

# 쾌적한 Clone Checker

## Stage 2030

Team 1.

201111388 조연호

201211374 이창오

201211379 장중훈

201314196 양동혁

# Contents

- 1. Revise Plan**
- 2. Define Essential Use Cases**
- 3. Define Domain Model**
- 4. Define System Sequence Diagrams**
- 5. Define Operation Contracts**
- 6. Define State Diagram**
- 7. Refine System Test Case**
- 8. Traceability**

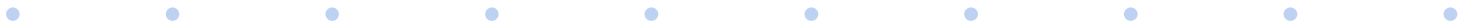


# 1. Revise Plan



# 1. Revise Plan

- Requirement Function 과 Use case 수정
- Use case diagram 수정
- Concept Model 수정



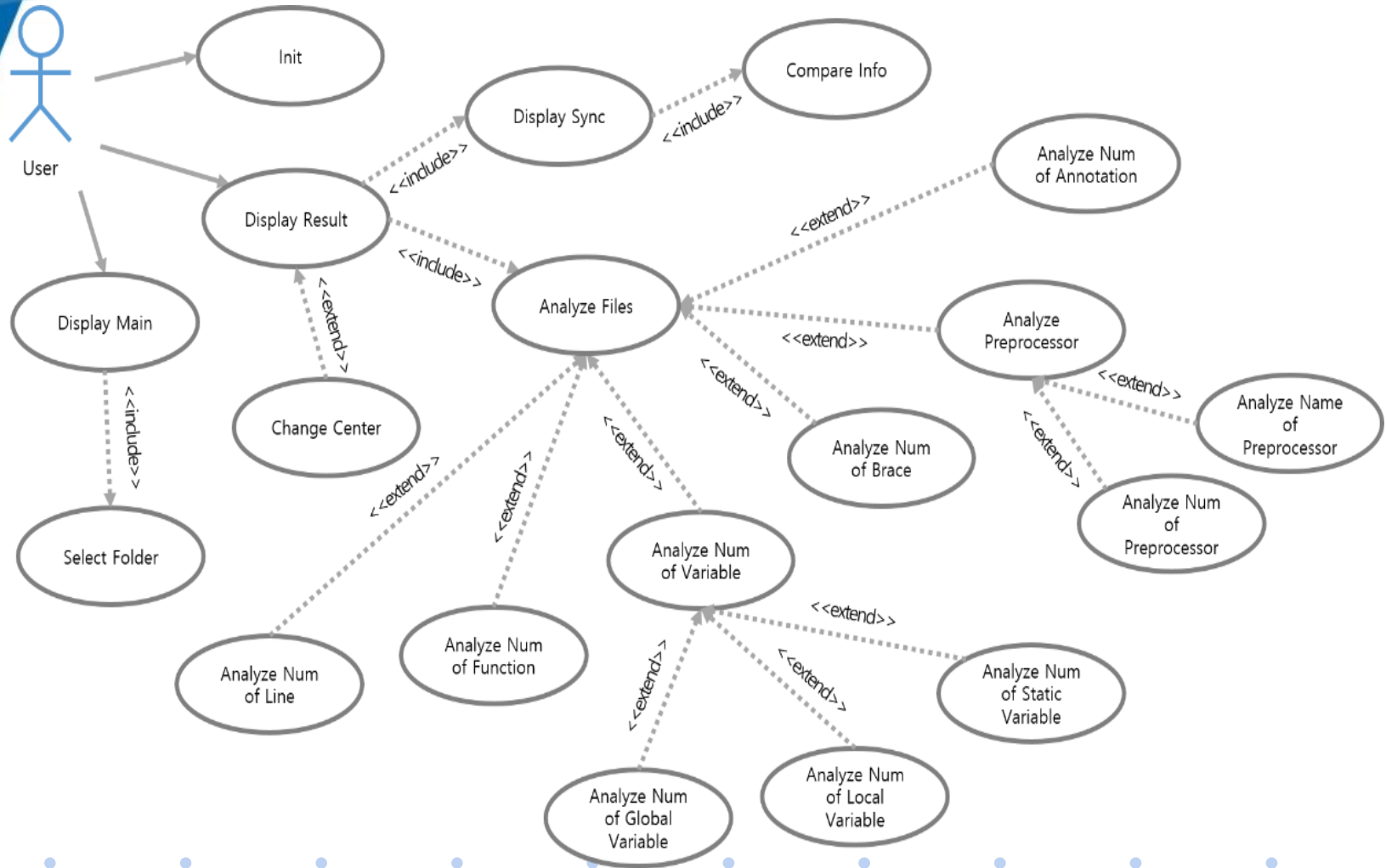
# 1. Revise Plan

Ref	Function	Ref	Function
R.1	Init	R.3.7	Analyze Num Of Brace
R.2	Display Main	R.3.8	Analyze Preprocessor
R.2.1	Select Folder	R.3.8.1	Analyze Num Of Preprocessor
R.3	Analyze Files	R.3.8.2	Analyze Name Of Preprocessor
R.3.1	Analyze Num Of Line	R.3.9	Analyze Num Of Annotation
R.3.2	Analyze Num Of Function	R.4	Display Result
R.3.3	Analyze Num Of Variable	R.4.1	Compare Info
R.3.4	Analyze Num Of Global Variable	R.4.2	Display Sync
R.3.5	Analyze Num Of Local Variable	R.4.3	Change Center
R.3.6	Analyze Num Of Static Variable		

