

Graphical Clone Checker

Team #3

201211341 김태현

201411259 고수창

200911411 이상규

Contents

2050 Construct

2051 Implement Class & Methods Definitions

2052 Implement Windows

2055 Write Unit Test Code

2060 Test

2061 Unit Testing

2063 System Testing

2067 Testing Traceability Analysis

2051 Implement Class & Methods Definitions

- Class Definitions

1. MainSystem

Type	Class
Name	MainSystem
Purpose	GUI를 포함한 메인 클래스로써 Actor와의 연결 관계를 가지며 다른 기능 등을 전체적으로 지원하기 위한 클래스
Overview	N/A
Cross References	Functions: All Use Cases: All
Exceptional Courses of Events	N/A

2. FileManager

Type	Class
Name	FileManager
Purpose	파일 목록 관리와 관련 기능에 대한 클래스
Overview	N/A
Cross References	Functions: R1.1, R1.2, R1.3, R1.4 Use Cases: Open Files, Delete Files, Display File List, Clear
Exceptional Courses of Events	N/A

3. ResultManager

Type	Class
Name	ResultManager
Purpose	표절 여부 검사, 결과와 관련 기능에 대한 클래스
Overview	N/A
Cross References	Functions: R3.1, R3.2, R3.3, R3.4, R3.5, R3.6, R3.7, R3.8 Use Cases: Compare, Check Raw Text, Parse File, Check Name, Check Comment, Check Condition, Check Loop, Calculate Scores
Exceptional Courses of Events	N/A

4. ViewManager

Type	Class
Name	ViewManager
Purpose	결과를 보여주는 기능에 대한 클래스
Overview	N/A
Cross References	Functions: R2.1, R4.1, R4.2 Use Cases: Configure, Display Result, Make Graph
Exceptional Courses of Events	N/A

- Method Definitions

1. openFiles()

Type	Method
Name	openFiles
Purpose	User가 검사 대상 파일을 열 수 있도록 한다
Cross References	Functions: R1.1, R1.3 Use Cases
Input (Method)	FileList: ArrayList
Output (Method)	FileList: ArrayList
Abstract Operation (Method)	1. 인자로 받아온 FileList를 FileManager.FileList에 추가하고, FileManger.FileList를 Return한다.
Exceptional Courses of Events	E1.이미 FileList에 동일한 이름을 가진 파일이 있을 경우, 경고창을 띄운다.

2. deleteFiles()

Type	Method
Name	deleteFiles
Purpose	검사 대상 파일을 파일 목록에서 삭제한다.
Cross References	Functions: R1.2 Use Cases:
Input (Method)	FileList: ArrayList
Output (Method)	

Abstract Operation (Method)	<ol style="list-style-type: none"> 인자로 받아온 FileList를 FileManager.FileList에서 삭제한다. displayFileList에 FileList를 인자로 호출한다.
Exceptional Courses of Events	E1. 선택된 파일이 없을 경우 아무것도 삭제하지 않는다.

3. displayFileList()

Type	Method
Name	displayFileList
Purpose	Open된 파일 목록을 보여준다.
Cross References	Functions: R1.3 Use Cases:
Input (Method)	FileList: ArrayList
Output (Method)	
Abstract Operation (Method)	<ol style="list-style-type: none"> 인자로 받아온 FileList를 GUI에 표시한다.
Exceptional Courses of Events	N/A

4. clear()

Type	Method
Name	clear
Purpose	파일 목록에 있는 Files를 모두 삭제한다.
Cross References	Functions: R1.4 Use Cases:
Input (Method)	
Output (Method)	
Abstract Operation (Method)	<ol style="list-style-type: none"> FileList의 내용을 모두 삭제하고 GUI에 파일 목록을 삭제한다.
Exceptional Courses of Events	E1. 파일 목록(7)에 파일이 없을 경우 아무것도 삭제하지 않는다.

