

Project Team

T4

201311265 김상원

201210194 김정환

201311269 김제헌

201311297 이상명

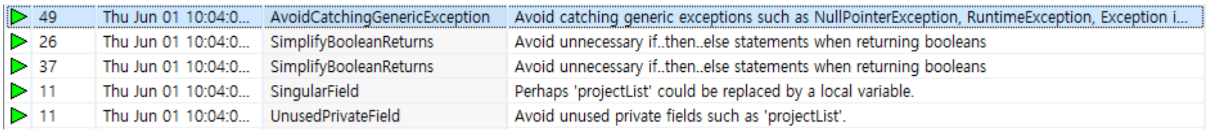
Date

2017-06-01

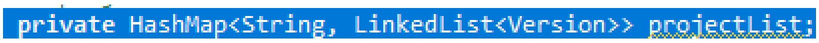
1. T2 코드 분석

- PMD Warnings

AdminProject.java

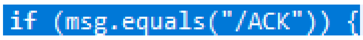
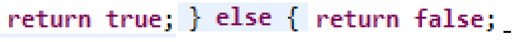


▶	49	Thu Jun 01 10:04:0...	AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPointerException, RuntimeException, Exception i...
▶	26	Thu Jun 01 10:04:0...	SimplifyBooleanReturns	Avoid unnecessary if..then..else statements when returning booleans
▶	37	Thu Jun 01 10:04:0...	SimplifyBooleanReturns	Avoid unnecessary if..then..else statements when returning booleans
▶	11	Thu Jun 01 10:04:0...	SingularField	Perhaps 'projectList' could be replaced by a local variable.
▶	11	Thu Jun 01 10:04:0...	UnusedPrivateField	Avoid unused private fields such as 'projectList'.

- line11 

```
private HashMap<String, LinkedList<Version>> projectList;
```

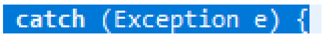
: SingularField, UnusedPrivateField - 전체 파일 내에서 생성 자에서의 초기화 외에 사용이 없음.

- line26, line37  

```
if (msg.equals("/ACK")) {
```

```
return true; } else { return false;
```

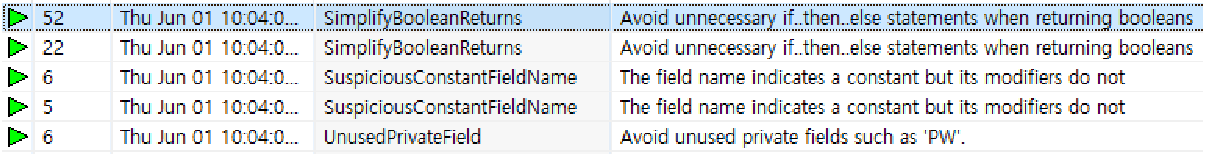
: SimplifyBooleanReturns – if문내에서 현재의 else부분을 잡고 else의 사용을 줄일 수 있으나 필요성은 잘 모르겠음.

- line 49 

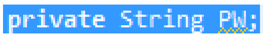
```
catch (Exception e) {
```

: AvoidCatchingGenericException – Exception을 특정해서 잡을 수 있다.

AdminUser.java

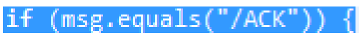
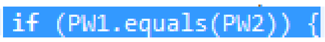


▶	52	Thu Jun 01 10:04:0...	SimplifyBooleanReturns	Avoid unnecessary if..then..else statements when returning booleans
▶	22	Thu Jun 01 10:04:0...	SimplifyBooleanReturns	Avoid unnecessary if..then..else statements when returning booleans
▶	6	Thu Jun 01 10:04:0...	SuspiciousConstantFieldName	The field name indicates a constant but its modifiers do not
▶	5	Thu Jun 01 10:04:0...	SuspiciousConstantFieldName	The field name indicates a constant but its modifiers do not
▶	6	Thu Jun 01 10:04:0...	UnusedPrivateField	Avoid unused private fields such as 'PW'.

- line6 

```
private String PW;
```

: UnusedPrivateField, SuspiciousConstantFieldName – 메소드 상에서 입력 받는 PW와 this.PW의 사용이 의미 없게 되어있다. This.PW를 사용하려면 client메소드에 입력값을 넣을 때 PW가 아니라 this.PW를 사용하면 된다.

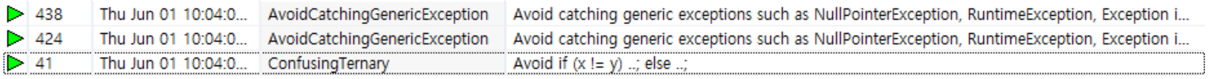
- line22, line52  

```
if (msg.equals("/ACK")) {
```

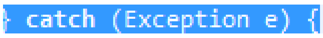
```
if (PW1.equals(PW2)) {
```

: SimplifyBooleanReturns - if문내에서 현재의 else부분을 잡고 else의 사용을 줄일 수 있으나 필요성은 잘 모르겠음.

AdminVer.java



▶	438	Thu Jun 01 10:04:0...	AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPointerException, RuntimeException, Exception i...
▶	424	Thu Jun 01 10:04:0...	AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPointerException, RuntimeException, Exception i...
▶	41	Thu Jun 01 10:04:0...	ConfusingTernary	Avoid if (x != y) ..; else ..;

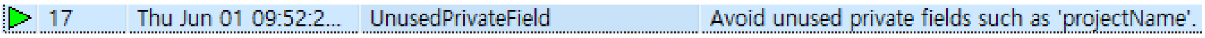
- line424, line438 

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

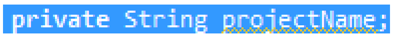
: Exception을 특정해서 잡을 수 있다.

Server

AdminProject.java



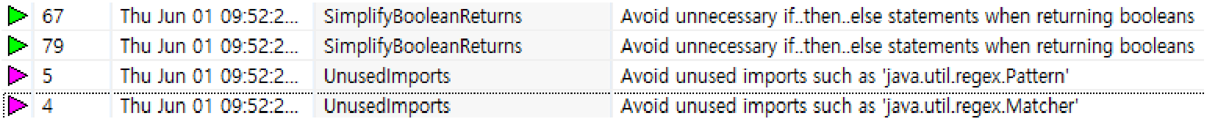
▶	17	Thu Jun 01 09:52:2...	UnusedPrivateField	Avoid unused private fields such as 'projectName'.

- line17 

```
private String projectName;
```

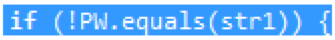
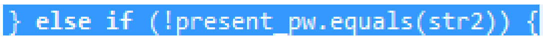
: 전체 파일 내에서 최초 선언 외에 사용이 없음.

AdminUser.java



▶	67	Thu Jun 01 09:52:2...	SimplifyBooleanReturns	Avoid unnecessary if..then..else statements when returning booleans
▶	79	Thu Jun 01 09:52:2...	SimplifyBooleanReturns	Avoid unnecessary if..then..else statements when returning booleans
▶	5	Thu Jun 01 09:52:2...	UnusedImports	Avoid unused imports such as 'java.util.regex.Pattern'
▶	4	Thu Jun 01 09:52:2...	UnusedImports	Avoid unused imports such as 'java.util.regex.Matcher'

- line4, line5 : 사용되지 않는 패키지들이 import되어 있다.

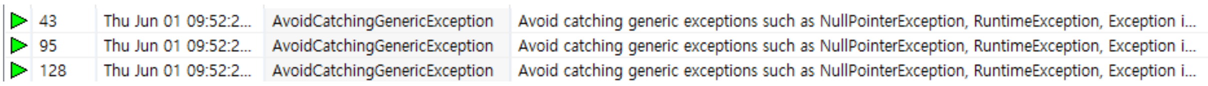
- line79, line67  

```
if (!PW.equals(str1)) {
```

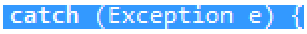
```
} else if (!present_pw.equals(str2)) {
```

: if문내에서 현재의 else부분을 잡고 else의 사용을 줄일 수 있으나 필요성은 잘 모르겠음.

MainServer.java



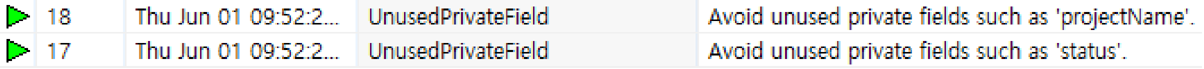
▶	43	Thu Jun 01 09:52:2...	AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPointerException, RuntimeException, Exception i...
▶	95	Thu Jun 01 09:52:2...	AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPointerException, RuntimeException, Exception i...
▶	128	Thu Jun 01 09:52:2...	AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPointerException, RuntimeException, Exception i...

- line43, line95, line128 

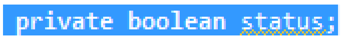
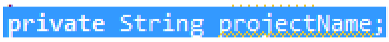
```
catch (Exception e) {
```

: Exception을 특정해서 잡을 수 있다.

ServerThread.java



18	Thu Jun 01 09:52:2...	UnusedPrivateField	Avoid unused private fields such as 'projectName'.
17	Thu Jun 01 09:52:2...	UnusedPrivateField	Avoid unused private fields such as 'status'.

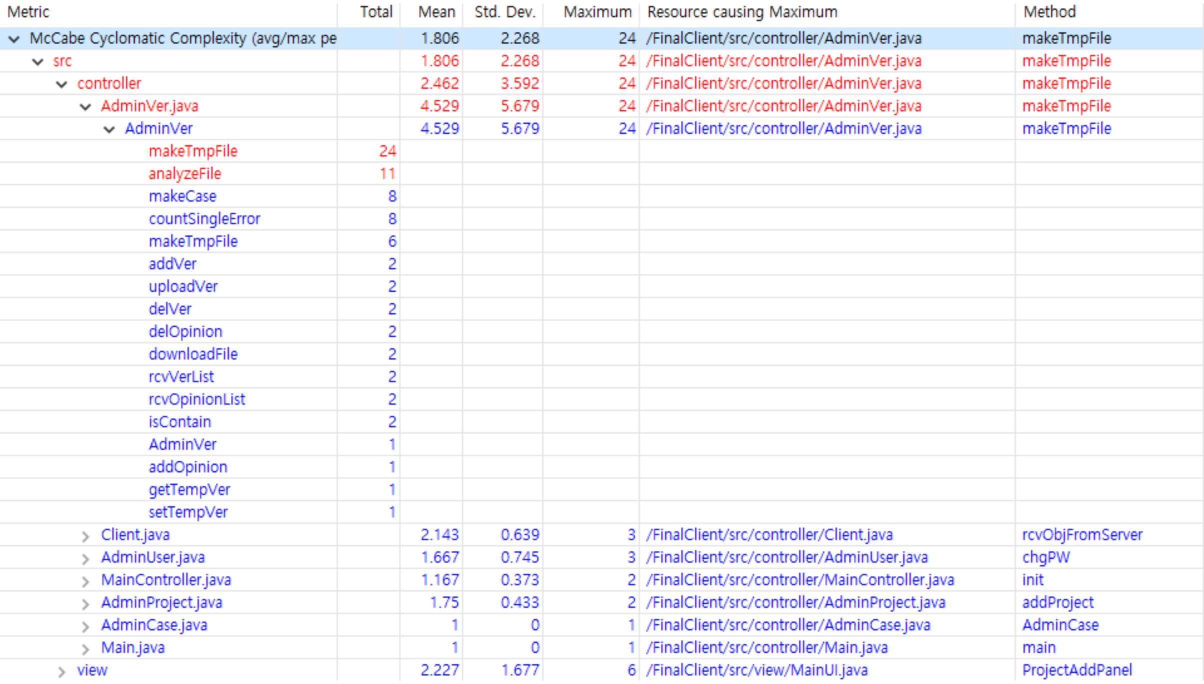
- line17, line18  `private boolean status;`  `private String projectName;`

: 선언 된 후 전혀 사용되지 않습니다.

