

S M A _ T 6

1To10 _ CPT Tool

Static Analysis

One to Ten

CPT 한번에 끝내기



didix@naver.com



010-9633-5320



trello.com//software-modelling

| 손성호 이광제 이용주 황준익

Table of Contents

Introduction

Level 1. IntelliJ

Level 2. IntelliJ

Level 3. IntelliJ



Introduction

Introduction

Static Analysis

T5. SSS CPT

Level 1 : IntelliJ

Level 2 : Eclipse metrics

Level 3 : Find bugs

T8. Feesual CPT

Level 1 : IntelliJ

Level 2 : Eclipse metrics

Level 3 : Find bugs

Static Analysis

Lv.1_IntelliJ

Static Analysis Lv1_IntelliJ

T5. SSS CPT

149개
검출

T8. Feesual CPT

20개
검출

Team 5

SSS CPT

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool

▼ **Class structure** 13 warnings

- ▼ **Field can be local** 12 warnings
 - ▶ **DataController** 1 warning
 - ▶ **DataValue** 3 warnings
 - ▶ **ResultPane** 3 warnings
 - ▶ **ValuePane** 5 warnings
- ▶ **Parameter can be local** 1 warning
- ▶ **Control flow issues** 16 warnings
- ▶ **Data flow issues** 3 warnings
- ▶ **Declaration redundancy** 55 warnings
- ▶ **Error handling** 1 warning
- ▶ **General** 2 weak warnings
- ▶ **Imports** 51 warnings

Convert to local Suppress

```
private Category headerCategory;  
private Category tailCategory;
```

▼ **Class structure** 13 warnings

- ▼ **Field can be local** 12 warnings
 - ▶ **DataController** 1 warning
 - ▶ **DataValue** 3 warnings
 - ▶ **ResultPane** 3 warnings
 - ▶ **ValuePane** 5 warnings
- ▶ **Parameter can be local** 1 warning
- ▼ **Control flow issues** 16 warnings
 - ▶ **Pointless boolean expression** 13 warnings
 - ▶ **Redundant 'if' statement** 3 warnings
- ▶ **Data flow issues** 3 warnings
- ▶ **Declaration redundancy** 55 warnings
- ▶ **Error handling** 1 warning

3 problems: Convert to local Suppress

```
private Attribute single_header;  
private Attribute single_tail;  
private Attribute error_header;  
private Attribute error_tail;  
private Attribute valid_header;  
private Attribute valid_tail;
```

▼ **Class structure** 13 warnings

- ▼ **Field can be local** 12 warnings
 - ▶ **DataController** 1 warning
 - ▶ **DataValue** 3 warnings
 - ▶ **ResultPane** 3 warnings
 - ▶ **ValuePane** 5 warnings
- ▶ **Parameter can be local** 1 warning
- ▼ **Control flow issues** 16 warnings
 - ▶ **Pointless boolean expression** 13 warnings
 - ▶ **Redundant 'if' statement** 3 warnings
- ▶ **Data flow issues** 3 warnings
- ▶ **Declaration redundancy** 55 warnings
- ▶ **Error handling** 1 warning

3 problems: Convert to local Suppress

```
public class ResultPane extends JPanel{  
    private JPanel resultPanel;  
    private JLabel resultLabel, printLabel;  
    //색상  
    private Color bg= new Color(255,255,255);  
    private Color resultC = new Color(126,135,191);  
    public ResultPane(){
```

▼ **Class structure** 13 warnings

- ▼ **Field can be local** 12 warnings
 - ▶ **DataController** 1 warning
 - ▶ **DataValue** 3 warnings
 - ▶ **ResultPane** 3 warnings
 - ▶ **ValuePane** 5 warnings
- ▶ **Parameter can be local** 1 warning
- ▼ **Control flow issues** 16 warnings
 - ▶ **Pointless boolean expression** 13 warnings
 - ▶ **Redundant 'if' statement** 3 warnings
- ▶ **Data flow issues** 3 warnings
- ▶ **Declaration redundancy** 55 warnings
- ▶ **Error handling** 1 warning

5 problems: Convert to local Suppress

```
public class ValuePane extends JPanel{  
    private JPanel valueListPane, btnPanel, temp;  
    private JPanel valueList[];  
    private JButton addBtn, delBtn, valueConfirm[];  
    private JCheckBox single[],error[];  
  
    private JComboBox propertyCombo[],IfPropertyCombo[];  
    private JScrollPane scrollIPane;  
    //설정 변수들  
    private int width = 200, height =700;// 크기  
    private int indexTemp =0;  
    private int listSize =50; //리스트 크기
```

변수가 하나의 메소드 에서만 사용됨 (지역변수 추천)

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool

▼ **Class structure** 13 warnings

- ▶ Field can be local 12 warnings
- ▼ Parameter can be local 1 warning
 - ▼ **Ⓢ** MainController 1 warning
 - Parameter can be converted to a local variable

▼ **Control flow issues** 16 warnings

- ▶ Pointless boolean expression 13 warnings
- ▶ Redundant 'if' statement 3 warnings

▶ **Data flow issues** 3 warnings

▶ **Declaration redundancy** 55 warnings

▶ **Error handling** 1 warning

▶ **General** 2 weak warnings

▶ **Imports** 51 warnings

Convert to local Suppress ▼

```
}  
public int getCategoryIndex(String categoryName, int index){  
    index = dc.getCategoryIndex(categoryName);  
}
```

9: Version Control Terminal Inspection Results

인자로 받은 Index값 사용 x

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool

3 problems: Simplify Suppress

```
    }  
    if (swap == true){  
        Category temp = category_list.get(i);  
  
        if (loc == true)  
            i++;  
  
        if (nothing_check == true)  
        {
```

10 problems: Simplify Suppress

```
public void addDataValue(Value v){  
    if (v.isSingle()==true){  
        addSingle(v.getName());  
    }else if (v.isError()==true){  
        addError(v.getName());  
    }else if (v.isProperty()==true || v.isIfproperty() == true){  
        addProperty(v);  
    }else if (v.isValid()==true){  
        addValid(v.getName());  
    }  
  
public void deleteDataValue(Value v){
```

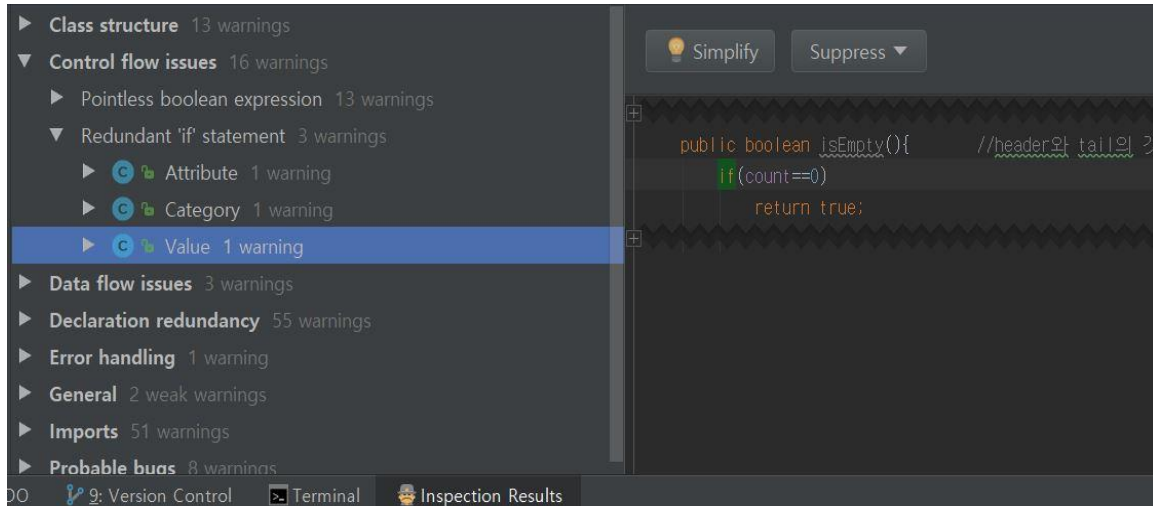
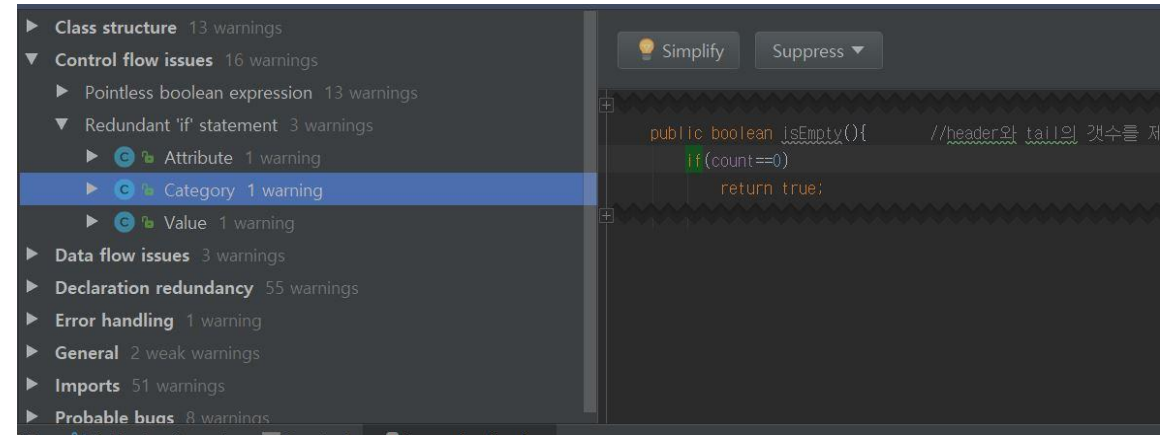
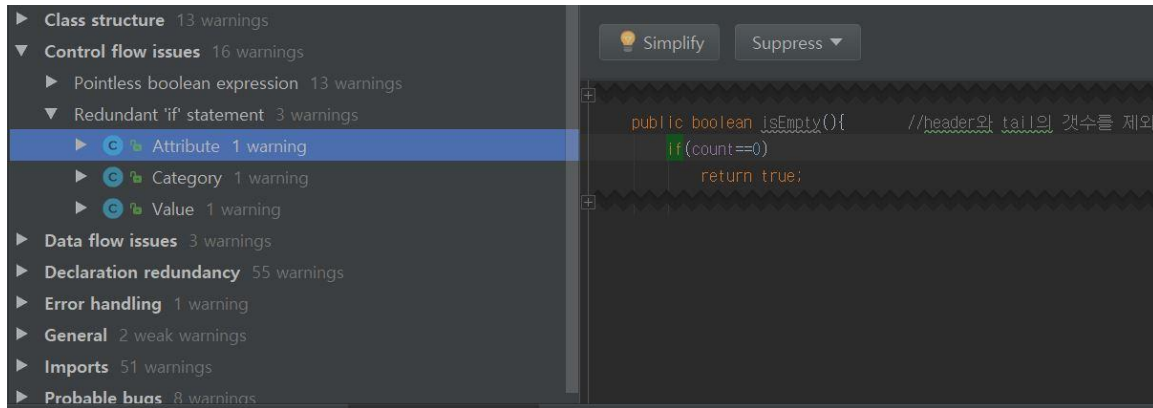
10 problems: Simplify Suppress

```
        addValid(v.getName());  
  
public void deleteDataValue(Value v){  
    if (v.isSingle()==true){  
        deleteSingle(v.getName());  
    }else if (v.isError()==true){  
        deleteError(v.getName());  
    }else if (v.isProperty()==true || v.isIfproperty() == true){  
        deleteProperty(v);  
    }else if (v.isValid()==true){  
        deleteValid(v.getName());  
    }
```

