

# SOFTWARE VERIFICATION

2nd System Test & Analysis Test

**TEAM 1**

컴퓨터공학부  
201011314 김민재  
201011356 이종찬  
201011376 한지승  
201111329 강성길

2016.06.09

# INDEX



1

**Specification**

2

**Brute Force Test**

3

**Category Partitioning test**

4

**Pairwise Combination Test**

5

**System Test Conclusion**

# INDEX

6

**CheckStyle**

7

**PMD**

8

**FindBugs**

9

**Code Coverage**



# Specification

# 1. Specification

## OSP 1000 Stage

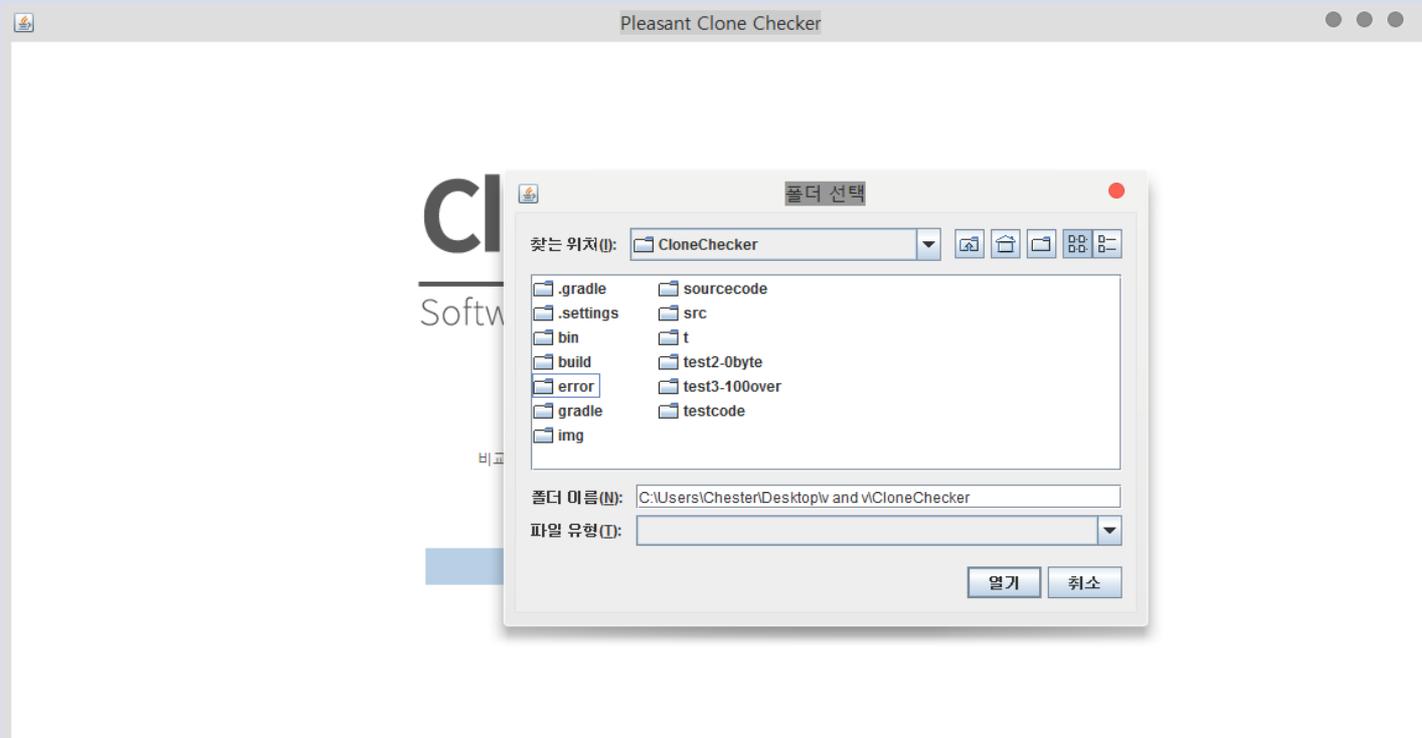
- 용어 정의가 명확하지 않음

Term	Description	Remarks
소스 코드	컴파일 가능 여부와 상관 없이 프로그래밍 언어로 작성된 코드	
소스 코드 파일	파일 크기와 상관 없이 소스 코드가 담긴 .c 확장자 파일 파일명 양식 : 학번_이름.c	

# 1. Specification

## OSP 2030 Stage

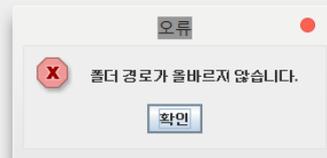
- Use Case “Start”
  - Exception 처리에 대한 오류메시지가 정확하지 않다.



# 1. Specification

## OSP 2030 Stage

- Use Case “Start”
  - Exception 처리에 대한 오류메시지가 정확하지 않다.



# 1. Specification

## OSP 2030 Stage

- Use Case “Analyze File”
  - Pre Requisites 내용 중에서 File Instance를 생성한다고 명시되어 있으나 생성되는 곳을 찾아 볼 수 없다.

<b>Use Case</b>	Analyze File
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일을 분석하여 일치율 계산에 필요한 정보로 가공한다.
<b>Overview</b>	시스템 내에서 파일을 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3.1(Analyze Line), R.3.2(Analyze Function), R.3.3(Analyze Variable), R.3.4(Analyze Preprocessor), R.3.5(Analyze Annotation)
<b>Pre-Requisites</b>	<i>Files</i> 인스턴스가 생성되어 있어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 시스템 내에서 파일을 분석한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2030 Stage

- Use Case “Analyze Annotation”
  - 결과값을 저장하는 과정이 명시되어 있지 않음

<b>Use Case</b>	Analyze Annotation
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 주석의 개수를 분석하여 결과값을 저장한다.
<b>Overview</b>	시스템 내에서 주석의 개수를 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	Change Annotation이 실행되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 시스템 내에서 주석의 개수를 분석한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2030 Stage

- Use Case “Analyze Line”
  - 결과값을 저장하는 과정이 명시되어 있지 않음

<b>Use Case</b>	Analyze Line
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 라인 수를 분석하여 결과값을 저장한다.
<b>Overview</b>	시스템 내에서 라인 수를 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 시스템 내에서 라인 수를 분석한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2030 Stage

- Use Case “File Type”
  - 새로 선언된 타입을 찾아 쓰는 Use Case가 없다.

<b>Use Case</b>	Find Type
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 내 새로 선언된 타입을 찾는다.
<b>Overview</b>	구조체 또는 typedef로 새로 선언된 타입을 찾는다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	Delete Annotation, Divide Code Line, Delete Printf가 실행되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 구조체 또는 typedef로 새로 선언된 타입을 찾는다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2030 Stage

- Use Case “Analyze Function”
  - 저장하는 과정이 명시되어 있지 않다.

<b>Use Case</b>	Analyze Function
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 함수의 개수와 이름을 분석하여 결과값을 저장한다.
<b>Overview</b>	시스템 내에서 함수의 개수와 이름을 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	Delete Printf, Delete Annotation, Find Type이 실행되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 시스템 내에서 함수의 개수와 이름을 분석한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2030 Stage

- Use Case “Analyze Variable”
  - 저장하는 과정이 명시되어 있지 않다.