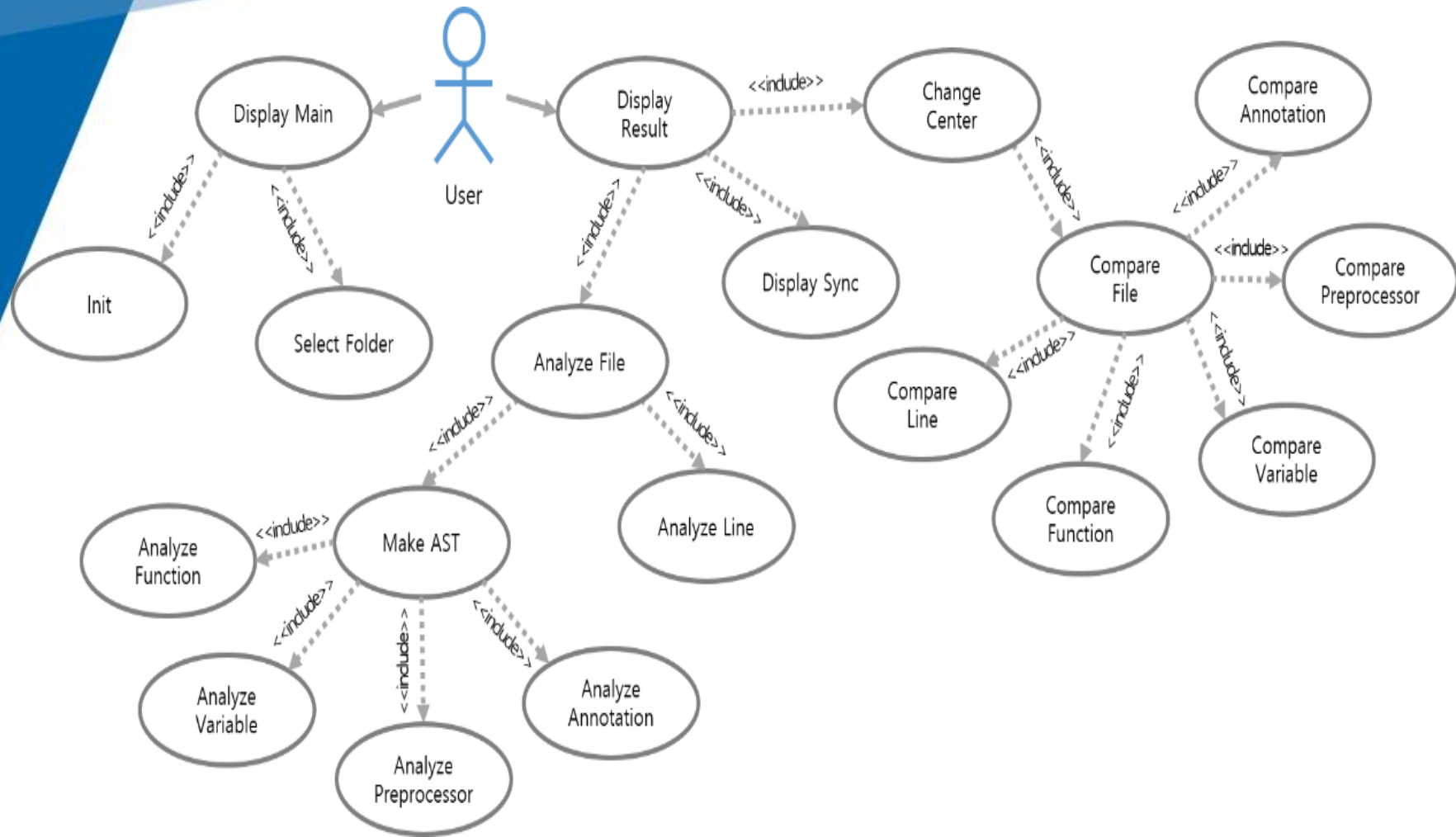
# 1. Revise Plan

Ref	Function	Ref	Function
R.1	Display Main	R.2.1.2.4	Analyze Annotation
R.1.1	Init	R.2.2	Change Center
R.1.2	Select Folder	R.2.2.1	Compare File
R.2	Display Result	R.2.2.1.1	Compare Line
R.2.1	Analyze File	R.2.2.1.2	Compare Function
R.2.1.1	Analyze Line	R.2.2.1.3	Compare Variable
R.2.1.2	Make AST	R.2.2.1.4	Compare Preprocessor
R.2.1.2.1	Analyze Function	R.2.2.1.5	Compare Annotation
R.2.1.2.2	Analyze Variable	R.2.3	Display Sync
R.2.1.2.3	Analyze Preprocessor		

# 1. Revise Plan



# 1. Revise Plan





# 1. Revise Plan

Upload	Analyze	Compare	Display



Interface	File	AST	Compare

# 2. Define Essential Use Cases



## 2. Define Essential Use case

<b>Use Case</b>	Display Main
<b>Actor</b>	User
<b>Purpose</b>	프로그램의 시작 화면을 출력한다.
<b>Overview</b>	사용자가 Clone Checker를 시작할 수 있도록 폴더 지정버튼과 '시작하기' 버튼을 출력한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.1.1(Init), R.1.2(Select Folder)
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A) : Actor , (S) : System <ol style="list-style-type: none"><li>1. (S) : Invoke 'Init'</li><li>2. (A) : 사용자가 폴더 지정 버튼을 누른다.</li><li>3. (S) : Invoke 'Select Folder'</li><li>4. (S) : 사용자가 지정한 폴더 경로를 출력한다.</li><li>5. (A) : 사용자가 '시작하기' 버튼을 누른다.</li></ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	폴더 경로가 정상적이지 않을 경우 오류 메시지를 팝업 창으로 띄운다.

## 2. Define Essential Use case

<b>Use Case</b>	Init
<b>Actor</b>	Event-based
<b>Purpose</b>	프로그램을 초기화 시킨다.
<b>Overview</b>	이미 저장되어 있는 정보들을 제거하는 과정을 통해 프로그램을 정상적으로 사용할 수 있는 환경을 만든다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.1(Display Main)
<b>Pre-Requisites</b>	R.1(Display Main)
<b>Typical Courses of Events</b>	(A) : Actor , (S) : System  1. (S) : 기존의 지정한 폴더 정보를 삭제한다. 2. (S) : 기존의 소스 코드를 분석한 정보들을 삭제한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	

## 2. Define Essential Use case

<b>Use Case</b>	Select Folder
<b>Actor</b>	User
<b>Purpose</b>	사용자가 폴더 경로를 지정할 수 있는 화면을 출력한다.
<b>Overview</b>	사용자가 Clone Checker에 사용될 소스 코드 파일이 있는 폴더 경로를 지정할 수 있도록 폴더 지정 화면을 출력한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.1 (Display Main)
<b>Pre-Requisites</b>	R.1 (Display Main)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (A) : 사용자가 소스 코드 파일이 있는 폴더를 지정하여 '선택' 버튼을 누른다. 2. (S) : 폴더 지정 화면을 닫고, 메인 화면으로 돌아간다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	폴더를 지정하지 않고 '선택' 버튼을 누를 경우 오류 메시지를 팝업 창으로 띄운다.

## 2. Define Essential Use case

<b>Use Case</b>	Display Result
<b>Actor</b>	User
<b>Purpose</b>	사용자가 소스 코드 파일의 비교 결과를 확인할 수 있는 화면을 출력한다.
<b>Overview</b>	기준 소스 코드 파일을 기준으로 태그 클라우드 형 비교 결과와 소스 코드 파일의 이름이 나열되어 있는 리스트 화면을 출력한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.1(Analyze File), R.2.2(Change Center), R.2.3(Display Sync)
<b>Pre-Requisites</b>	R.1(Display Main)
<b>Typical Courses of Events</b>	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>(S) : Invoke 'Analyze Files'</li> <li>(S) : Invoke 'Change Center'</li> <li>(S) : 비교한 결과가 담긴 태그 클라우드와 소스 코드 파일이 나열된 리스트, '이전 화면으로 돌아가기' 버튼을 화면에 출력한다.</li> </ol> <ol style="list-style-type: none"> <li>(A) : 태그 클라우드 또는 리스트에서 소스 코드 파일 이름을 선택한다.</li> <li>(S) : Invoke 'Change Center'</li> <li>(S) : 기준이 변경된 화면을 출력한다.</li> </ol> <ol style="list-style-type: none"> <li>(A) : 태그 클라우드에서 소스 코드 파일 이름 위에 마우스 커서를 올린다.</li> <li>(S) : Invoke 'Display Sync'</li> </ol> <ol style="list-style-type: none"> <li>(A) : '이전 화면으로 돌아가기' 버튼을 누른다.</li> <li>(S) : 결과 화면 창을 종료한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Analyze File
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일을 분석한다.
<b>Overview</b>	소스 코드 파일을 분석하여 일치율 계산에 필요한 정보로 가공한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2 (Display Result), R.2.1.1(Analyze Line), R.2.1.2(Make AST)
<b>Pre-Requisites</b>	R.2 (Display Result)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : Invoke 'Analyze Line' 2. (S) : Invoke 'Make AST'
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Analyze Line
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 라인 수를 분석한다.
<b>Overview</b>	소스 코드 파일의 라인 수를 분석하여 값으로 저장한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.1 (Analyze File)
<b>Pre-Requisites</b>	R.2.1 (Analyze File)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 소스 코드 파일의 라인 수를 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A