If 문의 조건으로 '변수 == true' 라고 표현

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool



if문 필요 x

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool

The screenshot shows the IntelliJ IDEA interface with the 'Inspection Results' panel on the left. The 'Data flow issues' section is expanded to show 'Redundant local variable' warnings. The 'Calculator' class is selected, showing a warning: 'Local variable 'result' is redundant'. The code editor displays the following code:

```
int result = external_count * internal_count + single_and_error;
return result;
```

The screenshot shows the IntelliJ IDEA interface with the 'Inspection Results' panel on the left. The 'Data flow issues' section is expanded to show 'Redundant local variable' warnings. The 'DataController' class is selected, showing a warning: 'Local variable 'temp' is redundant'. The code editor displays the following code:

```
public int getCategoryIndex(String categoryName){
    int temp = (headerCategory.whereCategory(headerCategory, categoryName)).getCategoryIndex();
    return temp;
}
```

The screenshot shows the IntelliJ IDEA interface with the 'Inspection Results' panel on the left. The 'Data flow issues' section is expanded to show 'Redundant local variable' warnings. The 'Value' class is selected, showing a warning: 'Local variable 'temp' is redundant'. The code editor displays the following code:

```
public Value remove(Value p){
    Value temp = p; //현재 p의 값을 저장한다.
    p.prev.next = p.next; //p의 전노드가 p의 다음 노드를 가리키게 한다.
}
```

Local 변수 선언 필요 x

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool

Class structure 13 warnings
Control flow issues 16 warnings
Data flow issues 3 warnings
Declaration redundancy 55 warnings
 Empty method 5 warnings
 MainController 1 warning
 The method is empty
 ValuePane 4 warnings
 Method can be void 2 warnings
 Unused declaration 48 warnings
Error handling 1 warning
General 2 weak warnings
Imports 51 warnings

```
//view 용 메소드  
public void displayCategoryList()  
{  
}  
public void printList()  
{  
}
```

Class structure 13 warnings
Control flow issues 16 warnings
Data flow issues 3 warnings
Declaration redundancy 55 warnings
 Empty method 5 warnings
 MainController 1 warning
 ValuePane 4 warnings
 The method is empty
 The method is empty
 The method is empty
 The method is empty
 Method can be void 2 warnings
 Unused declaration 48 warnings

```
@Override  
public void keyTyped(KeyEvent e) {  
}  
@Override  
public void keyReleased(KeyEvent e) {  
}
```

Class structure 13 warnings
Control flow issues 16 warnings
Data flow issues 3 warnings
Declaration redundancy 55 warnings
 Empty method 5 warnings
 MainController 1 warning
 ValuePane 4 warnings
 The method is empty
 The method is empty
 The method is empty
 The method is empty
 Method can be void 2 warnings
 Unused declaration 48 warnings

```
@Override  
public void keyTyped(KeyEvent e) {  
}  
@Override  
public void keyReleased(KeyEvent e) {  
}
```

메소드가 비어있음

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool

The screenshot shows the IntelliJ IDEA interface with a warning list on the left and a code editor on the right. The warning list is expanded to show 'Attribute 1 warning' under the 'Method can be void' category. The code editor shows the following code:

```
public String remove(Attribute p){  
    String val = p.name; //현재 p의 값
```

Buttons for 'Make method 'void'' and 'Suppress' are visible above the code editor.

The screenshot shows the IntelliJ IDEA interface with a warning list on the left and a code editor on the right. The warning list is expanded to show 'Category 1 warning' under the 'Method can be void' category. The code editor shows the following code:

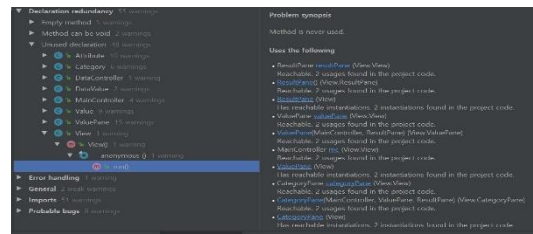
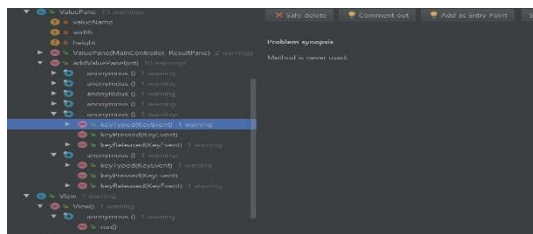
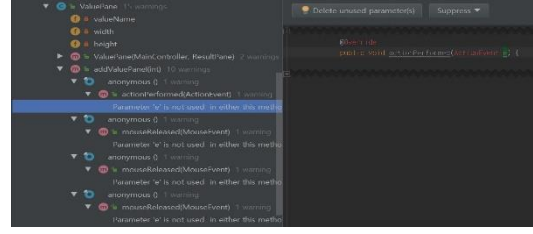
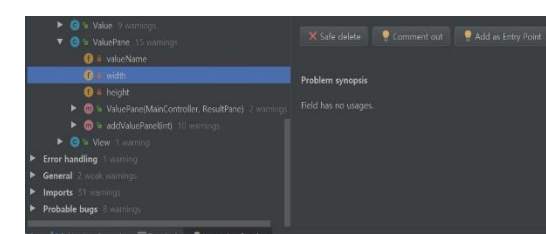
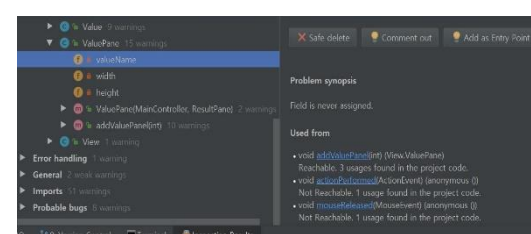
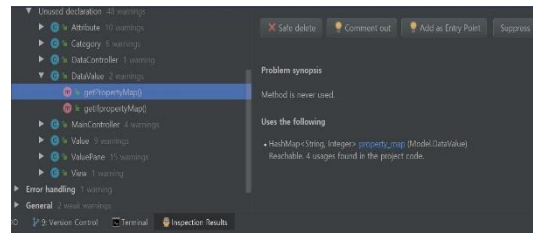
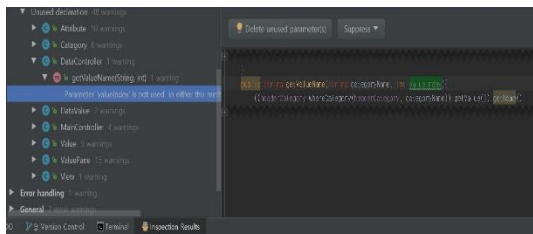
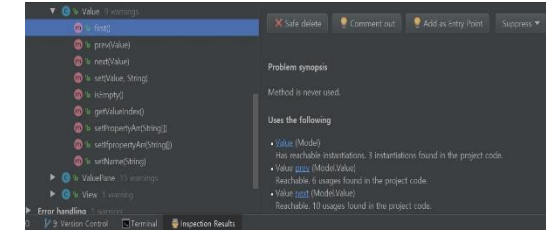
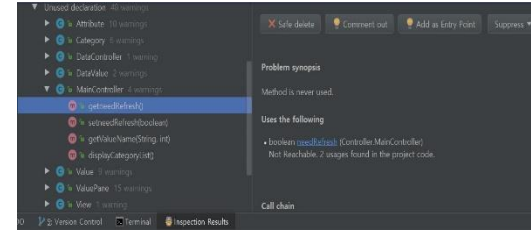
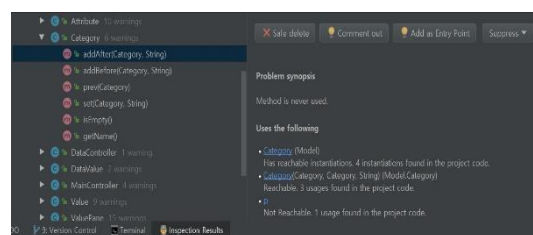
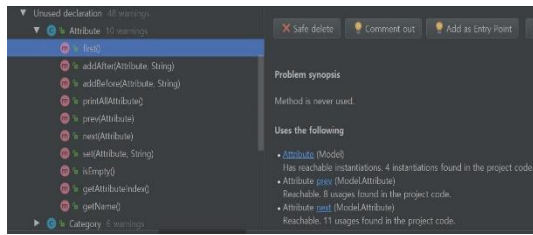
```
public String remove(Category p){  
    String val = p.name; //현재 p의 값을 저장한
```

Buttons for 'Make method 'void'' and 'Suppress' are visible above the code editor.

메소드 Return형 적을 필요 x

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool



메소드/ 인자가 사용되지 않음

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool

2 problems: [View duplicates like this](#) [Navigate to duplicate](#)

```
ifProperty[valueIndex].addKeyListener(new KeyListener(){
    @Override
    public void keyTyped(KeyEvent e){
    }
    @Override
    public void keyPressed(KeyEvent e){
        if(e.getKeyCode()==KeyEvent.VK_ENTER){//
            String s= ifProperty[valueIndex].getText();
            if(icount[valueIndex]<5){
                tempName[valueIndex][icount[valueIndex]] = s;
                ifPropertyCombo[valueIndex].addItem(s);
                icount[valueIndex]++;
            }else{
                //H}
            }
            //System.out.println(s);
        }
    }
    @Override
    public void keyReleased(KeyEvent e){
    }
});
```

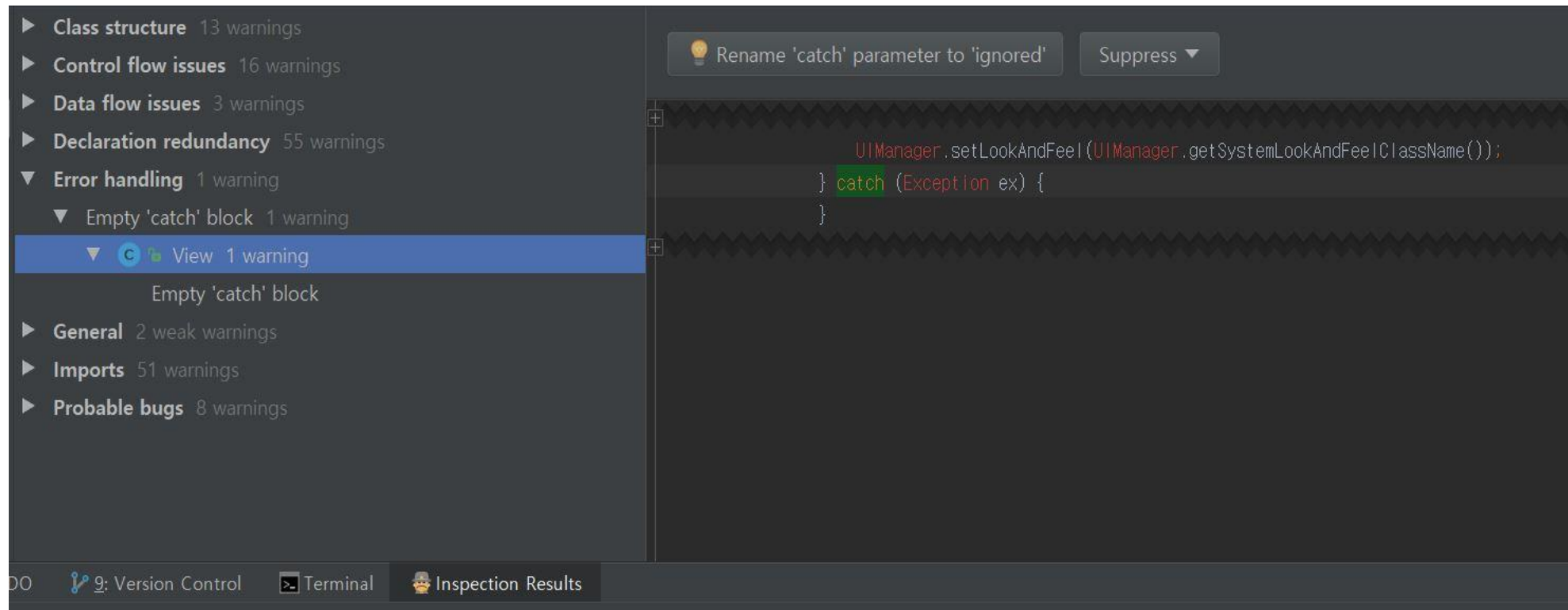
2 problems: [View duplicates like this](#) [Navigate to duplicate](#)

```
//textField
property[valueIndex].addKeyListener(new KeyListener(){
    @Override
    public void keyTyped(KeyEvent e){
    }
    @Override
    public void keyPressed(KeyEvent e){
        if(e.getKeyCode()==KeyEvent.VK_ENTER){//
            String s= property[valueIndex].getText();
            if(pcount[valueIndex]<5){
                tempName[valueIndex][pcount[valueIndex]] = s;
                propertyCombo[valueIndex].addItem(s);
                pcount[valueIndex]++;
            }else{
                //H}
            }
            //System.out.println(s);
        }
    }
    @Override
    public void keyReleased(KeyEvent e){
    }
});
```