<b>Use Case</b>	Analyze Variable
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 변수의 개수와 이름을 분석하여 결과값을 저장한다.
<b>Overview</b>	시스템 내에서 변수의 개수와 이름을 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	Delete Printf, Delete Annotation, Find Type이 실행되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 시스템 내에서 변수의 개수와 이름을 분석한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2030 Stage

- Use Case “Analyze Preprocessor”
  - 저장하는 과정이 명시되어 있지 않다.

<b>Use Case</b>	Analyze Preprocessor
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 전처리기의 개수와 이름을 분석하여 결과값을 저장한다.
<b>Overview</b>	시스템 내에서 전처리기의 개수와 이름을 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	Delete Annotation이 실행되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 시스템 내에서 전처리기의 개수와 이름을 분석한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2030 Stage

- Calculate와 관련된 Use case 모두
  - 각 Use case에서 일치율을 계산하고 저장하는 과정이 명시되어 있지 않다.

<b>Use Case</b>	Calculate File
<b>Actor</b>	-
<b>Purpose</b>	Files 인스턴스에 저장되어있는 변수를 비교하여 일치율을 계산한다.
<b>Overview</b>	Files 인스턴스에 저장되어있는 변수를 바탕으로 시스템 내에서 일치율을 계산한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.5.1(Calculate Line Sync-Rate), R.5.2(Calculate Function Sync-Rate), R.5.3(Calculate Variable Sync-Rate), R.5.4(Calculate Preprocessor Sync-Rate), R.5.5(Calculate Annotation Sync-Rate)
<b>Pre-Requisites</b>	소스 코드 파일 분석이 정상적으로 완료되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 분석된 데이터를 비교하여 시스템 내에서 일치율을 계산한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	Calculate Line Sync-Rate
<b>Actor</b>	-
<b>Purpose</b>	분석한 라인 수를 비교하여 일치율을 계산한다.
<b>Overview</b>	분석한 라인 수를 바탕으로 시스템 내에서 일치율을 계산한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.5(Calculate Sync-Rate)
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 분석한 라인 수를 비교하여 시스템 내에서 일치율을 계산한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2030 Stage

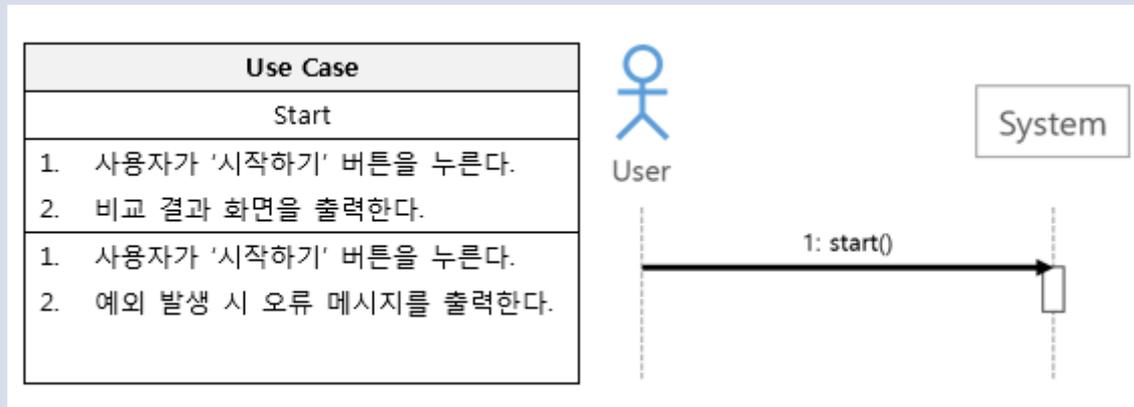
- Use Case “Display Sync”
  - Calculate와 관련된 Use case에서 계산한 일치율을 사용해야 하는데, 문서상에선 반영되어 있지 않다.

<b>Use Case</b>	Display Sync
<b>Actor</b>	User
<b>Purpose</b>	기준 소스 코드 파일과 비교할 소스 코드 파일의 일치율과 유사 항목을 출력한다.
<b>Overview</b>	사용자가 소스 코드 파일 이름에 마우스 커서를 올리면 말풍선 형태로 일치율과 유사 항목을 출력한다.
<b>Type</b>	Primary & Evident
<b>Cross Reference</b>	N/A
<b>Pre-Requisites</b>	소스 코드 파일 비교가 정상적으로 완료되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 사용자가 태그 클라우드의 소스 코드 파일 이름에 마우스 커서를 올린다. 2. (S) : 말풍선 형태로 일치율과 유사 항목을 출력한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2135 Stage

- Use Case “Start”
  - 예외 발생시 오류 메시지가 뜨지 않는 경우가 있다.



# 1. Specification

## OSP 2041 Stage

- 현재 결과화면과 다름



# 1. Specification

## OSP 2041 Stage

- Use Case “Select folder”
  - Actor의 역할 필요

<b>Use Case</b>	Select Folder
<b>Actor</b>	User
<b>Purpose</b>	비교할 소스 코드 파일들이 들어있는 폴더를 선택한다.
<b>Overview</b>	사용자가 소스 코드 파일들이 들어있는 폴더 경로를 선택한다.
<b>Type</b>	Primary & Evident
<b>Cross Reference</b>	N/A
<b>Pre-Requisites</b>	버튼(A)를 누른다.
<b>UI Widgets</b>	Window-1
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : JAVA에서 제공되는 폴더 선택 창을 출력한다. 2. (S) : 사용자가 선택한 폴더 경로를 <i>Controller.folderPath</i> 에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2041 Stage

- Use Case “Start”
  - Exception 처리에 대한 오류메시지가 정확하지 않다.

Use Case	Start
Actor	User
Purpose	비교 결과 화면을 출력한다.
Overview	사용자가 '시작하기' 버튼을 누르면 비교 결과 화면을 출력한다.
Type	Primary & Evident
Cross Reference	R.3(Analyze File)
Pre-Requisites	버튼(B)을 누른다.
UI Widgets	Window-1
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) : Files 인스턴스를 실제 소스 코드 파일 개수 만큼 생성한다. 2. (S) : 생성된 Files 인스턴스마다 <i>File:name</i> 에 파일 이름을 저장한다. 3. (S) : 소스 코드를 한 줄씩 <i>Controller:source</i> 에 저장한다. 4. (S) : 생성된 Files 인스턴스마다 <i>Analyze(source)</i> 를 선언하고, <i>Analyze:analyzeFile(fileNum)</i> 를 호출한다. 5. (S) : 결과 화면(Window-2)을 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	아래의 상황일 경우 폴더 지정 오류 메시지를 출력하고, 결과 화면을 출력하지 않는다. 1. 폴더 안에 .c 확장자를 가진 파일이 없는 경우 - ".c 확장자를 가진 파일이 없는 폴더입니다." 메시지 2. 폴더 안에 파일이 100개 초과일 경우 - "파일 100개 이상인 폴더입니다." 메시지 3. 폴더 안에 0바이트인 파일이 있는 경우 - "0Byte 파일이 있는 폴더입니다." 메시지 소스 코드를 읽지 못하는 경우 "폴더 경로가 올바르지 않습니다." 오류 메시지를 출력하고, 결과 화면을 출력하지 않는다.