5. exit()

Type	Method
Name	exit
Purpose	프로그램을 종료한다.
Cross References	N/A

Input (Method)	
Output (Method)	
Abstract Operation (Method)	1. 프로그램을 종료한다.
Exceptional Courses of Events	N/A

6. compare()

Type	Method
Name	compare
Purpose	선택된 파일들을 1:1로 비교 분석한다.
Cross References	Functions: R3.2, R3.3, R3.4, R3.5, R3.6, R3.7, R3.8, R4.1 Use Cases: Check Raw Text, Parse Files, Check Name, Check Comment, Check Condition, Check Loop, Calculate Scores, Display Result
Input (Method)	FileList: ArrayList
Output (Method)	
Abstract Operation (Method)	<ol style="list-style-type: none"> FileList의 파일들의 조합 A File, B File을 만들어 1:1로 비교를 한다. <ol style="list-style-type: none"> ResultManager.parseFile로 File을 String으로 바꾼다. ResultManager.checkRawText의 결과가 80을 초과하는 경우, calculateScores로 진행한다. ResultManager.checkRawText의 결과가 80 이하일 경우, ParseFile)를 진행한다. checkName, checkComment, checkCondition, checkLoop의 결과로 Scores를 업데이트 한다. ViewManager.displayResult와 ViewManager.makeGraph를 진행하여 Result window의 GUI를 업데이트 한다.
Exceptional Courses of Events	E1. 파일목록(7)에 파일이 없을 경우 에러를 띄운다.

7. checkRawText()

Type	Method
Name	checkRawText
Purpose	두 파일이 텍스트 단위에서 얼마나 비슷한지 검사한다.
Cross References	Functions: R3.1, R3.3, R3.8, R4.1 Use Cases: Compare Files, Parse File, Calculate Scores, Display Result
Input (Method)	a: String, b: String
Output (Method)	percentage: int

Abstract Operation (Method)	<ol style="list-style-type: none"> 1. 두 파일을 input으로 받아서 텍스트 단위에서 정량적으로 얼마나 일치하는지 검사한다. 2. 검사 결과 수치를 Scores 배열에 저장한다.
Exceptional Courses of Events	N/A

- 본 checkRawText() 메소드부터 이하 checkComment(), checkName(), checkLoop(), checkCondition() 메소드의 구현에는 Levenshtein distance(https://en.wikipedia.org/wiki/Levenshtein_distance)를 이용한 Percentage of Text Match 메소드를 사용한다.
(http://blog.sodhanalibrary.com/2014/03/text-comparison-in-java_27.html#.VzzQNPI97Dc)

8. parseFile()

Type	Method
Name	parseFile
Purpose	C 소스 파일을 파싱한다.
Cross References	Functions: R3.1, R3.2, R3.4, R3.5, R3.6, R3.7 Use Cases: Compare Files, Check Raw Text, Check Name, Check Comment, Check Condition, Check Loop
Input (Method)	FileList: ArrayList
Output (Method)	FileArray: ArrayList
Abstract Operation (Method)	<ol style="list-style-type: none"> 1. 하나의 소스파일을 파싱한다. 2. 파싱된 결과를 Result에 저장 후에 반환한다.
Exceptional Courses of Events	N/A

9. checkComment()

Type	Method
Name	checkComment
Purpose	주석을 그대로 복사했는지 검사한다.
Cross References	Functions: R3.1, R3.3, R3.8, R4.1 Use Cases: Compare Files, Parse Files, Calculate Scores, Display Result
Input (Method)	a: String, B: String
Output (Method)	percentage: int
Abstract Operation (Method)	<ol style="list-style-type: none"> 1. 두 파일을 input으로 받아서 주석을 찾아 정량적으로 얼마나 일치하는지 검사한다. 2. 검사 결과 수치를 Scores 배열에 저장한다.
Exceptional Courses of Events	N/A

10. checkName()

Type	Method
Name	checkName
Purpose	함수명, 변수명을 의도적으로 변경했는지 검사한다.
Cross References	Functions: R3.1, R3.3, R3.8, R4.1 Use Cases: Compare Files, Parse Files, Calculate Scores, Display Result
Input (Method)	a: String, B: String
Output (Method)	percentage: int
Abstract Operation (Method)	<ol style="list-style-type: none">1. 두 파일을 input으로 받아서 변수나 함수 이름을 변경한 여부를 찾아내고 정량적으로 얼마나 일치하는지 검사한다.2. 검사 결과 수치를 Scores 배열에 저장한다.
Exceptional Courses of Events	N/A

11. checkLoop()