- Metrics 분석

FinalClient

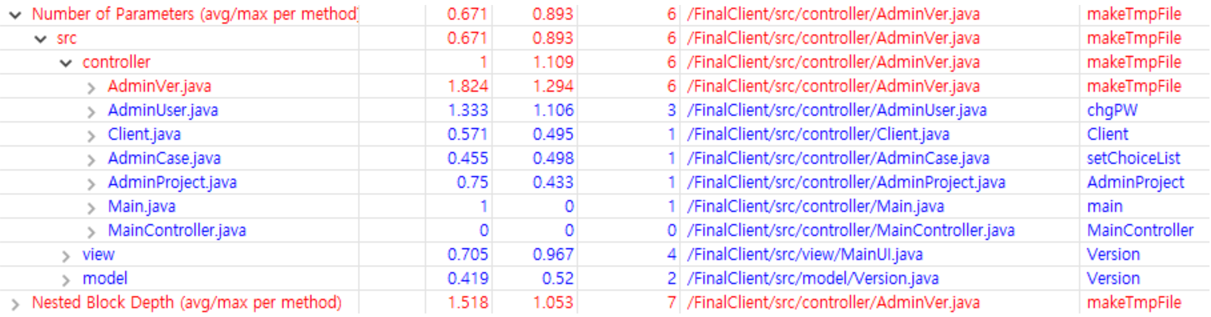
1. Cyclomatic complexity



Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max pe		1.806	2.268	24	/FinalClient/src/controller/AdminVer.java	makeTmpFile
src		1.806	2.268	24	/FinalClient/src/controller/AdminVer.java	makeTmpFile
controller		2.462	3.592	24	/FinalClient/src/controller/AdminVer.java	makeTmpFile
AdminVer.java		4.529	5.679	24	/FinalClient/src/controller/AdminVer.java	makeTmpFile
AdminVer		4.529	5.679	24	/FinalClient/src/controller/AdminVer.java	makeTmpFile
makeTmpFile	24					
analyzeFile	11					
makeCase	8					
countSingleError	8					
makeTmpFile	6					
addVer	2					
uploadVer	2					
delVer	2					
delOpinion	2					
downloadFile	2					
rcvVerList	2					
rcvOpinionList	2					
isContain	2					
AdminVer	1					
addOpinion	1					
getTempVer	1					
setTempVer	1					
Client.java		2.143	0.639	3	/FinalClient/src/controller/Client.java	rcvObjFromServer
AdminUser.java		1.667	0.745	3	/FinalClient/src/controller/AdminUser.java	chgPW
MainController.java		1.167	0.373	2	/FinalClient/src/controller/MainController.java	init
AdminProject.java		1.75	0.433	2	/FinalClient/src/controller/AdminProject.java	addProject
AdminCase.java		1	0	1	/FinalClient/src/controller/AdminCase.java	AdminCase
Main.java		1	0	1	/FinalClient/src/controller/Main.java	main
view		2.227	1.677	6	/FinalClient/src/view/MainUI.java	ProjectAddPanel

makeTmpFile, analyzeFile에서 각각 Maximum점수 11, 24의 높은 점수를 기록하였다.

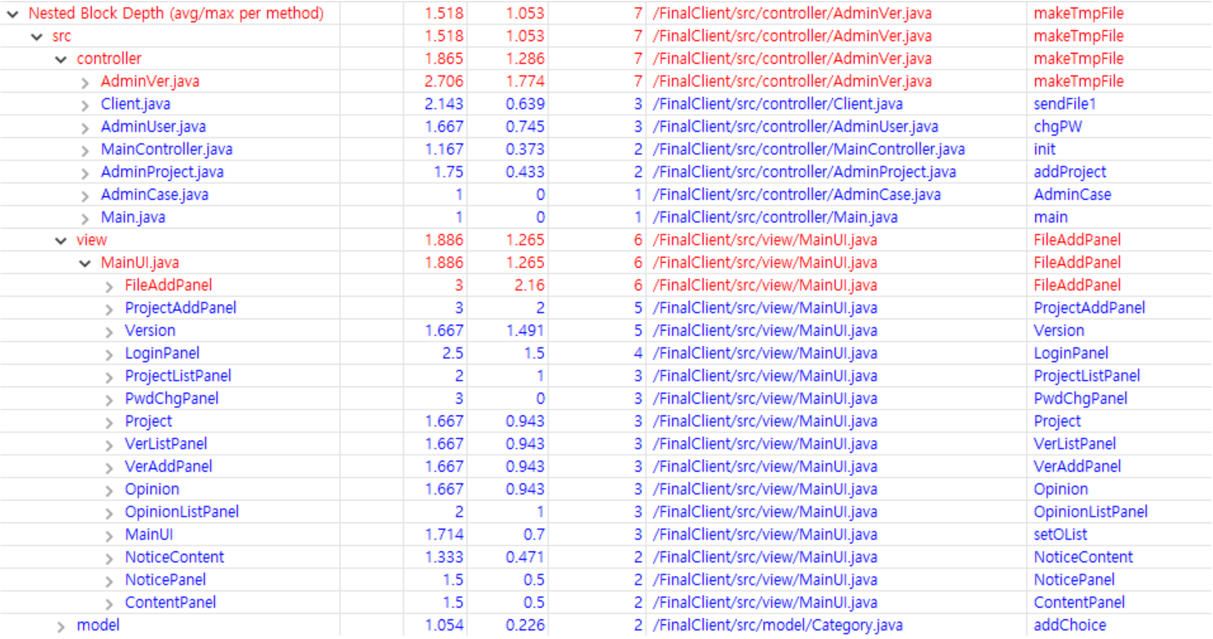
2. Number of Parameters



Number of Parameters (avg/max per method)		0.671	0.893	6	/FinalClient/src/controller/AdminVer.java	makeTmpFile
src		0.671	0.893	6	/FinalClient/src/controller/AdminVer.java	makeTmpFile
controller		1	1.109	6	/FinalClient/src/controller/AdminVer.java	makeTmpFile
AdminVer.java		1.824	1.294	6	/FinalClient/src/controller/AdminVer.java	makeTmpFile
AdminUser.java		1.333	1.106	3	/FinalClient/src/controller/AdminUser.java	chgPW
Client.java		0.571	0.495	1	/FinalClient/src/controller/Client.java	Client
AdminCase.java		0.455	0.498	1	/FinalClient/src/controller/AdminCase.java	setChoiceList
AdminProject.java		0.75	0.433	1	/FinalClient/src/controller/AdminProject.java	AdminProject
Main.java		1	0	1	/FinalClient/src/controller/Main.java	main
MainController.java		0	0	0	/FinalClient/src/controller/MainController.java	MainController
view		0.705	0.967	4	/FinalClient/src/view/MainUI.java	Version
model		0.419	0.52	2	/FinalClient/src/model/Version.java	Version
Nested Block Depth (avg/max per method)		1.518	1.053	7	/FinalClient/src/controller/AdminVer.java	makeTmpFile

AdminVer.java내의 makeTmpFile에서 Maximum 6의 높은 기록을 하였다.

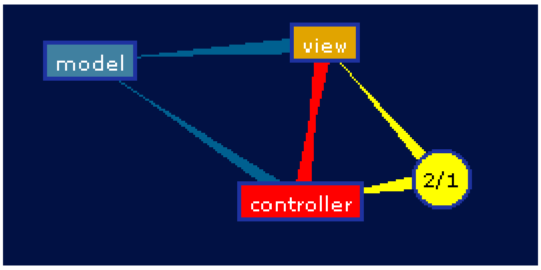
3. Nested Block Depth



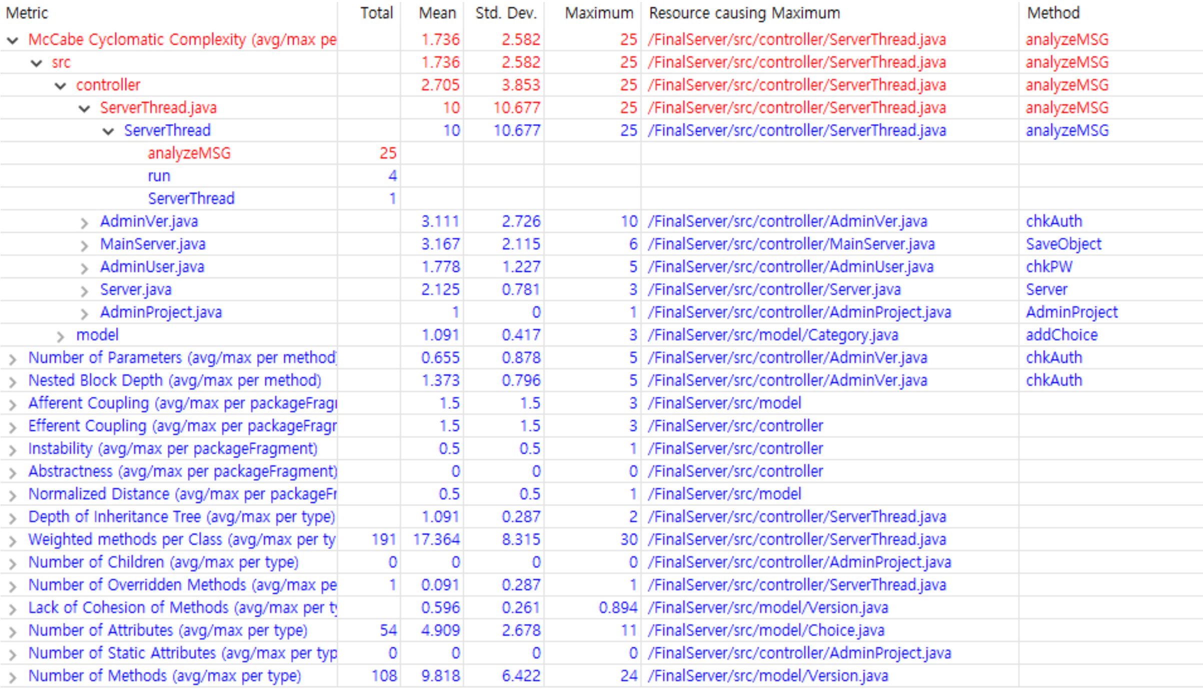
Nested Block Depth (avg/max per method)	1.518	1.053	7	/FinalClient/src/controller/AdminVer.java	makeTmpFile
src	1.518	1.053	7	/FinalClient/src/controller/AdminVer.java	makeTmpFile
controller	1.865	1.286	7	/FinalClient/src/controller/AdminVer.java	makeTmpFile
AdminVer.java	2.706	1.774	7	/FinalClient/src/controller/AdminVer.java	makeTmpFile
Client.java	2.143	0.639	3	/FinalClient/src/controller/Client.java	sendFile1
AdminUser.java	1.667	0.745	3	/FinalClient/src/controller/AdminUser.java	chgPW
MainController.java	1.167	0.373	2	/FinalClient/src/controller/MainController.java	init
AdminProject.java	1.75	0.433	2	/FinalClient/src/controller/AdminProject.java	addProject
AdminCase.java	1	0	1	/FinalClient/src/controller/AdminCase.java	AdminCase
Main.java	1	0	1	/FinalClient/src/controller/Main.java	main
view	1.886	1.265	6	/FinalClient/src/view/MainUI.java	FileAddPanel
MainUI.java	1.886	1.265	6	/FinalClient/src/view/MainUI.java	FileAddPanel
FileAddPanel	3	2.16	6	/FinalClient/src/view/MainUI.java	FileAddPanel
ProjectAddPanel	3	2	5	/FinalClient/src/view/MainUI.java	ProjectAddPanel
Version	1.667	1.491	5	/FinalClient/src/view/MainUI.java	Version
LoginPanel	2.5	1.5	4	/FinalClient/src/view/MainUI.java	LoginPanel
ProjectListPanel	2	1	3	/FinalClient/src/view/MainUI.java	ProjectListPanel
PwdChgPanel	3	0	3	/FinalClient/src/view/MainUI.java	PwdChgPanel
Project	1.667	0.943	3	/FinalClient/src/view/MainUI.java	Project
VerListPanel	1.667	0.943	3	/FinalClient/src/view/MainUI.java	VerListPanel
VerAddPanel	1.667	0.943	3	/FinalClient/src/view/MainUI.java	VerAddPanel
Opinion	1.667	0.943	3	/FinalClient/src/view/MainUI.java	Opinion
OpinionListPanel	2	1	3	/FinalClient/src/view/MainUI.java	OpinionListPanel
MainUI	1.714	0.7	3	/FinalClient/src/view/MainUI.java	setOList
NoticeContent	1.333	0.471	2	/FinalClient/src/view/MainUI.java	NoticeContent
NoticePanel	1.5	0.5	2	/FinalClient/src/view/MainUI.java	NoticePanel
ContentPanel	1.5	0.5	2	/FinalClient/src/view/MainUI.java	ContentPanel
model	1.054	0.226	2	/FinalClient/src/model/Category.java	addChoice