## 2. Define Essential Use case

<b>Use Case</b>	Make AST
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 AST(Abstract Syntax Tree) 를 생성한다.
<b>Overview</b>	소스 코드 파일의 함수와 변수, 전처리기, 주석을 분석할 수 있도록 AST를 생성한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.1 (Analyze File), R.2.1.2.1 (Analyze Function), R.2.1.2.2 (Analyze Variable) R.2.1.2.3 (Analyze Preprocessor) R.2.1.2.4 (Analyze Annotation)
<b>Pre-Requisites</b>	R.2.1 (Analyze File)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  <ol style="list-style-type: none"> <li>1. (S) : 소스 코드 파일을 AST로 변환한다.</li> <li>2. (S) : Invoke 'Analyze Function'</li> <li>3. (S) : Invoke 'Analyze Variable'</li> <li>4. (S) : Invoke 'Analyze Preprocessor'</li> <li>5. (S) : Invoke 'Analyze Annotation'</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Analyze Function
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 함수의 개수와 이름을 분석한다.
<b>Overview</b>	생성된 AST 에서 함수의 개수와 이름을 분석하여 값으로 저장한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.1.2 (Make AST)
<b>Pre-Requisites</b>	R.2.1.2 (Make AST)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 전달받은 AST에서 함수를 분석한 후 함수의 개수와 이름을 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Analyze Variable
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 변수의 개수와 이름을 분석한다.
<b>Overview</b>	생성된 AST에서 변수의 개수와 이름을 분석하여 값으로 저장한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.1.2 (Make AST)
<b>Pre-Requisites</b>	R.2.1.2 (Make AST)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 전달받은 AST에서 함수를 분석한 후 변수의 개수와 이름을 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Analyze Preprocessor
<b>Actor</b>	Event –based
<b>Purpose</b>	소스 코드 파일의 전처리기의 개수와 이름을 분석한다.
<b>Overview</b>	생성된 AST에서 전처리기의 개수와 이름을 분석하여 값으로 저장한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.1.2 (Make AST)
<b>Pre-Requisites</b>	R.2.1.2 (Make AST)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 전달받은 AST에서 함수를 분석한 후 전처리기의 개수와 이름을 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Analyze Annotation
<b>Actor</b>	Event –based
<b>Purpose</b>	소스 코드 파일의 주석의 개수를 분석한다.
<b>Overview</b>	생성된 AST에서 주석의 개수를 분석하여 값으로 저장한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.1.2 (Make AST)
<b>Pre-Requisites</b>	R.2.1.2 (Make AST)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 전달받은 AST에서 함수를 분석한 후 주석의 개수를 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Change Center
<b>Actor</b>	User
<b>Purpose</b>	기준이 되는 소스 코드 파일을 변경한다..
<b>Overview</b>	비교 기준이 되는 소스 코드 파일을 변경한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2 (Display Result), R.2.2.1(Compare File)
<b>Pre-Requisites</b>	R.2 (Display Result),
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (A) : 태그 클라우드 또는 리스트에서 소스 코드 파일 이름을 선택한다. 2. (S) : 사용자가 선택한 소스 코드 파일로 기준을 변경한다. 3. (S) : Invoke 'Compare File'
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Compare File
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일을 비교한다.
<b>Overview</b>	분석하여 가공한 정보를 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.2(Change Center), R.2.2.1.1(Compare Line), R.2.2.1.2(Compare Function), R.2.2.1.3(Compare Variable), R.2.2.1.4(Compare Preprocessor) , R.2.2.1.5(Compare Annotation)
<b>Pre-Requisites</b>	R.2.2 (Change Center)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : Invoke 'Compare Line' 2. (S) : Invoke 'Compare Function' 3. (S) : Invoke 'Compare Variable' 4. (S) : Invoke 'Compare Preprocessor' 5. (S) : Invoke 'Compare Annotation'
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Compare Line
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 라인 수를 비교한다.
<b>Overview</b>	라인 수를 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.2.1 (Compare File)
<b>Pre-Requisites</b>	R.2.2.1 (Compare File)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 기준 소스 코드 파일과 비교 소스 코드 파일의 라인 수를 비교하여 일치율을 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A



## 2. Define Essential Use case

<b>Use Case</b>	Compare Function
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 함수의 개수와 이름을 비교한다.
<b>Overview</b>	함수의 개수와 이름을 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.2.1 (Compare File)
<b>Pre-Requisites</b>	R.2.2.1 (Compare File)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 기준 소스 코드 파일과 비교 소스 코드 파일의 함수의 개수와 이름을 비교하여 일치율을 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Compare Variable
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 변수의 개수와 이름을 비교한다.
<b>Overview</b>	변수의 개수와 이름을 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.2.1 (Compare File)
<b>Pre-Requisites</b>	R.2.2.1 (Compare File)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 기준 소스 코드 파일과 비교 소스 코드 파일의 변수의 개수와 이름을 비교하여 일치율을 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Compare Preprocessor
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 전처리기의 개수와 이름을 비교한다.
<b>Overview</b>	전처리기의 개수와 이름을 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.2.1 (Compare File)
<b>Pre-Requisites</b>	R.2.2.1 (Compare File)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 기준 소스 코드 파일과 비교 소스 코드 파일의 전처리기의 개수와 이름을 비교하여 일치율을 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2. Define Essential Use case

<b>Use Case</b>	Compare Annotation
<b>Actor</b>	Event-based
<b>Purpose</b>	소스 코드 파일의 주석의 개수를 비교한다.
<b>Overview</b>	주석의 개수를 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2.2.1 (Compare File)
<b>Pre-Requisites</b>	R.2.2.1 (Compare File)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (S) : 기준 소스 코드 파일과 비교 소스 코드 파일의 주석의 개수를 비교하여 일치율을 변수에 저장한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