Type	Method
Name	checkLoop
Purpose	반복문의 의도적인 변경이 있는지 검사한다.
Cross References	Functions: R3.1, R3.3, R3.8, R4.1 Use Cases: Compare Files, Parse Files, Calculate Scores, Display Result
Input (Method)	a: String, B: String
Output (Method)	percentage: int
Abstract Operation (Method)	<ol style="list-style-type: none">1. 두 파일을 input으로 받아서 반복문을 의도적으로 변경한 여부를 찾아내고 정량적으로 얼마나 일치하는지 검사한다.2. 검사 결과 수치를 Scores 배열에 저장한다.
Exceptional Courses of Events	N/A

12. checkCondition()

Type	Method
Name	checkCondition
Purpose	조건문의 의도적인 변경이 있는지 확인한다.
Cross References	Functions: R3.1, R3.3, R3.8, R4.1 Use Cases: Compare Files, Parse Files,

	Calculate Scores, Display Result
Input (Method)	a: String, B: String
Output (Method)	percentage: int
Abstract Operation (Method)	<ol style="list-style-type: none"> 1. 두 파일을 input으로 받아서 조건문을 의도적으로 변경한 여부를 찾아내고 정량적으로 얼마나 일치하는지 검사한다. 2. 검사 결과 수치를 Scores 배열에 저장한다.
Exceptional Courses of Events	N/A

13. calculateScores()

Type	Method
Name	calculateScores
Purpose	결과 점수를 계산한다.
Cross References	Functions: R3.2, R3.4, R3.5, R3.6, R3.7 Use Cases: Check Raw Text, Check Name, Check Comment, Check Condition, Check Loop
Input (Method)	
Output (Method)	
Abstract Operation (Method)	<ol style="list-style-type: none"> 1. Scores의 전체 평균결과를 계산하고 평균점수를 Scores에 업데이트 한다.
Exceptional Courses of Events	N/A

14. makeGraph()

Type	Method
Name	makeGraph
Purpose	결과 그래프를 그린다.
Cross References	Functions: R3.8, R4.1 Use Cases: Calculate Scores, Display Result
Input (Method)	hashMap: HashMap, hashMap2: HashMap
Output (Method)	mainPanel: JPanel
Abstract Operation (Method)	<ol style="list-style-type: none"> 1. Scores 배열을 참고해서 그래프를 그린다.
Exceptional Courses of Events	N/A

15. displayResult()

Type	Method
Name	displayResult
Purpose	결과를 보여준다.
Cross References	Functions: R2.1, R3.8, R4.2 Use Case: Configure, Calculate Scores, Make Graph
Input (Method)	
Output (Method)	
Abstract Operation (Method)	1. Scores 배열과 int Percentage를 참고해서 결과를 그린다.
Exceptional Courses of Events	N/A

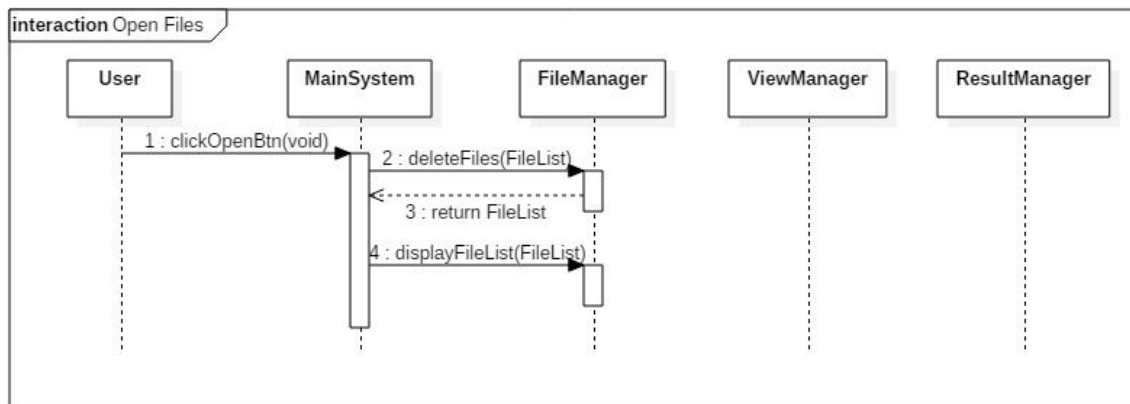
16. configure()

