

Java Term Project

프로그램 개발용 GUI 기반 IDE 개발

제출일 : 2017.12.17.

학번	이름	기여도
2016 xxxxx	박xx	33.3%
2016 xxxxx	송xx	33.3%
2013 xxxxx	김xx	33.3%
		%

1. S/W Requirements

- 자바 프로그램 개발용 GUI 기반의 IDE 프로그램이다.

[이미 존재하는 파일 입력 후 "Open" 버튼 선택]

- 파일이 존재할 경우 파일 내용을 "Editing Window"에 기술, 그렇지 않다면 "Result Window"에 오류 메시지 출력

[새로운 자바 프로그램을 "Editing Window"에서 작성]

- 새로운 코드를 작성한 후 저장할 파일명을 기입한 후 "Save" 버튼 선택
- 파일이 존재할 경우 "Result Window"에 오류 메시지 출력, 정상적으로 저장된 경우 성공 메시지를 "Result Window"에 출력

["Compile" 버튼 선택 시]

- "Compile" 버튼 선택 시, Open 한 자바 컴파일 실행
- 컴파일 실행 결과, 오류가 발생하면 오류 내용을 "Result Window"에 표시하고 정상적으로 컴파일이 된다면 컴파일 성공 메시지를 출력
- 컴파일 오류가 발생한 경우 "SaveError" 버튼 선택 시, "자바 파일 이름.error"로 저장

[파일 내용 변경]

- "Save" 버튼 시, 저장할 파일 이름이 제공되지 않을 경우 현재 Open된 파일에 overwrite함. 파일명이 주어진다면 그 파일명으로 저장.

[파일 삭제]

- .java 파일과 .class파일을 삭제. 정상적으로 삭제 된 경우 "Result Window"에 성공 메시지를 출력하고 오류 발생 시 오류 메시지를 출력.

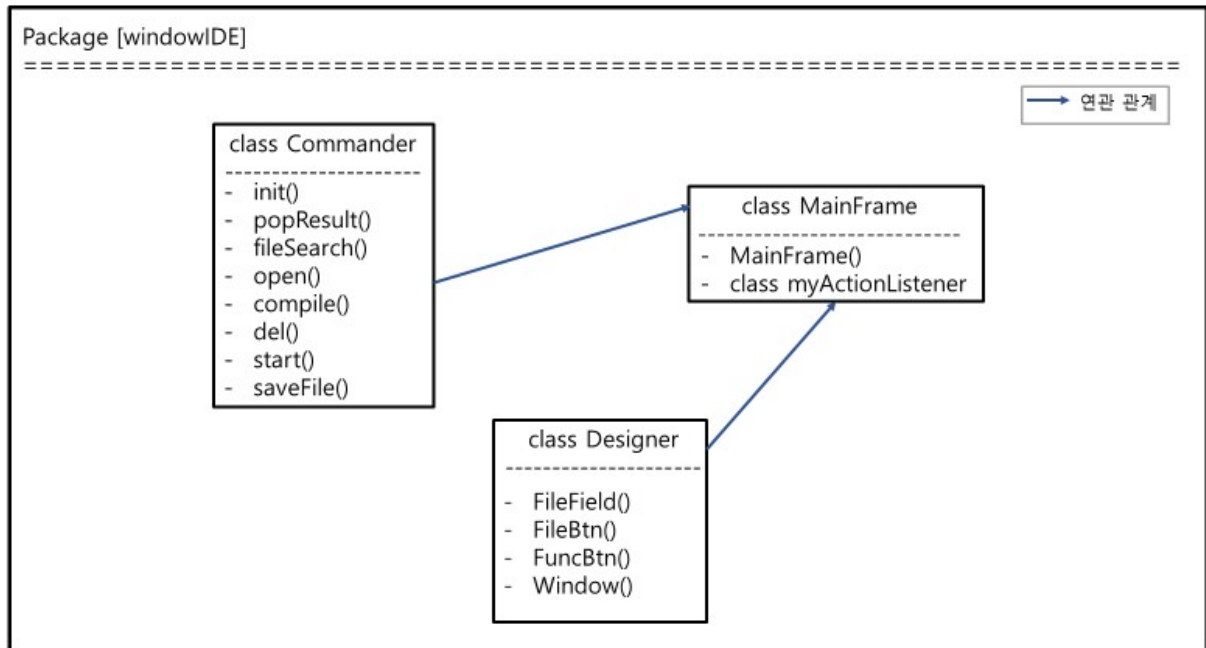
[초기화]

- "Clear" 버튼 선택 시 "Editing Window", "Result Window" 및 파일명을 입력하기 위한 텍스트 영역을 모두 초기화 함.