adminVer.java내의 makeTmpFile에서 Maximum 7을 기록하였고 UI부분의 FindAddPanel에서도 Maximum 6의 높은 점수를 기록하였다.

4. Dependency Graph View



FinalServer



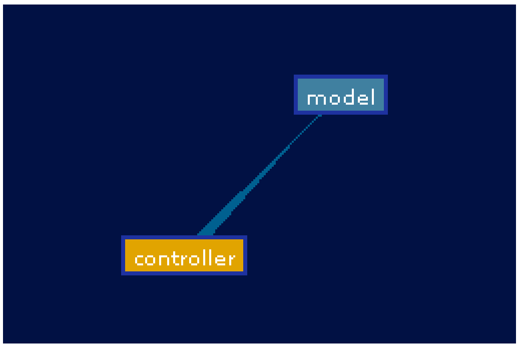
Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max pe		1.736	2.582	25	/FinalServer/src/controller/ServerThread.java	analyzeMSG
src		1.736	2.582	25	/FinalServer/src/controller/ServerThread.java	analyzeMSG
controller		2.705	3.853	25	/FinalServer/src/controller/ServerThread.java	analyzeMSG
ServerThread.java		10	10.677	25	/FinalServer/src/controller/ServerThread.java	analyzeMSG
ServerThread		10	10.677	25	/FinalServer/src/controller/ServerThread.java	analyzeMSG
analyzeMSG	25					
run	4					
ServerThread	1					
AdminVer.java		3.111	2.726	10	/FinalServer/src/controller/AdminVer.java	chkAuth
MainServer.java		3.167	2.115	6	/FinalServer/src/controller/MainServer.java	SaveObject
AdminUser.java		1.778	1.227	5	/FinalServer/src/controller/AdminUser.java	chkPW
Server.java		2.125	0.781	3	/FinalServer/src/controller/Server.java	Server
AdminProject.java		1	0	1	/FinalServer/src/controller/AdminProject.java	AdminProject
model		1.091	0.417	3	/FinalServer/src/model/Category.java	addChoice
Number of Parameters (avg/max per method		0.655	0.878	5	/FinalServer/src/controller/AdminVer.java	chkAuth
Nested Block Depth (avg/max per method)		1.373	0.796	5	/FinalServer/src/controller/AdminVer.java	chkAuth
Afferent Coupling (avg/max per packageFragi		1.5	1.5	3	/FinalServer/src/model	
Efferent Coupling (avg/max per packageFragr		1.5	1.5	3	/FinalServer/src/controller	
Instability (avg/max per packageFragment)		0.5	0.5	1	/FinalServer/src/controller	
Abstractness (avg/max per packageFragment)		0	0	0	/FinalServer/src/controller	
Normalized Distance (avg/max per packageFr		0.5	0.5	1	/FinalServer/src/model	
Depth of Inheritance Tree (avg/max per type)		1.091	0.287	2	/FinalServer/src/controller/ServerThread.java	
Weighted methods per Class (avg/max per ty	191	17.364	8.315	30	/FinalServer/src/controller/ServerThread.java	
Number of Children (avg/max per type)	0	0	0	0	/FinalServer/src/controller/AdminProject.java	
Number of Overridden Methods (avg/max pe	1	0.091	0.287	1	/FinalServer/src/controller/ServerThread.java	
Lack of Cohesion of Methods (avg/max per t		0.596	0.261	0.894	/FinalServer/src/model/Version.java	
Number of Attributes (avg/max per type)	54	4.909	2.678	11	/FinalServer/src/model/Choice.java	
Number of Static Attributes (avg/max per typ	0	0	0	0	/FinalServer/src/controller/AdminProject.java	
Number of Methods (avg/max per type)	108	9.818	6.422	24	/FinalServer/src/model/Version.java	

Cyclomatic Complexity부분을 제외한 나머지 항목들에서는 양호한 점수가 기록되었다.

1. Cyclomatic Complexity

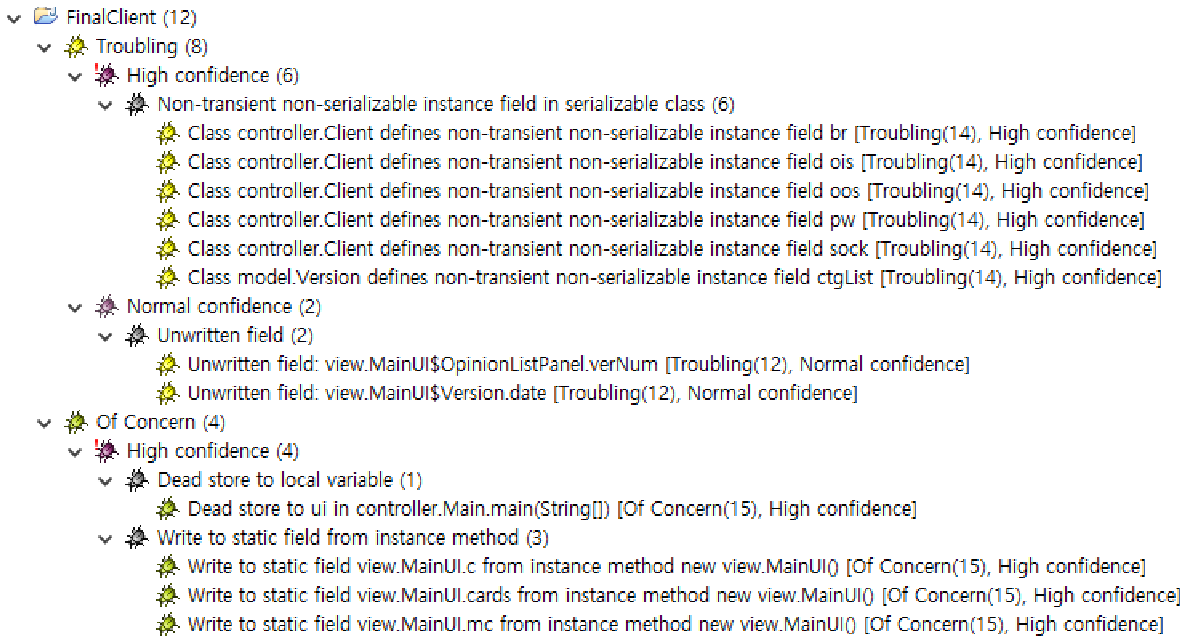
- ServerThread.java내의 analyzeMSG에서 Maximum 25의 아주 높은 점수가 기록되었다.

2. Dependency Graph View



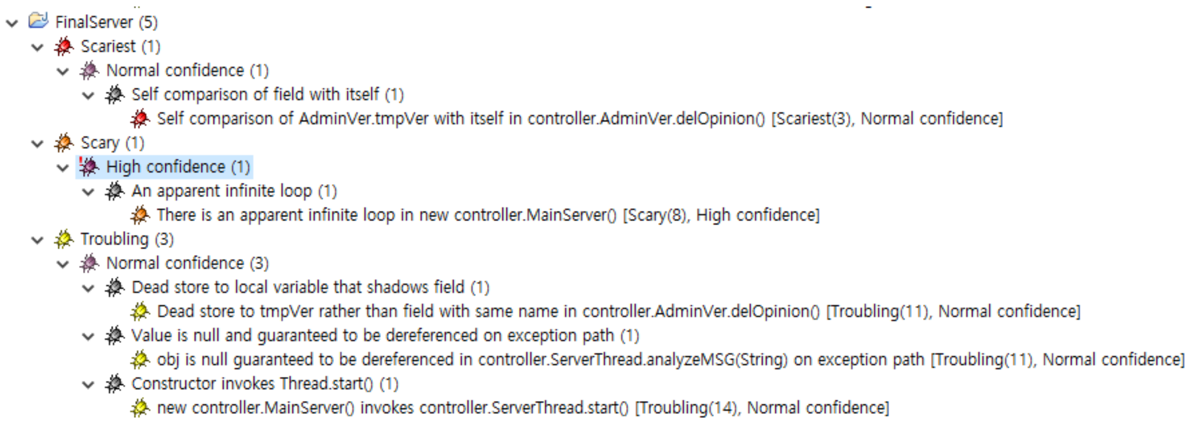
- FindBugs report분석

1. FinalClient



```
FinalClient (12)
  Troubling (8)
    High confidence (6)
      Non-transient non-serializable instance field in serializable class (6)
          Class controller.Client defines non-transient non-serializable instance field br [Troubling(14), High confidence]
          Class controller.Client defines non-transient non-serializable instance field ois [Troubling(14), High confidence]
          Class controller.Client defines non-transient non-serializable instance field oos [Troubling(14), High confidence]
          Class controller.Client defines non-transient non-serializable instance field pw [Troubling(14), High confidence]
          Class controller.Client defines non-transient non-serializable instance field sock [Troubling(14), High confidence]
          Class model.Version defines non-transient non-serializable instance field ctgList [Troubling(14), High confidence]
    Normal confidence (2)
      Unwritten field (2)
          Unwritten field: view.MainUI$OpinionListPanel.verNum [Troubling(12), Normal confidence]
          Unwritten field: view.MainUI$Version.date [Troubling(12), Normal confidence]
  Of Concern (4)
    High confidence (4)
      Dead store to local variable (1)
          Dead store to ui in controller.Main.main(String[]) [Of Concern(15), High confidence]
      Write to static field from instance method (3)
          Write to static field view.MainUI.c from instance method new view.MainUI() [Of Concern(15), High confidence]
          Write to static field view.MainUI.cards from instance method new view.MainUI() [Of Concern(15), High confidence]
          Write to static field view.MainUI.mc from instance method new view.MainUI() [Of Concern(15), High confidence]
```