Type	Method
Name	configure
Purpose	몇 퍼센트보다 높은 결과를 보여줄 것인지 설정하도록 한다.
Cross References	Functions: R2.1, R4.1 Use Cases: Display Result
Input (Method)	Percentage: int
Output (Method)	N/A
Abstract Operation (Method)	<ol style="list-style-type: none"> 1. 초기값은 40으로 설정되어 있다. 2. Configure window에서 0-100 사이의 정수를 받아 Apply를 받았을 경우 int Percentage를 업데이트 한다. 3. Display Result 시 업데이트 된 Percentage 이상의 결과만을 표시한다.
Exceptional Courses of Events	E1. 설정한 Filter range(10)가 0 미만이거나 100을 초과할 경우 에러를 띄운다.

2052 Implement Windows

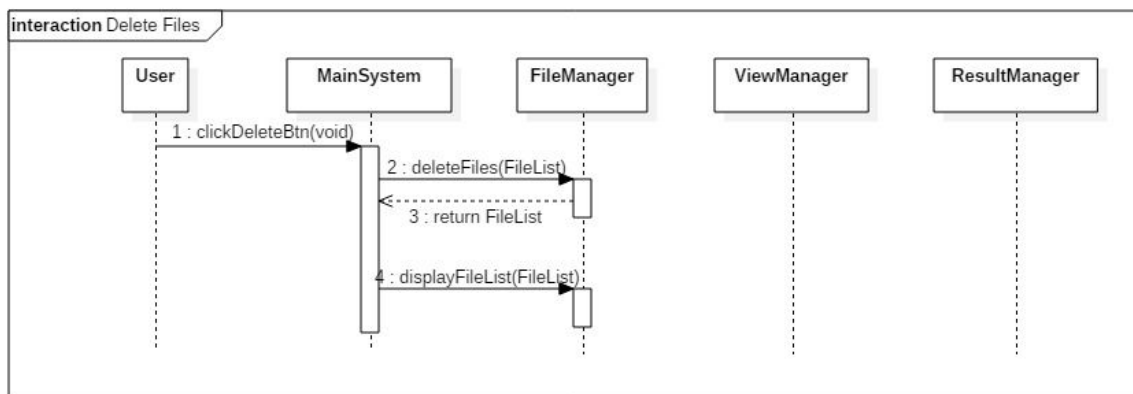
1. clickOpenBtn

Name	clickOpenBtn
Responsibilities	Open button을 누른다.
Type	GUI
Cross References	R1.1
Notes	Open button을 눌러 openFiles 메소드를 호출한다.
Pre-Conditions	
Post-Conditions	FileManager 클래스의 openFiles 메소드를 호출하여 FileList를 업데이트하고, displayFileList를 호출한다.



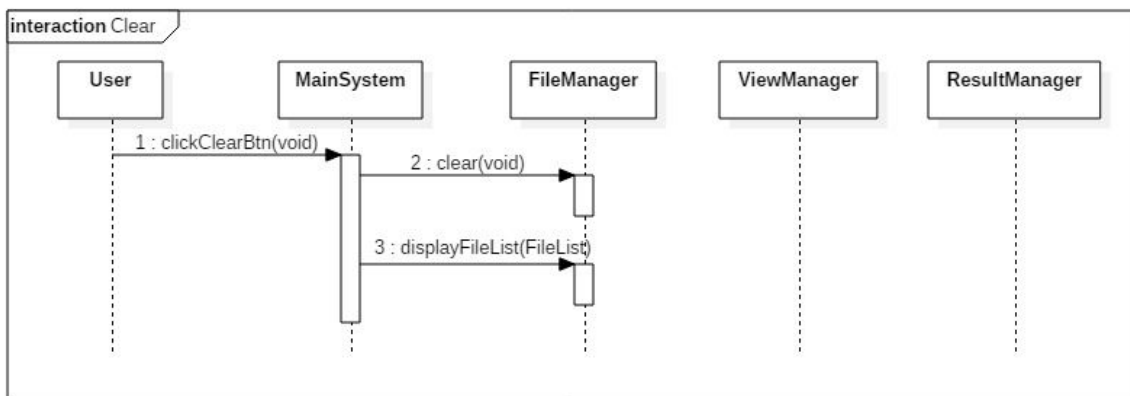
2. clickDeleteBtn

Name	clickDeleteBtn
Responsibilities	Delete button을 누른다.
Type	GUI
Cross References	R1.2
Notes	Delete button을 눌러 선택된 deleteFiles 메소드를 호출하고 선택된 FileList를 전달한다.
Pre-Conditions	선택된 파일이 있어야 한다.
Post-Conditions	FileManager 클래스의 deleteFiles 메소드를 호출하여 FileList를 업데이트하고, displayFileList를 호출한다.