코드 중복

Static Analysis Lv1. IntelliJ

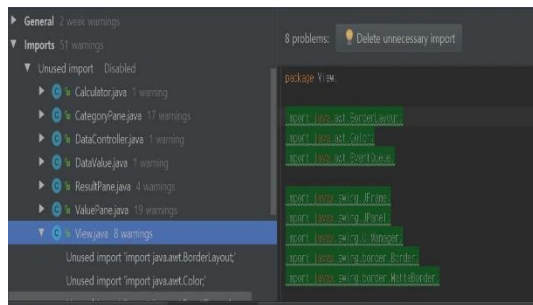
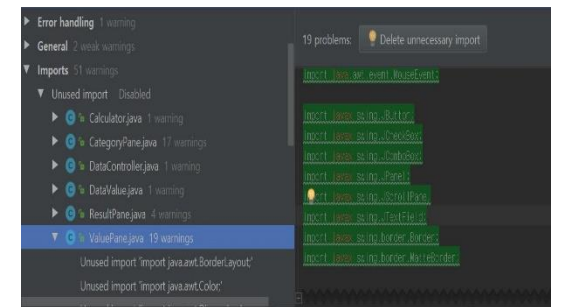
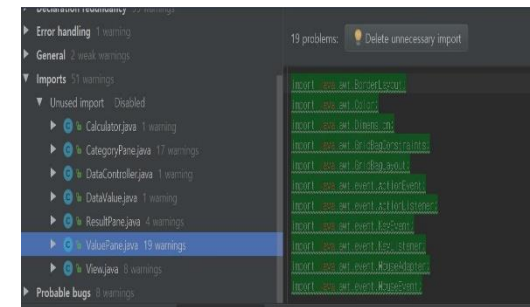
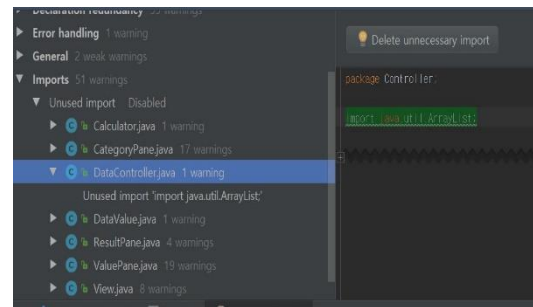
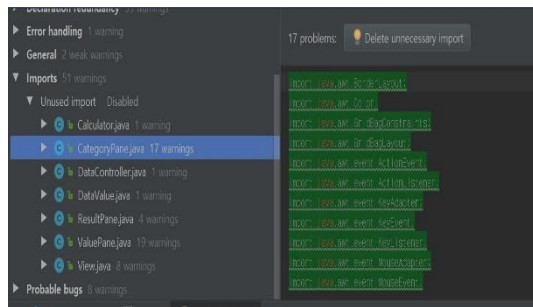
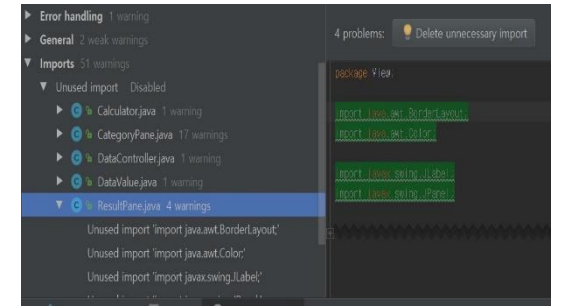
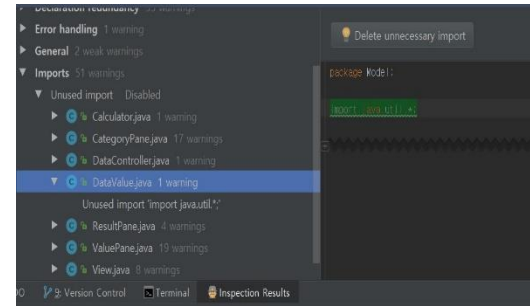
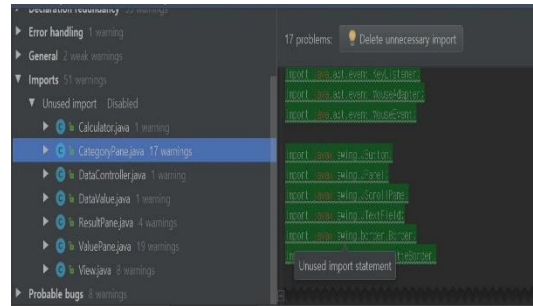
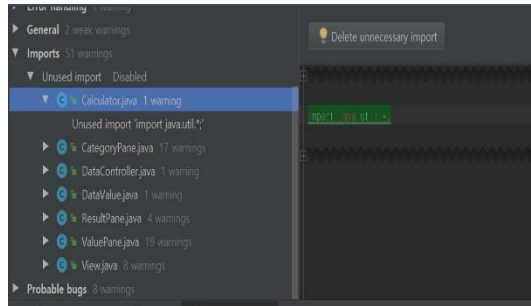
T5. SSS CPT Tool



블록이 비어있음

Static Analysis Lv1. IntelliJ

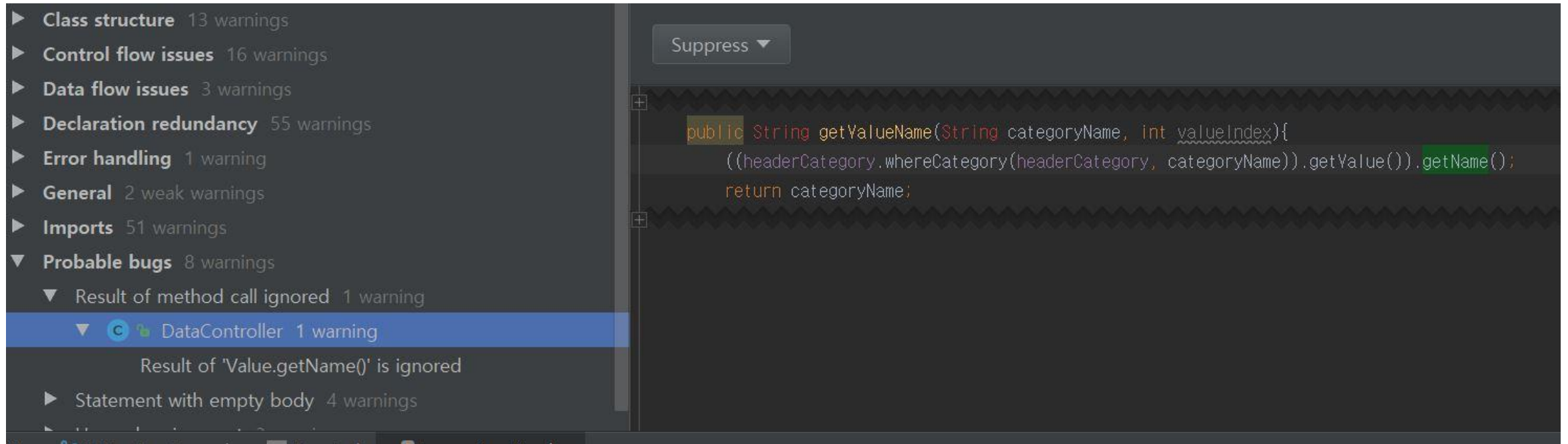
T5. SSS CPT Tool



Import한 것들이 사용되지 않음

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool



The screenshot displays the IntelliJ IDEA interface with a static analysis warning. On the left, the 'Probable bugs' section is expanded to show 'DataController' with 1 warning: 'Result of method call ignored'. The main editor shows the following code snippet:

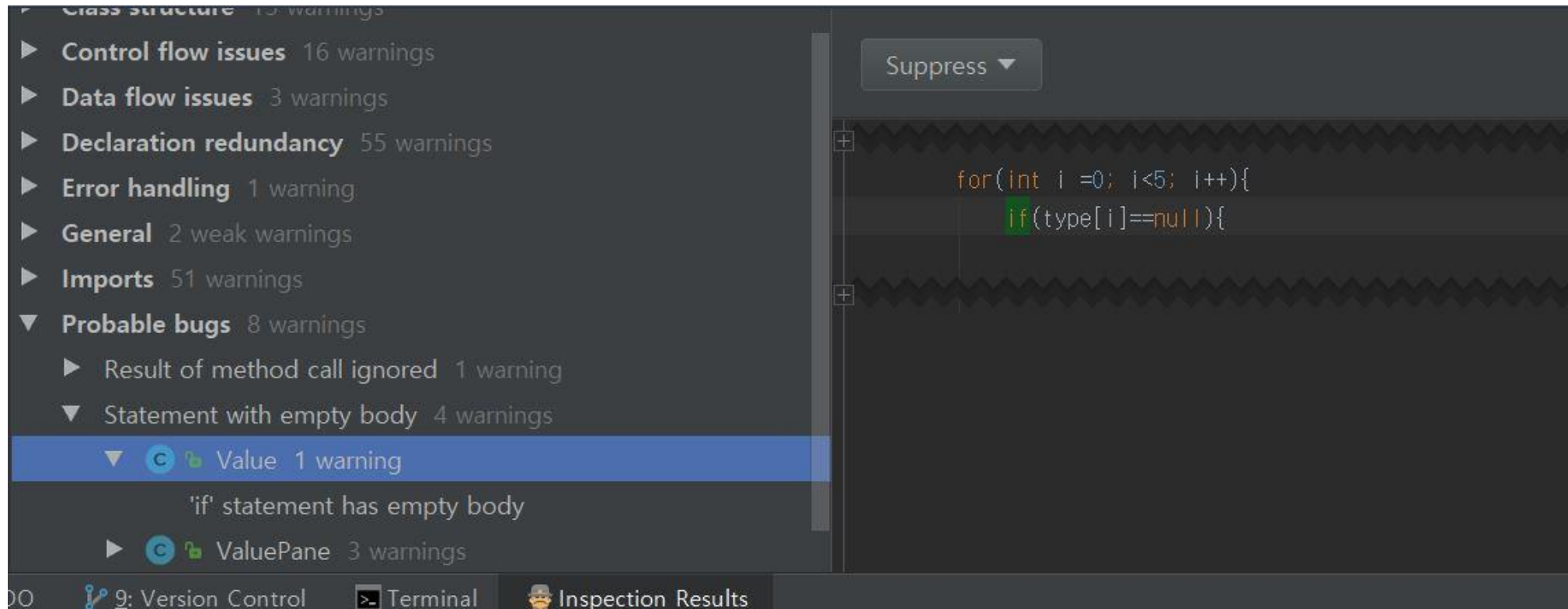
```
public String getValueName(String categoryName, int valueIndex){  
    ((headerCategory.whereCategory(headerCategory, categoryName)).getValue()).getName();  
    return categoryName;  
}
```