2. FinalServer



```
FinalServer (5)
  Scariest (1)
    Normal confidence (1)
      Self comparison of field with itself (1)
          Self comparison of AdminVer.tmpVer with itself in controller.AdminVer.delOpinion() [Scariest(3), Normal confidence]
  Scary (1)
    High confidence (1)
      An apparent infinite loop (1)
          There is an apparent infinite loop in new controller.MainServer() [Scary(8), High confidence]
  Troubling (3)
    Normal confidence (3)
      Dead store to local variable that shadows field (1)
          Dead store to tmpVer rather than field with same name in controller.AdminVer.delOpinion() [Troubling(11), Normal confidence]
      Value is null and guaranteed to be dereferenced on exception path (1)
          obj is null guaranteed to be dereferenced in controller.ServerThread.analyzeMSG(String) on exception path [Troubling(11), Normal confidence]
      Constructor invokes Thread.start() (1)
          new controller.MainServer() invokes controller.ServerThread.start() [Troubling(14), Normal confidence]
```

2. T3 코드 분석

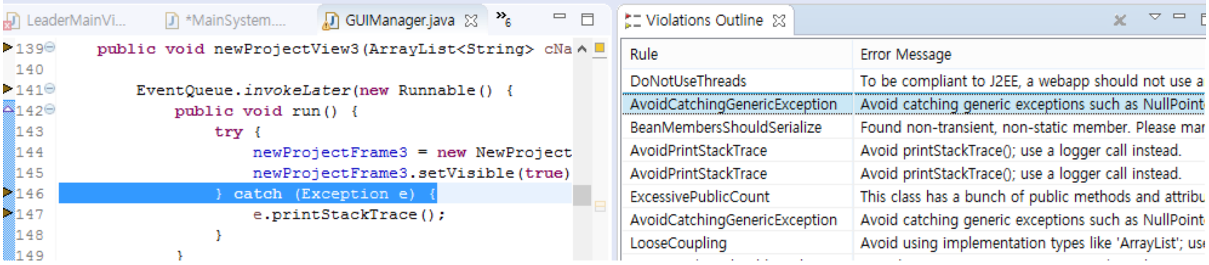
- PMD warnings

1. Control

1.1 GUIManager

Control.GUIManager.newProjectView3

⇨ AvoidCatchingGenericException : Exception 클래스로 뭉뚱그려 잡지 말고 NullPointer Exception 등의 세부적인 Exception으로 잡아줘야 한다.

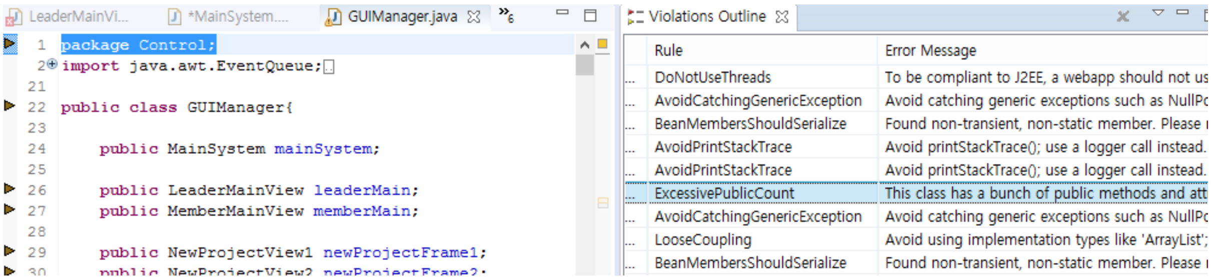


```
public void newProjectView3(ArrayList<String> cNa

    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                newProjectFrame3 = new NewProject
                newProjectFrame3.setVisible(true)
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
```

Rule	Error Message
DoNotUseThreads	To be compliant to J2EE, a webapp should not use a
AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPoint
BeanMembersShouldSerialize	Found non-transient, non-static member. Please mar
AvoidPrintStackTrace	Avoid printStackTrace(); use a logger call instead.
AvoidPrintStackTrace	Avoid printStackTrace(); use a logger call instead.
ExcessivePublicCount	This class has a bunch of public methods and attribu
AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPoint
LooseCoupling	Avoid using implementation types like 'ArrayList'; use

Control.GUIManager

⇨ ExcessivePublicCount : 해당 클래스에 너무 많은 퍼블릭 메소드 및 변수가 존재한다. 적절히 private등으로 캡슐화 해주어야 한다.

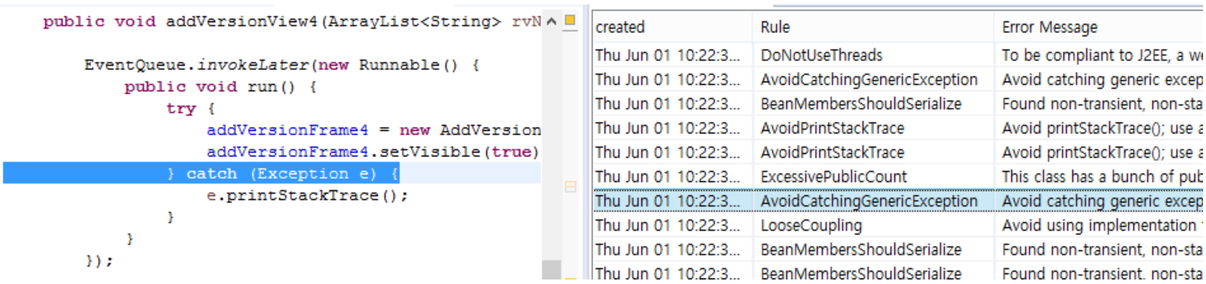


```
package Control;
import java.awt.EventQueue;

public class GUIManager{

    public MainSystem mainSystem;

    public LeaderMainView leaderMain;
    public MemberMainView memberMain;

    public NewProjectView1 newProjectFrame1;
    public NewProjectView2 newProjectFrame2;
```

Rule	Error Message
DoNotUseThreads	To be compliant to J2EE, a webapp should not us
AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPo
BeanMembersShouldSerialize	Found non-transient, non-static member. Please i
AvoidPrintStackTrace	Avoid printStackTrace(); use a logger call instead.
AvoidPrintStackTrace	Avoid printStackTrace(); use a logger call instead.
ExcessivePublicCount	This class has a bunch of public methods and att
AvoidCatchingGenericException	Avoid catching generic exceptions such as NullPo
LooseCoupling	Avoid using implementation types like 'ArrayList';
BeanMembersShouldSerialize	Found non-transient, non-static member. Please i

Control.GUIManager.newProjectView4

⇨ AvoidCatchingGenericException : Exception 클래스로 뭉뚱그려 잡지 말고 NullPointer Exception 등의 세부적인 Exception으로 잡아줘야 한다.

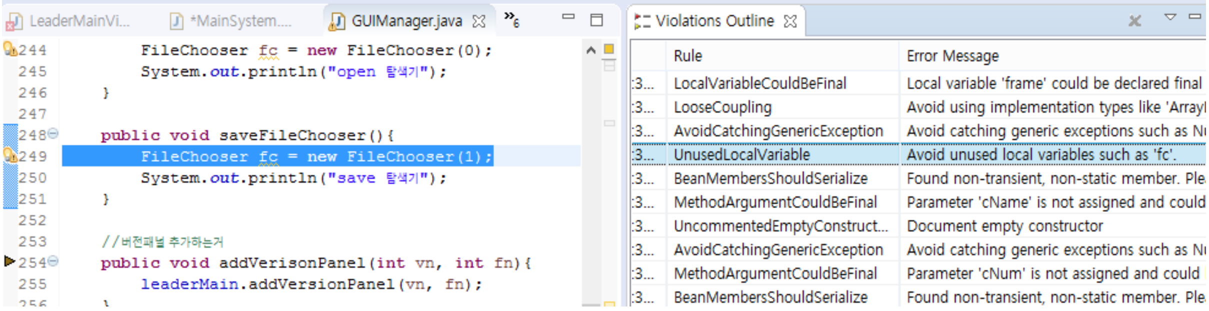


```
public void addVersionView4(ArrayList<String> rvN

    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                addVersionFrame4 = new AddVersion
                addVersionFrame4.setVisible(true)
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
```

created	Rule	Error Message
Thu Jun 01 10:22:3...	DoNotUseThreads	To be compliant to J2EE, a we
Thu Jun 01 10:22:3...	AvoidCatchingGenericException	Avoid catching generic excep
Thu Jun 01 10:22:3...	BeanMembersShouldSerialize	Found non-transient, non-sta
Thu Jun 01 10:22:3...	AvoidPrintStackTrace	Avoid printStackTrace(); use a
Thu Jun 01 10:22:3...	AvoidPrintStackTrace	Avoid printStackTrace(); use a
Thu Jun 01 10:22:3...	ExcessivePublicCount	This class has a bunch of pub
Thu Jun 01 10:22:3...	AvoidCatchingGenericException	Avoid catching generic excep
Thu Jun 01 10:22:3...	LooseCoupling	Avoid using implementation
Thu Jun 01 10:22:3...	BeanMembersShouldSerialize	Found non-transient, non-sta
Thu Jun 01 10:22:3...	BeanMembersShouldSerialize	Found non-transient, non-sta

Control.GUIManager.saveFileChooser

⇨ UnusedLocalVariable : 사용하지 않는 전역변수 fc를 선언하였다. 객체 생성을 위해서 였다면, new FileChooser(1); 만 사용해도 될 것이다.