# 1. Specification

## OSP 2041 Stage

- Use Case “Analyze Line”
  - 문장 개수 파악이라는 말이 적절하지 못함

<b>Use Case</b>	Analyze Line
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 라인 수를 분석하여 결과값을 저장한다.
<b>Overview</b>	시스템 내에서 라인 수를 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	N/A
<b>UI Widgets</b>	-
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 생성된 <i>Analyze:source</i> 의 문장 개수를 파악하여 소스 코드 파일의 라인 수를 분석한다. 2. (S) : 라인 수를 <i>File:numOfLine</i> 에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2041 Stage

- Use Case “Delete Annotation”
  - 문장 개수 파악이라는 말이 적절하지 못함

<b>Use Case</b>	Delete Annotation
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 주석을 삭제한다.
<b>Overview</b>	소스 코드 파일의 주석을 삭제한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	Analyze Annotation이 실행되어야 한다.
<b>UI Widgets</b>	-
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 생성된 <i>Analyze:source</i> 를 확인한다. 2. (S) : <i>Analyze:source</i> 에 <i>///</i> 을 포함한 문장을 삭제한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 1. Specification

## OSP 2041 Stage

- Use Case “Analyze Function” , “Analyze Variable”
  - Analyze : listType에는 구조체만 저장하는데 함수개수와 변수개수를 listType에서 찾는 것이 모순

<b>Use Case</b>	Analyze Function
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 함수의 개수와 이름을 분석하여 결과값을 저장한다.
<b>Overview</b>	시스템 내에서 함수의 개수와 이름을 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	Delete Printf, Delete Annotation, Find Type이 실행되어야 한다.
<b>UI Widgets</b>	-
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 생성된 <i>Analyze:source</i> 를 확인한다. 2. (S) : 리턴 타입( <i>Analyze:listType</i> ) 발견 시 문장 내에 '(' 가 들어가고 ';'가 들어있지 않다면 함수의 개수( <i>File:numOfFunction</i> )를 증가시킨다. 3. (S) : <i>File:listFunction</i> 에 함수의 이름을 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional</b>	N/A

# 1. Specification

## OSP 2041 Stage

- Use Case “Analyze Function” , “Analyze Variable”
  - Analyze : listType에는 구조체만 저장하는데 함수개수와 변수개수를 listType에서 찾는 것이 모순

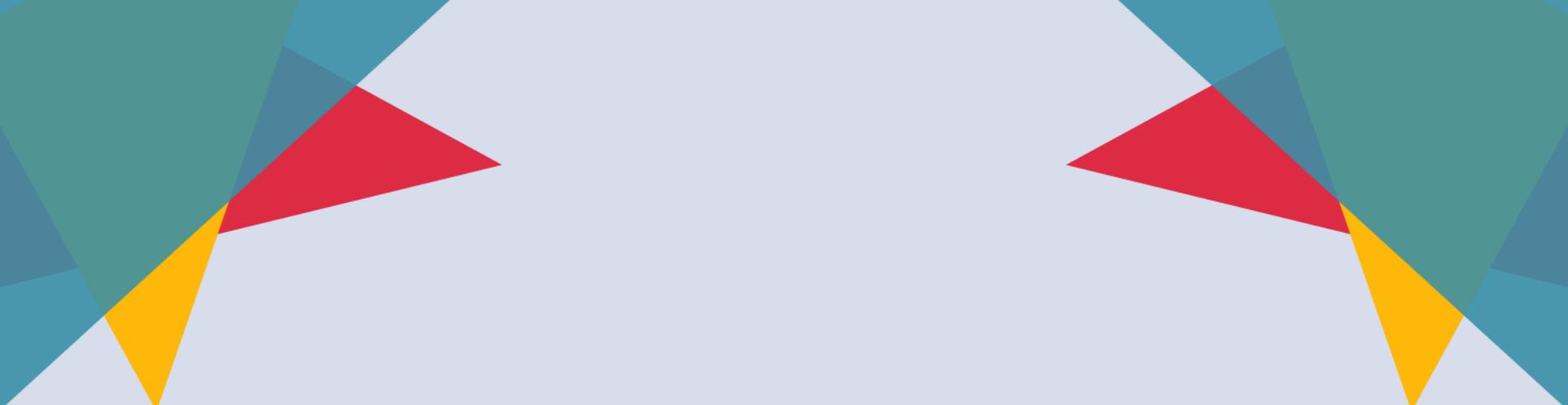
<b>Use Case</b>	Analyze Function
<b>Actor</b>	-
<b>Purpose</b>	소스 코드 파일의 함수의 개수와 이름을 분석하여 결과값을 저장한다.
<b>Overview</b>	시스템 내에서 함수의 개수와 이름을 분석한다.
<b>Type</b>	Primary & Hidden
<b>Cross Reference</b>	R.3(Analyze File)
<b>Pre-Requisites</b>	Delete Printf, Delete Annotation, Find Type이 실행되어야 한다.
<b>UI Widgets</b>	-
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 생성된 <i>Analyze:source</i> 를 확인한다. 2. (S) : 리턴 타입( <i>Analyze:listType</i> ) 발견 시 문장 내에 '(' 가 들어가고 ';'가 들어있지 않다면 함수의 개수( <i>File:numOfFunction</i> )를 증가시킨다. 3. (S) : <i>File:listFunction</i> 에 함수의 이름을 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional</b>	N/A

# 1. Specification

## OSP 2041 Stage

- Use Case “Display Sinc”
  - Calculate와 관련된 Use case에서 계산한 일치율을 사용해야 하는데, 문서상에선 반영되어 있지 않다.

<b>Use Case</b>	Display Sync
<b>Actor</b>	User
<b>Purpose</b>	기준 소스 코드 파일과 비교할 소스 코드 파일의 일치율과 유사 항목을 출력한다.
<b>Overview</b>	사용자가 소스 코드 파일 이름에 마우스 커서를 올리면 말풍선 형태로 일치율과 유사 항목을 출력한다.
<b>Type</b>	Primary & Evident
<b>Cross Reference</b>	N/A
<b>Pre-Requisites</b>	소스 코드 파일 비교가 정상적으로 완료되어야 한다.
<b>UI Widgets</b>	Window-3
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 사용자가 태그 클라우드의 소스 코드 파일 이름에 마우스 커서를 올린다. 2. (S) : 말풍선 형태로 일치율과 각 항목의 일치율을 출력한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A