The warning message 'Result of 'Value.getName()' is ignored' is visible below the code. A 'Suppress' button is located above the code.

getname 호출 후 사용 x

Static Analysis Lv1. IntelliJ

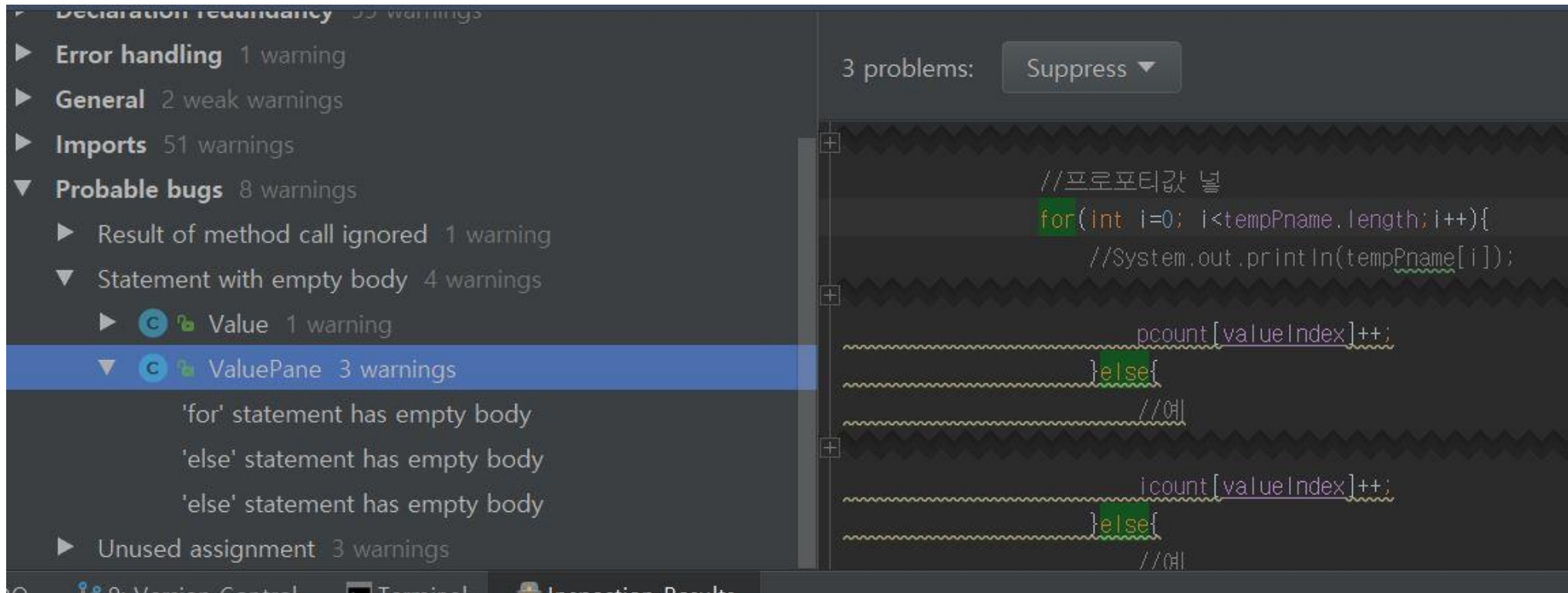
T5. SSS CPT Tool



If문의 body가 없음

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool



For / else문의 body가 없음

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool

The screenshot displays the IntelliJ IDEA interface with the 'Declaration redundancy' warning expanded. The left sidebar shows a tree view of warnings, with 'Declaration redundancy' (55 warnings) selected. The right pane shows the code snippet with two problems highlighted: 'Remove redundant initializer' and 'Suppress'. The code snippet is as follows:

```
//sort
int i = 0, j = 0;
while (i < category_list.size()){

    int result = 0;
    int i = 0;
    if ((i = dv.getValidCount()) != 0){
```

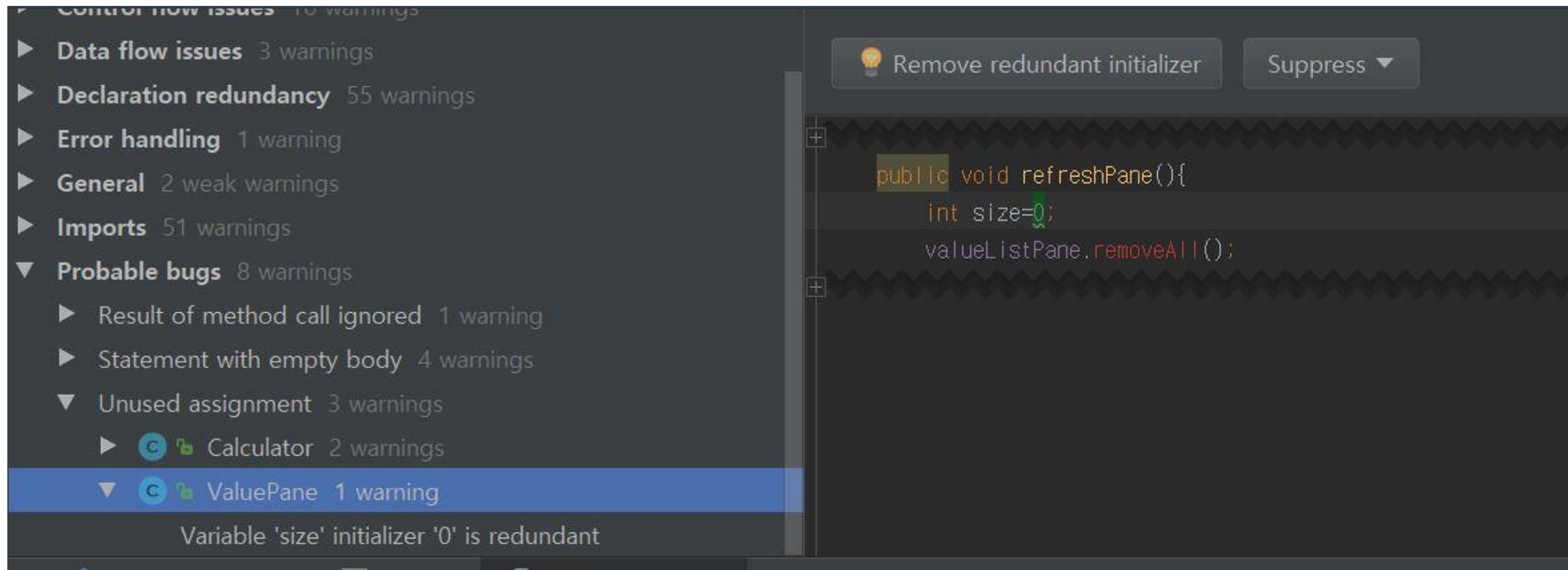
The warning details for 'Declaration redundancy' are:

- Variable 'j' initializer '0' is redundant
- Variable 'i' initializer '0' is redundant

j / i 초기화 후 사용 x

Static Analysis Lv1. IntelliJ

T5. SSS CPT Tool



The screenshot displays the IntelliJ IDEA interface with a static analysis warning. On the left, the 'Probable bugs' section is expanded to show 'ValuePane' with 1 warning. The warning message is 'Variable 'size' initializer '0' is redundant'. On the right, the code editor shows a method named 'refreshPane()' with the following code:

```
public void refreshPane(){  
    int size=0;  
    valueListPane.removeAll();  
}
```

At the top of the code editor, there are two buttons: 'Remove redundant initializer' and 'Suppress'. The warning message is also visible at the bottom of the screenshot.

