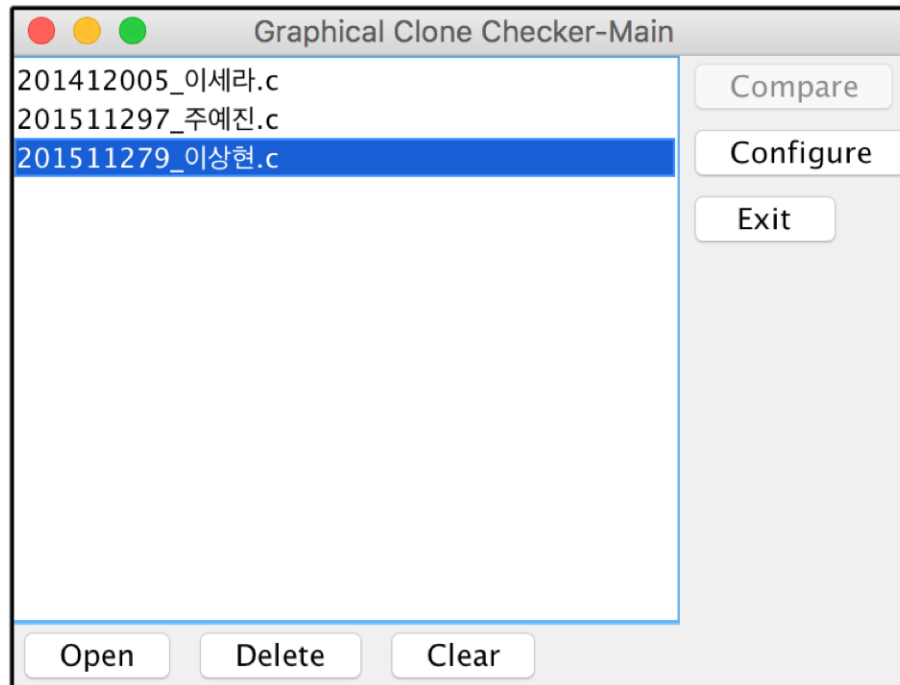


Graphical Clone Checker

T3. 김태현. 교수창. 이상규.