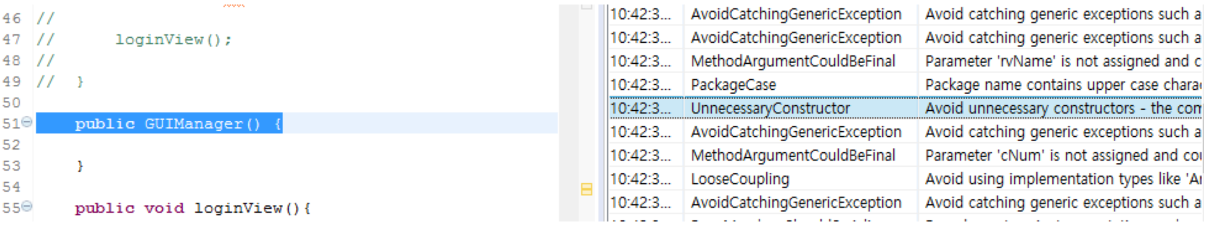


```
        FileChooser fc = new FileChooser(0);
        System.out.println("open 탐색기");
    }

    public void saveFileChooser(){
        FileChooser fc = new FileChooser(1);
        System.out.println("save 탐색기");
    }

    //버전패널 추가하는거
    public void addVerisonPanel(int vn, int fn){
        leaderMain.addVersionPanel(vn, fn);
```

	Rule	Error Message
:3...	LocalVariableCouldBeFinal	Local variable 'frame' could be declared final
:3...	LooseCoupling	Avoid using implementation types like 'Arrayl
:3...	AvoidCatchingGenericException	Avoid catching generic exceptions such as N
:3...	UnusedLocalVariable	Avoid unused local variables such as 'fc'.
:3...	BeanMembersShouldSerialize	Found non-transient, non-static member. Ple
:3...	MethodArgumentCouldBeFinal	Parameter 'cName' is not assigned and could
:3...	UncommentedEmptyConstruct...	Document empty constructor
:3...	AvoidCatchingGenericException	Avoid catching generic exceptions such as N
:3...	MethodArgumentCouldBeFinal	Parameter 'cNum' is not assigned and could
:3...	BeanMembersShouldSerialize	Found non-transient, non-static member. Ple

Control.GUIManager.GUIManager

⇨ UnnecessaryConstructor : 굳이 기본 생성자는 만들지 않아도 알아서 만들어준다.



```
//
//      loginView();
//
//  }

    public GUIManager() {

    }

    public void loginView(){
```

10:42:3...	AvoidCatchingGenericException	Avoid catching generic exceptions such a
10:42:3...	AvoidCatchingGenericException	Avoid catching generic exceptions such a
10:42:3...	MethodArgumentCouldBeFinal	Parameter 'rvName' is not assigned and c
10:42:3...	PackageCase	Package name contains upper case chara
10:42:3...	UnnecessaryConstructor	Avoid unnecessary constructors - the con
10:42:3...	AvoidCatchingGenericException	Avoid catching generic exceptions such a
10:42:3...	MethodArgumentCouldBeFinal	Parameter 'cNum' is not assigned and co
10:42:3...	LooseCoupling	Avoid using implementation types like 'A
10:42:3...	AvoidCatchingGenericException	Avoid catching generic exceptions such a

1.2 MainSystem

⇨ Violation이 일부 존재하긴 하지만 크게 영향을 미치는 문제는 발견되지 않음.

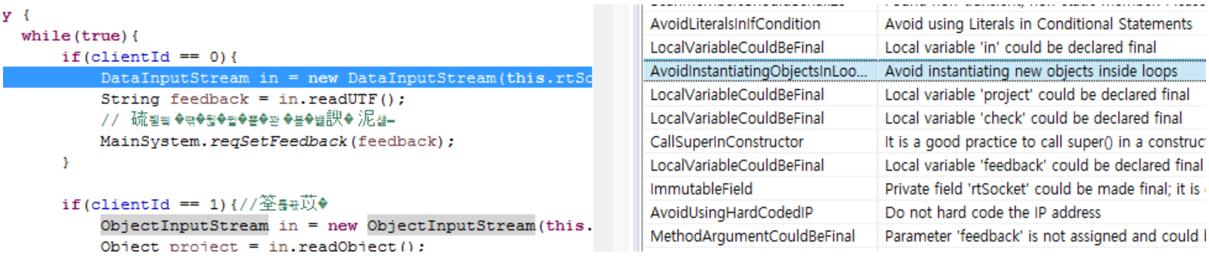
1.3 ProjectManager

⇨ Violation이 일부 존재하긴 하지만 크게 영향을 미치는 문제는 발견되지 않음.

1.4 NetWorkManager

Control.ReceiveThreadInNetworkManager.run

⇨ AvoidInstantiatingObjectsInLoops : 루프 내에서 new를 통해 새 인스턴스를 만드는 것은 권장하지 않는다.

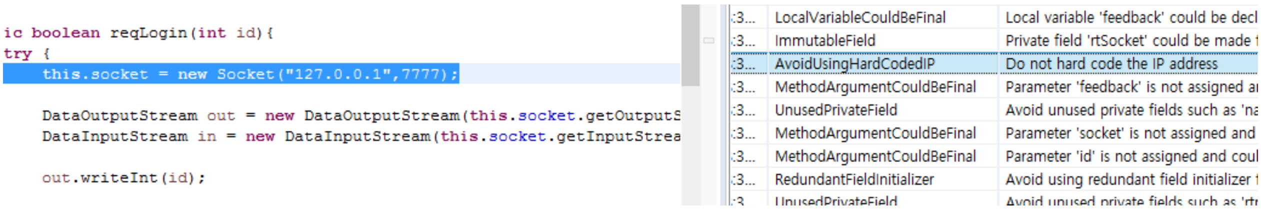


```
y {
  while(true){
      if(clientId == 0){
          DataInputStream in = new DataInputStream(this.rtSc
          String feedback = in.readUTF();
          // 硫뷸씤 ◆뭐◆됨◆뽑◆꽈 ◆블◆벌訳◆ 泥섎–
          MainSystem.reqSetFeedback(feedback);
      }

      if(clientId == 1){//筌뤿쑬苡◆
          ObjectInputStream in = new ObjectInputStream(this.
          Object project = in.readObject();
```

AvoidLiteralsInIfCondition	Avoid using Literals in Conditional Statements
LocalVariableCouldBeFinal	Local variable 'in' could be declared final
AvoidInstantiatingObjectsInLoo...	Avoid instantiating new objects inside loops
LocalVariableCouldBeFinal	Local variable 'project' could be declared final
LocalVariableCouldBeFinal	Local variable 'check' could be declared final
CallSuperInConstructor	It is a good practice to call super() in a construc
LocalVariableCouldBeFinal	Local variable 'feedback' could be declared final
ImmutableField	Private field 'rtSocket' could be made final; it is
AvoidUsingHardCodedIP	Do not hard code the IP address
MethodArgumentCouldBeFinal	Parameter 'feedback' is not assigned and could

Control.NetworkManager.reqLogin

⇨ AvoidUsingHardCodedIP : IP 주소를 직접 손으로 작성하여 입력하는 것은 지양해야한다.

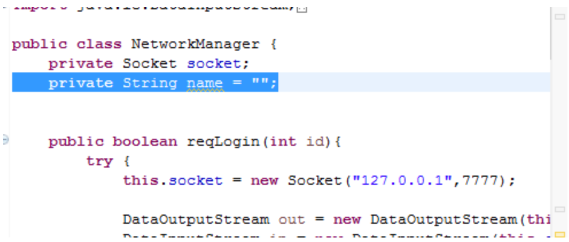


```
ic boolean reqLogin(int id){
try {
    this.socket = new Socket("127.0.0.1",7777);

    DataOutputStream out = new DataOutputStream(this.socket.getOutputS
    DataInputStream in = new DataInputStream(this.socket.getInputStrea

    out.writeInt(id);
```

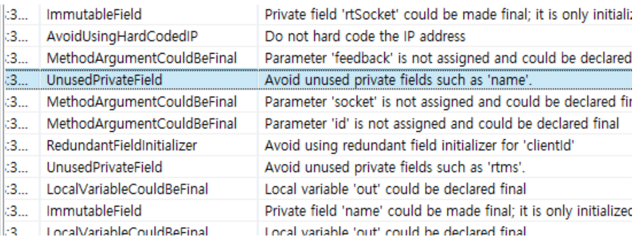
:3...	LocalVariableCouldBeFinal	Local variable 'feedback' could be decl
:3...	ImmutableField	Private field 'rtSocket' could be made f
:3...	AvoidUsingHardCodedIP	Do not hard code the IP address
:3...	MethodArgumentCouldBeFinal	Parameter 'feedback' is not assigned a
:3...	UnusedPrivateField	Avoid unused private fields such as 'na
:3...	MethodArgumentCouldBeFinal	Parameter 'socket' is not assigned and
:3...	MethodArgumentCouldBeFinal	Parameter 'id' is not assigned and coul
:3...	RedundantFieldInitializer	Avoid using redundant field initializer f
:3	UnusedPrivateField	Avoid unused private fields such as 'rt

Control.NetworkManager

⇨ UnusedPrivateField : 사용하지 않는 변수 name이 남아있다. 삭제해줄 것.



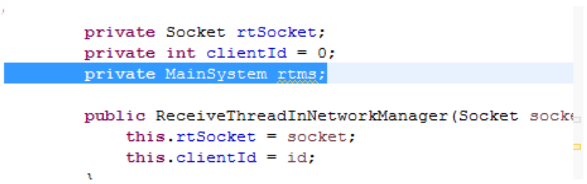
```
import java.io.DataInputStream;

public class NetworkManager {
    private Socket socket;
    private String name = "";


    public boolean reqLogin(int id){
        try {
            this.socket = new Socket("127.0.0.1",7777);

            DataOutputStream out = new DataOutputStream(thi
            DataInputStream in = new DataInputStream(this.s
```



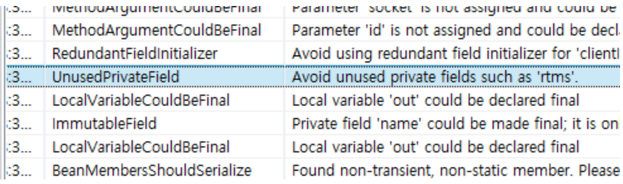
:3...	ImmutableField	Private field 'rtSocket' could be made final; it is only initiali:
:3...	AvoidUsingHardCodedIP	Do not hard code the IP address
:3...	MethodArgumentCouldBeFinal	Parameter 'feedback' is not assigned and could be declared
:3...	UnusedPrivateField	Avoid unused private fields such as 'name'.
:3...	MethodArgumentCouldBeFinal	Parameter 'socket' is not assigned and could be declared fi
:3...	MethodArgumentCouldBeFinal	Parameter 'id' is not assigned and could be declared final
:3...	RedundantFieldInitializer	Avoid using redundant field initializer for 'clientId'
:3...	UnusedPrivateField	Avoid unused private fields such as 'rtms'.
:3...	LocalVariableCouldBeFinal	Local variable 'out' could be declared final
:3...	ImmutableField	Private field 'name' could be made final; it is only initializec
:3	LocalVariableCouldBeFinal	Local variable 'out' could be declared final

Control.ReceiveThreadInNetworkManager

⇨ UnusedPrivateField : 사용하지 않는 변수 rtms가 남아있다. 삭제해줄 것.