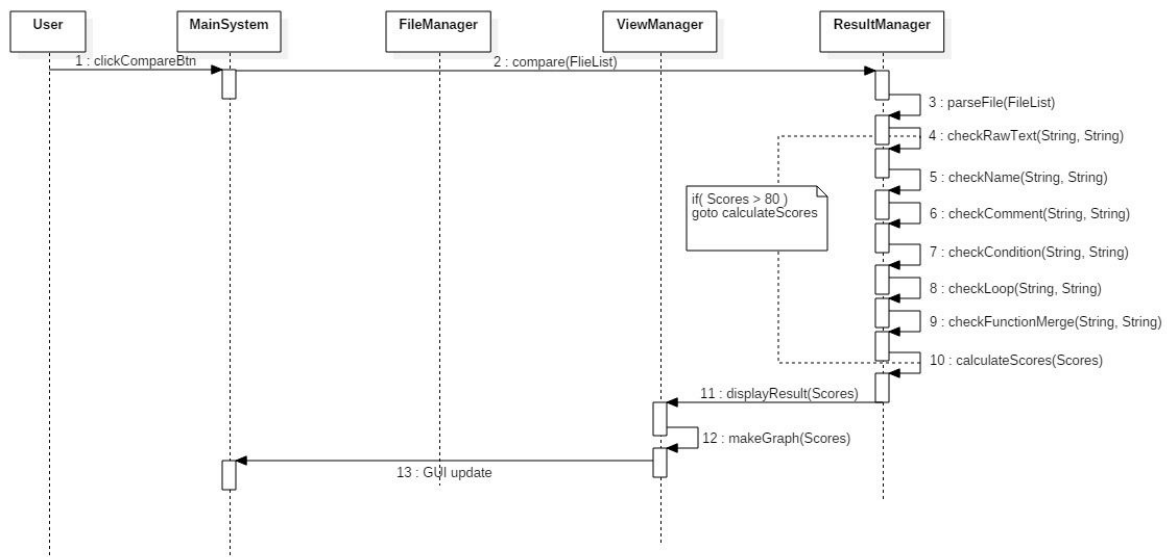
3. clickClearBtn

Name	clickClearBtn
Responsibilities	Clear button을 누른다.
Type	GUI
Cross References	R1.4
Notes	Clear button을 눌러 clear 메소드를 호출한다.
Pre-Conditions	
Post-Conditions	FileManager 클래스의 clear 메소드를 호출하여 FileList를 업데이트하고, displayFileList를 호출한다.



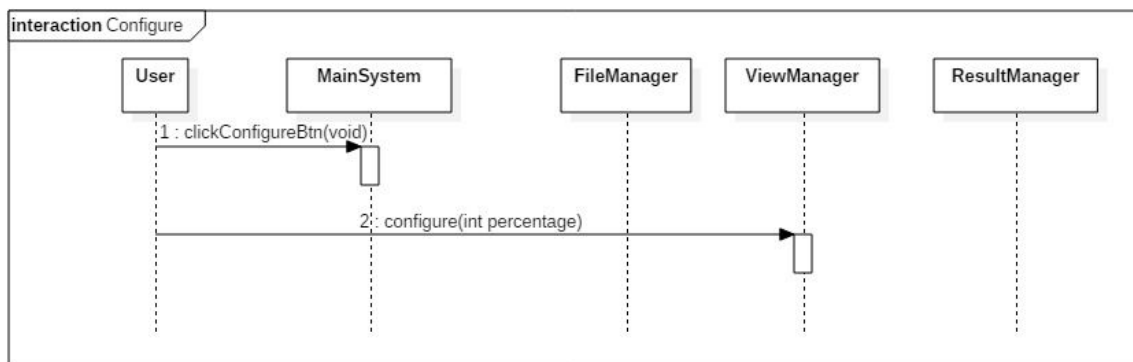
4. clickCompareBtn

Name	clickCompareBtn
Responsibilities	Compare button을 누른다.
Type	GUI
Cross References	R3.1
Notes	Compare button을 눌러 compare 메소드를 호출하고 FileList를 전달한다.
Pre-Conditions	FileList의 Size가 2 이상이어야 한다.
Post-Conditions	ResultManager 클래스의 checkRawText, parseFile, checkName, checkComment, checkCondition, checkLoop, calculateScores 메소드를 통해 업데이트 된 Scores를 ViewManager 클래스의 displayResult, makeGraph 메소드를 거쳐 GUI를 업데이트 한다.