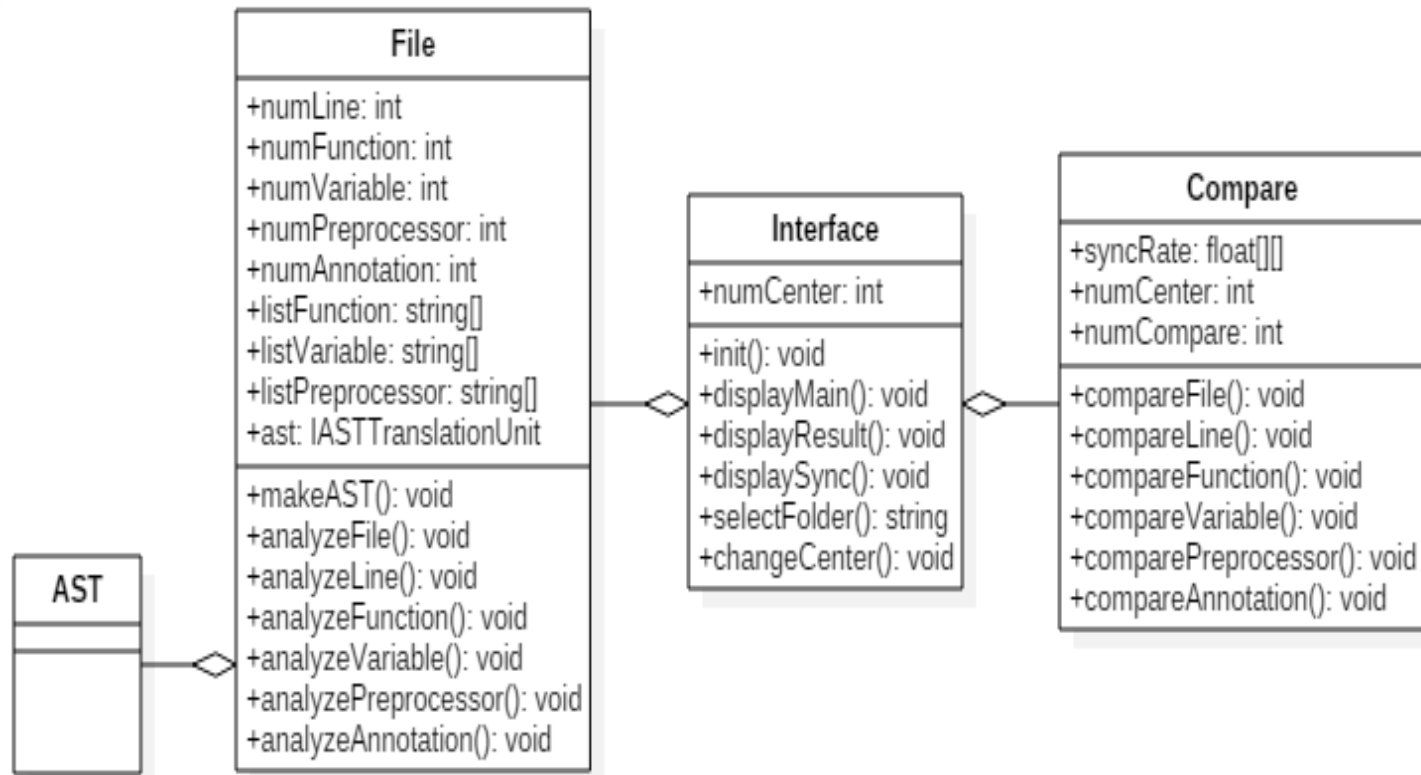
## 2. Define Essential Use case

<b>Use Case</b>	Display sync
<b>Actor</b>	User
<b>Purpose</b>	일치율과 유사 항목을 출력한다.
<b>Overview</b>	기준 소스 코드 파일과 비교 기준 소스 코드 파일의 일치율과 유사 항목을 말풍선 형태로 출력한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	R.2(Display Result)
<b>Pre-Requisites</b>	R.2(Display Result)
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System  1. (A) : 사용자가 태그 클라우드에 있는 소스 코드 파일 이름 위에 마우스 커서를 올린다.  2. (S) : 기준 소스 코드 파일과의 일치율과 유사항목을 결과 화면 창에 말풍선으로 출력한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