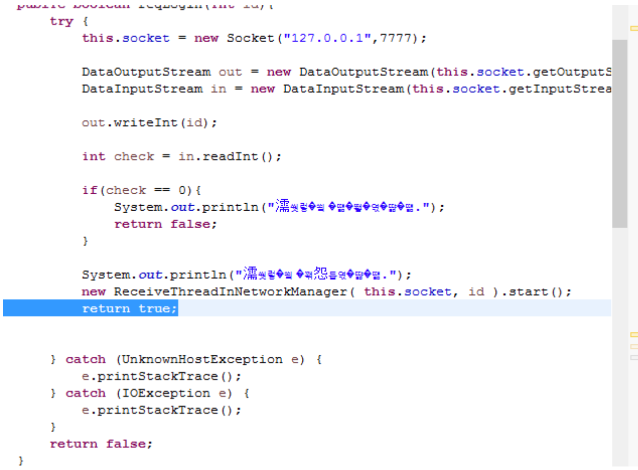
```
    private Socket rtSocket;
    private int clientId = 0;
    private MainSystem rtms;

    public ReceiveThreadInNetworkManager(Socket socke
        this.rtSocket = socket;
        this.clientId = id;
    }
```



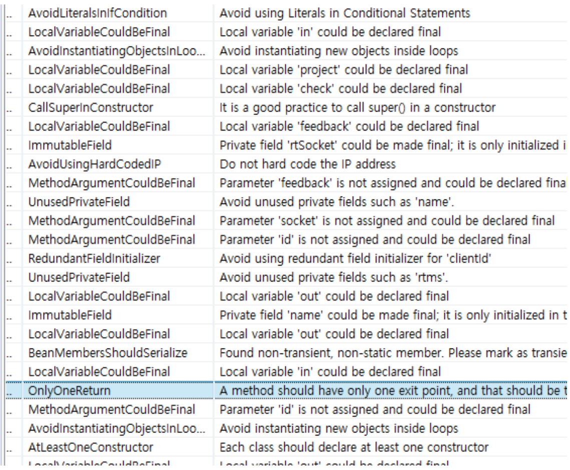
:3...	MethodArgumentCouldBeFinal	Parameter 'socket' is not assigned and could be
:3...	MethodArgumentCouldBeFinal	Parameter 'id' is not assigned and could be decl
:3...	RedundantFieldInitializer	Avoid using redundant field initializer for 'clientI
:3...	UnusedPrivateField	Avoid unused private fields such as 'rtms'.
:3...	LocalVariableCouldBeFinal	Local variable 'out' could be declared final
:3...	ImmutableField	Private field 'name' could be made final; it is on
:3...	LocalVariableCouldBeFinal	Local variable 'out' could be declared final
:3...	BeanMembersShouldSerialize	Found non-transient, non-static member. Please

Control.NetworkManager.reqLogin

⇨ OnlyOneReturn : 메소드 내에 조건문으로 분기가 나뉜 것도 아닌데, try문 끝부분과 메소드 자체 끝 부분에 return이 각각 존재한다. 하나를 삭제해줄 것.



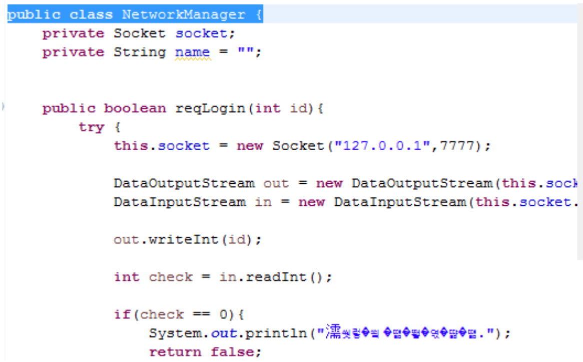
```
public boolean reqLogin(int id){
    try {
        this.socket = new Socket("127.0.0.1",7777);

        DataOutputStream out = new DataOutputStream(this.socket.getOutputS
        DataInputStream in = new DataInputStream(this.socket.getInputStrea

        out.writeInt(id);

        int check = in.readInt();

        if(check == 0){
            System.out.println("濡쒓렇�씤 �떎�뙣�뻽�뒿�땲�떎.");
            return false;
        }

        System.out.println("濡쒓렇�씤 �꽦怨듯뻽�뒿�땲�떎.");
        new ReceiveThreadInNetworkManager( this.socket, id ).start();
        return true;


    } catch (UnknownHostException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return false;
}
```



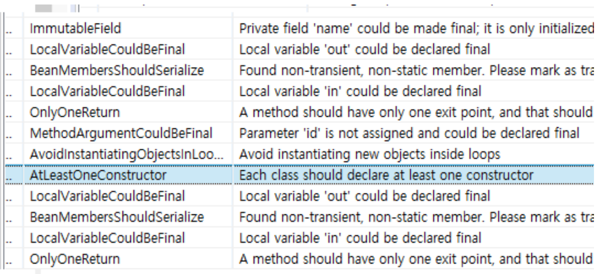
..	AvoidLiteralsInIfCondition	Avoid using Literals in Conditional Statements
..	LocalVariableCouldBeFinal	Local variable 'in' could be declared final
..	AvoidInstantiatingObjectsInLoo...	Avoid instantiating new objects inside loops
..	LocalVariableCouldBeFinal	Local variable 'project' could be declared final
..	LocalVariableCouldBeFinal	Local variable 'check' could be declared final
..	CallSuperInConstructor	It is a good practice to call super() in a constructor
..	LocalVariableCouldBeFinal	Local variable 'feedback' could be declared final
..	ImmutableField	Private field 'rtSocket' could be made final; it is only initialized i
..	AvoidUsingHardCodedIP	Do not hard code the IP address
..	MethodArgumentCouldBeFinal	Parameter 'feedback' is not assigned and could be declared fina
..	UnusedPrivateField	Avoid unused private fields such as 'name'.
..	MethodArgumentCouldBeFinal	Parameter 'socket' is not assigned and could be declared final
..	MethodArgumentCouldBeFinal	Parameter 'id' is not assigned and could be declared final
..	RedundantFieldInitializer	Avoid using redundant field initializer for 'clientId'
..	UnusedPrivateField	Avoid unused private fields such as 'rtms'.
..	LocalVariableCouldBeFinal	Local variable 'out' could be declared final
..	ImmutableField	Private field 'name' could be made final; it is only initialized in t
..	LocalVariableCouldBeFinal	Local variable 'out' could be declared final
..	BeanMembersShouldSerialize	Found non-transient, non-static member. Please mark as transie
..	LocalVariableCouldBeFinal	Local variable 'in' could be declared final
..	OnlyOneReturn	A method should have only one exit point, and that should be t
..	MethodArgumentCouldBeFinal	Parameter 'id' is not assigned and could be declared final
..	AvoidInstantiatingObjectsInLoo...	Avoid instantiating new objects inside loops
..	AtLeastOneConstructor	Each class should declare at least one constructor
..	LocalVariableCouldBeFinal	Local variable 'out' could be declared final

Control.NetworkManager

➪ AtLeastOneConstructor : 각 클래스마다 적어도 하나의 생성자는 생성해주는 것이 좋다.



```
public class NetworkManager {
    private Socket socket;
    private String name = "";


    public boolean reqLogin(int id){
        try {
            this.socket = new Socket("127.0.0.1",7777);

            DataOutputStream out = new DataOutputStream(this.sock
            DataInputStream in = new DataInputStream(this.socket.

            out.writeInt(id);

            int check = in.readInt();

            if(check == 0){
                System.out.println("濡쒓렇�씤 �떎�뙣�뻽�뒿�땲�떎.");
                return false;
```

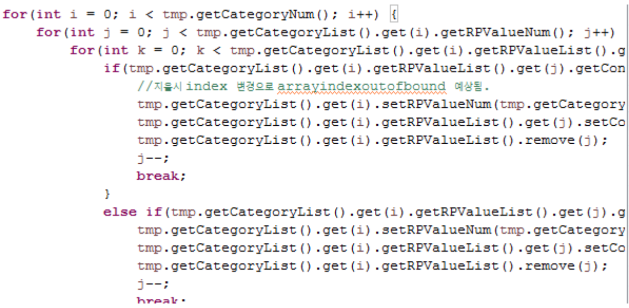


..	ImmutableField	Private field 'name' could be made final; it is only initialized
..	LocalVariableCouldBeFinal	Local variable 'out' could be declared final
..	BeanMembersShouldSerialize	Found non-transient, non-static member. Please mark as tra
..	LocalVariableCouldBeFinal	Local variable 'in' could be declared final
..	OnlyOneReturn	A method should have only one exit point, and that should
..	MethodArgumentCouldBeFinal	Parameter 'id' is not assigned and could be declared final
..	AvoidInstantiatingObjectsInLoo...	Avoid instantiating new objects inside loops
..	AtLeastOneConstructor	Each class should declare at least one constructor
..	LocalVariableCouldBeFinal	Local variable 'out' could be declared final
..	BeanMembersShouldSerialize	Found non-transient, non-static member. Please mark as tra
..	LocalVariableCouldBeFinal	Local variable 'in' could be declared final
..	OnlyOneReturn	A method should have only one exit point, and that should

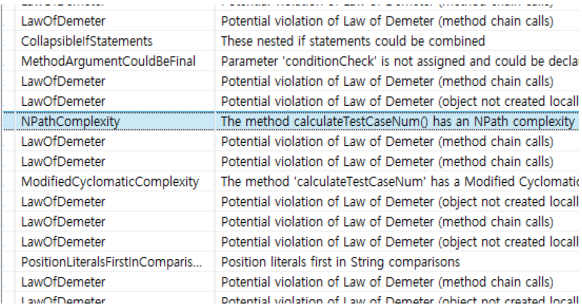
1.4 ResultManager

Control.ResultManager.calculateTestCaseNum

➪ NPathComplexity : 현재 3중 for문을 통해 갈라지는 경우의 수가 너무 많음. 복잡도를 줄이기 위해서 알고리즘을 새로 구상해보는 것을 추천.