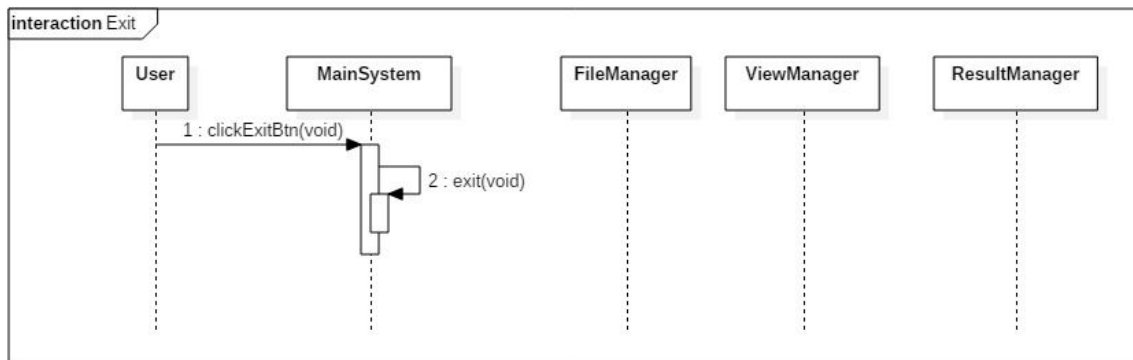
5. clickConfigureBtn

Name	clickConfigureBtn
Responsibilities	Configure button을 누른다.
Type	GUI
Cross References	R2.1
Notes	Configure button을 눌러 Configure window를 호출한다.
Pre-Conditions	
Post-Conditions	Apply 버튼을 누르면 ViewManager 클래스의 configure 메소드를 호출하고, Configure Window를 닫는다. Cancel 버튼을 누르면 창을 닫는다.



6. clickExitBtn

Name	clickExitBtn
Responsibilities	Exit button을 누른다.
Type	GUI
Cross References	R5
Notes	Exit button을 눌러 exit 메소드를 호출한다.
Pre-Conditions	
Post-Conditions	프로그램을 종료한다.



2055 Write Unit Test Code

```
@org.junit.Test
```

```
public void TestCheckRawText(){  
    ResultManager resultManager = new ResultManager();  
    assertEquals(100,resultManager.CheckRawText("int a = 9;", "int a = 9;"));  
}
```

```
@org.junit.Test
```

```
public void TestCheckName(){  
    ResultManager resultManager = new ResultManager();  
    assertEquals(100,resultManager.CheckName("int a = 9;", "int b = 9;"));  
}
```

```
@org.junit.Test
```

```
public void TestCheckLoop(){  
    ResultManager resultManager = new ResultManager();  
    assertEquals(100,resultManager.CheckLoop("while", "for"));  
}
```

```
@org.junit.Test
```

```
public void TestCheckCondition(){  
    ResultManager resultManager = new ResultManager();  
    assertEquals(100,resultManager.CheckCondition("if", "switch"));  
}
```

```
@org.junit.Test
```

```
public void TestCheckComment(){  
    ResultManager resultManager = new ResultManager();  
    assertEquals(100,resultManager.CheckComment("//aa", "//aa"));  
}
```

2061 Unit Testing

The screenshot displays an IDE window titled "AppTest.java - [test] - test - [~/github/2016sv/test]". The editor shows the following Java code for `AppTest.java`:

```
16 {
17
18     @org.junit.Test
19     public void TestCheckRawText(){
20         ResultManager resultManager = new ResultManager();
21         assertEquals(100, resultManager.CheckRawText("int a = 9;"));
22     }
23
24
25     @org.junit.Test
26     public void TestCheckName(){
27         ResultManager resultManager = new ResultManager();
28         assertEquals(100, resultManager.CheckName("int a = 9;", "int a = 9;"));
29     }
30
31     @org.junit.Test
32     public void TestCheckLoop(){
33         ResultManager resultManager = new ResultManager();
34         assertEquals(100, resultManager.CheckLoop("while", "for"));
35     }
36
37     @org.junit.Test
38     public void TestCheckCondition(){
39         ResultManager resultManager = new ResultManager();
40         assertEquals(100, resultManager.CheckCondition("if", "switch"));
41     }
42 }
```