# 3. Define Domain Model



# 3. Define Domain Model



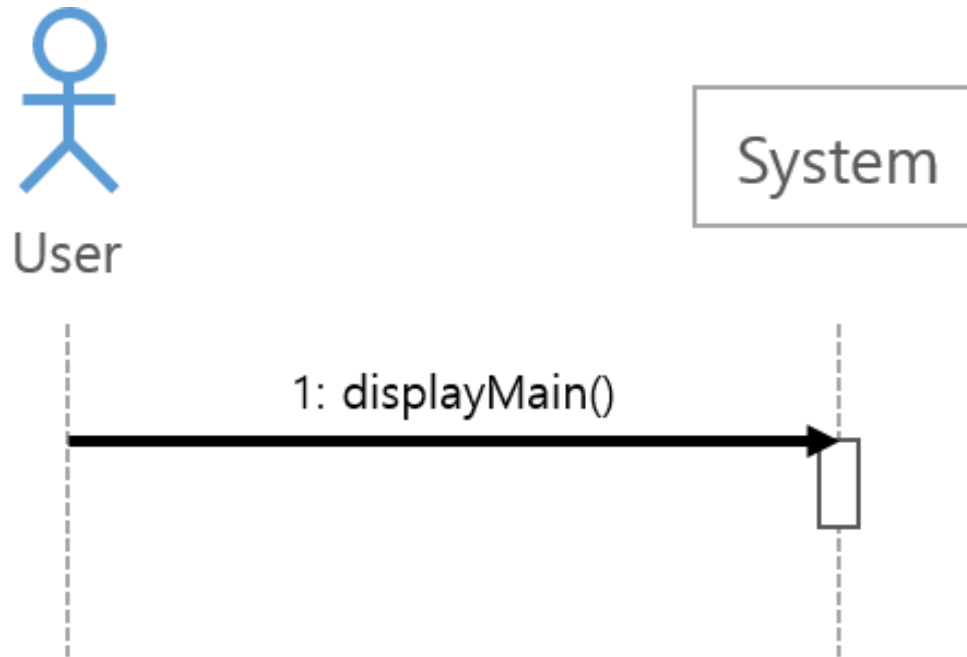
# 4. Define System Sequence Diagrams





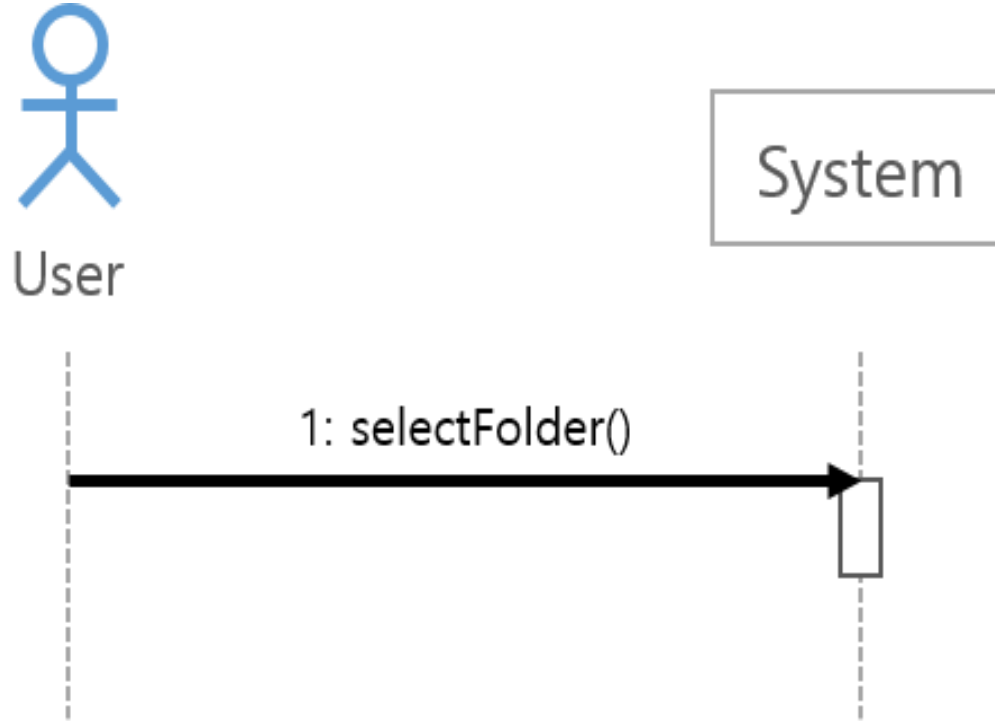
# 4. Define System Sequence Diagrams

## Use case – Display Main



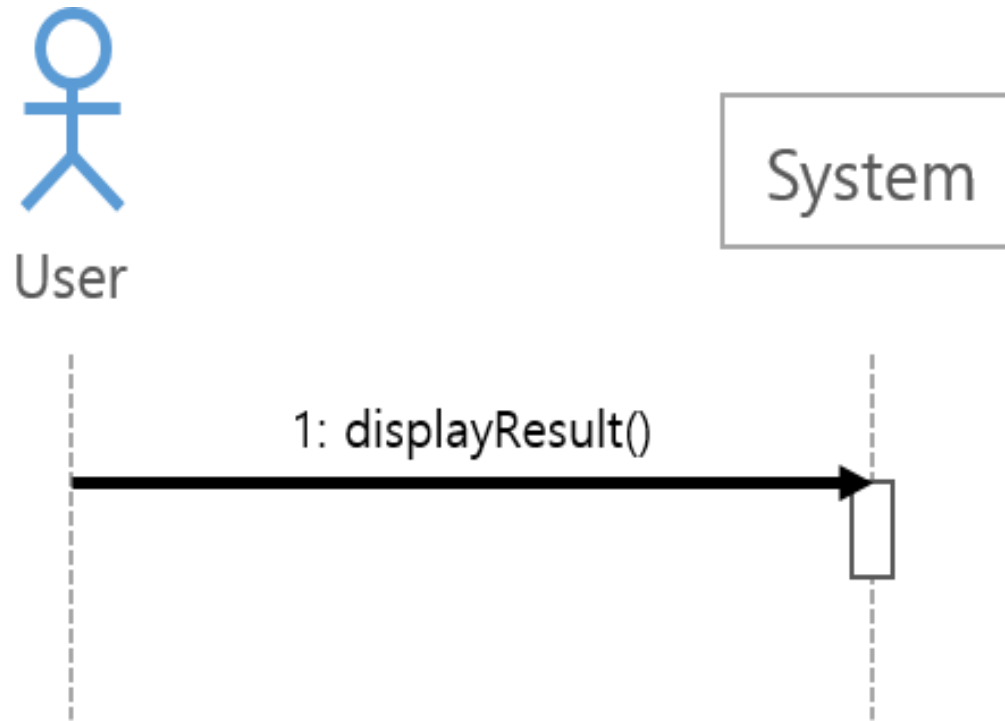
# 4. Define System Sequence Diagrams

## Use case – Select Folder



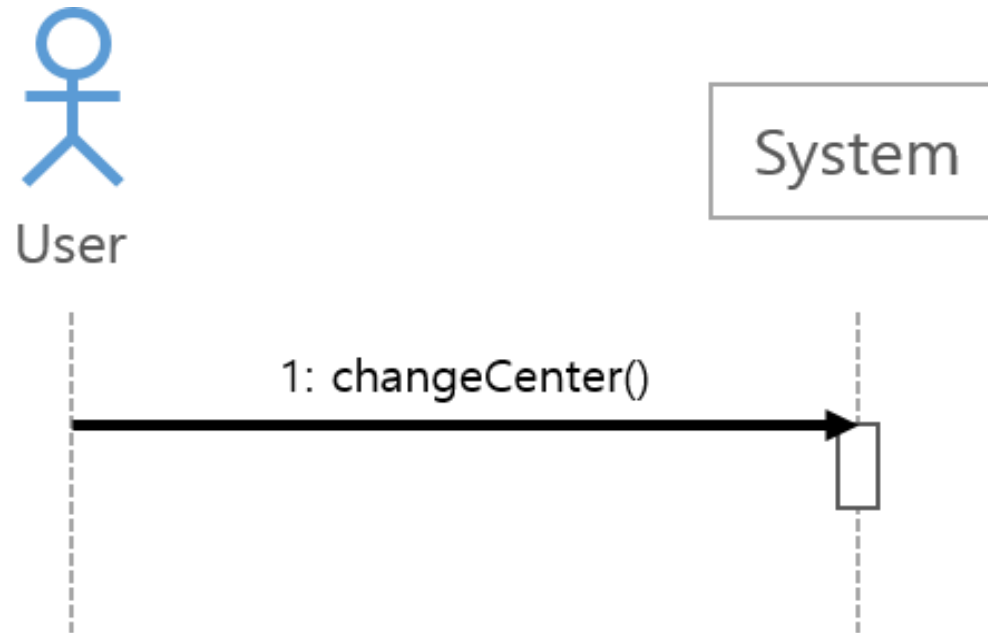
# 4. Define System Sequence Diagrams

## Use case – Display Result



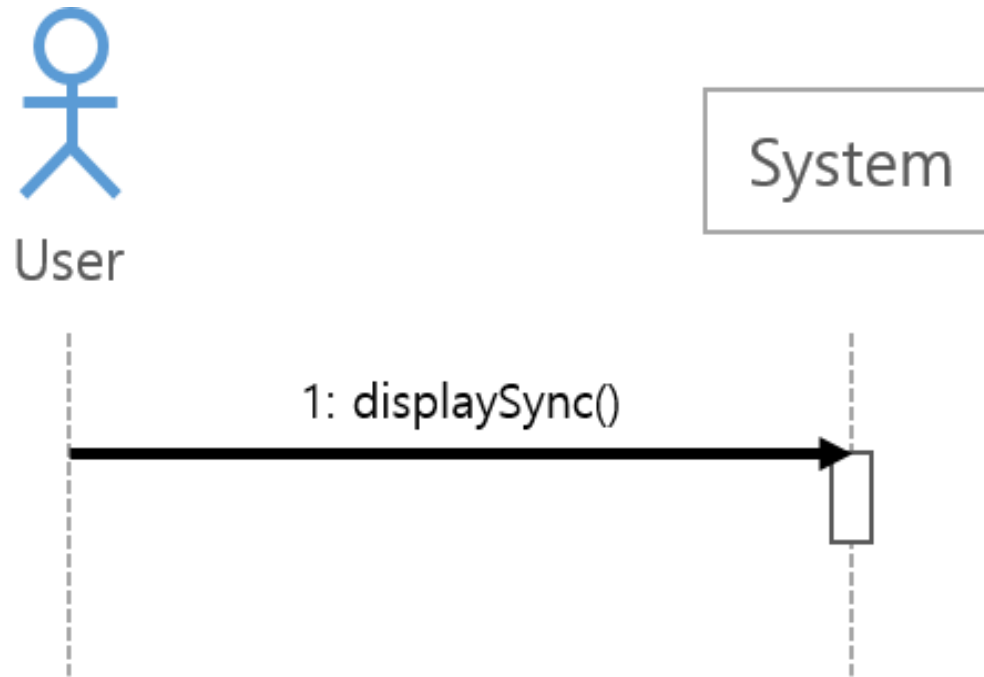
# 4. Define System Sequence Diagrams

## Use case – Change Center



# 4. Define System Sequence Diagrams

## Use case – Display Sync



# 5. Define Operation Contracts



## 6. Define Operation Contracts

<b>Name</b>	displayMain()
<b>Responsibilities</b>	폴더를 지정할 수 있는 버튼과 '시작하기' 버튼이 제공되는 메인 화면을 출력한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.1.1, R.1.2 Use Case : Init, Select Folder
<b>Notes</b>	화면 중앙에 폴더 지정 창을 불러오는 버튼이 있다. 폴더 지정이 정상적으로 완료될 경우 폴더 경로가 출력되며, 하단에 있는 '시작하기' 버튼이 활성화된다.
<b>Exception</b>	비정상적인 폴더 경로일 경우 '시작하기' 버튼을 누르면 오류 메시지가 팝업 창으로 나타난다.
<b>Output</b>	폴더를 지정할 수 있는 버튼과 '시작하기' 버튼이 제공되는 메인 화면을 출력한다.
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	Invoke 'Init' 결과 화면 창을 출력한다.



## 6. Define Operation Contracts

<b>Name</b>	Init()
<b>Responsibilities</b>	프로그램 을 초기화한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.1 Use Case : Display Main
<b>Notes</b>	Clone Checker 가 정상적으로 수행할 수 있도록 프로그램을 초기화한다. 이전에 Clone Checker가 수행된 적이 있다면 다시 수행할 수 있도록 프로그램을 초기화 한다.
<b>Exception</b>	N/A
<b>Output</b>	프로그램의 각종 변수들이 초기화된다.
<b>Pre-Conditions</b>	'Display Main'에 의해 호출된다.
<b>Post-Conditions</b>	N/A





## 6. Define Operation Contracts

<b>Name</b>	selectFolder()
<b>Responsibilities</b>	비교할 소스 코드 파일들이 들어있는 폴더를 지정하는 창을 출력한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.1 Use Case : Display Main
<b>Notes</b>	사용자가 소스 코드 파일이 들어있는 폴더를 지정한다.
<b>Exception</b>	비정상적인 폴더 경로일 경우 폴더 지정 창이 닫히지 않는다.
<b>Output</b>	메인 화면 창에 사용자가 지정한 경로가 나타난다.
<b>Pre-Conditions</b>	사용자가 메인 화면 창에서 폴더 지정 버튼을 누른다.
<b>Post-Conditions</b>	N/A