2. S/W 설계

/* Class Diagram 및 (또는) 전체 S/W 설계 내용 기술 */

1. Class Diagram



2. 클래스 관련 주요 설명

■ windowIDE Package

- **Commander 클래스** : Commander 클래스는 외부 명령을 위한 리스트 관리 및 실행과 관련된 클래스이다. 외부 명령은 컴파일, 실행, 파일 삭제, 파일 출력 등이 있다. Uploader 객체를 사용하여, getPath 메소드, getName 메소드를 사용한다.

➔ init 메소드 : 명령어 리스트를 "cmd /c"로 초기화하는 메소드이다.

➔ popResult 메소드 : Commamder 객체의 result 스트링 값을 pop 시키는 메소드이다.

➔ fileSearch 메소드 : path를 인자로 받는다. 인자로 받은 파일이 .java의 확장명을 가지고 있지 않거나, 파일이 존재하지 않다면 false를 반환하면서 변수 result에 "error : File not founded"를 넣는다. 파일이 존재하고 .java로 끝난다면 true를 반환한다.

➔ open 메소드 : "type path"을 사용하여 파일 내용을 확인할 수 있다.

➔ compile 메소드 : "javac -d path"을 사용하여 .java를 .class 파일로 컴파일 해주는 역할을 한다. 컴파일이 제대로 되었다면 변수 result에 "compile success"를 넣는다. 그리고 popResult 메소드를 리턴한다.

➔ del 메소드 : reset 기능을 수행할 때 사용된다.

"del 파일명" 명령을 통해 .java파일, .class파일 그리고 .error파일을 삭제한다.

➔ start 메소드 : 명령어가 담긴 리스트를 인자로, ProcessBuilder를 생성해 외부 명령을 실행한다. 예외가 생기면 변수 result에 오류 메시지를 넣는다.

➔ saveFile 메소드 : 파일을 저장하기 위해 구현된 메소드이다. 파일이 이미 존재한다면 변수 result에 "file already exist"를 넣고 반환한다. 저장에 성공 했을 시, "save success"를 넣고 리턴한다. 해당 메소드의 option 인자는 경우에 따라서 기능이 제한되는 파일 덮어쓰기 때문에 사용되었다. 저장경로가 제공되지 않아 기존의 파일에 덮어쓰기 하는 경우가 option.0에 속하며, 저장경로가 주어진 상황이 option.1에 속한다. 그리고 error파일을 생성하는 경우가 option.2에 속한다.

- **Designer 클래스** : 시각적 요소 디자인을 위해 만들어진 클래스이다.

➔ FileField 메소드 : JTextField를 사용하여 만들어진 메소드이다.

➔ FileBtn 메소드, FuncBtn 메소드 : JButton을 사용하여 만들어진 메소드이다.

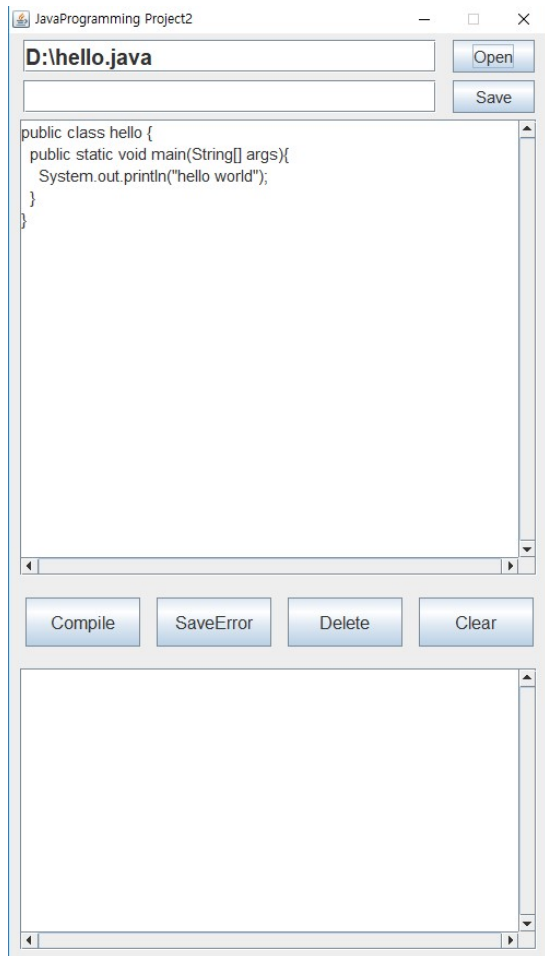
➔ Winodw 메소드 : JTextArea를 사용하여 만들어진 메소드이다.

- **MainFrame 클래스** : 이 프로그램의 주요 기능 OpenPanel, SavePanel, EditWindow, FunctionPanel, ResultWindow를 구현한 클래스이다. ActionListener를 implements한 myActionListener를 사용하여 이벤트를 처리한다.