2

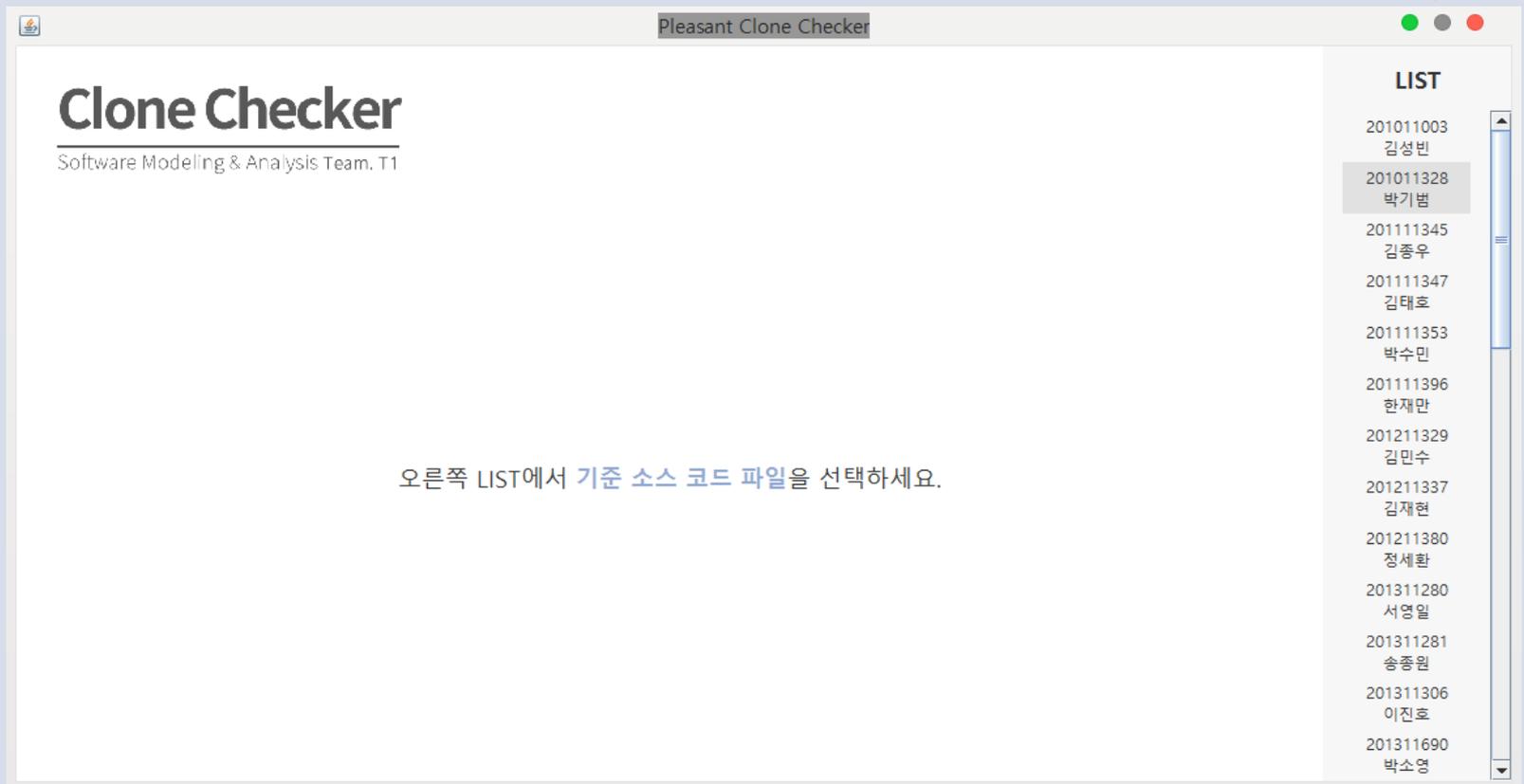
## **Brute Force Test**

## 2. Brute Force Test

- Brute Force Testing Report
  - Fail – 9개, Pass – 18개, 총 Case 27개, 66% Pass

## 2. Brute Force Test

- Fail Case
  - 맨 위나 맨 아래에 선택된 상태에서 키보드가 작동됨



## 2. Brute Force Test

- Fail Case
  - 주석의 개수가 서로 다른데 서로 유사도 비교 값이 다르다 (하나는 96%, 다른 하나는 100%로 서로 상이함)
  - 5개의 유사도 척도가 100%지만 평균이 96%로 표시되는 경우가 있다.
  - 크기가 0인 파일이 하나라도 있어도 다른 코드 분석은 아예 안한다.
  - 이름 양식을 맞추지 않았을 경우 오류가 난다.
  - 가상머신 위에서 실행 할 경우 Heap 사이즈가 부족하다는 오류가 발생한다.
  - 마우스를 드래그 해 놓을 경우 포인트는 이동하지만 선택은 되지 않는다.
  - ppt 확장자를 c로 바꾸면 프로그램에 오류가 발생한다.
  - 파일이 하나만 존재하는 폴더를 선택 할 경우 여전히 실행된다.

## 2. Brute Force Test

- Pass Case
  - 숨김파일 - 클론체킹 대상으로 포함하지 않음
  - 빈폴더 - 예외처리 진행
  - 빠르게 중앙으로 이동 - 정상적으로 동작
  - 파일 내용을 크게 늘림 - 시간은 다소 걸리지만 정상적으로 동작
  - 여러 종류의 파일 혼합 - 확장자가 .c인 파일만 비교
  - 확장자가 .c가 아닌 파일 - 비교대상에서 제외
  - 빈파일 - 예외처리 진행
  - 폴더에 다수의 파일을 포함 - 파일이 100개 초과했다고 예외처리 진행
  - 폴더 안의 폴더는 비교하지 않음
  - 확장자를 지운 파일 - 대상으로 포함하지 않음



3

## **Category Partition Test**

# 3. Category Partition Test

- Category Partition Test Case

Category	Kind	#	value	Error Constraints	Property Constraints	Single Constraints	
기초 기능	파일 디렉토리	100	소스파일만 있는 경우				
		101	물려만 있는 경우	error			
		102	소스파일과 출력이 섞인 경우	error			
파일갯수		110	파일이 존재 하지 않을 경우	error			
		111	1개	error			
		112	여러개				
		120	C언어 확장자 소스파일 만 있을 때		[Property ONCC]		
파일종류		121	C언어와 다른 확장자 소스파일들이 섞여있을 때				
		122	모든 파일이 출길 파일일 경우	error			
		123	모든 파일이 같은 이름 다른 형식 파일	error			
		124	읽기 권한이 없는 파일	error			
시작하기	코드라인	130	시작하기 버튼 클릭				
		200	모든 파일이 한글코드		[Property OLC]		
분석 및 계산		201	모두 다 빈문서				
		202	모두 다 여러 라인		[Property MLC]		
		203	파일들이 한글코드 or 빈문서 or 여러라인 섞인 코드 라인 스타일		[Property MTC]		
		210	모든 파일이 /* */만 있는 경우		if OLC    MLC		
		211	모든 파일이 //만 있는 경우		if OLC    MLC		
		212	모든 파일이 주석이 없는 경우		if OLC    MLC		
		213	모든 파일이 /* */ or // or 주석이 없는 경우		if MTC		
printf문		220	모든 파일이 printf문 존재		if OLC    MLC  ONCC		
		221	모든 파일이 printf문 없는 경우				
		222	파일들의 printf문의 존재 여부가 섞인 경우		if MTC		
함수		230	모든 파일이 함수 존재		if OLC    MLC		
		231	모든 파일이 함수가 없는 경우		[Property NF]		
		232	모든 파일이 함수가 다른 함수의 매개변수로 들어갈때		if OLC    MLC		
		233	모든 파일이 함수가 다른 함수의 매개변수로 들어가지 않을 때				
		234	파일들이 함수가 존재 or 함수가 없거나 or 함수의 매개변수로 들어가거나 or 매개변수로 들어가지 않을때		if MTC		
변수		240	모든 파일이 변수 존재		if OLC    MLC		
		241	모든 파일이 변수가 없을 경우				
		242	모든 파일이 매개변수가 쓰일 경우		if OLC    MLC & INF		
		243	모든 파일이 매개변수가 쓰이지 않을 경우		if NF		
		244	모든 파일이 변수존재 or 매개변수가 쓰이거나 or 매개변수가 안쓰이는 경우 중 한가지 경우		if MTC		
		250	모든 파일이 똑같은 전처리기 선언		if OLC    MLC    ONCC		
전처리		251	모든 파일이 다른 전처리기 선언		if OLC    MLC    ONCC		
		252	모든 파일이 전처리기가 없는 경우				
		253	파일들이 똑같은 전처리기 선언 or 모든 파일이 다른 전처리기 선언 or 전처리기가 없는 경우		if MTC		
		253	파일들이 똑같은 전처리기 선언 or 모든 파일이 다른 전처리기 선언 or 전처리기가 없는 경우				
UI	첫화면 출력	400	파일이름이 문자 10개 이상			single	
		410	마우스로 List 목록 선택			single	
	기온변경	Tag Cloud 선택	411	키보드로 List 목록 선택			single
			420	상세 우사드 보기			single
			421	올렸다 내리기			single
			422	내리고 올리기			single
			423	종류기			single
			424	여러 번 클릭			single