The Run console at the bottom shows the following output:

```
Run: MainSystem All in test (2)
Done: 5 of 5 (in 0.037 s)
/Library/Java/JavaVirtualMachines/jdk1.7.0_45.jdk/Contents/Home/bin/java
Process finished with exit code 0
```

At the bottom of the IDE, the status bar indicates: "Tests Passed: 5 passed in 0.037 s (moments ago) 1:1 LF UTF-8 307 of 741M".

2063 System Testing

Test Number	Test 항목	Description	Use Case	System Function	Pass / Fail
1-1	File Open Test	Select한 파일이 하이라이팅이 잘 되는지 확인한다.	Open Files	R1.1	P
1-2	File Open Test	Open을 한 File이 List에 잘 뜨는지 확인한다.	Open Files	R1.1	P
1-3	File Delete Test	파일을 선택하지 않고 삭제하려 할 때 경고창이 잘 뜨는지 확인한다.	Delete Files	R1.2	P
1-4	File Delete Test	선택한 파일을 삭제했을 때 파일 목록에서 없어졌는지 확인한다.	Delete Files	R1.2	P
1-5	File Delete Test	다수의 파일을 한꺼번에 삭제 가능한지 확인한다.	Delete Files	R1.2	P
1-6	Display File List Test	파일들이 파일 목록에 뜨는지 확인한다.	Display File List	R1.3	P
1-7	Clear Test	Clear를 눌렀을 때 모든 파일들이 파일 목록에서 잘 사라졌는지 확인한다.	Clear	R1.4	P
2-1	Configure Test	설정된 값이 결과에 잘 적용되는지 확인한다.	Configure	R2.1	P
2-2	Configure Test	설정값이 음수이거나 100을 넘어갔을 때 넘어가지 않는지 확인한다.	Configure	R2.1	P
3-1	Compare Test	파일 목록에 파일이 없거나 하나밖에 없을 경우에 경고창이 뜨는지 확인한다.	Compare	R3.1	P
3-2	Compare Test	파일 비교가 제대로 되는지 확인한다.	Compare	R3.1	P
3-3	Check Raw Text Test	Raw text 비교가 제대로 되는지 확인한다.	Check Raw Text	R3.2	P
3-4	Parse File Test	Source code의 parsing이 제대로 되는지 확인한다.	Parse File	R3.3	P
3-5	Check Name Test	대상 파일들이 함수명이나 변수명만 바꾼 것을 잘 찾아내는지 확인한다.	Check Name	R3.4	P
3-6	Check Comment Test	대상 파일들이 주석만 복사한 것을 잘 찾아내는지 확인한다.	Check Comment	R3.5	P
3-7	Check Condition Test	대상 파일들이 if-switch문을 변경한 것을 잘 찾아내는지 확인한다.	Check Condition	R3.6	P
3-8	Check Loop Test	대상 파일들이 for-while을 변경한 것을 잘 찾아내는지 확인한다.	Check Loop	R3.7	P
3-9	Calculate Scores Test	결과 점수가 잘 계산되는지 확인한다.	Calculate Scores	R3.8	P
4-1	Display Result Test	결과 목록의 값과 그래프의 값이 일치하는지 확인한다.	Display Result	R4.1	P
4-2	Display Result Test	그래프가 표시되는지 확인한다.	Display Result	R4.1	P

4-3	Make Graph Test	측정한 값이 꺾은선 그래프로 잘 표시되는지 확인한다.	Make Graph	R4.2	P
4-4	Make Graph Test	측정한 값의 평균 값이 꺾은선 그래프로 잘 표시되는지 확인한다.	Make Graph	R4.2	P
5-1	Exit Test	프로그램이 정상적으로 종료되는지 확인한다.	Exit	R5	P

2067 Testing Traceability Analysis