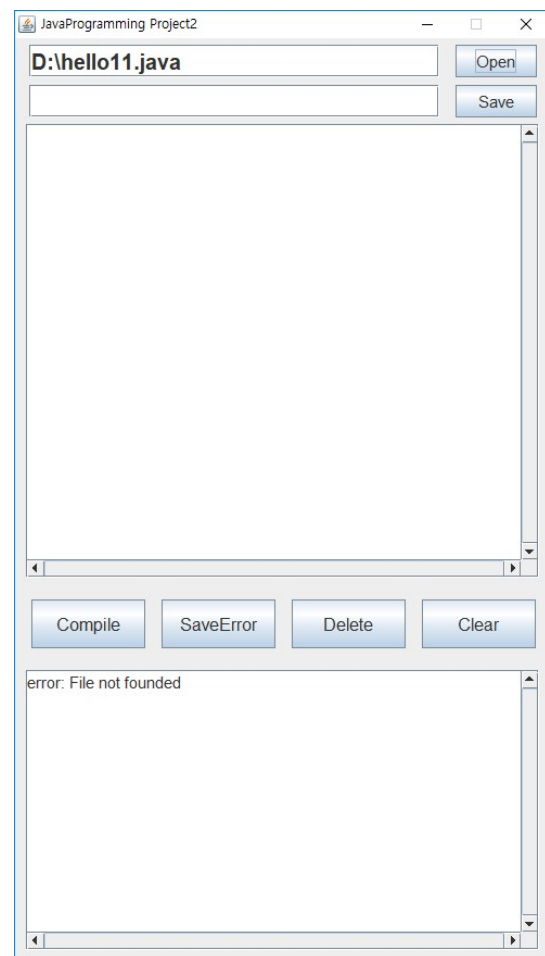
3. 실행 결과

/* 실행 결과를 screen capture하여 삽입 */

1. 존재하는 파일 Open

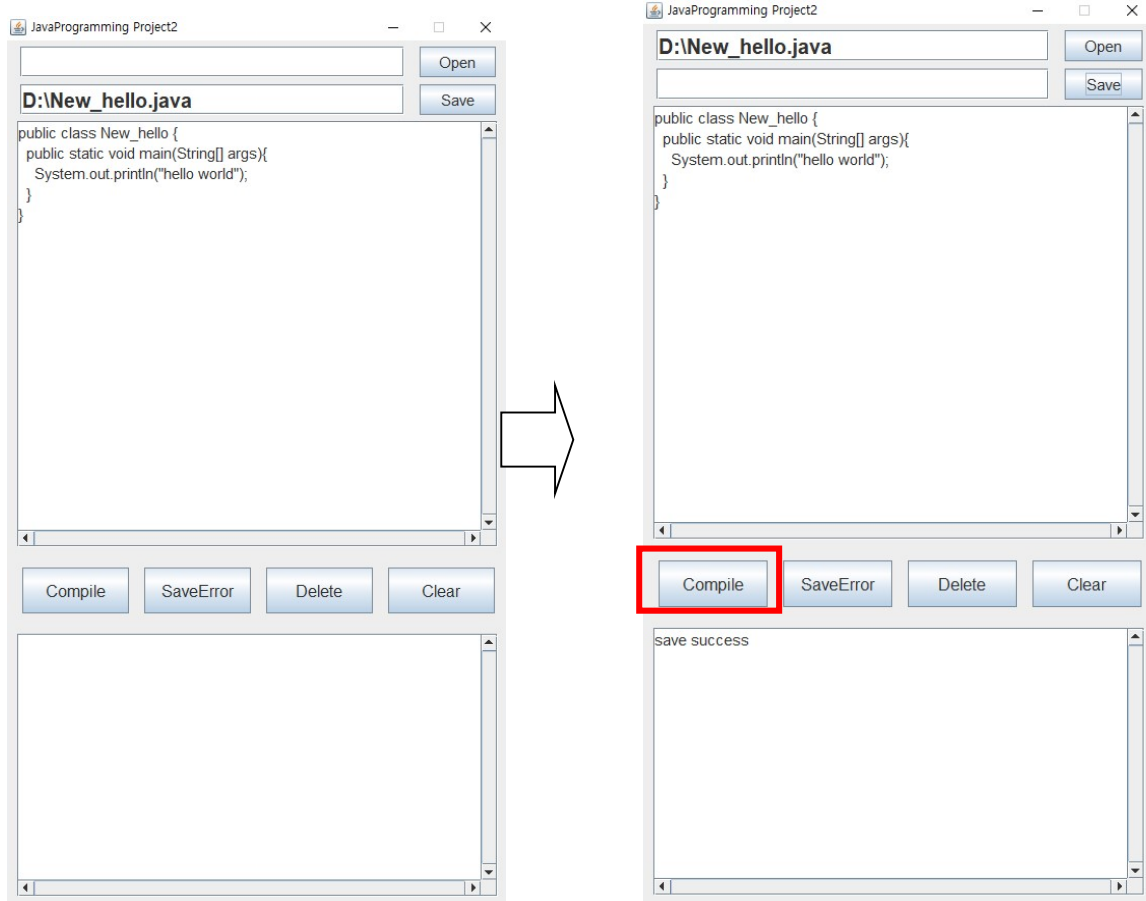


1-1. 존재하지 않는 파일 Open




error : File not founded 출력