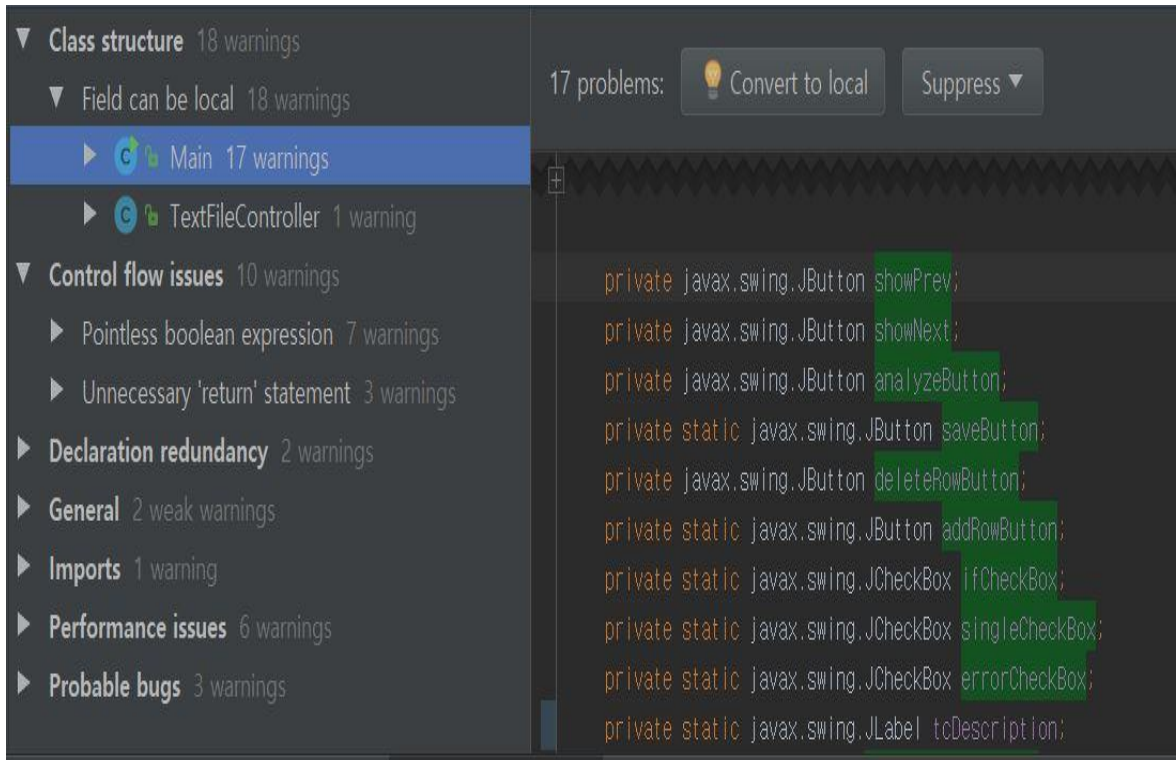
Size 초기화 후 사용 x

Team 8

Feesual CPT

Static Analysis Lv1. IntelliJ

T8. Feesual CPT

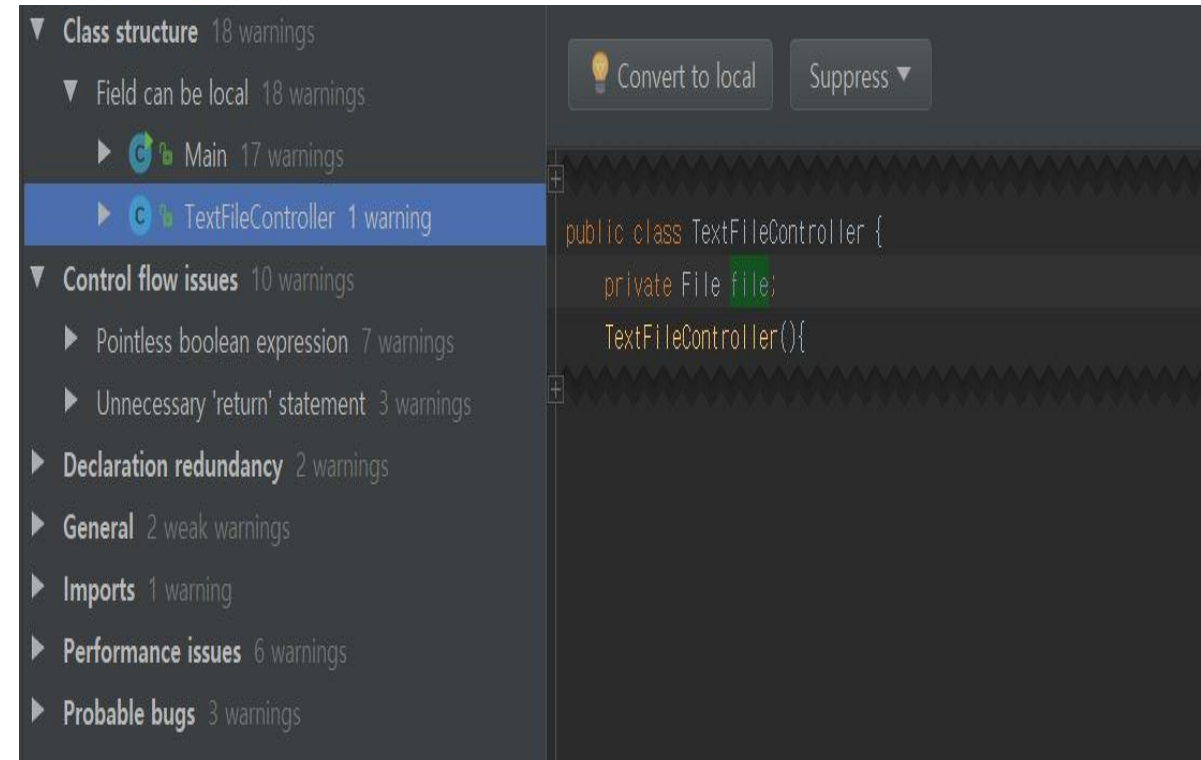


Class structure 18 warnings

- Field can be local 18 warnings
 - Main 17 warnings
 - TextFileController 1 warning
- Control flow issues 10 warnings
 - Pointless boolean expression 7 warnings
 - Unnecessary 'return' statement 3 warnings
- Declaration redundancy 2 warnings
- General 2 weak warnings
- Imports 1 warning
- Performance issues 6 warnings
- Probable bugs 3 warnings

17 problems:

```
private javax.swing.JButton showPrev;  
private javax.swing.JButton showNext;  
private javax.swing.JButton analyzeButton;  
private static javax.swing.JButton saveButton;  
private javax.swing.JButton deleteRowButton;  
private static javax.swing.JButton addRowButton;  
private static javax.swing.JCheckBox ifCheckBox;  
private static javax.swing.JCheckBox singleCheckBox;  
private static javax.swing.JCheckBox errorCheckBox;  
private static javax.swing.JLabel tcDescription;
```



Class structure 18 warnings

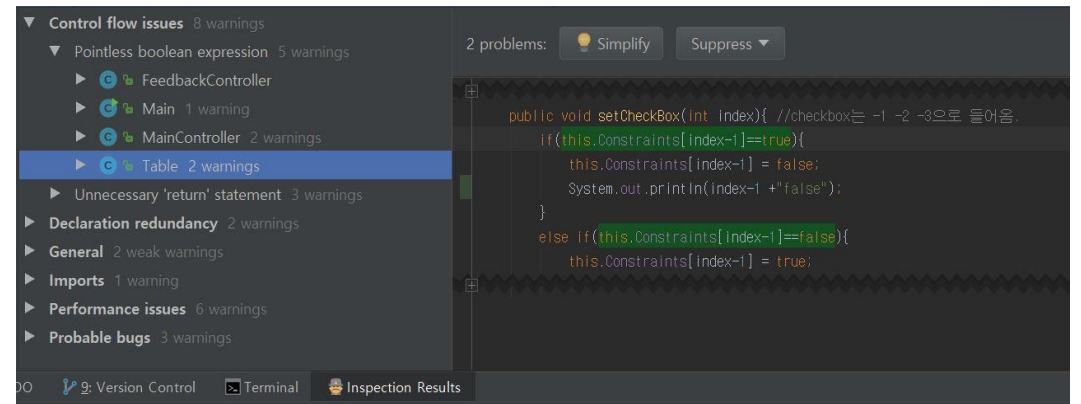
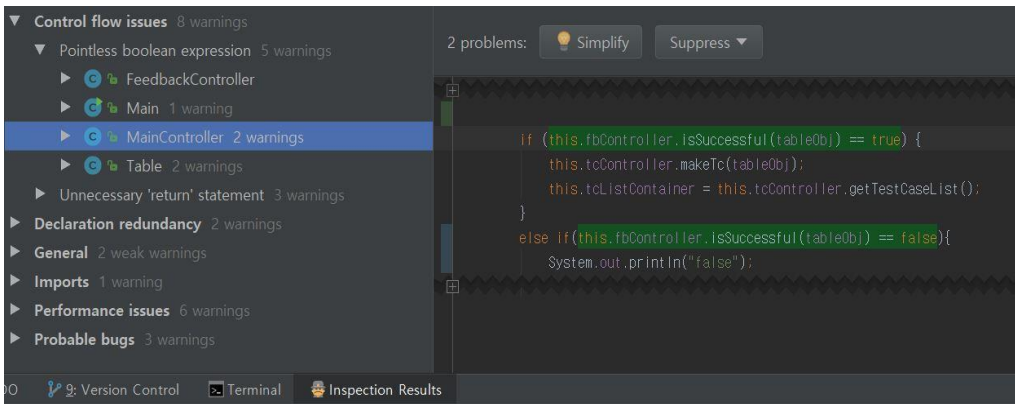
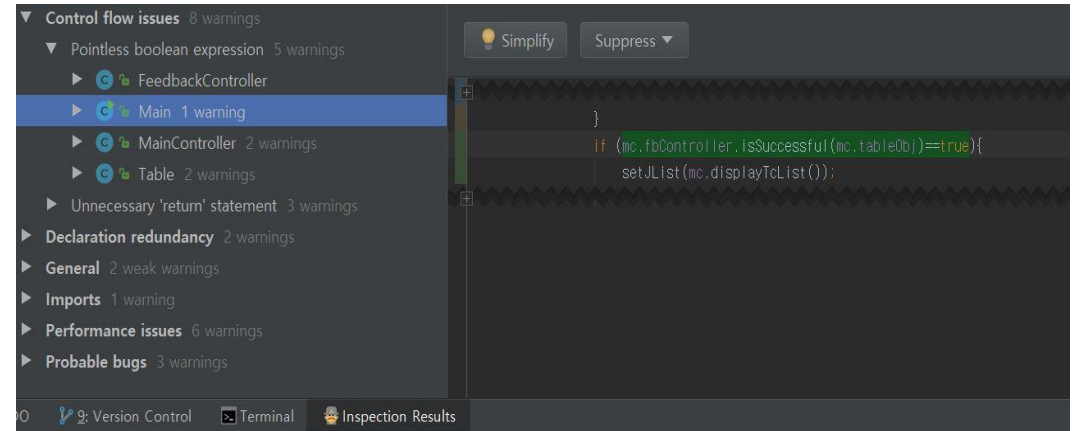
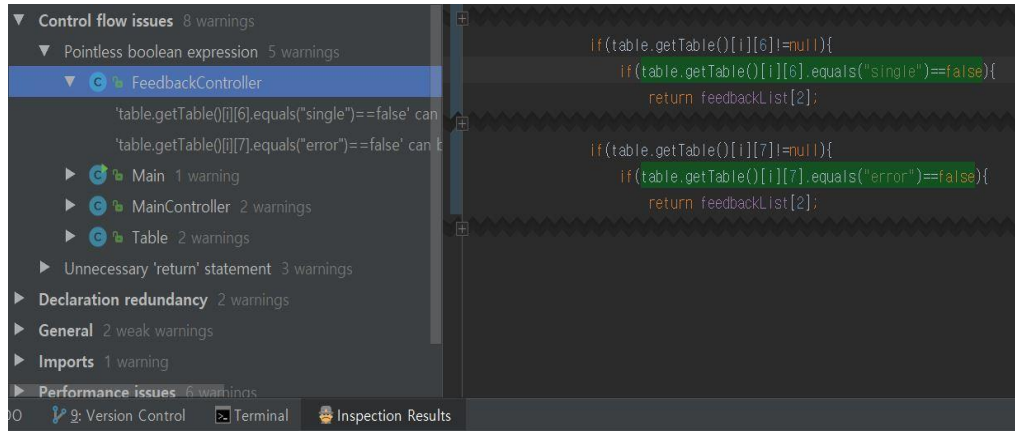
- Field can be local 18 warnings
 - Main 17 warnings
 - TextFileController 1 warning
- Control flow issues 10 warnings
 - Pointless boolean expression 7 warnings
 - Unnecessary 'return' statement 3 warnings
- Declaration redundancy 2 warnings
- General 2 weak warnings
- Imports 1 warning
- Performance issues 6 warnings
- Probable bugs 3 warnings

```
public class TextFileController {  
    private File file;  
    TextFileController(){  
    }  
}
```

변수가 하나의 메소드에서만 사용 (지역변수 추천)

Static Analysis Lv1. IntelliJ

T8. Feesual CPT



‘ == true / false ‘ 사용 (단순히 expression 추천)

Static Analysis Lv1. IntelliJ

T8. Feesual CPT

The screenshot displays the IntelliJ IDEA interface with the following elements:

- Left Panel (Inspection Results):**
 - Control flow issues: 8 warnings
 - Pointless boolean expression: 5 warnings
 - Unnecessary 'return' statement: 3 warnings
 - TestCaseController: 3 warnings
 - 'return' is unnecessary as the last statement in a 'void' method
 - 'return' is unnecessary as the last statement in a 'void' method
 - 'return' is unnecessary as the last statement in a 'void' method
 - Declaration redundancy: 2 warnings
 - General: 2 weak warnings
 - Imports: 1 warning
 - Performance issues: 6 warnings
 - Probable bugs: 3 warnings
- Right Panel (Code Editor):**
 - 3 problems: Remove redundant statement, Suppress
 - Code snippet showing three instances of `return;` statements at the end of method blocks, each highlighted in green to indicate the inspection results.
- Bottom Panel:** Version Control, Terminal, and Inspection Results tabs.

불필요한 return문 사용

Static Analysis Lv1. IntelliJ

T8. Feesual CPT

The screenshot shows the IntelliJ IDEA interface with a warning for an unused parameter. The left sidebar shows a tree view of warnings, with 'Unused declaration' expanded to show 'FeedbackController' and 'getFeedback(ArrayList<TestCase>)' selected. The main editor shows the code for the 'getFeedback' method, with a tooltip indicating that the parameter 'tcList' is not used. The tooltip text is: "Parameter 'tcList' is not used in either this method or an enclosing method." The code snippet is:

```
public Feedback getFeedback(ArrayList<TestCase> tcList) {  
    return feedbackList[6];  
}
```

The screenshot shows the IntelliJ IDEA interface with a warning for an unused constructor. The left sidebar shows a tree view of warnings, with 'Unused declaration' expanded to show 'SingleTestCase()' selected. The right sidebar shows the 'Problem synopsis' for the warning, which states: "Constructor is never used." The 'Uses the following' section lists 'SingleTestCase()' with the note: "Has reachable instantiations. 2 instantiations found in the project code." The bottom status bar shows 'Inspection Results'.