# 3. Category Partition Test

- Category Partition Test Case

	전처리	250	같은 전처리기 존재		property SPN	
		251	전처리기 선언이 상이			
		252	전처리기가 없는 경우		property PPN	3
		253	전처리기가 있는 것 과 없는 것		property MPPN	
계산	주석 유사도	300	똑같은 주석 수		if not MCM	
		301	비교 파일 모두 주석 존재 무		if NC	
		302	주석이 없는 파일과 있는 파일 비교		if MCM	
		303	비교 파일의 주석 수 차이		if CTML    CTSL    CTB	
	라인 유사도	310	똑같은 라인 수			
		311	비교파일 모두 빈 파일		if ED	
		312	라인이 없는 파일과 있는 파일 비교		if EDMC	
		313	비교 파일의 라인 수 차이		if not ED    STC	
	함수 유사도	320	똑같은 함수 수		if not PEF	
		321	비교파일 모두 함수 존재 무		if NEF	
		322	함수가 있는 파일과 없는 파일 비교		if MEF	
		323	비교 파일의 함수 수 차이		if EF	
	전처리기 유사도	330	똑같은 전처리기 수		if SPN    PPN	
		331	비교파일 모두 전처리기 존재 무		if PPN	
		332	전처리기가 있는 파일과 없는 파일 비교		if MPPN	
		333	비교 파일의 전처리기 수 차이			
	변수 유사도	340	똑같은 변수 수		if not MEV    not EP	
		341	비교파일 모두 변수 존재 무		if NEV	
		342	변수가 있는 파일과 없는 파일 비교		if MEV	
		343	비교 파일의 변수 수 차이		if EV    MEV	
UI	첫화면 출력	400	파일 이름이 문자 10개 이상			single 1
	기준변경	410	마우스로 List 목록 선택			single
		411	키보드로 List 목록 선택			single 2
	Tag Cloud 선택	420	상세 유사도 보기			single
		421	올렸다 내리기			single
		422	내리고 올리기			single
		423	흔들기			single
		424	여러 번 클릭			single 5
	Total		2,764,800,000	921,600,000	1200	128

2,160,000 → 144,000 → 2068 → 44



### 3. Category Partition Test

- Category Partition Test Case
  - 다른 전처리기이어도 유사도 100%
  - 전처리가 없으면 아예 진행되지 않음
  - 주석이 없는 경우 주석에 관한 유사도가 0%
  - 빈문서인 C파일과 함께 있으면 오류로 진행 불가
  - 한 줄에 길게 쓸 경우에 변수, 함수, 라인 수 계산에 오류

### 3. Category Partition Test

- Category Partition Test Case

**44개의 Test Case 중 12개 Fail로  
73%의 성공을 보임**





4

# Pairwise Combination Test

# 4. Pairwise Combination Test

## Pairwise Test Report

### Parameters:

▲▼ 파일디렉토리	소스파일만있는경우	폴더만 있는경우	소스파일과 폴더가 섞인		
▲▼ 파일갯수	파일이 존재 하지 않을?	1개	여러개		
▲▼ 파일종류	C언어 확장자 소스파일	C언어와 다른 확장자 소	숨김 파일의 경우	같은 이름 다른 형식 파	읽기 권한이 없는 파일
▲▼ 시작하기	시작하기 버튼 클릭				
▲▼ 코드라인	다단적코드	한줄 코드	다단적 코드 + 한줄 코드	빈문서	빈문서 + 다단적코드
▲▼ 주석	/* */만 있는 경우	//만 있는 경우	두가지 주석같이 있는경	주석이 없는 경우	주석이 없는 파일과 있는
▲▼ printf문	printf문 존재	printf문 없는경우	printf문의 존재가 섞인		
▲▼ 함수	기본적인 함수 존재	함수가 없는 경우	함수가 있는 파일과 없는	함수가 매개변수에 들어	
▲▼ 변수	기본적인 변수존재	타입이 같고 변수명이 다	변수가 없는 경우	매개변수가 쓰일때	변수가 있는 것과 없는
▲▼ 전처리	같은 전처리기 존재	전처리기 선언이 상이	전처리가 없는경우	전처리가 있는 것과	
▲▼ 주석 유사도	똑같은 주석수	비교파일 모두 주석 존재	주석이 없는 파일과 있는	비교 파일의 라인 수 차	
▲▼ 라인 유사도	똑같은 라인수	비교파일 모두 빈 파일	라인이 없는 파일과 있는	비교 파일의 라인 수 차	
▲▼ 함수 유사도	똑같은 함수 수	비교파일 모두 함수 존재	함수가 있는 파일과 없는	비교파일의 함수 수 차	
▲▼ 전처리기 유사도	똑같은 전처리기 수	비교파일 모두 전처리기	전처리가 있는 파일과	비교 파일의 전처리기	
▲▼ 변수유사도	똑같은 변수 수	비교파일 모두 변수 존재	변수가 있는 파일과 없는	비교 파일의 변수 수 차	

parameter

### Constraints:

# 4. Pairwise Combination Test

## Pairwise Test Report

### Constraints:

▲ ▼ custom constraint: ✕

파일디렉토리	is not	폴더만 있는 경우	and	+	✕
파일디렉토리	is not	소스파일과 폴더가 섞인 경우	and	+	✕
파일갯수	is not	파일이 존재 하지 않을 경우	and	+	✕
파일갯수	is not	1개	and	+	✕
파일종류	is not	숨김 파일의 경우	and	+	✕
파일종류	is not	읽기 권한이 없는 파일		+	✕

▲ ▼ if ✕

코드라인	is not	빈문서	then	+	✕
주석	is	/* */만 있는 경우	or	+	✕
주석	is	//만 있는 경우	or	+	✕
주석	is	두가지 주석같이 있는 경우	and	+	✕
함수	is	기본적인 함수 존재	or	+	✕
함수	is	함수가 매개변수에 들어갈때	and	+	✕
라인 유사도	is	비교 파일의 라인 수 차이		+	✕

▲ ▼ if ✕

코드라인	is	빈문서	then	+	✕
주석	is	주석이 없는 경우	and	+	✕
printf문	is	printf문 없는 경우	and	+	✕
함수	is	함수가 없는 경우	and	+	✕
변수	is	변수가 없는 경우	and	+	✕
라인 유사도	is	비교파일 모두 빈 파일		+	✕



