Analysis & Design, Implementation, and Test

Document

개체이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

TEAM 16

**김은경 (20163927) 김소연 (20165417) 김수진 (20160342) 남유선 (20163228)**

**손승표 (20162581) 윤신영 (20163657)**

1. **Domain model (class diagram)**

텍스트이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

(you can see more clear diagram images at Github ‘Diagram’ folder!)

1. **Software architecture & Design model**

텍스트이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

텍스트, 지도이(가) 표시된 사진

높은 신뢰도로 생성된 설명[System sequence diagram]

1. **Major design decision**

// 수진, 유선 part

[How to compare the texts?]

By using the algorithms for longest common subsequence(LCS) problem and sequence alignment, we compute the matrix C which stores the LCS length and compare two panels based on the matrix.

[How to distinguish two cases? 1. Different strings are in same line index in each panel 2. The string is in only one panel]

For case 1, we multiply (-1) to the index of string. For case 2, the string is stored in the panel which the string is in and the blank value(0) is stored in another panel. This is for computing the start and end indices of blocks. When the getter function is called, returned data is the arraylist that all elements of left panel are changed to negative integer and all elements of right panel are changed to positive integer to distinguish two panel’s indices.

[What data type would be returned to Merge class after comparing in FileComparator class?]

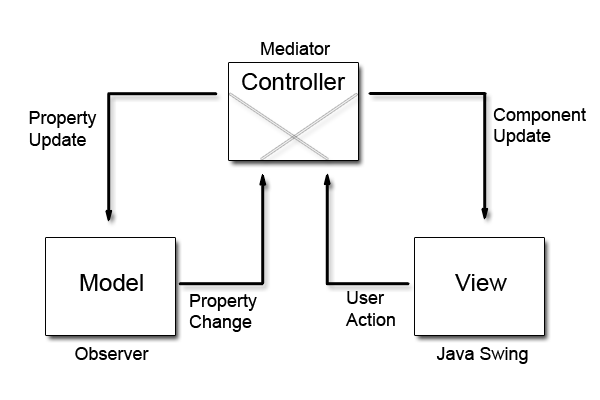
In FileComparator class, the difference of two panel is stored in each integer type arraylist and the block, which store the block’s start and end index, is stored in integer array type arraylist. When merge call getter functions by FileComparator object, all elements in left panel’s arraylists are changed to negative integer, and all in right panel’s are changed to positive integer.

[What data type would be returned after calling traverse function?]

There is a cursor for traverse function which is an integer type index for the block array and it represents current location of cursor. If traverse function is called, the block array and the cursor are returned.

1. **Explanation about how MVC concept was applied**

Here is the picture of MVC model. When we design this program, we referred to this picture. The point is that ‘Model’ can’t be directly connected to ‘View’.



We define ‘view’ classes as {MainView, PanelView, TextLineNumber, CompareTable, CompareTableRenderer}, ‘model’ classes as {Merge, TextEditorModel, FileComparator}, and ‘controller’ as {MergeController, TextEditorController}.

As you can see these two class diagrams (domain model and design model), all the classes in view are not directly connected to any class in model. In other words, controller is the only mediator between model and view. Also, when state of model is changed, controller receives the signal and sends it to the view so that the view can apply the change and show it to the user.

So, concept of MVC model is applied to our program.

1. **Explanation about how OO design principles were applied (with code examples)**

[Encapsulation]

We set member variables as private or protected (there are only few exceptions), also we made public getter & setter methods to access those private or protected member variables. For some methods which are used only within in one class, we set them as private, otherwise, we set methods as private.

[Inheritance]

We use inheritance when we design ‘view’ classes. Here are the pictures of code.

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

1. **Explanation about how our program was designed to be testable by Unit-test tools**
2. **Usage of program & screen shots of examples**