메소드 / 생성자 사용 x

Static Analysis Lv1. IntelliJ

T8. Feesual CPT

The screenshot displays the IntelliJ IDEA interface with the following components:

- Left Panel (Inspection Results):**
 - Control flow issues: 8 warnings
 - Declaration redundancy: 2 warnings
 - General: 2 weak warnings
 - Duplicated Code: 2 weak warnings
 - TestCaseController.java: 2 weak warnings
 - Found duplicated code in this file
 - Found duplicated code in this file
 - Imports: 1 warning
 - Performance issues: 6 warnings
 - Probable bugs: 3 warnings
- Right Panel (Code Editor):**
 - 2 problems: View duplicates like this, Navigate to duplicate
 - Code snippet showing duplicated code blocks:

```
}else if(propertyTc.size() != 0){  
    for(int i = 0 ; i < propertyTc.size() ; i++){  
        TestCase tc = new TestCase();  
        ArrayList<SingleTestCase> temp = propertyTc.get(i).getSingleTcList();  
        for(int j = 0 ; j < temp.size() ; j++){  
            tc.add(temp.get(j).getRefNum(), temp.get(j).getDesc());  
        }  
        tcList.add(tc);  
    }  
}  
}else if(nonPropertyTc.size() != 0){  
    for(int i = 0 ; i < nonPropertyTc.size() ; i++){
```
- Bottom Panel:** 9: Version Control, Terminal, Inspection Results

동일한 코드 중복

Static Analysis Lv1. IntelliJ

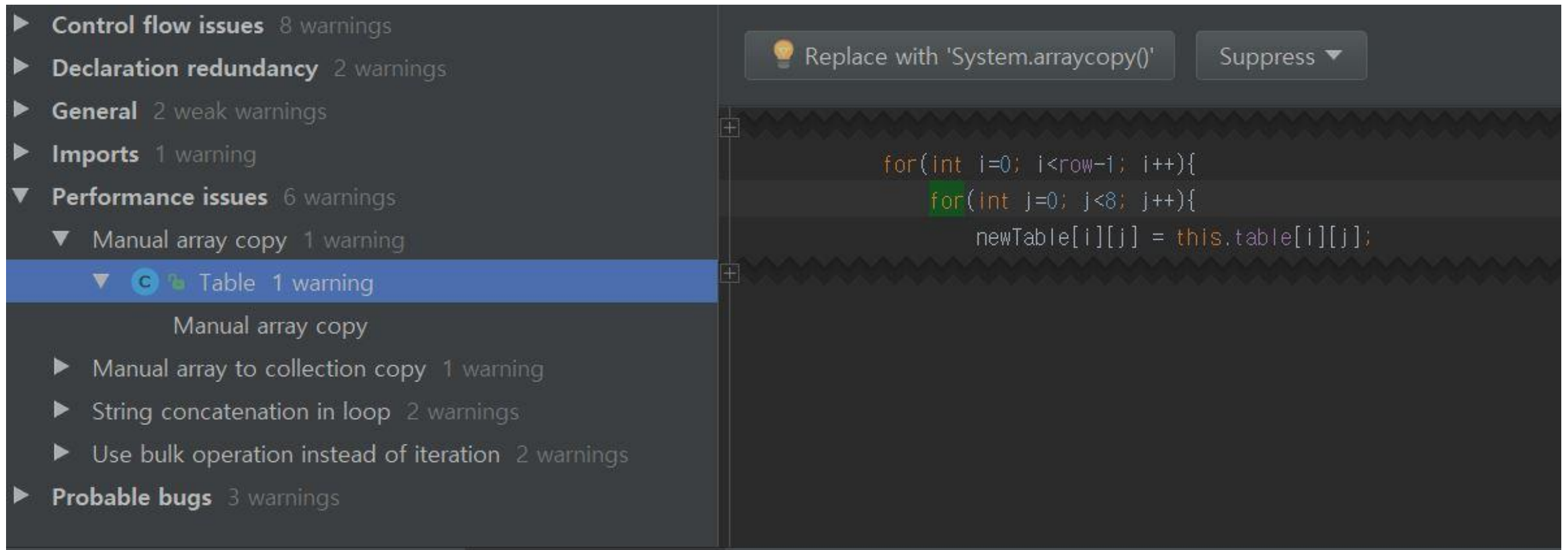
T8. Feesual CPT

The screenshot displays the IntelliJ IDEA interface. On the left, the 'Inspection Results' tool window is open, showing a tree view of warnings. The 'Imports' category is expanded, and a warning for 'Table.java' is selected, indicating an 'Unused import'. The main editor area shows the code for 'Table.java', with the line `import javax.swing.JTable;` highlighted in green. A tooltip above the code reads 'Delete unnecessary import'. The bottom status bar shows the 'Inspection Results' tab is active.

Import한 내용 사용 x

Static Analysis Lv1. IntelliJ

T8. Feesual CPT



The screenshot displays the IntelliJ IDEA interface. On the left, the 'Warnings' panel is open, showing a tree view of issues. Under 'Performance issues', 'Manual array copy' is selected, with a sub-item 'Table' (1 warning) highlighted. The main editor shows a nested loop:

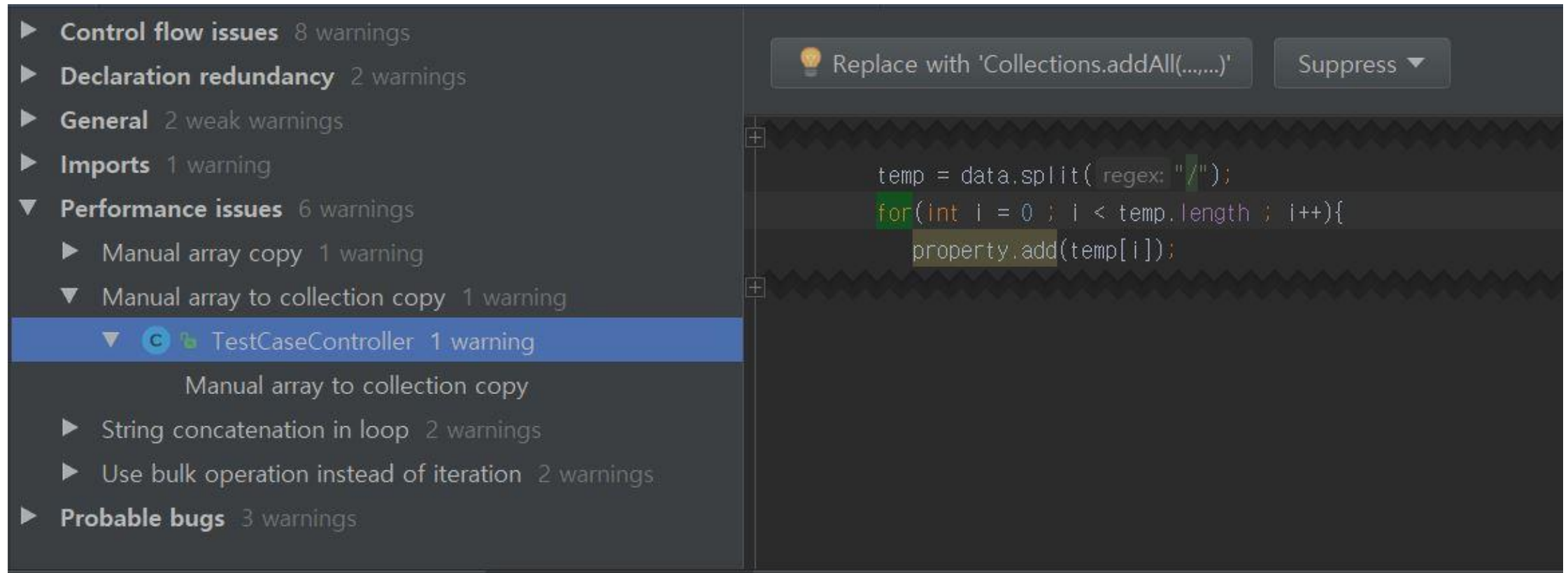
```
for(int i=0; i<row-1; i++){  
    for(int j=0; j<8; j++){  
        newTable[i][j] = this.table[i][j];  
    }  
}
```

A tooltip above the code suggests 'Replace with 'System.arraycopy()'' and includes a 'Suppress' button.

단순 array 복사 (arraycopy 추천)

Static Analysis Lv1. IntelliJ

T8. Feesual CPT



The screenshot shows the IntelliJ IDEA static analysis interface. On the left, a tree view displays the following categories and counts:

- ▶ **Control flow issues** 8 warnings
- ▶ **Declaration redundancy** 2 warnings
- ▶ **General** 2 weak warnings
- ▶ **Imports** 1 warning
- ▼ **Performance issues** 6 warnings
 - ▶ Manual array copy 1 warning
 - ▼ Manual array to collection copy 1 warning
 - ▼ **TestCaseController** 1 warning
 - Manual array to collection copy
 - ▶ String concatenation in loop 2 warnings
 - ▶ Use bulk operation instead of iteration 2 warnings
- ▶ **Probable bugs** 3 warnings

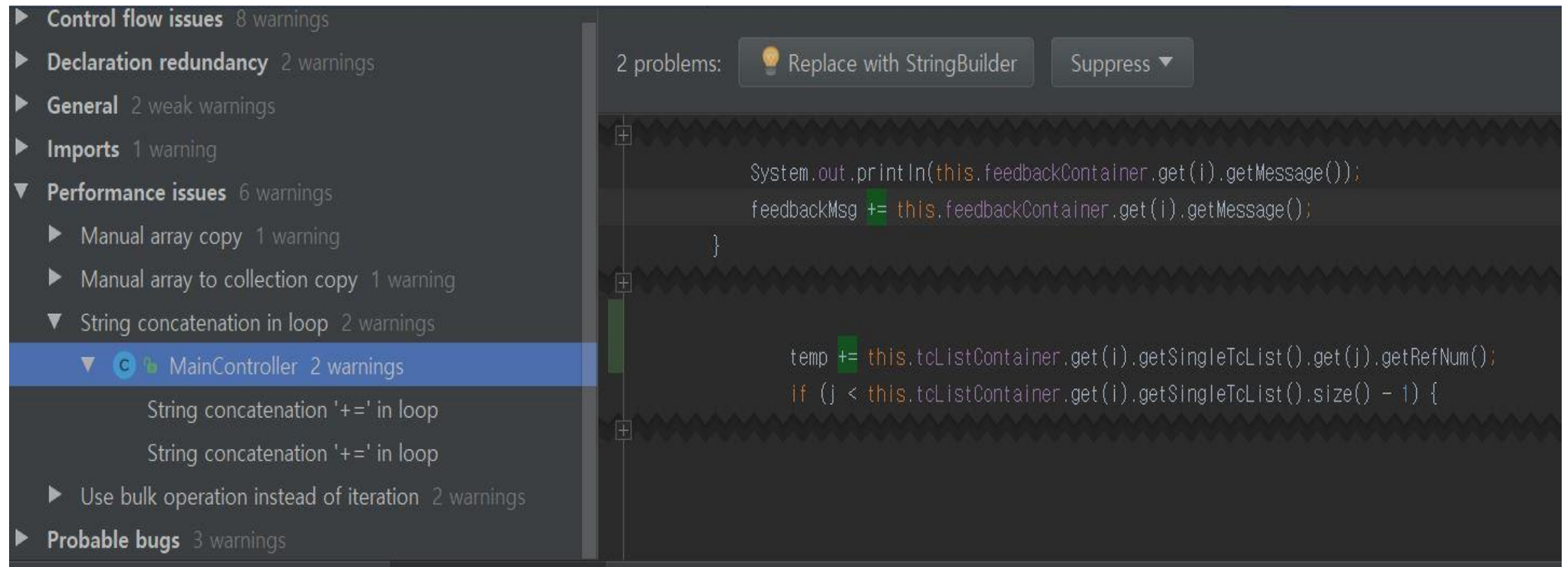
On the right, a code snippet is shown with a warning icon and a lightbulb icon. The warning text is "Replace with 'Collections.addAll(...,....)'" and there is a "Suppress" button. The code snippet is:

```
temp = data.split( regex: "/" );  
for( int i = 0 ; i < temp.length ; i++ ) {  
    property.add( temp[ i ] );  
}
```

Collection에 반복문으로 하나씩 array 내용 삽입
(collection.addAll 메소드 추천)

Static Analysis Lv1. IntelliJ

T8. Feesual CPT



The screenshot displays the IntelliJ IDEA interface with a static analysis tool. On the left, a sidebar lists various warning categories: Control flow issues (8 warnings), Declaration redundancy (2 warnings), General (2 weak warnings), Imports (1 warning), Performance issues (6 warnings), and Probable bugs (3 warnings). The 'Performance issues' section is expanded, showing 'String concatenation in loop' (2 warnings) and 'MainController' (2 warnings). The 'MainController' entry is selected, showing two warnings: 'String concatenation '+' in loop'.