## 4. Pairwise Combination Test

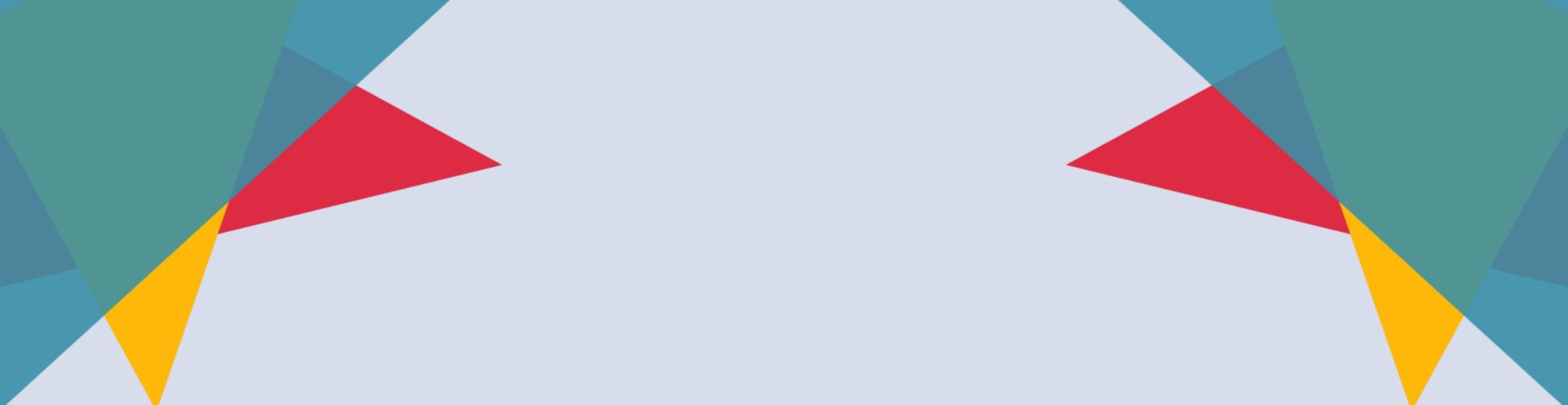
### Pairwise Test Report

- 파일 모두에 전처리가 존재하지 않으므로 유사도 비교의 지표로 삼으면 안 되는게, 그렇게 진행하여 같은 내용의 파일이더라도 유사도가 낮다고 표시된다. 또한 전혀 다른 파일을 추가하더라도 유사도 비교에 큰 영향을 끼치지 않는다.
- 주석을 제외한 나머지 부분을 한줄로 만들 경우(minifier), 함수와 변수, 라인 수, 전처리기 전 부분에 대해서 정확한 유사도 분석을 하지 못한다 / 파일 모두에 전처리가 존재하지 않으므로 유사도 비교의 지표로 삼으면 안 되는게, 그렇게 진행하여 같은 내용의 파일이더라도 유사도가 낮다고 표시된다. 또한 전혀 다른 파일을 추가하더라도 유사도 비교에 큰 영향을 끼치지 않는다.

## 4. Pairwise Combination Test

- Pairwise Test Case

**41개의 Test Case 중 9개 Fail로  
79%의 성공을 보임**



5

# Conclusion

## 5. Conclusion

Total Report

Testing or Review	Result
Spec Review	21
Category Partition Testing	12/44
Pairwise Testing	9/41
Brute Force Testing	9/27

# 5. Conclusion

## Communitication

The screenshot shows a Slack interface for the #report channel. The channel has 6 members. The conversation is in Korean and discusses a bug or issue related to a 'break' condition in code. The messages are as follows:

- Anonymous:** 안에서 if문으로 조건 맞으면 break되는 부분이 있는데 거꾸로 break가 안일어나는 인풋을 넣어줘서 for 문을 모두 순회하게 해준다든지
- jonghun 5:01 PM:** 아 그럼 테스트해야하는 메소드는 다 해놓은 건데 테스트 케이스가 부족해서 커버리지가 맞은 거예요? 제가 이해한 게 맞는지 잘 모르겠어요ㅠ
- jinh574 5:02 PM:** 코드 커버리지를 메소드별로 커버리지를 확인 하는 것이 아니라 명령어당 커버리지를 확인하는거라서 메소드 안에서 모든 명령어가 실행되고 지나갈 수 있도록
- ksch123200 5:02 PM:** 01094594089
- jinh574 5:02 PM:** 조건을 넣어서 확인을 해주어야해
- ksch123200 5:02 PM:** 이거로 전화해
- jonghun 5:04 PM:** 아 이해했어요! 메소드 안에 모든 라인이 통과할수 있어야 된다는 거네요
- ksch123200 5:04 PM:** 응응 야스
- jonghun 5:05 PM:** 일단 바로 말해볼게요 잠시만요..
- ksch123200 5:14 PM:** 레드마인에 일단 3개 일감 올려놨어! 확인하세요~~
- slipknot1com 5:14 PM:** 냐~

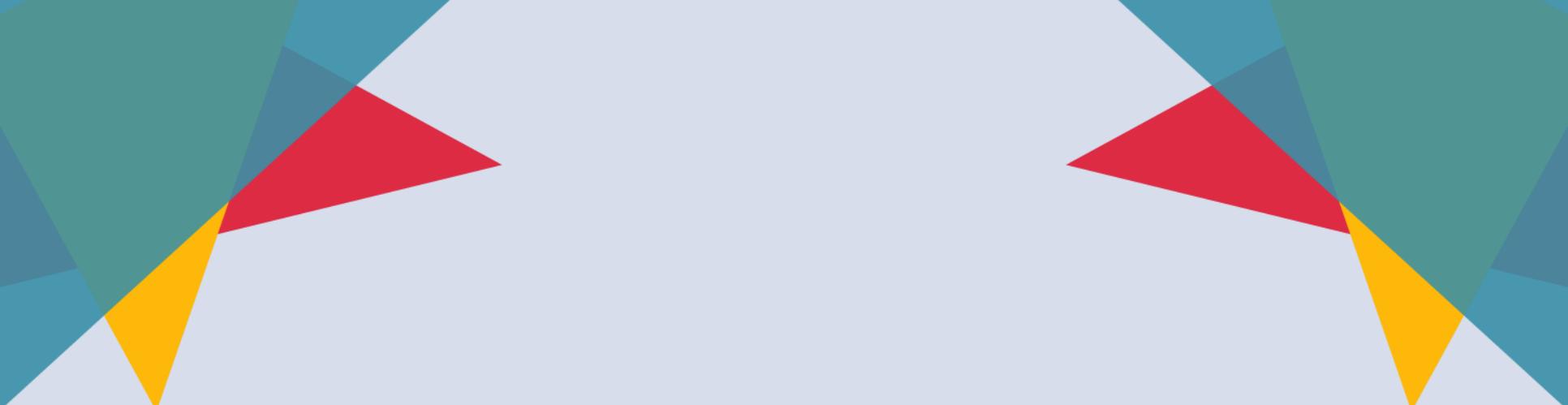
A separator line indicates the start of 'Yesterday'.