## 6. Define Operation Contracts

<b>Name</b>	displayResult()
<b>Responsibilities</b>	기준 소스 코드 파일을 기준으로 태그 클라우드형 비교 결과 화면을 출력한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.1, R.2.2, R.2.3 Use Case : Analyze File, Change Center, Display Sync
<b>Notes</b>	<ul style="list-style-type: none"><li>-화면 왼쪽에는 소스 코드 파일 간의 일치율을 쉽게 보여주는 태그 클라우드가 있다.</li><li>-화면 오른쪽에는 소스 코드 파일 이름이 나열된 리스트가 있다.</li><li>-화면 하단에는 메인 화면으로 돌아갈 수 있는 '이전 화면으로 돌아가기' 버튼이 있다.</li><li>-태그 클라우드 가운데에는 기준이 되는 소스 코드 파일 이름이 나타나며, 그 주위는 일치율이 높은 순으로 소스 코드 파일 이름이 나타난다.</li><li>-사용자가 태그 클라우드 또는 리스트에서 소스코드 파일 이름을 누르면 기준 소스 코드 파일이 변경되고, 그에 따라 비교 결과도 변경된다.</li><li>-사용자가 태그 클라우드에 있는 소스 코드 파일 이름 위에 마우스 커서를 올리면 기준 소스 코드 파일과의 일치율과 유사 항목이 말풍선 형태로 표시된다.</li></ul>
<b>Exception</b>	N/A
<b>Output</b>	태그 클라우드와 리스트, '이전 화면으로 돌아가기' 버튼이 있는 결과 화면 창을 출력한다.
<b>Pre-Conditions</b>	사용자가 메인 화면 창에서 '시작하기' 버튼을 누른다.
<b>Post-Conditions</b>	Invoke 'Change Center'

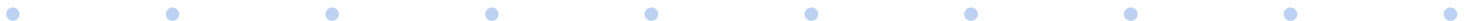
## 6. Define Operation Contracts

<b>Name</b>	analyzeFile()
<b>Responsibilities</b>	소스 코드 파일을 분석하여 일치율 계산에 필요한 정보로 가공한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2, R.2.1.1., R.2.1.2 Use Case : Display Result, Analyze Line, Make AST
<b>Notes</b>	소스 코드 파일을 분석하는 다양한 메소드를 호출한다.
<b>Exception</b>	소스 코드 파일의 경로가 비정상적일 경우 오류 메시지가 팝업 창으로 나타난다.
<b>Output</b>	N/A
<b>Pre-Conditions</b>	'Display Result'에 의해 호출된다.
<b>Post-Conditions</b>	Invoke 'Analyze Line' Invoke 'Make AST'



## 6. Define Operation Contracts

<b>Name</b>	analyzeLine()
<b>Responsibilities</b>	소스 코드 파일의 라인 수를 분석한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.1 Use Case : Analyze File
<b>Notes</b>	소스 코드 파일의 라인 수를 분석하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	소스 코드 파일의 라인 수를 변수에 저장한다.
<b>Pre-Conditions</b>	'Analyze File'에 의해 호출된다.
<b>Post-Conditions</b>	File.numLine에 소스 코드 파일의 라인 수를 저장한다.



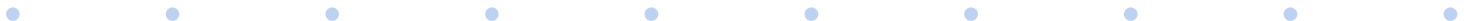
# 6. Define Operation Contracts

<b>Name</b>	makeAST()
<b>Responsibilities</b>	소스 코드 파일의 AST(Abstract Syntax Tree)를 생성한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.1, R.2.1.2.1, R.2.1.2.2, R.2.1.2.3, R.2.1.2.4 Use Case : Analyze File, Analyze Function, Analyze Variable, Analyze Preprocessor, Analyze Annotation
<b>Notes</b>	소스 코드 파일의 AST(Abstract Syntax Tree)를 생성하여 다양한 메소드에서 활용할 수 있도록 한다.
<b>Exception</b>	N/A
<b>Output</b>	소스 코드 파일의 AST(Abstract Syntax Tree)를 만들어 변수에 저장한다.
<b>Pre-Conditions</b>	'Analyze File'에 의해 호출된다.
<b>Post-Conditions</b>	File.ast에 소스 코드 파일의 AST(Abstract Syntax Tree)를 저장한다. Invoke 'Analyze Function' Invoke 'Analyze Variable' Invoke 'Analyze Preprocessor' Invoke 'Analyze Annotation'



## 6. Define Operation Contracts

<b>Name</b>	analyzeFunction()
<b>Responsibilities</b>	생성된 AST(Abstract Syntax Tree)에서 함수의 개수와 이름을 분석한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.1.2 Use Case : Make AST
<b>Notes</b>	소스 코드 파일에 있는 함수의 개수와 이름을 분석하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	소스 코드 파일에 있는 함수의 개수와 이름을 변수에 저장한다.
<b>Pre-Conditions</b>	'Make AST'에 의해 호출된다.
<b>Post-Conditions</b>	File.numFunction에 소스 코드 파일에 있는 함수의 개수를 저장한다. File.listFunction에 소스 코드 파일에 있는 함수의 이름을 저장한다.



## 6. Define Operation Contracts

<b>Name</b>	analyzeVariable()
<b>Responsibilities</b>	생성된 AST(Abstract Syntax Tree)에서 변수의 개수와 이름을 분석한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.1.2 Use Case : Make AST
<b>Notes</b>	소스 코드 파일에 있는 변수의 개수와 이름을 분석하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	소스 코드 파일에 있는 변수의 개수와 이름을 변수에 저장한다.
<b>Pre-Conditions</b>	'Make AST'에 의해 호출된다.
<b>Post-Conditions</b>	File.numVariable에 소스 코드 파일에 있는 변수의 개수를 저장한다. File.listVariable에 소스 코드 파일에 있는 변수의 이름을 저장한다.



## 6. Define Operation Contracts

<b>Name</b>	analyzePreprocessor()
<b>Responsibilities</b>	생성된 AST(Abstract Syntax Tree)에서 전처리기의 개수와 이름을 분석한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.1.2 Use Case : Make AST
<b>Notes</b>	소스 코드 파일에 있는 전처리기의 개수와 이름을 분석하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	소스 코드 파일에 있는 전처리기의 개수와 이름을 변수에 저장한다.
<b>Pre-Conditions</b>	'Make AST'에 의해 호출된다.
<b>Post-Conditions</b>	File.numPreprocessor에 소스 코드 파일에 있는 전처리기의 개수를 저장한다. File.listPreprocessor에 소스 코드 파일에 있는 전처리기의 이름을 저장한다.





## 6. Define Operation Contracts

<b>Name</b>	analyzeAnnotation()
<b>Responsibilities</b>	생성된 AST(Abstract Syntax Tree)에서 주석의 개수를 분석한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.1.2 Use Case : Make AST
<b>Notes</b>	소스 코드 파일에 있는 주석의 개수를 분석하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	소스 코드 파일에 있는 주석의 개수를 변수에 저장한다.
<b>Pre-Conditions</b>	'Make AST'에 의해 호출된다.
<b>Post-Conditions</b>	File.numAnnotation에 소스 코드 파일에 있는 주석의 개수를 저장한다.