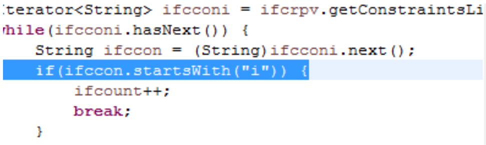
```
for(int i = 0; i < tmp.getCategoryNum(); i++) {
    for(int j = 0; j < tmp.getCategoryList().get(i).getRPValueNum(); j++)
        for(int k = 0; k < tmp.getCategoryList().get(i).getRPValueList().g
            if(tmp.getCategoryList().get(i).getRPValueList().get(j).getCon
                //지울시 index 변경으로 arrayindexoutofbound 예상됨.
                tmp.getCategoryList().get(i).setRPValueNum(tmp.getCategory
                tmp.getCategoryList().get(i).getRPValueList().get(j).setCo
                tmp.getCategoryList().get(i).getRPValueList().remove(j);
                j--;
                break;
            }
            else if(tmp.getCategoryList().get(i).getRPValueList().get(j).g
                tmp.getCategoryList().get(i).setRPValueNum(tmp.getCategory
                tmp.getCategoryList().get(i).getRPValueList().get(j).setCo
                tmp.getCategoryList().get(i).getRPValueList().remove(j);
                j--;
                break;
```



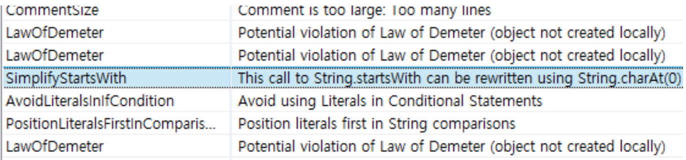
LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
CollapsibleIfStatements	These nested if statements could be combined
MethodArgumentCouldBeFinal	Parameter 'conditionCheck' is not assigned and could be decla
LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
LawOfDemeter	Potential violation of Law of Demeter (object not created locall
NPathComplexity	The method calculateTestCaseNum() has an NPath complexity
LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
ModifiedCyclomaticComplexity	The method 'calculateTestCaseNum' has a Modified Cyclomatic
LawOfDemeter	Potential violation of Law of Demeter (object not created locall
LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
LawOfDemeter	Potential violation of Law of Demeter (object not created locall
PositionLiteralsFirstInComparis...	Position literals first in String comparisons
LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
LawOfDemeter	Potential violation of Law of Demeter (object not created locall

Control.ResultManager.calculateTestCaseNum

➪ SimplifyStartsWith : startWith을 쓰기보다 charAt(0)를 쓰는 것이 더 심플하다.



```
:terator<String> ifcconi = ifcrpv.getConstraintsLi
/hile(ifcconi.hasNext()) {
    String ifccon = (String)ifcconi.next();
    if(ifccon.startsWith("i")) {
        ifcount++;
        break;
    }
```



CommentSize	Comment is too large: Too many lines
LawOfDemeter	Potential violation of Law of Demeter (object not created locally)
LawOfDemeter	Potential violation of Law of Demeter (object not created locally)
SimplifyStartsWith	This call to String.startsWith can be rewritten using String.charAt(0)
AvoidLiteralsInIfCondition	Avoid using Literals in Conditional Statements
PositionLiteralsFirstInComparis...	Position literals first in String comparisons
LawOfDemeter	Potential violation of Law of Demeter (object not created locally)

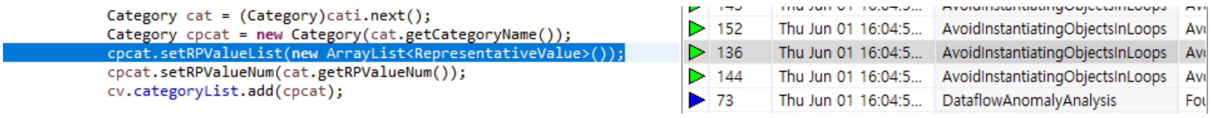
2. GUI

GUI는 창의 개수가 매우 많고, 시스템 내부 작동과 외적인 것이라고 판단하여 제외한다.

3. Model

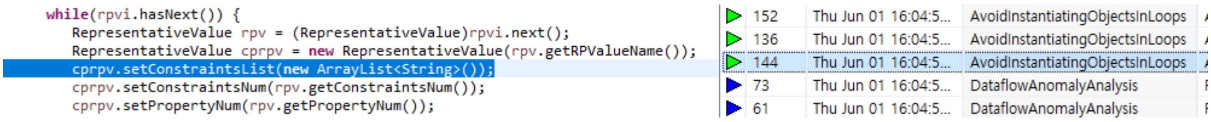
Model.Version

⇨ AvoidInstantiationgObjectsInLoops: 루프 내에서 new를 통해 새 인스턴스를 만드는 것은 권장하지 않는다.



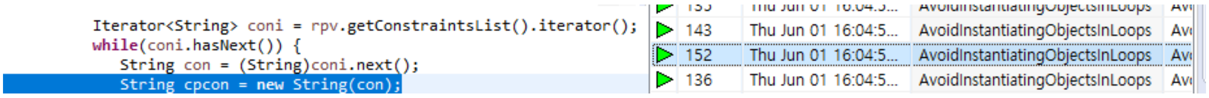
```
        Category cat = (Category)cati.next();
        Category cpcat = new Category(cat.getCategoryName());
        cpcat.setRPValueList(new ArrayList<RepresentativeValue>());
        cpcat.setRPValueNum(cat.getRPValueNum());
        cv.categoryList.add(cpcat);
```

▶	152	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	136	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	144	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	73	Thu Jun 01 16:04:5...	DataflowAnomalyAnalysis	Fo



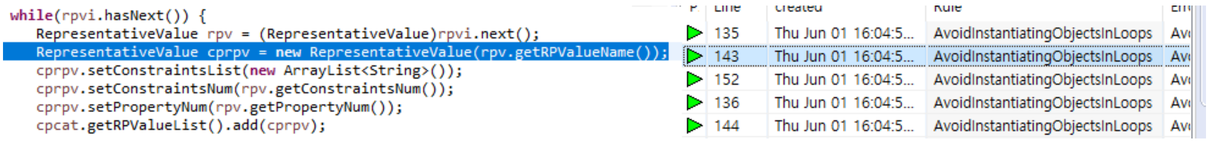
```
    while(rpvi.hasNext()) {
      RepresentativeValue rpv = (RepresentativeValue)rpvi.next();
      RepresentativeValue cprpv = new RepresentativeValue(rpv.getRPValueName());
      cprpv.setConstraintsList(new ArrayList<String>());
      cprpv.setConstraintsNum(rpv.getConstraintsNum());
      cprpv.setPropertyNum(rpv.getPropertyNum());
```

▶	152	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	136	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	144	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	73	Thu Jun 01 16:04:5...	DataflowAnomalyAnalysis
▶	61	Thu Jun 01 16:04:5...	DataflowAnomalyAnalysis



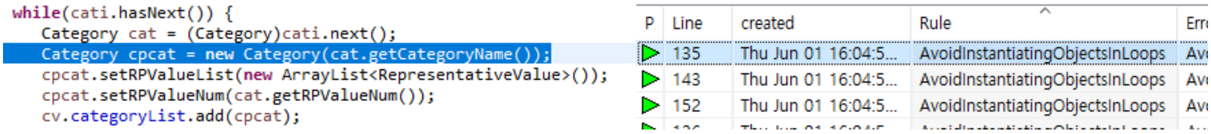
```
      Iterator<String> coni = rpv.getConstraintsList().iterator();
      while(coni.hasNext()) {
        String con = (String)coni.next();
        String cpcon = new String(con);
```

▶	143	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	152	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	136	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av



```
while(rpvi.hasNext()) {
   RepresentativeValue rpv = (RepresentativeValue)rpvi.next();
   RepresentativeValue cprpv = new RepresentativeValue(rpv.getRPValueName());
   cprpv.setConstraintsList(new ArrayList<String>());
   cprpv.setConstraintsNum(rpv.getConstraintsNum());
   cprpv.setPropertyNum(rpv.getPropertyNum());
   cpcat.getRPValueList().add(cprpv);
```

P	Line	created	Rule	Err
▶	135	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	143	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	152	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	136	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	144	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av

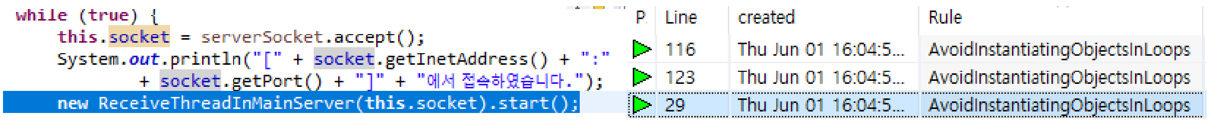


```
while(cati.hasNext()) {
  Category cat = (Category)cati.next();
  Category cpcat = new Category(cat.getCategoryName());
  cpcat.setRPValueList(new ArrayList<RepresentativeValue>());
  cpcat.setRPValueNum(cat.getRPValueNum());
  cv.categoryList.add(cpcat);
```

P	Line	created	Rule	Err
▶	135	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	143	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av
▶	152	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops	Av

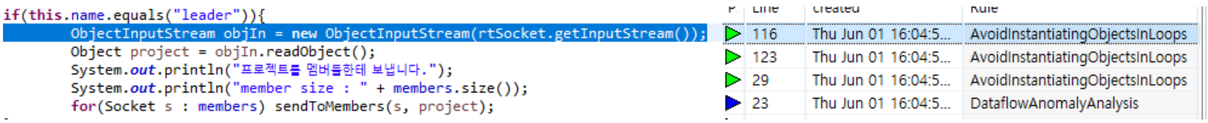
Model.Version

⇨ AvoidInstantiationgObjectsInLoops: 루프 내에서 new를 통해 새 인스턴스를 만드는 것은 권장하지 않는다.



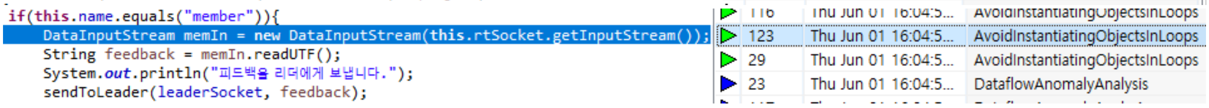
```
 while (true) {
     this.socket = serverSocket.accept();
     System.out.println("[" + socket.getInetAddress() + ":"
             + socket.getPort() + "]" + "에서 접속하였습니다.");
     new ReceiveThreadInMainServer(this.socket).start();
```

P	Line	created	Rule
▶	116	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	123	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	29	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops



```
if(this.name.equals("leader")){
        ObjectInputStream objIn = new ObjectInputStream(rtSocket.getInputStream());
        Object project = objIn.readObject();
        System.out.println("프로젝트를 멤버들한테 보냅니다.");
        System.out.println("member size : " + members.size());
        for(Socket s : members) sendToMembers(s, project);
```

P	Line	created	Rule
▶	116	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	123	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	29	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	23	Thu Jun 01 16:04:5...	DataflowAnomalyAnalysis