- Git BOT 12:18 PM:** Changoh pushed to branch master of 한지승/konkukcxse\_T1 (Compare changes)
  - 82d98a4a: junit 수정
  - chang92

## 5. Conclusion

### Total Report

- 전체적인 문서의 모호함 및 생략
  - 문서와 프로그램간의 상이
  - 지난번에 코드를 수정하면서 발생한 변동을 문서에 적용하지 않음
- 프로그램의 완성도 부족
  - 프로그램에 대한 정보 부족
    - 프로그램 사용 매뉴얼 필요
  - 다양한 코드로 테스트 필요
  - 오류 핸들링 필요



6

# CheckStyle

## 6. CheckStyle

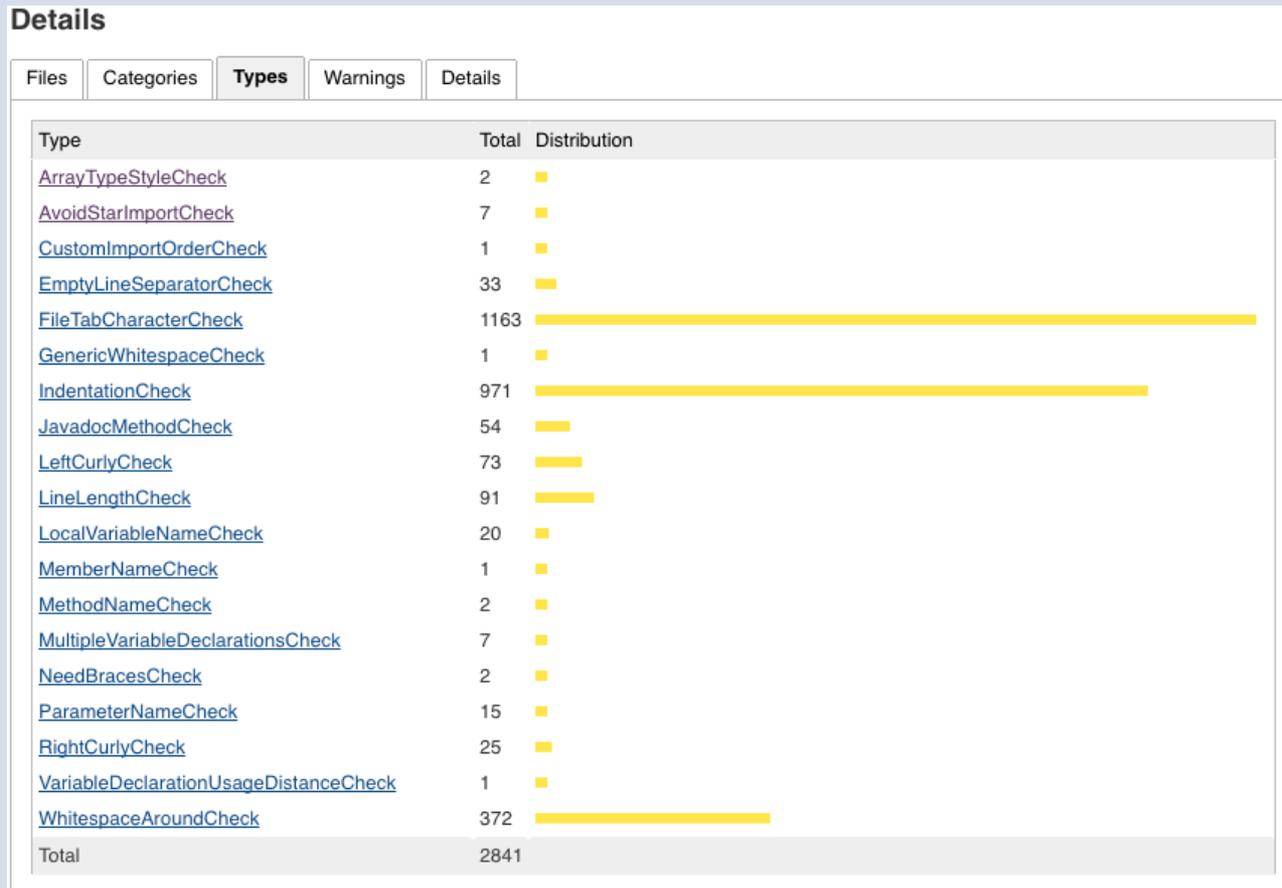
### About

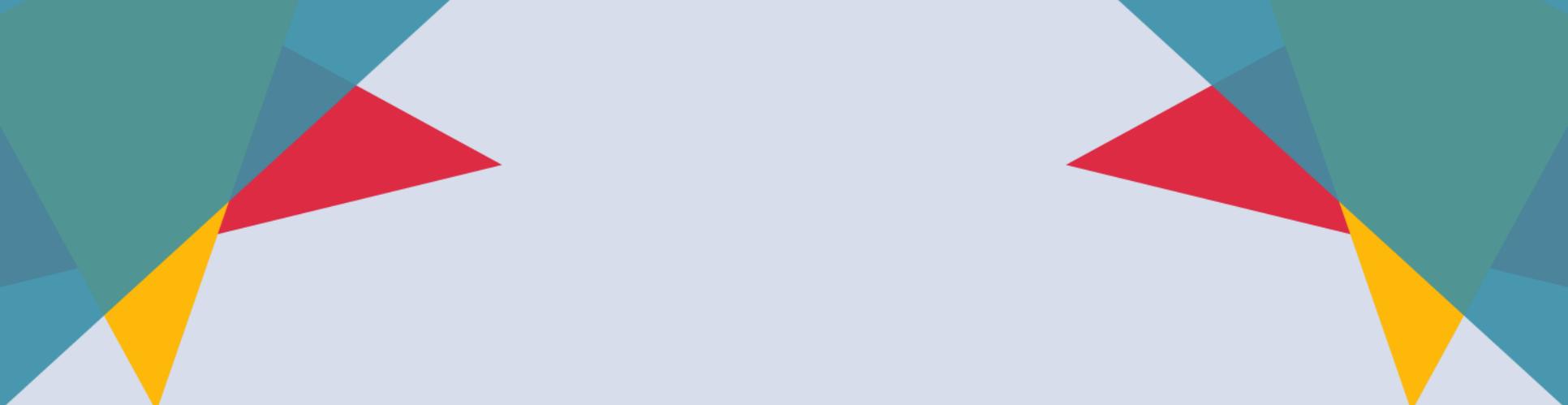
- CheckStyle은 코딩 표준에 맞게 소스코드를 작성하도록 도와주는 도구
- 개발자들이 잘 지키지는 않지만 중요한 것들을 지적해준다.  
(ex : private 설정)
- 협업 시에 코딩 스타일을 맞출 수 있다.



