

객체지향 언어와 실습



담당 : 오 세 만 교수님

학과 : 컴퓨터 공학과

학 번 : 2015112098

성 명 : 허정원

제출일 : 2018 년 6 월 17일



	최종프로젝트	2015112098 허정원
<pre>//프로젝트 명세 Editor 클래스의 기능을 구현하시오. 단어 검색기능 단어 변환 기능 검색결과 출력기능 단어 삽입 기능 기존 콘솔 메뉴를 잡아 Swing을 이용하여 GUI로 구현하시오. 메뉴바 구현 검색 진행상태 레이블 입/출력 파일 경로 입력 텍스트 필드 검색 단어 / 변환 단어 입력 텍스트 필드 검색결과 출력 리스트 //문제 분석 단어 검색기능은 텍스트 파일에서 검색하는 것으로 단어 사이는 스페이스로만 구분되어 줄 사이에 단어 걸침을 처리해야 한다. 단어 변환 기능은 검색된 단어를 변환할 단어로 바꿔 새로운 파일로 출력한다. 검색결과 출력기능은 ReportWriter.jar를 이용하여 출력한다. 단어 삽입 기능은 행과 열을 입력해 입력한 행과 열에 단어를 삽입한다. 자바 Swing과 awt를 이용하여 메뉴바, 텍스트 필드, 레이블, 상태 바, 버튼, 스크롤 바를 만들어 해당 기능을 구현한다. 상태 바는 스레드로 구현해 검색 실행 중에 다른 메소드를 실행할 수 있게 해준다. //프로그램 설계, 알고리즘 설명 실제 검색에 관한 기능을 수행할 Editor 클래스와 그 기능을 확장한 ImprovedEditor 클 래스가 존재한다. Editor 클래스 안에는 단어 검색을 할 search 메소드, 단어 변환을 할 replace 메소드, 검색결과를 저장할 ReportWriter 객체가 존재한다. 위 기능을 그래픽 사용자 인터페이스로 나타낼 Frame 클래스가 존재한다. Frame 클래스는 생성자로 메뉴바, 상태 바, 텍스트 필드, 스크롤, ImprovedEditor 객체 를 생성하며 ActionListener 메소드로 사용자 입력 시 기능이 수행되게 한다. 위 Frame 객체를 생성 관리할 Main 클래스가 존재한다.</pre>		

//주요 메소드 설명

```
//Editor 클래스 생성자
public Editor(JTextArea log, JProgressBar progress) {
    this.log=log;
    this.progressBar=progress;
}
//결과를 GUI에 출력할 JTextArea와 진행상태를 나타낼 JProgressBar를 매개변수로
받는다
```

