

---

---

# Feedback Category Partitioning Test Tool

---

201311272 나경수  
201311300 이원재  
201311306 이진호  
201310672 한석호

# 목차

## CONTENTS

Static Analysis 대응

Demonstration

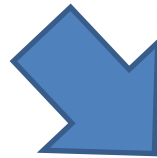
Epilogue

# Static Analysis 대응

## 1. PMD 대응

```
public void openFileChooser(){  
    FileChooser fc = new FileChooser(0);  
}
```

```
public void saveFileChooser(){  
    FileChooser fc = new FileChooser(1);  
}
```



사용하지 않는  
객체 변수 제거

```
public void openFileChooser(){  
    new FileChooser(0);  
}  
  
public void saveFileChooser(){  
    new FileChooser(1);  
}
```

# Static Analysis 대응

## 1. PMD 대응

```
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(41);

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

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

: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 fir
: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...	LocalVariableCouldBeFinal	Private field 'name' could be made final; it is only initializec
:3...	LocalVariableCouldBeFinal	Local variable 'out' could be declared final
: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

# Static Analysis 대응

## 1. PMD 대응

```
public LeaderMainView leaderMain;
public MemberMainView memberMain;

public NewProjectView1 newProjectFrame1;
public NewProjectView2 newProjectFrame2;
public NewProjectView3 newProjectFrame3;
public NewProjectView4 newProjectFrame4;

public AddVersionView1 addVersionFrame1;
public AddVersionView2 addVersionFrame2;
public AddVersionView3 addVersionFrame3;
public AddVersionView4 addVersionFrame4;

public LeaderFeedbackView leaderFeedbackView;
```



정보의 은닉화

```
private NewProjectView1 newProjectFrame1;
private NewProjectView2 newProjectFrame2;
private NewProjectView3 newProjectFrame3;
private NewProjectView4 newProjectFrame4;

private AddVersionView1 addVersionFrame1;
private AddVersionView2 addVersionFrame2;
private AddVersionView3 addVersionFrame3;
private AddVersionView4 addVersionFrame4;

private LeaderFeedbackView leaderFeedbackView;

private LeaderMainView leaderMain;
private MemberMainView memberMain;

public LeaderMainView getLeaderMain() {
    return leaderMain;
}

public MemberMainView getMemberMain() {
    return memberMain;
}
```

# Static Analysis 대응

## 1. PMD 대응

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

        DataOutputStream out = new DataOutputStream(this.socket.getOutputStream());
        DataInputStream in = new DataInputStream(this.socket.getInputStream());

        out.writeInt(id);

        int check = in.readInt();

        if(check == 0) {
            System.out.println("정상적으로 연결되었습니다.");
            return false;
        }

        System.out.println("클라이언트 연결되었습니다.");
        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
.. AvoidInstantiatingObjectsInLoops	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
.. <b>OnlyOneReturn</b>	<b>A method should have only one exit point, and that should be t</b>
.. MethodArgumentCouldBeFinal	Parameter 'id' is not assigned and could be declared final
.. AvoidInstantiatingObjectsInLoops	Avoid instantiating new objects inside loops
.. AtLeastOneConstructor	Each class should declare at least one constructor
.. LocalVariableCouldBeFinal	Local variable 'out' could be declared final

# Static Analysis 대응

## 1. PMD 대응

```
public boolean reqLogin(int id) {  
    int check = 0;  
  
    try {  
        this.socket = new Socket("127.0.0.1", 7777);  
  
        DataOutputStream out = new DataOutputStream(this.socket.getOutputStream());  
        DataInputStream in = new DataInputStream(this.socket.getInputStream());  
  
        out.writeInt(id);  
  
        check = in.readInt();  
  
    } catch (UnknownHostException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
  
    if (check == 0) {  
        System.out.println("로그인 실패입니다.");  
        return false;  
    }  
  
    System.out.println("로그인 성공입니다.");  
    new ReceiveThreadInNetworkManager(this.socket, id).start();  
    return true;  
}
```



# Static Analysis 대응

## 1. PMD 대응

```
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  
                //지음시 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 local
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 local
LawOfDemeter	Potential violation of Law of Demeter (method chain calls)
LawOfDemeter	Potential violation of Law of Demeter (object not created local
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 local



# Static Analysis 대응

## 1. PMD 대응

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

CommentSize	Comment is too large: 100 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)



```
while(ifcconi.hasNext()) {  
    String ifccon = (String)ifcconi.next();  
    if(ifccon.charAt(0) == 'i') {  
        ifcount++;  
        break;  
    }  
}
```

# Static Analysis 대응

## 2. Eclipse Matrix

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
▼ McCabe Cyclomatic Complexity (avg/max per		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

# Static Analysis 대응

## 2. Eclipse Matrix

▼ 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

# Static Analysis 대응

## 2. Eclipse Matrix

▼ 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					

---

# Demonstration

---

---

# Epilogue

---

## 느낀점

- 프로그램 구현 전까지 oopt를 따라 가며 보고서 작성 및 설계를 하느라 시간이 오래 걸렸지만 막상 구현 해 보니 구현 자체는 매우 빨랐다.
- 무작정 코딩을 시작하는 것보다 초기에 원하는 요구사항을 분석하고 체계적으로 프로그램을 만들 수 있어서 좋다.
- 여럿이서 한 프로그램을 만들 때 충돌이 생기기 마련인데, oopt를 통한 충분한 분석과 설계 그리고 diagram 작성으로 좀 더 유기적인 프로그래밍을 경험할 수 있었다.
- '고생하기' 라는 목표에 달성 하였다.



---

---

감사합니다.

---