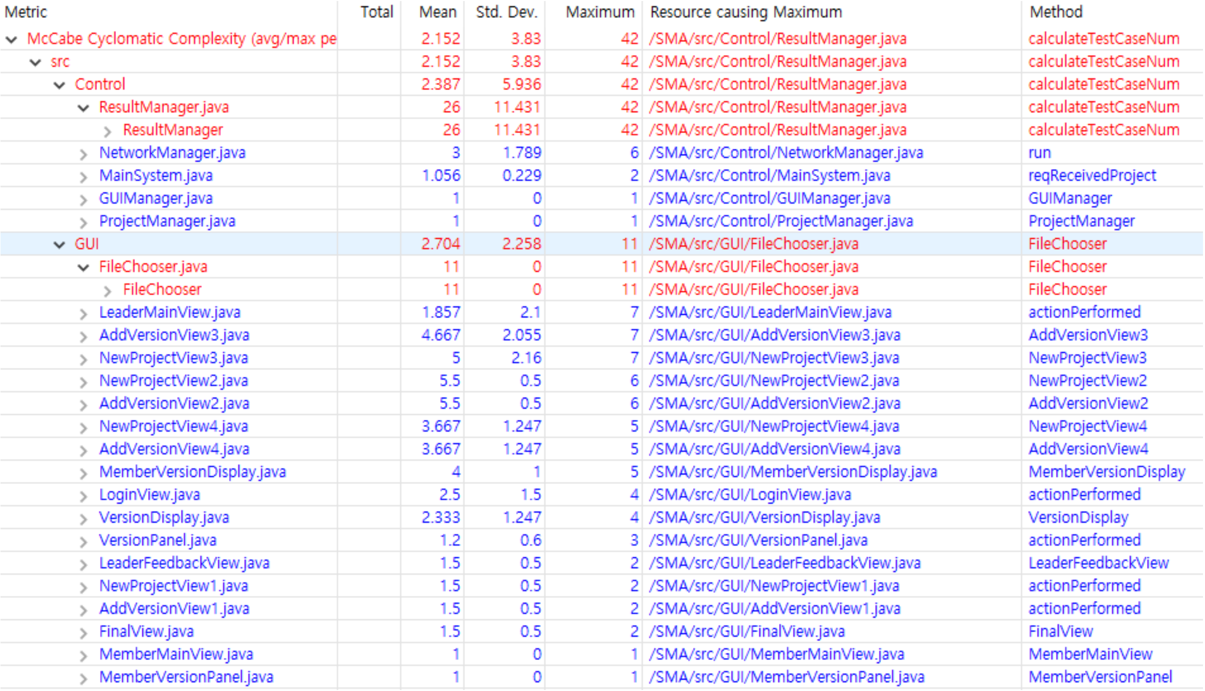


```
 if(this.name.equals("member")){
    DataInputStream memIn = new DataInputStream(this.rtSocket.getInputStream());
    String feedback = memIn.readUTF();
    System.out.println("피드백을 리더에게 보냅니다.");
    sendToLeader(leaderSocket, feedback);
```

▶	116	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	123	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	29	Thu Jun 01 16:04:5...	AvoidInstantiatingObjectsInLoops
▶	23	Thu Jun 01 16:04:5...	DataflowAnomalyAnalysis

- Metrics

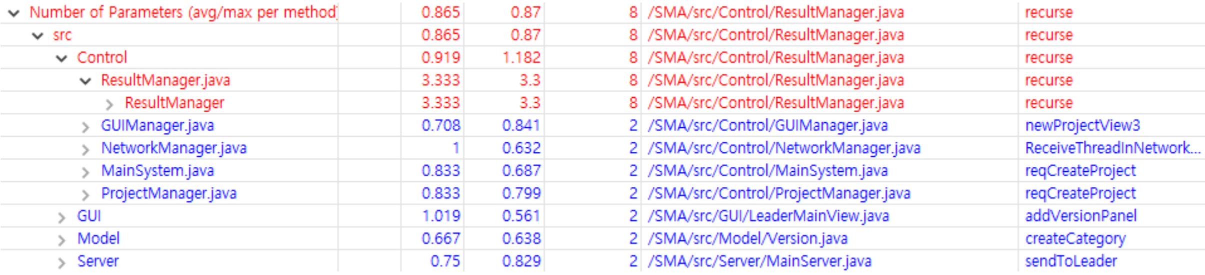
1. Cyclomatic Complexity



Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max pe		2.152	3.83	42	/SMA/src/Control/ResultManager.java	calculateTestCaseNum
src		2.152	3.83	42	/SMA/src/Control/ResultManager.java	calculateTestCaseNum
Control		2.387	5.936	42	/SMA/src/Control/ResultManager.java	calculateTestCaseNum
ResultManager.java		26	11.431	42	/SMA/src/Control/ResultManager.java	calculateTestCaseNum
ResultManager		26	11.431	42	/SMA/src/Control/ResultManager.java	calculateTestCaseNum
NetworkManager.java		3	1.789	6	/SMA/src/Control/NetworkManager.java	run
MainSystem.java		1.056	0.229	2	/SMA/src/Control/MainSystem.java	reqReceivedProject
GUIManager.java		1	0	1	/SMA/src/Control/GUIManager.java	GUIManager
ProjectManager.java		1	0	1	/SMA/src/Control/ProjectManager.java	ProjectManager
GUI		2.704	2.258	11	/SMA/src/GUI/FileChooser.java	FileChooser
FileChooser.java		11	0	11	/SMA/src/GUI/FileChooser.java	FileChooser
FileChooser		11	0	11	/SMA/src/GUI/FileChooser.java	FileChooser
LeaderMainView.java		1.857	2.1	7	/SMA/src/GUI/LeaderMainView.java	actionPerformed
AddVersionView3.java		4.667	2.055	7	/SMA/src/GUI/AddVersionView3.java	AddVersionView3
NewProjectView3.java		5	2.16	7	/SMA/src/GUI/NewProjectView3.java	NewProjectView3
NewProjectView2.java		5.5	0.5	6	/SMA/src/GUI/NewProjectView2.java	NewProjectView2
AddVersionView2.java		5.5	0.5	6	/SMA/src/GUI/AddVersionView2.java	AddVersionView2
NewProjectView4.java		3.667	1.247	5	/SMA/src/GUI/NewProjectView4.java	NewProjectView4
AddVersionView4.java		3.667	1.247	5	/SMA/src/GUI/AddVersionView4.java	AddVersionView4
MemberVersionDisplay.java		4	1	5	/SMA/src/GUI/MemberVersionDisplay.java	MemberVersionDisplay
LoginView.java		2.5	1.5	4	/SMA/src/GUI/LoginView.java	actionPerformed
VersionDisplay.java		2.333	1.247	4	/SMA/src/GUI/VersionDisplay.java	VersionDisplay
VersionPanel.java		1.2	0.6	3	/SMA/src/GUI/VersionPanel.java	actionPerformed
LeaderFeedbackView.java		1.5	0.5	2	/SMA/src/GUI/LeaderFeedbackView.java	LeaderFeedbackView
NewProjectView1.java		1.5	0.5	2	/SMA/src/GUI/NewProjectView1.java	actionPerformed
AddVersionView1.java		1.5	0.5	2	/SMA/src/GUI/AddVersionView1.java	actionPerformed
FinalView.java		1.5	0.5	2	/SMA/src/GUI/FinalView.java	FinalView
MemberMainView.java		1	0	1	/SMA/src/GUI/MemberMainView.java	MemberMainView
MemberVersionPanel.java		1	0	1	/SMA/src/GUI/MemberVersionPanel.java	MemberVersionPanel

ResultManager.java내의 caculateTestCaseNum에서 무려 42의 아주 심하게 높은 점수를 기록하였다. 또한 FileChooser.java의 FileChooser에서도 11의 높은 점수가 기록되었다.

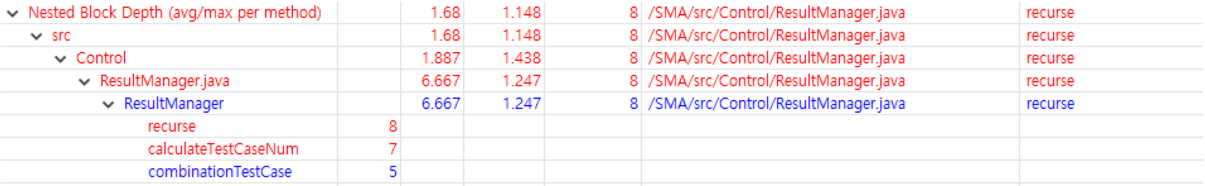
2. Number of Parameters



Number of Parameters (avg/max per method)		0.865	0.87	8	/SMA/src/Control/ResultManager.java	recurse
src		0.865	0.87	8	/SMA/src/Control/ResultManager.java	recurse
Control		0.919	1.182	8	/SMA/src/Control/ResultManager.java	recurse
ResultManager.java		3.333	3.3	8	/SMA/src/Control/ResultManager.java	recurse
ResultManager		3.333	3.3	8	/SMA/src/Control/ResultManager.java	recurse
GUIManager.java		0.708	0.841	2	/SMA/src/Control/GUIManager.java	newProjectView3
NetworkManager.java		1	0.632	2	/SMA/src/Control/NetworkManager.java	ReceiveThreadInNetwork...
MainSystem.java		0.833	0.687	2	/SMA/src/Control/MainSystem.java	reqCreateProject
ProjectManager.java		0.833	0.799	2	/SMA/src/Control/ProjectManager.java	reqCreateProject
GUI		1.019	0.561	2	/SMA/src/GUI/LeaderMainView.java	addVersionPanel
Model		0.667	0.638	2	/SMA/src/Model/Version.java	createCategory
Server		0.75	0.829	2	/SMA/src/Server/MainServer.java	sendToLeader

ResultManager.java의 recurse에서 8의 높은 점수가 기록되었다.

3. Nested Block Depth

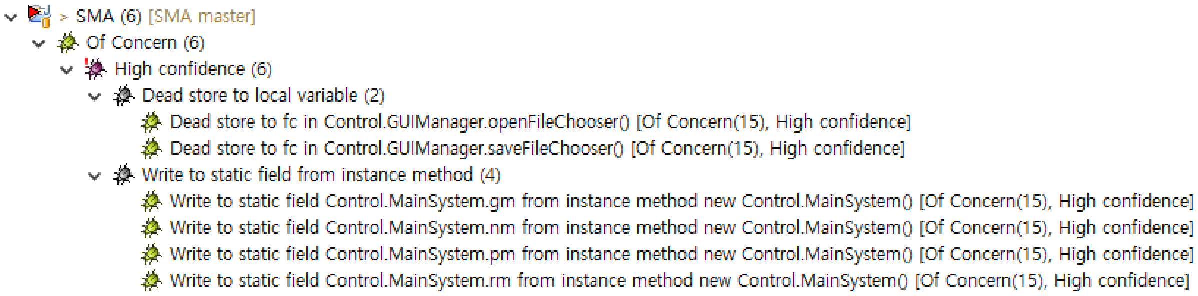


Nested Block Depth (avg/max per method)		1.68	1.148	8	/SMA/src/Control/ResultManager.java	recurse
src		1.68	1.148	8	/SMA/src/Control/ResultManager.java	recurse
Control		1.887	1.438	8	/SMA/src/Control/ResultManager.java	recurse
ResultManager.java		6.667	1.247	8	/SMA/src/Control/ResultManager.java	recurse
ResultManager		6.667	1.247	8	/SMA/src/Control/ResultManager.java	recurse
recurse	8					
calculateTestCaseNum	7					
combinationTestCase	5					

ResultManager.java내의 recurse에서 8의 높은 점수가 기록되었다.

전반적으로 ResultManager.java에 대하여 많은 항목에서 높은 점수가 기록되었다.

- FindBugs report



```
> SMA (6) [SMA master]
  Of Concern (6)
    High confidence (6)
      Dead store to local variable (2)
        Dead store to fc in Control.GUIManager.openFileChooser() [Of Concern(15), High confidence]
        Dead store to fc in Control.GUIManager.saveFileChooser() [Of Concern(15), High confidence]
      Write to static field from instance method (4)
        Write to static field Control.MainSystem.gm from instance method new Control.MainSystem() [Of Concern(15), High confidence]
        Write to static field Control.MainSystem.nm from instance method new Control.MainSystem() [Of Concern(15), High confidence]
        Write to static field Control.MainSystem.pm from instance method new Control.MainSystem() [Of Concern(15), High confidence]
        Write to static field Control.MainSystem.rm from instance method new Control.MainSystem() [Of Concern(15), High confidence]
```