## 6. Define Operation Contracts

<b>Name</b>	changeCenter()
<b>Responsibilities</b>	비교 기준이 되는 소스 코드 파일을 변경한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2, R.2.2.1 Use Case : Display Result, Compare File
<b>Notes</b>	사용자가 선택한 소스 코드 파일을 기준으로 설정한다. 만약 처음 프로그램을 실행하였을 경우 첫 번째 소스 코드 파일을 기준으로 설정한다.
<b>Exception</b>	N/A
<b>Output</b>	결과 화면에 나타나는 비교 결과가 변경된다.
<b>Pre-Conditions</b>	사용자가 태그 클라우드 또는 리스트에서 소스 코드 파일 이름을 누른다.
<b>Post-Conditions</b>	Interface.numCenter에 기준이 되는 소스 코드 번호가 저장된다. Invoke 'Compare File'



## 6. Define Operation Contracts

<b>Name</b>	compareFile()
<b>Responsibilities</b>	분석하여 가공한 정보를 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.2, R.2.2.1.1, R.2.2.1.2, R.2.2.1.3, R.2.2.1.4, R.2.2.1.5 Use Case : Change Center, Compare Line, Compare Function, Compare Variable, Compare Preprocessor, Compare Annotation
<b>Notes</b>	소스 코드 파일을 비교하는 다양한 메소드를 호출한다.
<b>Exception</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	'Change Center'에 의해 호출된다.
<b>Post-Conditions</b>	Invoke 'Compare Line' Invoke 'Compare Function' Invoke 'Compare Variable' Invoke 'Compare Preprocessor' Invoke 'Compare Annotation'



## 6. Define Operation Contracts

<b>Name</b>	compareLine()
<b>Responsibilities</b>	라인 수를 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.2.1 Use Case : Compare File
<b>Notes</b>	기준 소스 코드 파일과 현재 비교중인 소스 코드 파일의 라인 수를 비교하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	기준 소스 코드 파일과 현재 비교 중인 소스 코드 파일의 라인 수 일치율을 변수에 저장한다.
<b>Pre-Conditions</b>	'Compare File'에 의해 호출된다.
<b>Post-Conditions</b>	Compare.syncRate에 라인 수 일치율을 저장한다.



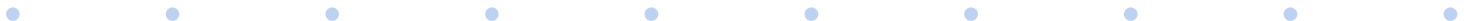
## 6. Define Operation Contracts

<b>Name</b>	compareFunction()
<b>Responsibilities</b>	함수의 개수와 이름을 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.2.1 Use Case : Compare File
<b>Notes</b>	기준 소스 코드 파일과 현재 비교중인 소스 코드 파일의 함수 개수와 이름을 비교하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	기준 소스 코드 파일과 현재 비교 중인 소스 코드 파일의 함수의 개수와 이름 일치율을 변수에 저장한다.
<b>Pre-Conditions</b>	'Compare File'에 의해 호출된다.
<b>Post-Conditions</b>	Compare.syncRate에 함수의 개수와 이름 일치율을 저장한다.



## 6. Define Operation Contracts

<b>Name</b>	compareVariable()
<b>Responsibilities</b>	변수의 개수와 이름을 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.2.1 Use Case : Compare File
<b>Notes</b>	기준 소스 코드 파일과 현재 비교중인 소스 코드 파일의 변수의 개수와 이름을 비교하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	기준 소스 코드 파일과 현재 비교 중인 소스 코드 파일의 변수의 개수와 이름 일치율을 변수에 저장한다.
<b>Pre-Conditions</b>	'Compare File'에 의해 호출된다.
<b>Post-Conditions</b>	Compare.syncRate에 변수의 개수와 이름 일치율을 저장한다.



## 6. Define Operation Contracts

<b>Name</b>	comparePreprocessor()
<b>Responsibilities</b>	전처리기의 개수와 이름을 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.2.1 Use Case : Compare File
<b>Notes</b>	기준 소스 코드 파일과 현재 비교중인 소스 코드 파일의 전처리기의 개수와 이름을 비교하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	기준 소스 코드 파일과 현재 비교 중인 소스 코드 파일의 전처리기의 개수와 이름 일치율을 변수에 저장한다.
<b>Pre-Conditions</b>	'Compare File'에 의해 호출된다.
<b>Post-Conditions</b>	Compare.syncRate에 전처리기의 개수와 이름 일치율을 저장한다.



## 6. Define Operation Contracts

<b>Name</b>	compareAnnotation()
<b>Responsibilities</b>	주석의 개수를 활용(비교)하여 일치율을 계산한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2.2.1 Use Case : Compare File
<b>Notes</b>	기준 소스 코드 파일과 현재 비교 중인 소스 코드 파일의 주석의 개수를 비교하여 변수에 저장한다.
<b>Exception</b>	N/A
<b>Output</b>	기준 소스 코드 파일과 현재 비교 중인 소스 코드 파일의 주석의 개수 일치율을 변수에 저장한다.
<b>Pre-Conditions</b>	'Compare File'에 의해 호출된다.
<b>Post-Conditions</b>	Compare.syncRate에 주석의 개수 일치율을 저장한다.





## 6. Define Operation Contracts

<b>Name</b>	displaySync()
<b>Responsibilities</b>	일치율과 유사 항목을 출력한다.
<b>Type</b>	System
<b>Cross Reference</b>	System Function : R.2 Use Case : Display Result
<b>Notes</b>	말풍선 왼쪽에는 기준 소스 코드 파일과 비교 소스 코드 파일의 일치율이 출력된다. 말풍선 오른쪽에는 기준 소스 코드 파일과 비교 소스 코드 파일의 유사 항목이 출력된다.
<b>Exception</b>	N/A
<b>Output</b>	일치율과 유사 항목을 말풍선 형태로 출력된다.
<b>Pre-Conditions</b>	사용자가 소스 코드 파일 이름 위에 마우스 커서를 올린다.
<b>Post-Conditions</b>	N/A

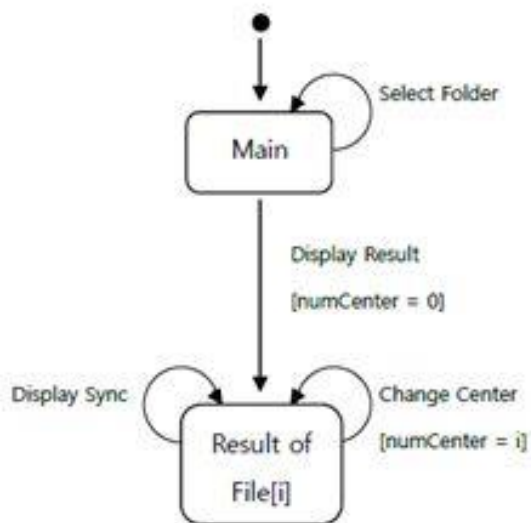


# 6. Define State Diagram

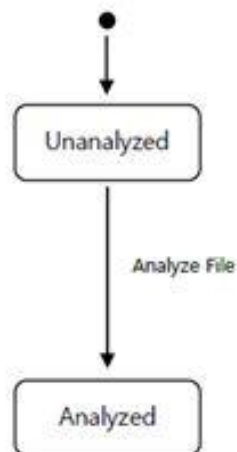


# 6. Define State Diagram

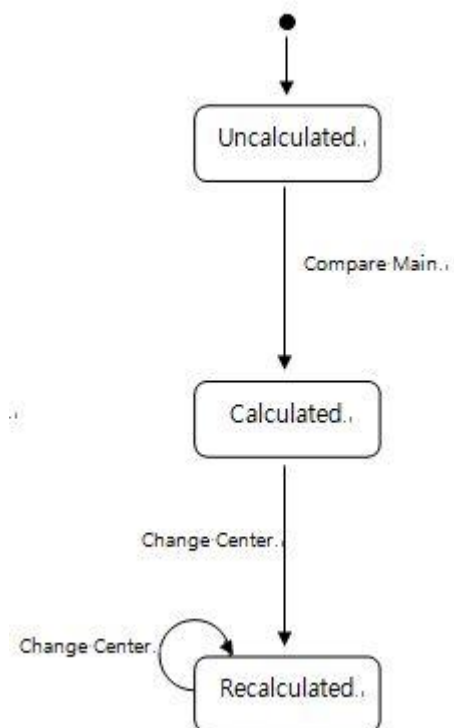
<State Diagram for "InterFace">



<State Diagram for "File">



<State Diagram for "Compare">

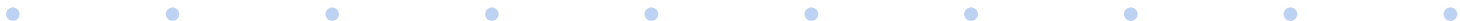


# 7. Refine System Test Case



# 7. Refine System Test Case

Use Case	Description
<b>1. Display Main</b>	폴더 지정 버튼과 '시작하기' 버튼이 출력되는지 확인한다.
	폴더 지정이 완료되었을 경우 지정한 폴더 경로가 정상적으로 출력되는지 확인한다.
	폴더 지정이 완료되었을 경우 '시작하기' 버튼이 활성화되는지 확인한다.
	비정상적인 폴더를 지정하였을 경우 오류 메시지가 팝업창으로 나타나는지 확인한다.
	'시작하기' 버튼을 눌렀을 경우 정상적으로 다음 화면으로 넘어가는지 확인한다.
	메인 화면이 1초 이내에 나타나는지 확인한다.
<b>2. Init</b>	메인 화면에서 이전에 지정했던 폴더 경로가 나타나지 않는지 확인한다.
	'시작하기' 버튼이 비활성화 되어있는지 확인한다.