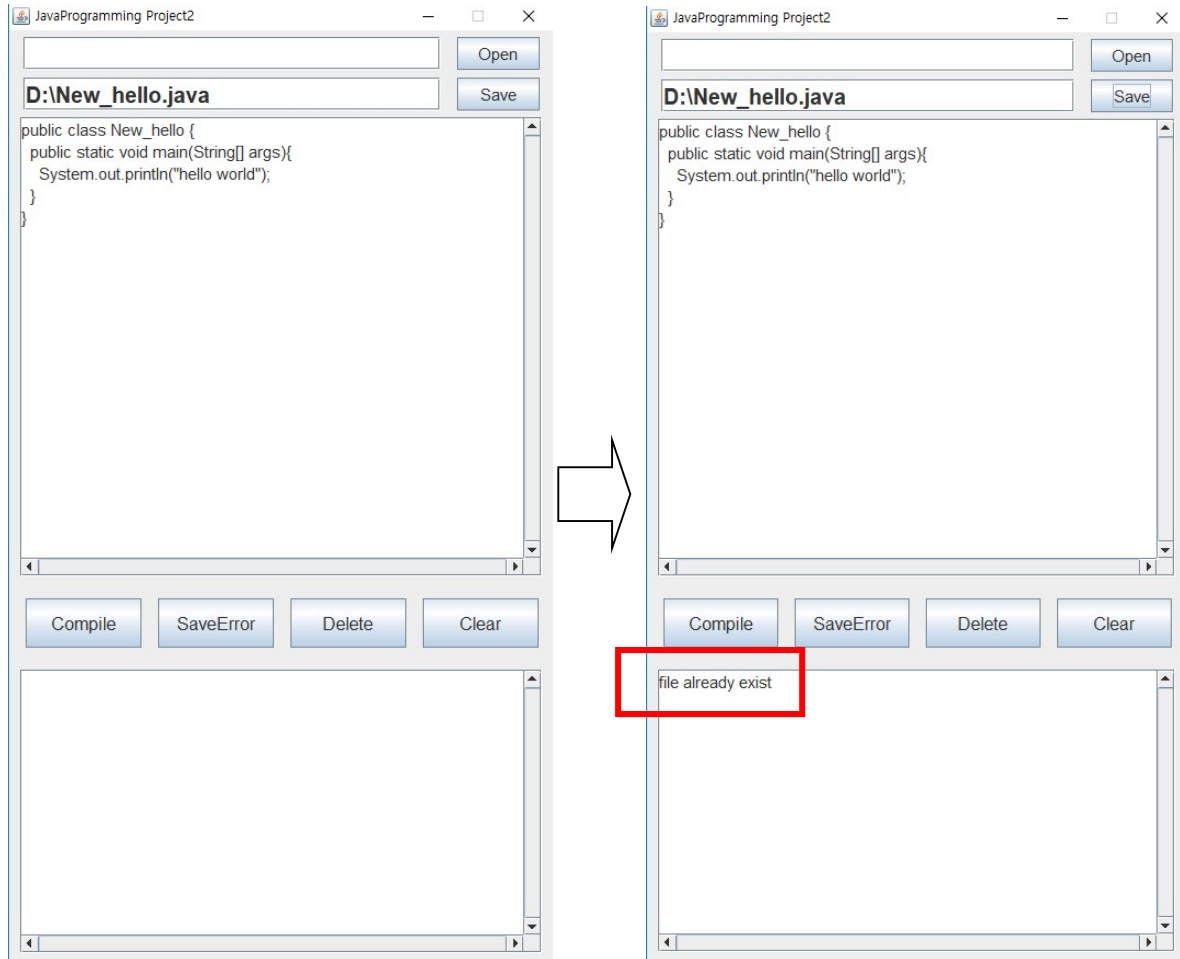
2. 새로운 파일 저장



새로운 파일 생성 완료

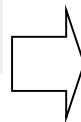
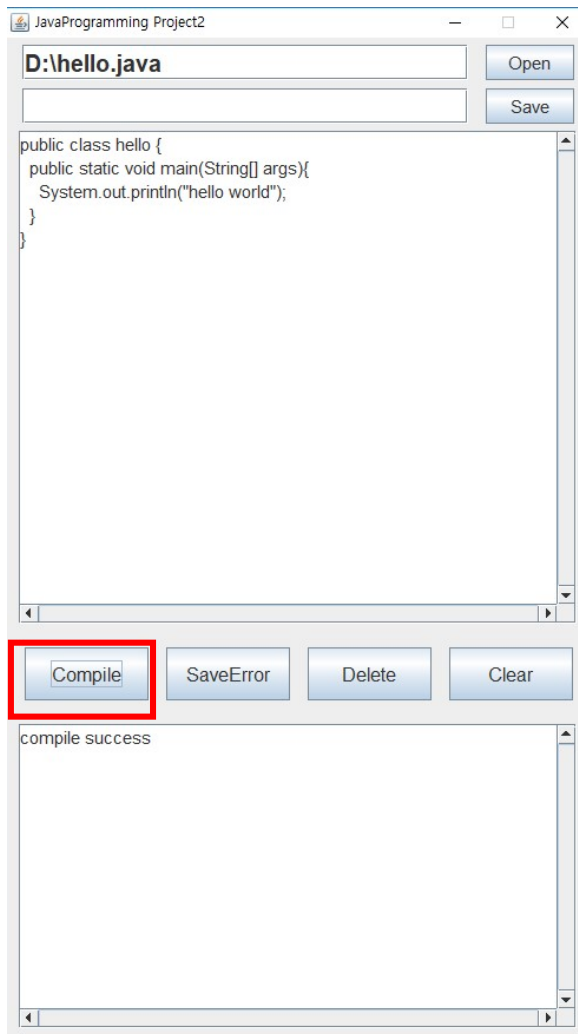
 New_hello.java 2017-11-27 오전... JAVA 파일