The main editor area shows two code snippets. The first snippet is a loop where a string is concatenated to a variable: `feedbackMsg += this.feedbackContainer.get(i).getMessage();`. A warning icon is present next to the '+' operator. The second snippet is a nested loop where a variable is incremented: `temp += this.tcListContainer.get(i).getSingleTcList().get(j).getRefNum();`. A warning icon is present next to the '+' operator. Above the code, there are two buttons: 'Replace with StringBuilder' and 'Suppress'.

반복문으로 String 복사

Static Analysis Lv1. IntelliJ

T8. Feesual CPT

2 problems: Replace with bulk method call Suppress ▾

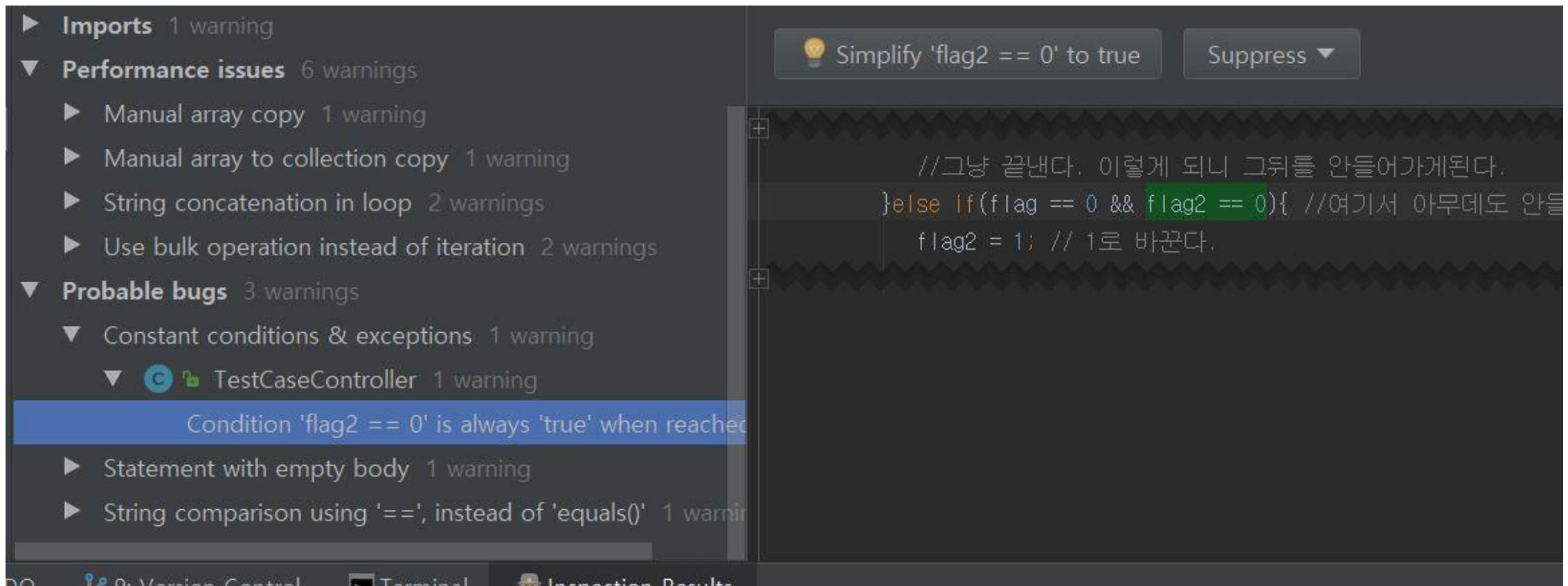
```
for(int i = 0 ; i < temp.length ; i++){  
    property.add(temp[i]);  
}
```

```
for(int j = 0 ; j < property.size() ; j++){  
    tempProperty.add(property.get(j));  
} // j < number 하는 이유가 혹시나 밑의 데이터가 이걸 침범할 경우를 여
```

Collection 복사 메소드 (collection.addAll 추천)

Static Analysis Lv1. IntelliJ

T8. Feesual CPT



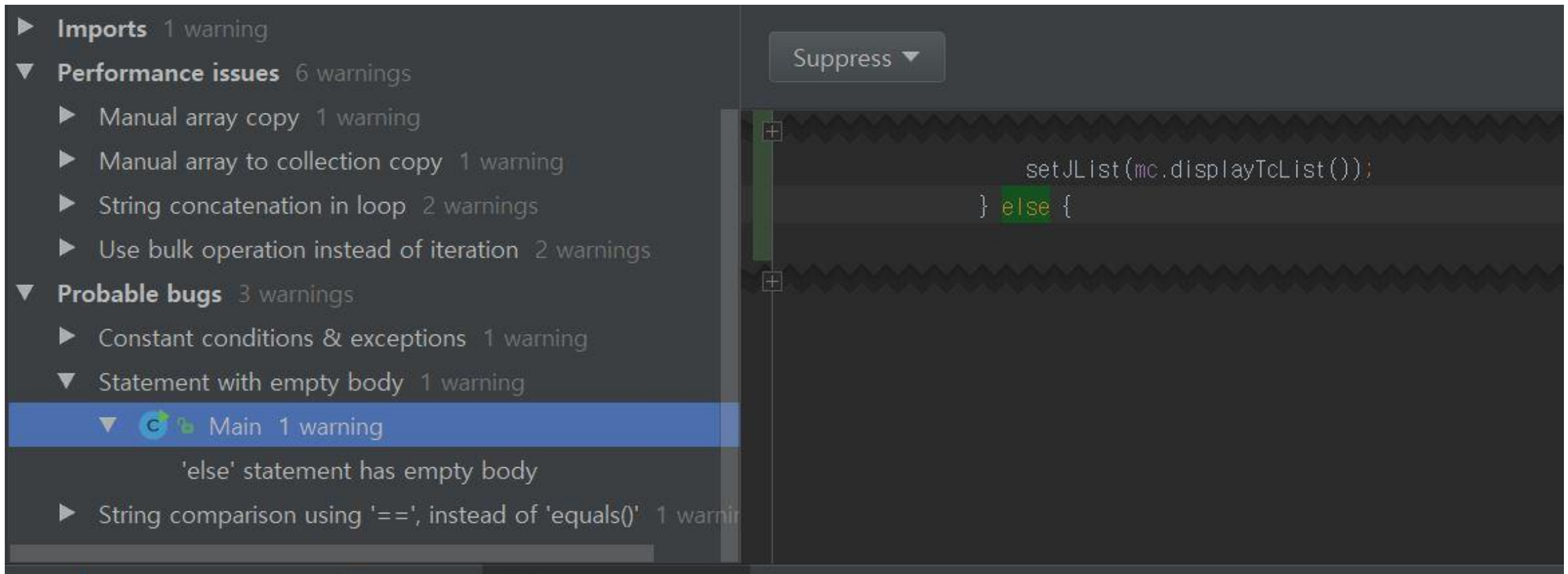
The screenshot shows the IntelliJ IDEA interface. On the left, the 'Inspection Results' tool window is open, displaying a tree view of warnings. Under 'Probable bugs', the warning 'Condition 'flag2 == 0' is always 'true' when reached' is highlighted. On the right, the code editor shows a Java snippet with a comment in Korean: '//그냥 끝낸다. 이렇게 되니 그뒤를 안들어가게된다.' and an 'else if' condition: 'else if(flag == 0 && flag2 == 0)'. The condition 'flag2 == 0' is highlighted in green. Above the code, there is a lightbulb icon and the text 'Simplify 'flag2 == 0' to true', along with a 'Suppress' button.

```
    //그냥 끝낸다. 이렇게 되니 그뒤를 안들어가게된다.  
    }else if(flag == 0 && flag2 == 0){ //여기서 아무데도 안들  
        flag2 = 1; // 1로 바꾼다.
```

Flag2는 이 line에서 항상 true

Static Analysis Lv1. IntelliJ

T8. Feesual CPT



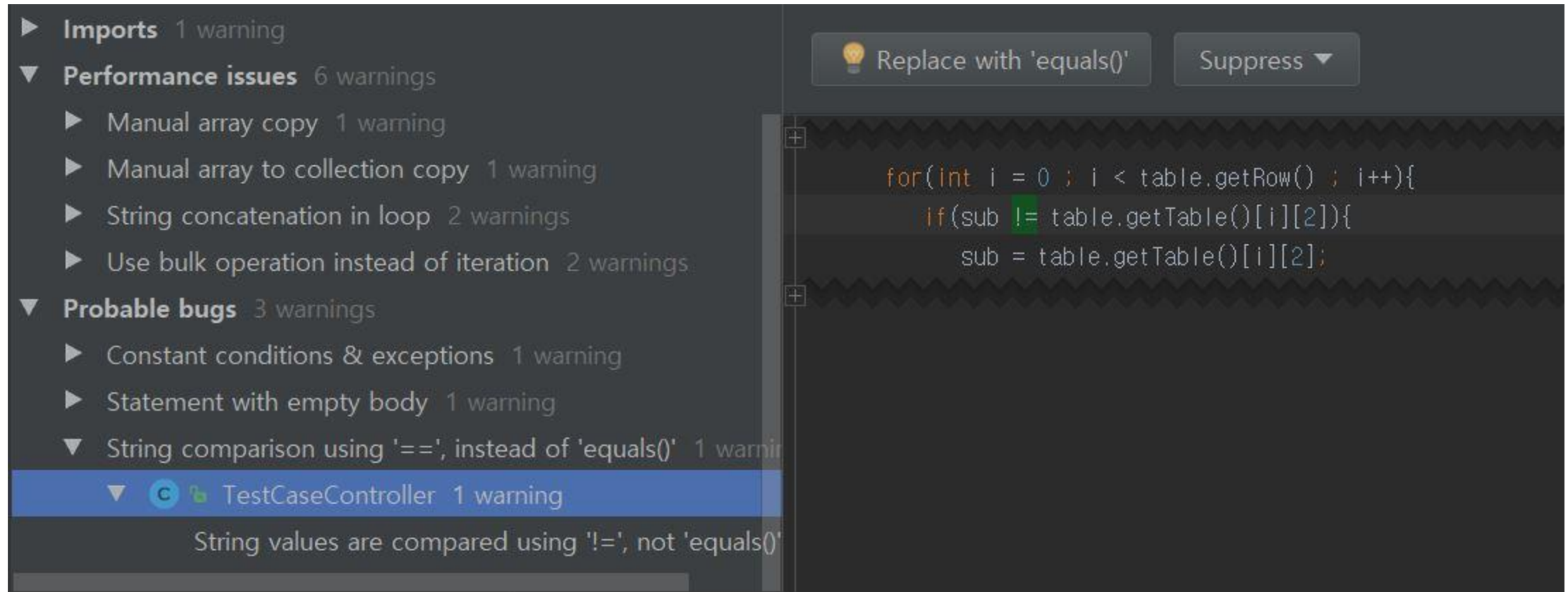
The screenshot displays the IntelliJ IDEA interface. On the left, the 'Performance issues' and 'Probable bugs' sections are visible. Under 'Probable bugs', the 'Main' method is highlighted with a warning: ''else' statement has empty body'. On the right, the code editor shows a snippet of Java code with an 'else' block that is currently empty. A 'Suppress' button is located above the code.

```
setJList(mc.displayTcList());  
} else {  
  
}
```

else문 body가 비어있음

Static Analysis Lv1. IntelliJ

T8. Feesual CPT



The screenshot displays the IntelliJ IDEA interface. On the left, the 'Performance issues' section is expanded, showing a warning for 'String comparison using !=, instead of equals()' under the 'Probable bugs' category. The warning is highlighted in blue. On the right, a code snippet is shown with a lightbulb icon and a 'Replace with equals()' button. The code snippet is as follows:

```
for(int i = 0 ; i < table.getRow() ; i++){  
    if(sub != table.getTable()[i][2]){  
        sub = table.getTable()[i][2];  
    }  
}
```