# 7. Refine System Test Case

Use Case	Description
<b>3. Select Folder</b>	폴더 지정 화면이 나타나고, 정상적으로 지정되었는지 확인한다.
	비정상적인 폴더를 지정하였을 경우 메인 화면으로 돌아가지 않는지 확인한다.
	폴더 지정 화면이 1초 이내에 나타나는지 확인한다.
<b>4. Display Result</b>	화면 왼쪽에 태그 클라우드 형으로 비교 결과가 출력되는지 확인한다.
	비교 결과에 맞게 태그 클라우드에 소스코드 파일 이름과 일치율이 정상적으로 출력되는지 확인한다.
	화면 오른쪽에 소스 코드 파일 이름이 리스트로 출력되는지 확인한다.
	화면 아래쪽에 '이전 화면으로 돌아가기' 버튼이 출력되는지 확인한다.
	'이전 화면으로 돌아가기' 버튼을 눌렀을 때 경로가 화면이 사라지고, 메인 화면이 나타나는지 확인한다.
	태그 클라우드 또는 리스트에서 소스 코드 파일 이름을 선택했을 때 기준과 비교 결과가 변경 되는지 확인한다.
	태그 클라우드에서 소스 코드 파일 이름이 겹치게 출력되지 않는지 확인한다.

# 7. Refine System Test Case

Use Case	Description
	리스트에 소스 코드 파일 이름이 오름차순 정렬이 되어 있는지 확인한다.
	비교 결과 화면이 8초 이내에 나타나는지 확인한다.
<b>5. Analyze File</b>	소스 코드 분석 결과가 올바른지 확인한다.
	소스 코드 분석 시간이 5초 이내인지 확인한다.
<b>6. Analyze Line</b>	소스 코드 파일의 라인 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>7. Make AST</b>	소스 코드 파일의 함수와 변수, 전처리기, 주석의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
	소스 코드 파일의 함수와 변수, 전처리기의 이름을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>8. Analyze Function</b>	소스 코드 파일의 함수의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.



# 7. Refine System Test Case

Use Case	Description
	소스 코드 파일의 함수의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>9. Analyze Variable</b>	소스 코드 파일의 변수의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
	소스 코드 파일의 변수의 이름을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>10. Analyze Preprocessor</b>	소스 코드 파일의 전처리기의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
	소스 코드 파일의 전처리기의 이름을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>11. Analyze Annotation</b>	소스 코드 파일의 주석의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>12. Change Center</b>	태그 클라우드 또는 리스트에서 소스코드 파일 이름을 선택했을 경우 기준이 정상적으로 변경되는지 확인한다.
	기준이 변경되면서 비교 결과가 정상적으로 변경되는지 확인한다.



# 7. Refine System Test Case

Use Case	Description
	현재 기준인 소스 코드 파일 이름을 다시 선택했을 경우 비교가 다시 되지 않는지 확인한다.
	기준 소스 코드 파일이 2초 이내에 변경되는지 확인한다.
<b>13. Compare File</b>	비교 결과가 올바른지 확인한다.
	비교 결과 화면이 8초 이내에 나타나는지 확인한다.
<b>14. Compare Line</b>	소스 코드 파일의 라인 개수만 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>15. Compare Function</b>	소스 코드 파일의 함수의 개수를 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.



# 7. Refine System Test Case

Use Case	Description
	소스 코드 파일의 함수의 이름을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>16. Compare Variable</b>	소스 코드 파일의 변수의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
	소스 코드 파일의 변수의 이름을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>17. Compare Preprocessor</b>	소스 코드 파일의 전처리기의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
	소스 코드 파일의 전처리기의 이름을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
<b>18. Compare Annotation</b>	소스 코드 파일의 주석의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.



# 7. Refine System Test Case

Use Case	Description
19. Display Sync	비교 소스 코드 파일 이름에 마우스 커서를 올리면 일치율과 유사 항목이 말풍선 형태로 출력되는지 확인한다.
	마우스 커서를 올린 소스 코드 파일의 일치율과 유사 항목이 맞는지 확인한다.
	출력되는 문구가 말풍선을 벗어나지는 않는지 확인한다.
	일치율과 유사 항목이 1초 이내에 나타나는지 확인한다.



# 8. Traceability



# 8. Traceability

System Test Case		System Function		Use Case		Operation
1. 메인화면 출력 시험	→	R.1 Display Main	→	Display Main	→	1: displayMain()
2. 초기화 시험	→	R.1.1 Init	→	Init	→	2: init()
3. 폴더 지정 시험	→	R.1.2 Select Folder	→	Select Folder	→	3: selectFolder()
4. 결과 화면 출력 시험	→	R.2 Display Result	→	Display Result	→	4: displayResult()
5. 파일 분석 시험	→	R.2.1 Analyze File	→	Analyze File	→	5: analyzeFile()
6. 라인 개수 분석 시험	→	R.2.1.1 Analyze Line	→	Analyze Line	→	6: analyzeLine()
7. AST 생성 시험	→	R.2.1.2 Make AST	→	Make AST	→	7: makeAST()
8. 함수 분석 시험	→	R.2.1.2.1 Analyze Function	→	Analyze Function	→	8: analyzeFunction()
9. 변수 분석 시험	→	R.2.1.2.2 Analyze Variable	→	Analyze Variable	→	9: analyzeVariable()
10. 전처리기 분석 시험	→	R.2.1.2.3 Analyze Preprocessor	→	Analyze Preprocessor	→	10: analyzePreprocessor()
11. 주석 분석 시험	→	R.2.1.2.4 Analyze Annotation	→	Analyze Annotation	→	11: analyzeAnnotation()
12. 기준 파일 변경 시험	→	R.2.2 Change Center	→	Change Center	→	12: changeCenter()
13. 파일 비교 시험	→	R.2.2.1 Compare File	→	Compare File	→	13: compareFile()
14. 라인 개수 비교 시험	→	R.2.2.1.1 Compare Line	→	Compare Line	→	14: compareLine()
15. 함수 비교 시험	→	R.2.2.1.2 Compare Function	→	Compare Function	→	15: compareFunction()
16. 변수 비교 시험	→	R.2.2.1.3 Compare Variable	→	Compare Variable	→	16: compareVariable()
17. 전처리기 비교 시험	→	R.2.2.1.4 Compare Preprocessor	→	Compare Preprocessor	→	17: comparePreprocessor()
18. 주석 비교 시험	→	R.2.2.1.5 Compare Annotation	→	Compare Annotation	→	18: compareAnnotation()
19. 일치율 및 유사 항목 출력 시험	→	R.2.3 Display Sync	→	Display Sync	→	19: displaySync()



Thank you  
Any Question?