2-1. 새로운 파일 저장 시, 이미 존재하는 파일명일 경우



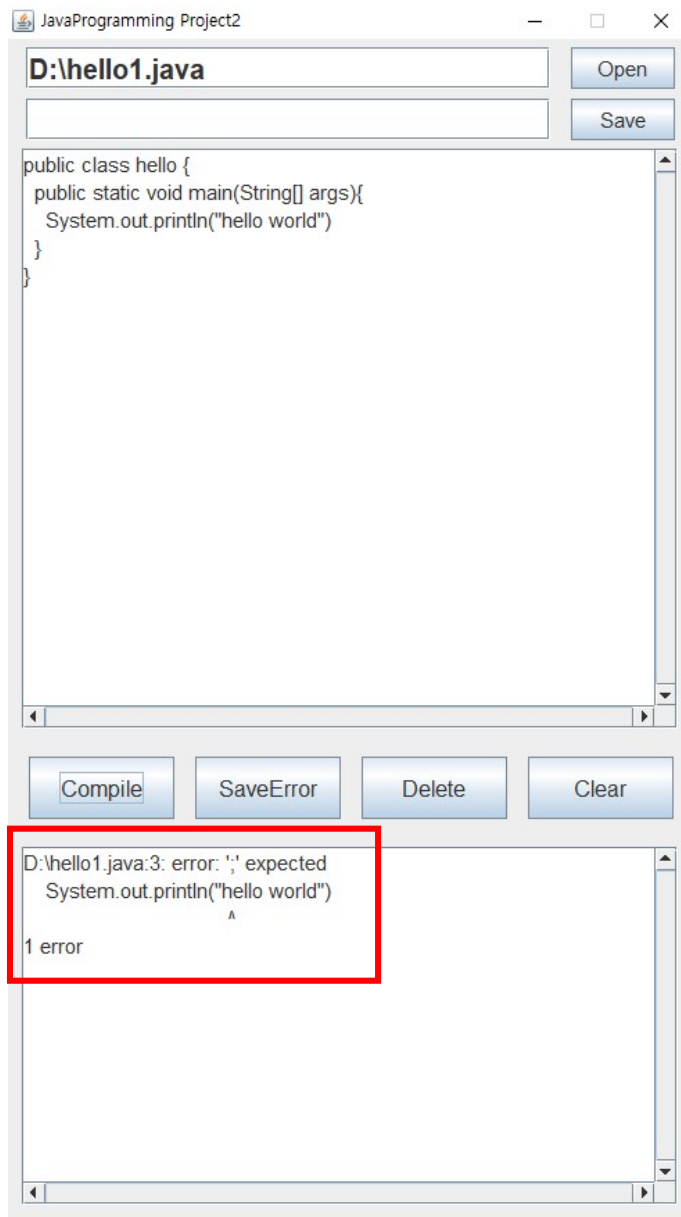
file already exist 출력

3. 컴파일 실행 성공



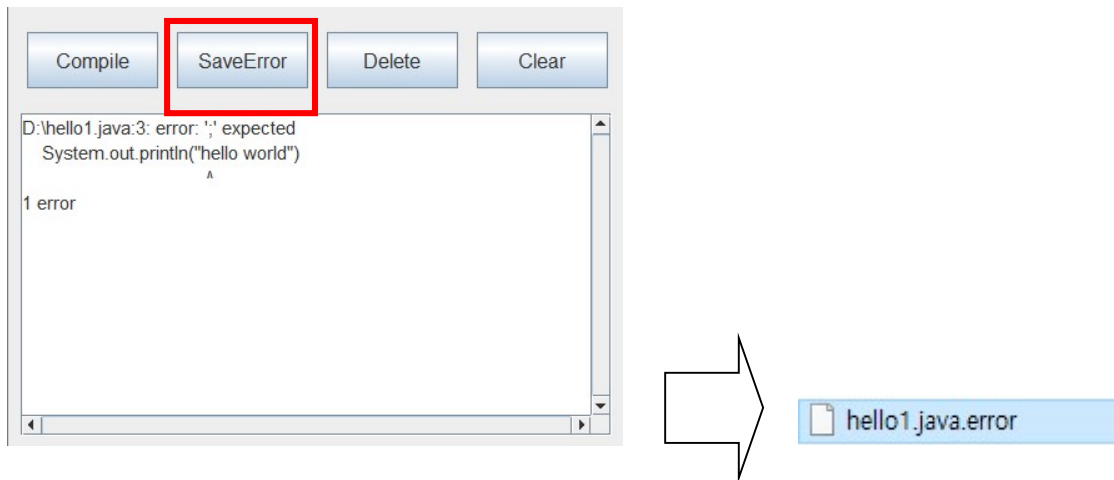
hello.class