String 비교문에 != / == 사용 (equals()함수 사용)

Static Analysis

Lv.2_Eclipse metrics

Team 5

SSS CPT

Static Analysis Lv2_Eclipse metrics

T5. SSS CPT Tool

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
▼ McCabe Cyclomatic Complexity (avg/max per method)		1.552	1.769	12	/ThreeS/src/Controller/Calculator.java	internal_calculate
▼ src		1.552	1.769	12	/ThreeS/src/Controller/Calculator.java	internal_calculate
▼ Controller		1.719	2.465	12	/ThreeS/src/Controller/Calculator.java	internal_calculate
▼ Calculator.java		8.667	3.399	12	/ThreeS/src/Controller/Calculator.java	internal_calculate
▼ Calculator		8.667	3.399	12	/ThreeS/src/Controller/Calculator.java	internal_calculate
internal_calculate	12					
calculate	10					
findKey	4					
> MainController.java		1	0	1	/ThreeS/src/Controller/MainController.java	MainController
> DataController.java		1	0	1	/ThreeS/src/Controller/DataController.java	DataController
▼ Model		1.551	1.592	12	/ThreeS/src/Model/Value.java	Value
▼ Value.java		1.576	1.939	12	/ThreeS/src/Model/Value.java	Value
▼ Value		1.576	1.939	12	/ThreeS/src/Model/Value.java	Value
Value	12					
whereValue	4					
first	2					

Calulator.internal calculate 메소드
Value.Value 생성자



Cyclomatic Complexity가 높게 나옴

Static Analysis Lv2_Eclipse metrics

T5. SSS CPT Tool

▼ Number of Parameters (avg/max per method)		0.958	1.09	6	/ThreeS/src/Model/Value.java	Value
▼ src		0.958	1.09	6	/ThreeS/src/Model/Value.java	Value
▼ Model		0.837	1.007	6	/ThreeS/src/Model/Value.java	Value
▼ Value.java		0.848	1.234	6	/ThreeS/src/Model/Value.java	Value
▼ Value		0.848	1.234	6	/ThreeS/src/Model/Value.java	Value
Value	6					
addLast	4					
whereValue	2					
set	2					
setPrev	1					
setNext	1					
remove	1					
.....	.					

Value의 Private생성자가
private변수 6개에 대한 초기화



Parameter가 많다고 판정

Team 8

Feesual CPT

Static Analysis Lv2_Eclipse metrics

T8. Feesual CPT

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
▼ McCabe Cyclomatic Complexity (avg/max per method)		3.259	6.435	40	/SMA2017_FeesualCPT/src/TestCaseController.java	makeTc
▼ src		3.259	6.435	40	/SMA2017_FeesualCPT/src/TestCaseController.java	makeTc
▼ (default package)		3.259	6.435	40	/SMA2017_FeesualCPT/src/TestCaseController.java	makeTc
▼ TestCaseController.java		10.571	13.468	40	/SMA2017_FeesualCPT/src/TestCaseController.java	makeTc
▼ TestCaseController		10.571	13.468	40	/SMA2017_FeesualCPT/src/TestCaseController.java	makeTc
makeTc	40					
recursive2	20					
checkProperty	5					
recursive1	4					
saveProperty	3					
TestCaseController	1					
getTestCaseList	1					
▼ FeedbackController.java		6.25	8.526	21	/SMA2017_FeesualCPT/src/FeedbackController.java	getFeedback
▼ FeedbackController		6.25	8.526	21	/SMA2017_FeesualCPT/src/FeedbackController.java	getFeedback
getFeedback	21					
isSuccessful	2					
FeedbackController	1					
getFeedback	1					
▼ TextFileController.java		7	6	13	/SMA2017_FeesualCPT/src/TextFileController.java	saveRequest
▼ TextFileController		7	6	13	/SMA2017_FeesualCPT/src/TextFileController.java	saveRequest
saveRequest	13					
TextFileController	1					
> MainController.java		1.7	1.269	5	/SMA2017_FeesualCPT/src/MainController.java	displayTcList
> Main.java		2.167	1.344	4	/SMA2017_FeesualCPT/src/Main.java	getTableData

TestCaseController.makeTc 메소드
TestCaseController.recursive2 메소드
FeedbackController.getFeedback 메소드
TextFileController.saveRequest 메소드



Cyclomatic Complexity가 높게 나옴

Static Analysis Lv2_Eclipse metrics

T8. Feesual CPT

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
▼ Nested Block Depth (avg/max per method)		1.907	1.531	7	/SMA2017_FeesualCPT/src/FeedbackController.java	getFeedback
▼ src		1.907	1.531	7	/SMA2017_FeesualCPT/src/FeedbackController.java	getFeedback
▼ (default package)		1.907	1.531	7	/SMA2017_FeesualCPT/src/FeedbackController.java	getFeedback
▼ FeedbackController.java		2.75	2.487	7	/SMA2017_FeesualCPT/src/FeedbackController.java	getFeedback
▼ FeedbackController		2.75	2.487	7	/SMA2017_FeesualCPT/src/FeedbackController.java	getFeedback
getFeedback	7					
isSuccessful	2					
FeedbackController	1					
getFeedback	1					
▼ TestCaseController.java		3.571	2.129	7	/SMA2017_FeesualCPT/src/TestCaseController.java	makeTc
▼ TestCaseController		3.571	2.129	7	/SMA2017_FeesualCPT/src/TestCaseController.java	makeTc
makeTc	7					
recursive2	6					
checkProperty	4					
recursive1	3					
saveProperty	3					
TestCaseController	1					
getTestCaseList	1					
> TextFileController.java		3	2	5	/SMA2017_FeesualCPT/src/TextFileController.java	saveRequest
> MainController.java		1.5	0.922	4	/SMA2017_FeesualCPT/src/MainController.java	displayTcList

FeedbackController.getFeedback 메소드
TestCaseController.makeTc 메소드
TestCaseController.recursive2 메소드



중첩된 Block이 많음

Static Analysis

Lv.3_Find Bugs

Team 5

SSS CPT

Static Analysis Lv3_Find Bugs

T5. SSS CPT Tool

The screenshot displays the Eclipse IDE interface with the following components:

- Top Bar:** workspace - FindBugs - ThreeS/src/Controller/DataController.java - Eclipse
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Project Explorer:** Shows the project structure with 'CPT (2)' and 'ThreeS (2)' folders.
- Code Editor:** Displays the source code for DataController.java. The method `getValueName` at line 37 is highlighted in blue. The code includes methods like `addValue`, `deleteValue`, `getValueName`, `getValueSize`, `printCategory`, `printValue`, and `getCategoryHeader`.
- Bug Explorer:** Located on the left, it shows a list of bugs under 'CPT (2)'. The selected bug is 'Return value of Model.Value.getName() ignored, but method has no side effect'.
- Bug Info Panel:** Located at the bottom right, it provides details for the selected bug. It includes a navigation pane, a description of the bug, and a warning that the code calls a method and ignores the return value, which can be removed. It also notes that the analysis aims to reduce false positives and lists common false-positive cases, such as methods designed to be overridden for side effects.

Static Analysis Lv3_Find Bugs

T5. SSS CPT Tool

1. String 파라미터에 대한 == , != 연산자 사용

2. 함수에서 입력인자가 변형 없이 출력인자로 나옴

```
public String getValueName(String categoryName, int valueIndex) {  
    ((headerCategory.whereCategory(headerCategory, categoryName)).getValue()).getName();  
    return categoryName;  
}
```


Team 8

Feesual CPT

Static Analysis Lv3_Find Bugs

T8. Feesual CPT

workspace - FindBugs - SMA2017_FeesualCPT/src/TestCaseController.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Bug Explorer

- CPT (2)
- SMA2017_FeesualCPT (2) [SMA2017_FeesualCPT master ↑ 2]
- Scariest (1)
- High confidence (1)
- Call to equals() comparing different types (1)
- Call to java.util.ArrayList<java.lang.String>.equals(java.util.ArrayList<SingleTest...)
- Troubling (1)
- Normal confidence (1)
- Comparison of String objects using == or != in TestCaseController.makeTc(Ta...
- ThreeS (2)
- Troubling (1)
- High confidence (1)
- Comparison of String parameter using == or != (1)
- Of Concern (1)

```
155 }
156 } //single 占位符 占位符 error 占位符 占位符 占位符 占位符 占位符.
157
158 String sub = table.getTable()[0][2];
159 int check = 0;
160 int tempNum = 0;
161 for(int i = 0 ; i < table.getRow() ; i++){
162     if(sub != table.getTable()[i][2]){
163         sub = table.getTable()[i][2];
164         check++;
165         if(check != 1){
166             tempNum++;
167         }
168
169         int flag = 0;
170         TestCase tempTc = new TestCase();
171
172         for(int j = i - tempNum ; j < i ; j++){
173             if(((table.getTable()[j][4] != null || table.getTable()[j][5] != null) && table.getConstraints()[0]) && table.getTab
174                 flag = 1;
175                 break;
176             }
177         }
178     }
```

Bug Info

TestCaseController.java: 162

Navigation

Comparison of String objects using == or != in TestCaseController.makeTc(Table)

Actual type String

Value loaded from sub

Bug: Comparison of String objects using == or != in TestCaseController.makeTc(Table)

This code compares java.lang.String objects for reference equality using the == or != operators. Unless both strings are either constants in a source file, or have been interned using the String.intern() method, the same string value may be represented by two different String objects. Consider using the equals(Object) method instead.

Rank: Troubling (11), **confidence:** Normal

Pattern: ES_COMPARING_STRINGS_WITH_EQ

Type: ES, **Category:** BAD_PRACTICE (Bad practice)

Comparison of String objects using == or != in TestCaseController.makeTc(Table) [Troubling(11), Normal confidence] | Writable | Smart Insert | 162 : 44

Static Analysis Lv3_Find Bugs

T8. Feesual CPT

1. String 파라미터에 == , != 연산자 사용
2. 서로 다른 성격의 자료형간 비교

Q & A