# 6. CheckStyle

## Report





**7** **PMD**

## 7. PMD

### About

- Program May Dependable의 약자
- 정해진 규칙에 따라 소스코드를 검사하여 위반사항을 찾아낼 수 있다.
- Eclipse-plugin으로 사용할 수 있으며, 별도로 커맨드 라인에서 사용 가능하다.
- 분석 대상은 java, javaScript, XML 등이 있다.
- PMD에서 제공하는 카테고리로부터 분석 대상 프로그램 언어인 Java에 맞는 카테고리 선정

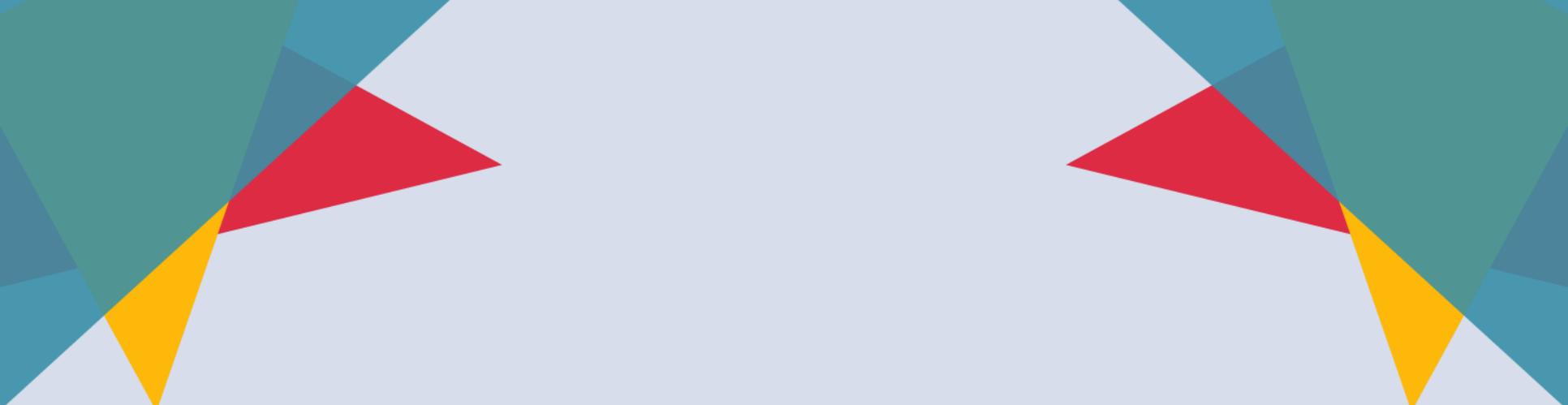
# 7. PMD

## Report

Type	Total	Distribution
<a href="#">CollapsibleIfStatements</a>	3	
<a href="#">JumbledIncrementer</a>	2	
<a href="#">UselessParentheses</a>	11	
Total	16	

### Summary

Total	High Priority	Normal Priority	Low Priority
16	0	<a href="#">16</a>	0



8

# FindBugs

## 8. FindBugs

### About

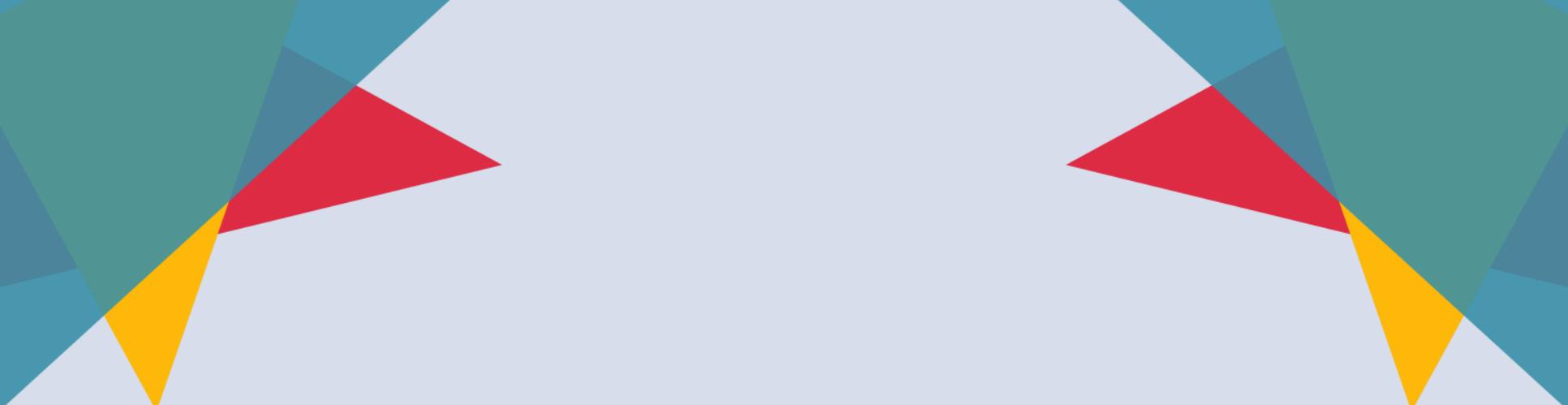
- 자바프로그램 분석도구로 메릴랜드 대학에서 개발
- Data flow, Control Flow 분석 등을 이용한다.
- 버그 패턴을 자동으로 찾아서 알려 준다.



# 8. FindBugs

## Report

Type	Total	Distribution
<a href="#">DM_DEFAULT_ENCODING</a>	1	
<a href="#">DM_STRING_CTOR</a>	3	
<a href="#">NM_METHOD_NAMING_CONVENTION</a>	2	
<a href="#">NP_NULL_ON_SOME_PATH_FROM_RETURN_VALUE</a>	1	
<a href="#">SIC_INNER_SHOULD_BE_STATIC</a>	1	
<a href="#">UUF_UNUSED_FIELD</a>	1	
<a href="#">UWF_UNWRITTEN_FIELD</a>	1	
Total	10	



9

# Code Coverage

## 9. Code Coverage

### About

- 자바를 위한 무료 코드 커버리지 라이브러리
- 코드 상에서 Green line, Yellow line, Red line으로 표시가 되며 Green line은 직접 코드를 통과한 경우, Yellow line은 분기에서 모든 조건을 체크하지 않고 지나가는 경우, Red line은 아예 코드를 통과하지 않은 경우이다.

# 9. Code Coverage

## Report

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
default		38%		51%	119	200	382	602	43	77	8	11

### default

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
<a href="#">Files</a>		92%		n/a	2	19	3	33	2	19	0	1
<a href="#">Main</a>		0%		n/a	2	2	3	3	2	2	1	1
<a href="#">Controller.new ActionListener() {...}</a>		0%		n/a	2	2	3	3	2	2	1	1
<a href="#">Controller.new ActionListener() {...}</a>		0%		n/a	2	2	7	7	2	2	1	1
<a href="#">Controller.new MouseAdapter() {...}</a>		0%		0%	4	4	5	5	2	2	1	1
<a href="#">Controller.new KeyAdapter() {...}</a>		0%		0%	4	4	6	6	2	2	1	1
<a href="#">Controller.RoundedBorder</a>		0%		n/a	5	5	13	13	5	5	1	1
<a href="#">Controller.new MouseAdapter() {...}</a>		0%		0%	9	9	23	23	4	4	1	1
<a href="#">Analyze</a>		72%		75%	24	73	36	160	1	12	0	1
<a href="#">Calculate</a>		63%		66%	24	39	28	94	8	14	0	1
<a href="#">Controller</a>		0%		0%	41	41	260	260	13	13	1	1
Total	2,665 of 4,269	38%	121 of 246	51%	119	200	382	602	43	77	8	11

## 9. Code Coverage

### Result

- Junit Code가 Analyze 코드와 Calculate Class에 관해서 작성이 되어서 해당 클래스 커버리지 계산만 의미가 있다. 또한 다른 Class들의 Junit Test가 되지 않은 상황이기 때문에 Junit테스트 결과에 대해서 신뢰하기 어렵다 판단된다.



THANK YOU

2nd Testing

2016.06.09