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

1. 프로그램의 시작화면. 활성화되어 있는 Load 버튼을 클릭하여 파일 load

스크린샷, 컴퓨터이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

1-1. load한 파일이 중복인 경우 중복임을 알리는 message dialog 출력

스크린샷, 컴퓨터, 실내이(가) 표시된 사진

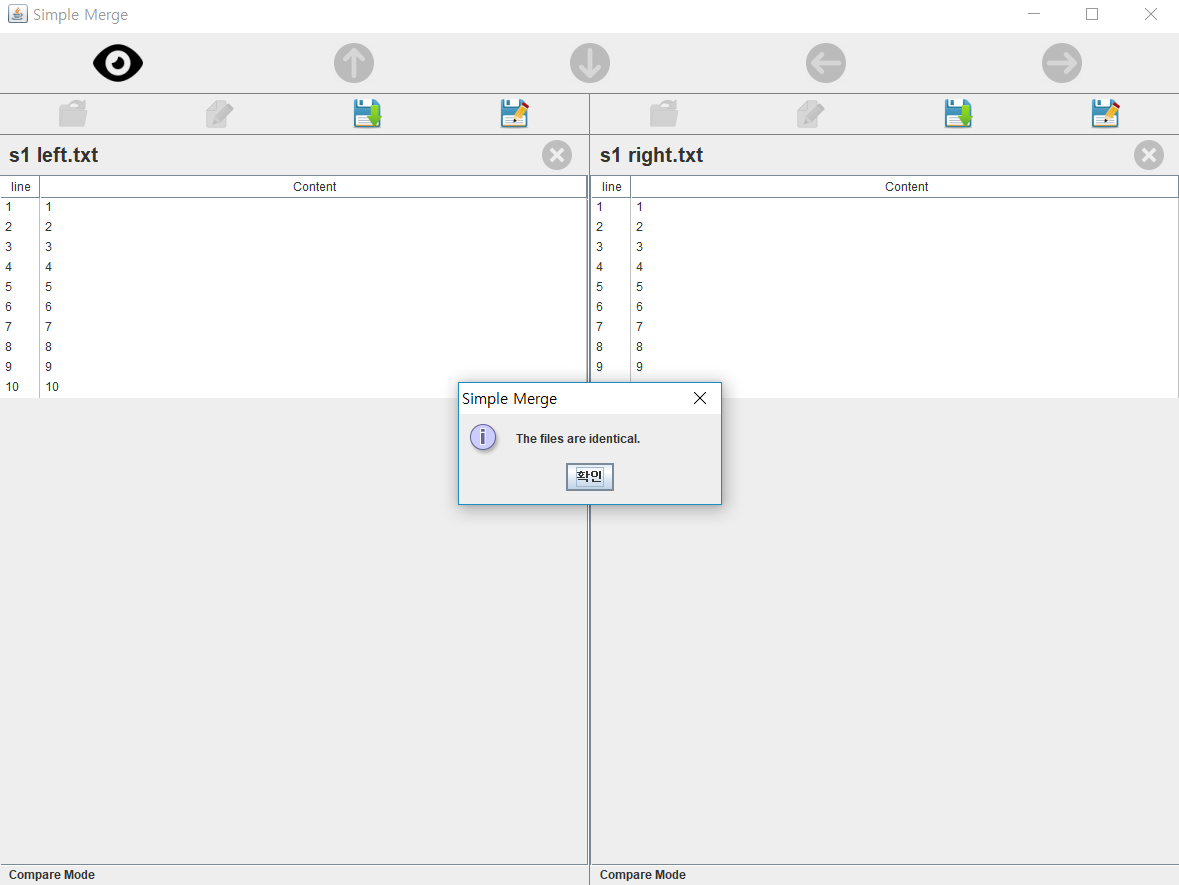
매우 높은 신뢰도로 생성된 설명

2. 파일이 Load된 상태. 패널의 밑부분에는 현재 패널의 모드가 표시되고(스크린 샷은 View Mode) 현재 커서가 있는 줄의 index는 빨간색으로 표시됨.

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

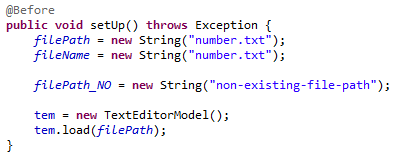
3. compare 버튼을 클릭한 후, Traverse와 Merge 버튼이 활성화됨. 줄 정렬을 위해 추가된 공백줄은 회색으로 표시되고, index는 ‘-‘로 표시됨. 더 이상 traverse할 수 있는 블록이 없으면 해당 버튼 비활성화.



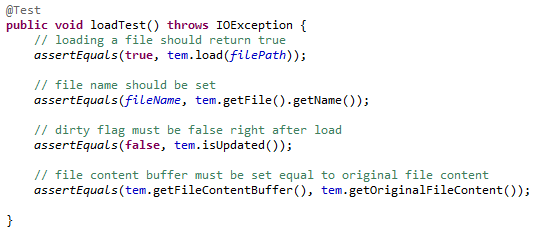
3-1. compare 연산 결과 두 텍스트파일이 동일한 경우 이를 알리는 message dialog 출력

1. **Functional unit test cases and results**

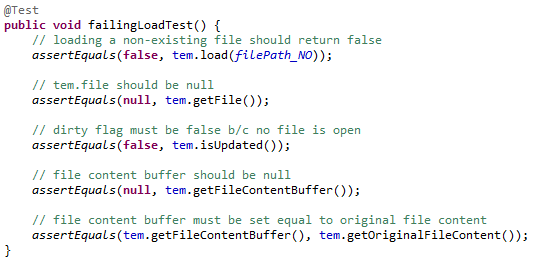
* setUp – test case로 사용할 파일의 경로와 이름을 설정하고, 테스트 하고자 하는 클래스의 객체를 만든다.