3-2. 오류가 있는 파일 컴파일 실행

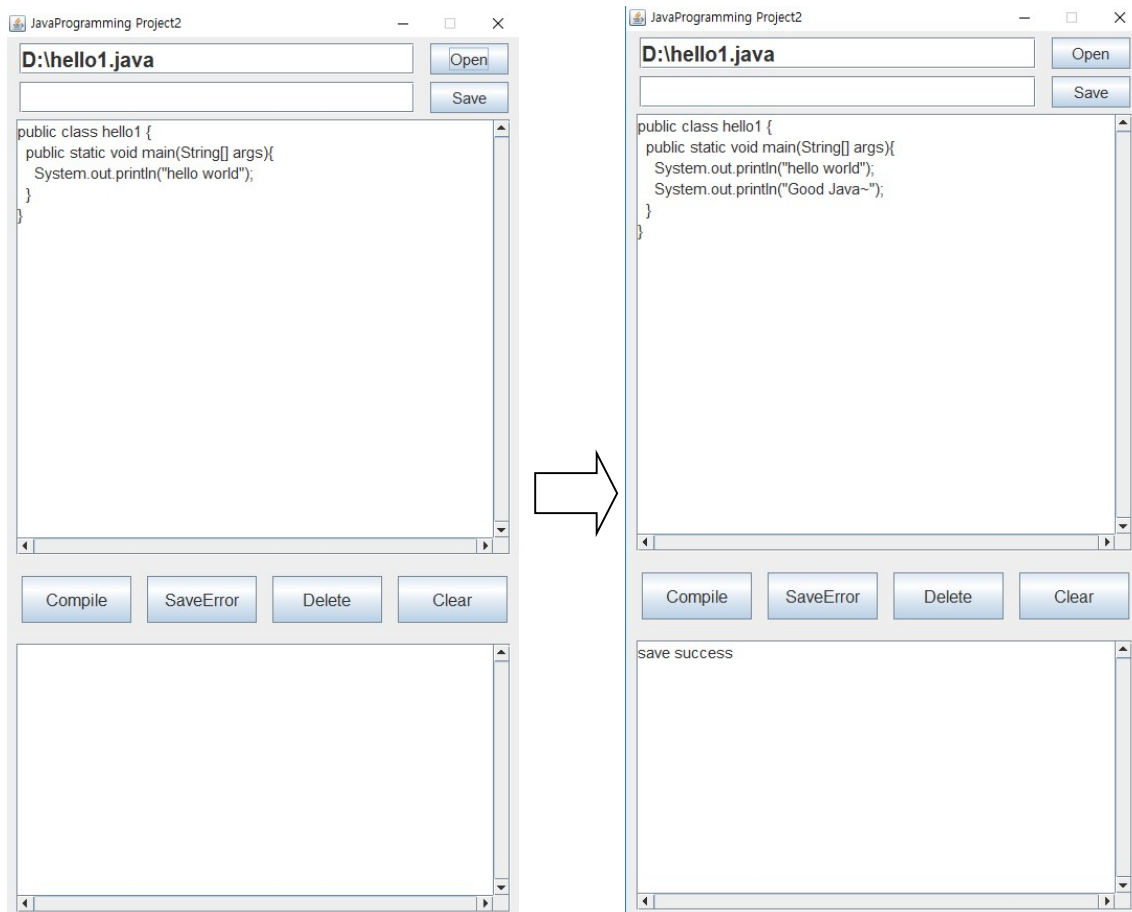


컴파일 에러 표시

4. SaveError 버튼 선택 시 오류 내용 저장 (.error)

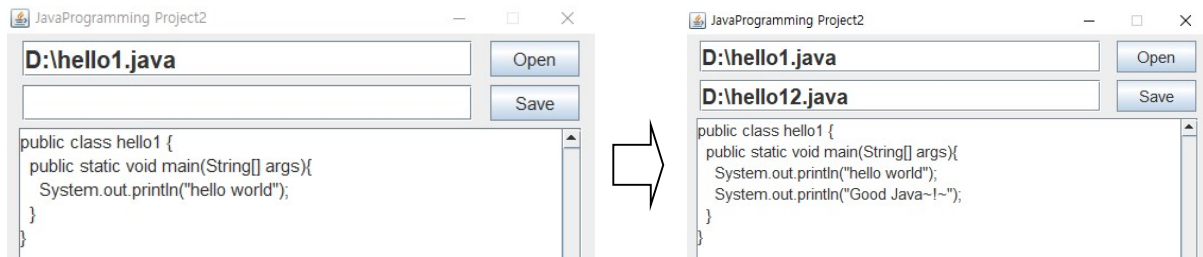