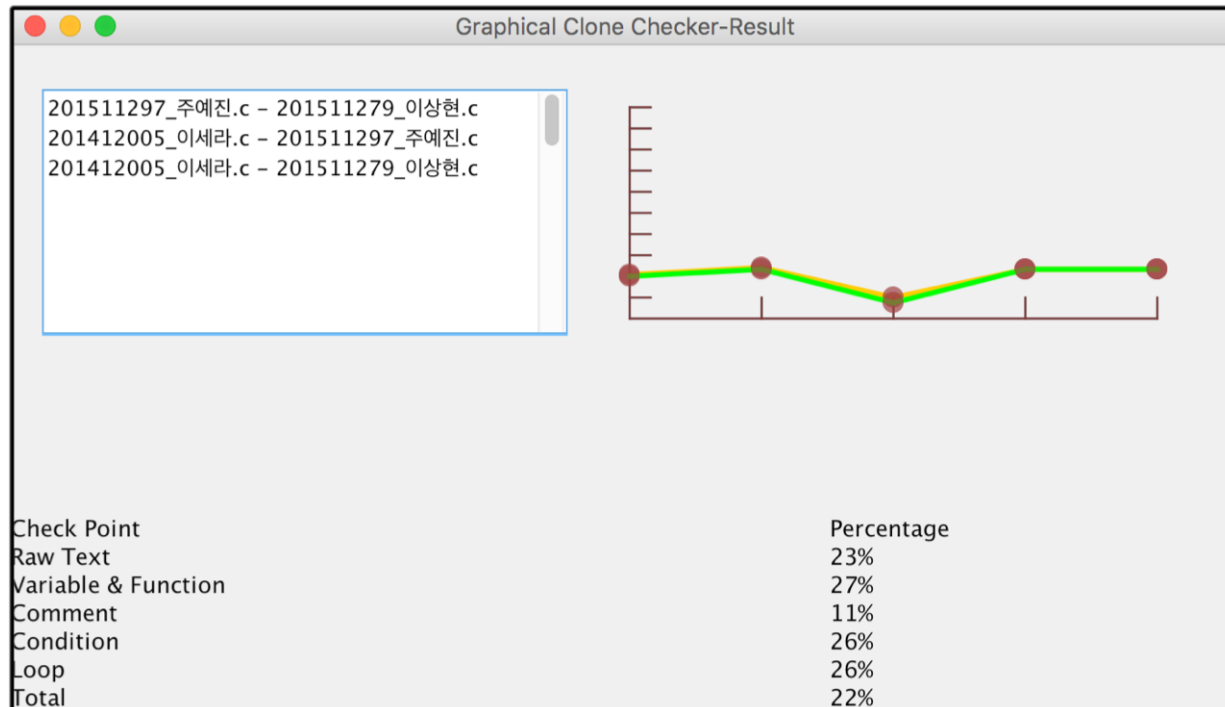
System Test 대응

Category	4.2 Failed Case Report 10. Compare 2번 이상 수행
Problem	2회 이상 Compare를 수행하였을 때부터 GUI에 오류가 생긴다.
Solution	한 번 Compare 버튼을 누른 후에는 버튼을 비활성화 시켰다.



System Test 대응

Category	4.2 Failed Case Report 4. 누구나 쉽게 사용할 수 있는 단순한 UI 인지 확인한다.
Problem	파일명이 길어지면 결과 창에서 확인 할 수가 없다.
Solution	결과 목록의 가로 길이를 늘린다.



Static Analysis 대응

1. Check Style 대응

Rule	Variable Declaration Usage Distance
Problem	Checks the distance between declaration of variable and its first usage.
Solution	코드의 가독성을 위해 수정하지 않는다.

Rule	Custom Import Order
Problem	Checks that the groups of import declarations appear in the order specified by the user. If there is an import but its group is not specified in the configuration such an import should be placed at the end of the import list.
Solution	import의 순서에 관련되어 필요성을 느끼지 못하여 수정하지 않는다.

Static Analysis 대응

2. Find Bug 대응

Rule	Dodgy - Dead store to local variable
Problem	<p>This instruction assigns a value to a local variable, but the value is not read or used in any subsequent instruction. Often, this indicates an error, because the value computed is never used.</p> <p>Note that Sun's javac compiler often generates dead stores for final local variables. Because FindBugs is a bytecode-based tool, there is no easy way to eliminate these false positives.</p>
Solution	사용되지 않는 지역 변수를 제거한다.

Rule	Dodgy - Write to static field from instance method
Problem	<p>This instance method writes to a static field. This is tricky to get correct if multiple instances are being manipulated, and generally bad practice.</p>
Solution	구현상에 어려움이 많기 때문에 수정하지 않는다.

OOPT 소감

- 단계별 문서화를 통해 소프트웨어 모델링 기법을 배우며, 소프트웨어 제작에 문서화의 역할이 꽤 용이하고 필수적이라는 점을 배울 수 있었다.
- 기존에 소프트웨어 제작시에 따로 문서를 정리하지 않고 머리속으로 설계하고 바로 작업에 들어가다 보면 프로젝트를 뒤엎거나 리팩토링 하는 경우가 많이 생겼는데, 문서 작업 후 개발을 하면서 소프트웨어 모델링의 중요성이 와닿았다.

OOPT 장단점

- 장점

1. 프로그래밍을 들어가기 전에 문서를 작성하여 기능에 대한 구체적인 정의와 이해가 가능하다.
2. 객체지향 언어에 특화되어 클래스 다이어그램, 시퀀스 다이어그램으로 문서를 통해 프로그램 제작시 흐름을 직관적으로 알 수 있다.

- 단점

1. 문서의 양이 많으므로 프로젝트의 크기에 따라 오히려 문서 작성 Cost가 커질 수 있다.
2. 문서 상의 내용과 실제 프로그램 구현시의 내용이 차이가 있을 수 있으므로 단계별 문서에 대한 Confirm이 정확히 이루어져야 한다.

SV팀과의 Co-Work 소감

생각지도 못한 부분에 대해 Testing이 있어서
당황스러운 부분이 많았지만,
그만큼 빠진 부분을 한번 더 생각해 보게 되었다.

또한 SV팀의 적절한 피드백 & 일정관리로 원만하게
프로젝트 수행이 가능하였다.

SV팀과의 Co-Work 장단점

- 장점

1. 제 3자 입장에서 테스트를 하기 때문에
생각지도 못한 오류를 찾는데 큰 도움이 되었다.

- 단점

1. 의사소통에 대한 오류로, 서로의 관점이 달라 이해의 차이가 생겼다.
2. 작은 항목들에 대해 너무나 많은 내용을 잡아내면서 대응이 쉽지 않았다.

감사합니다