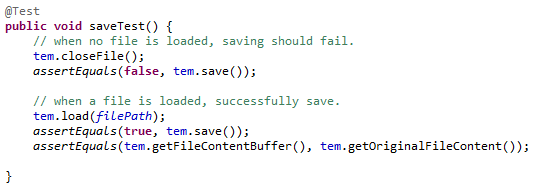
* load() 함수를 테스트한다. 파일을 성공적으로 불러오면 true를 반환하며, 파일 이름, dirty flag, 그리고 파일의 내용이 알맞게 설정됐는지 확인한다.



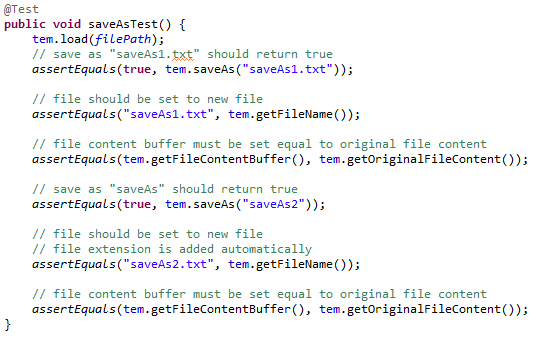
* load() 함수가 실패하는 경우를 테스트한다. 존재하지 않는 파일을 불러오려고 하였을 때 false를 반환하고, 파일의 이름과 내용이 적절하게 null로 설정됐는지 확인한다.



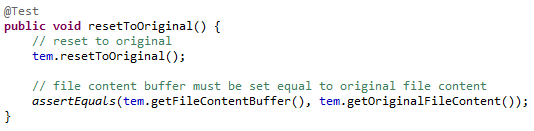
* save() 함수를 테스트한다. 파일이 열려 있지 않은 경우에는 false를 리턴하고, 그렇지 않으면 저장하고 true를 리턴한다. fileContentBuffer가 잘 설정되었는지 확인한다.



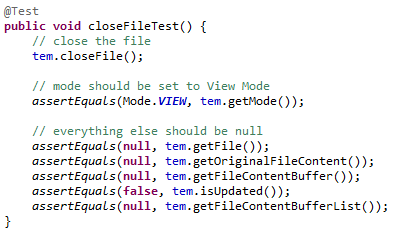
* saveAs() 함수를 테스트한다. Save()함수와 같은 방식으로 테스트하며, 저장 시 파일 확장자를 입력하지 않은 경우 열려 있던 파일의 확장자를 자동으로 붙여 주는지 확인한다.



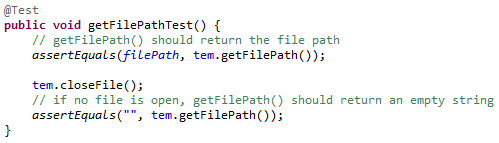
* resetToOriginal() 함수를 테스트한다. 함수가 호출되었을 때 fileContentBuffer가 originalFileConten로 set 되었는지 확인한다.



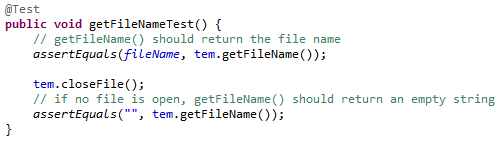
* closeFile() 함수를 테스트한다. 파일이 닫혔을 때 모드가 제대로 설정 되었는 지 확인하고, 나머지 필드들이 null로 지정되었는지 확인한다.



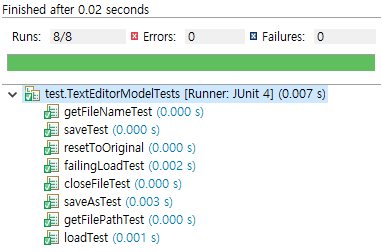
* getFilePath() 함수를 확인한다. 파일이 없을 때 null을 반환하지 않는지 테스트한다..



* getFileName() 함수를 확인한다. 파일이 없을 때 null을 반환하지 않는지 테스트한다.



* 테스트 결과 모든 테스트가 성공적으로 종료되었다.



1. **System test cases and results**
2. **(If any) functional limitations**