5-1. 파일 내용 변경 (파일 이름이 제공되지 않고 Save버튼을 누를 때)

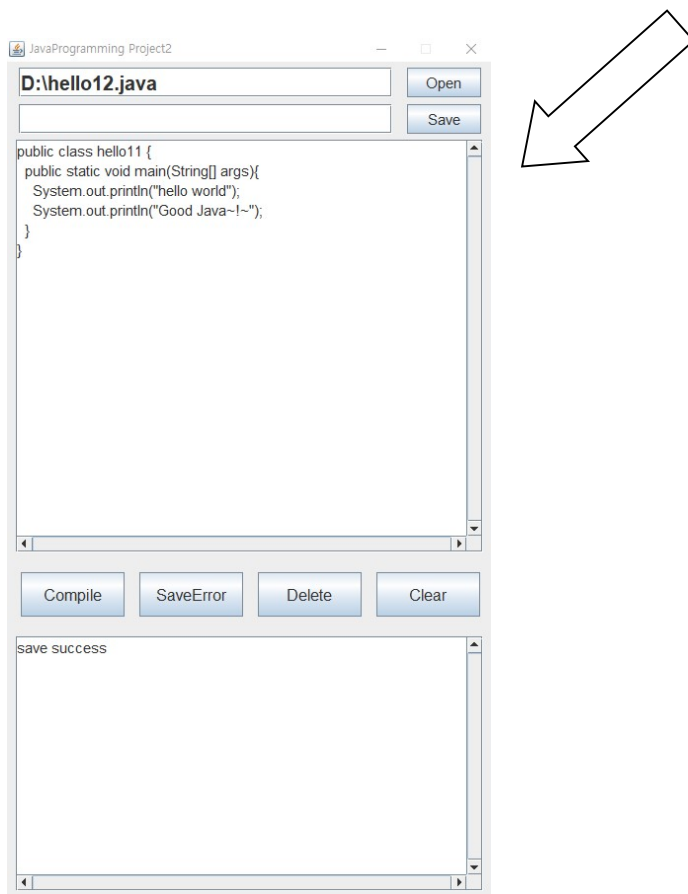


Save 선택 시, hello1.java에 저장

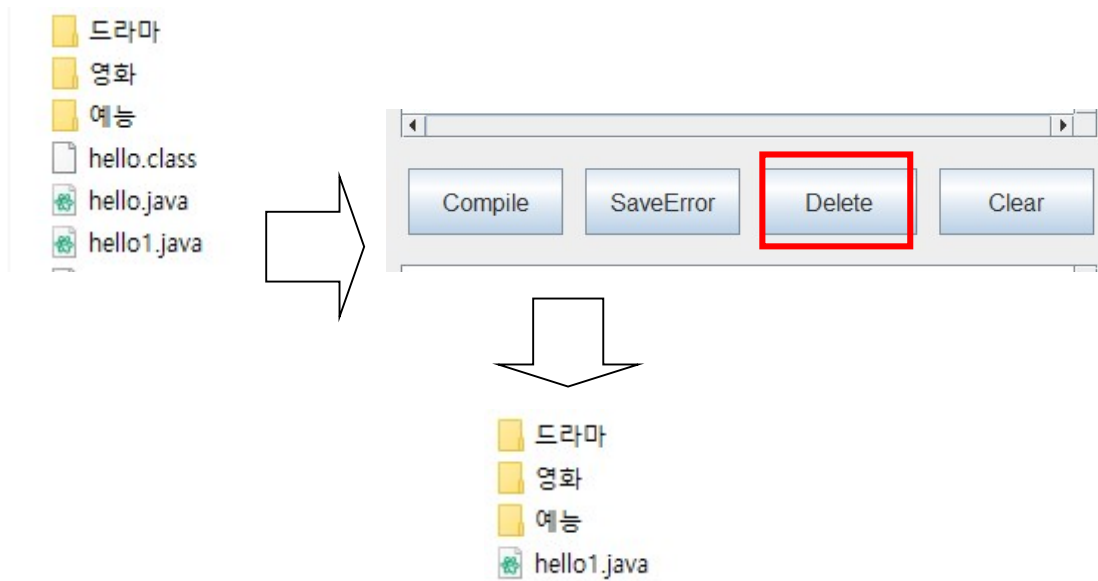
5-2. 파일 내용 변경 (파일 이름이 제공될 경우)



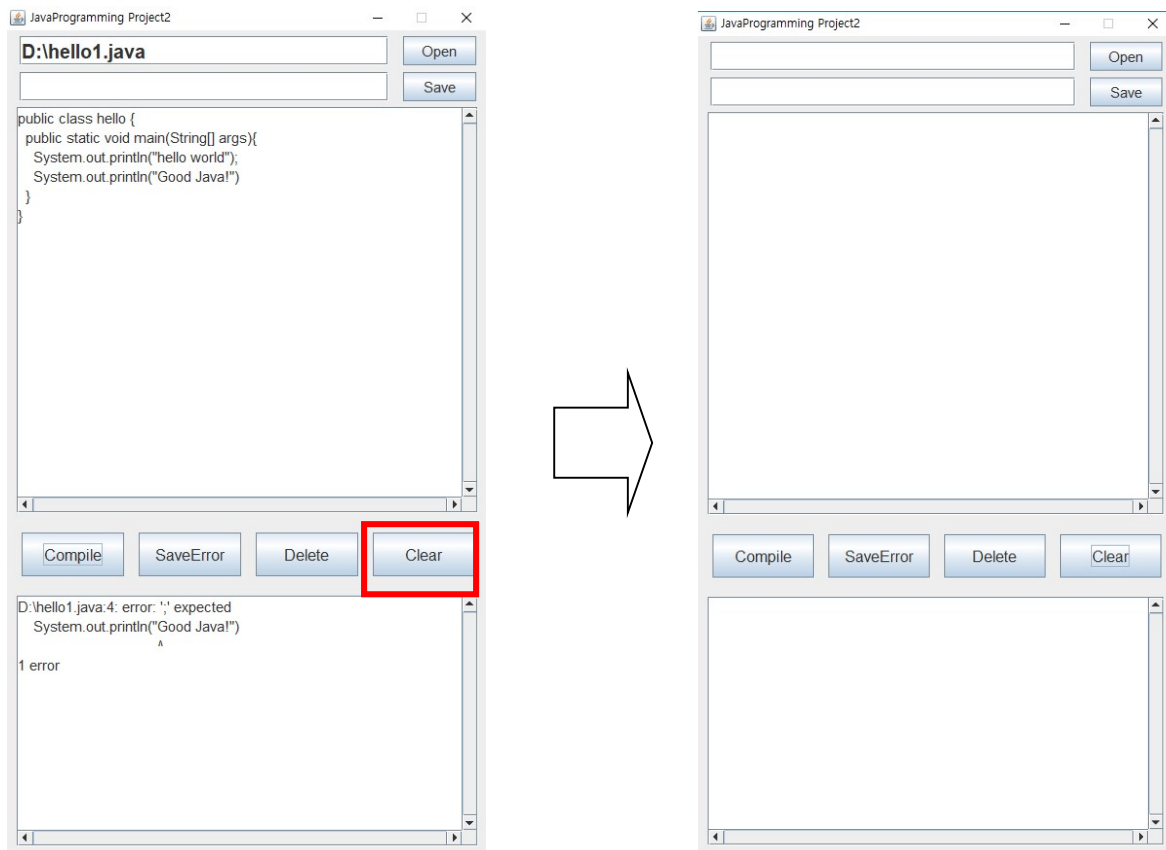
파일 내용 변경 후 새로운 파일명 입력



6. Delete 버튼 선택 시






7. Clear 버튼 선택 시



4. 특이 사항

/* 작성된 코드의 특이 사항 또는 추가 점수 획득을 위해 포함된 사항이 있을 경우 작성 */

- 컴파일 시, .class 파일, 에러 저장 시, .error 파일이 .java파일과 같은 위치에 생성된다.

 hello1.java.error hello1.java hello1.class