//Editor 클래스 searchWord메소드

```
public void searchWord(String find) {
    String line, next, words[];
    int lineCnt=1, wordCnt=0, found=0;
    try {
        bufferedReader.resetToFirst();//읽은 파일을 처음위치로
        System.out.println("...");//콘솔과
        log.append("...");//GUI에 상태 표시
        log.setCaretPosition(log.getDocument().getLength());//스크롤내리기
        line=bufferedReader.readLine();
        int length=line.length();
        int nextL=0,progress=0;
        while(true) {
            if(line==null)break;//EOF일 때 종료
            next=bufferedReader.readLine();
            if(next!=null)nextL=next.length();
            if(!line.endsWith(" ")) { //공백으로 끝나지 않을 때
                if(next!=null) {
                    line+=next.split(" ")[0]+" ";
                    next=next.substring(next.indexOf(" ")+1);
                }
            }
            //다음줄의 공백까지로 처리
            words=line.split(" "); //줄을 잘라 단어로 만듦
            for(wordCnt=0; wordCnt<words.length; wordCnt++) {
                if(words[wordCnt].equals(find)) { //단어를 찾음
                    System.out.println("...");
                    log.append("...");
                    log.setCaretPosition(log.getDocument().getLength());//콘솔과 GUI표시
                    CSearchItem item = new CSearchItem();
                    item.setIndexNo(wordCnt+1);
                    item.setLineNo(lineCnt);
                    report.addSearchItem(item);//검색결과저장
                    found++;
                }
            }
            lineCnt++;
            line=next;
            progress+=length;//읽은 줄을 더해 진행상태 계산
            progressBar.setValue((int) ((progress*100)/fileSize));
            //진행 상태 갱신
            length=nextL;
        }
        progressBar.setValue(100);//검색완료시 진행상태 100%
        System.out.println("...");
        log.append("...");
        log.setCaretPosition(log.getDocument().getLength());
        //콘솔과 GUI 표시
        report.setTotalSearchedNo(found);//검색결과 저장
    }
    catch (Exception err) {
        System.err.println("파일 검색 실패"+err);
    }
}
```

```

//Editor 클래스 replaceWord 메소드
public void replaceWord(String find, String change, String path) {
    String line, next, words[];
    int lineCnt=1, wordCnt=0, found=0;
    try {

        //...searchWord와 동일한 알고리즘...//

        changeline+=line.replaceAll(" "+find+" ", " "+change+" ");
        //단어를 바꿔줌
        while(changeline.length()>length) {
            outbild+=changeline.substring(0,length)+"\r\n";
            changeline=changeline.substring(length);
        }
        //변경 전 줄 길이로 라인을 잘라
        //줄 길이를 만족하면 출력할 outbild에 저장
        //줄 길이를 만족하지 못하면 changeline에 저장
        if(outbild.length()>4096) {
            printToFile(path, outbild);
            outbild=new String();
        }//여러번 출력하면 느려져 4096byte이상일 때 파일 출력
        //출력 후 내용 비워줌

        //...searchWord와 동일한 알고리즘...//

    }
    printToFile(path, outbild);
    printToFile(path, changeline);
    outbild=new String();
    changeline=new String();
    //남은 string 출력

    //...searchWord와 동일한 알고리즘...//

}
catch (Exception err) {
    System.err.println("파일 검색 실패"+err);
}
}

```

```

//Improved Editor 클래스에 appendStr 메소드
public void appendStr(String str, int lineNum, int colNum, String path) {
    String line;
    int lineCnt=0, found=0, length=0, progress=0;
    try {
        bufferedReader.resetToFirst();//읽은 파일을 처음위치로
        System.out.println("...");//콘솔과
        log.append("...");//GUI에 상태 표시
        log.setCaretPosition(log.getDocument().getLength());//스크롤
        while(true) {
            lineCnt++;
            line=bufferedReader.readLine();
            if(line==null) break;
            length=line.length();
            //찾는 행까지 읽어
            if(lineCnt==lineNum) {
                outbuild+=line.substring(0, colNum)+" "+str+"
"+line.substring(colNum)+"\r\n";//찾는 열에 단어를 붙여줌
                found++;
            }
            else
                outbuild+=line+"\r\n";
            if(outbuild.length()>4096) {
                printToFile(path, outbuild);
                outbuild=new String();
            }
            //4096byte 이상일 때 출력 후 초기화
            progress+=length;
            progressBar.setValue((int) ((progress*100)/fileSize));
            //진행률 계산
        }
        printToFile(path, outbuild);
        outbuild=new String();
        //남은 문자열 복사
        progressBar.setValue(100);
        //진행률 100%
        if(found==0) {
            System.err.println("해당 라인을 찾을 수 없습니다.");
            log.append("해당 라인을 찾을 수 없습니다.\n");
            log.setCaretPosition(log.getDocument().getLength());
            return;
        }
        //error 표시
        System.out.println("변경완료");
        log.append("변경완료\n");
        log.setCaretPosition(log.getDocument().getLength());
    }
    catch (Exception err) {
        System.err.println("파일 검색 실패"+err);
    }
}

```

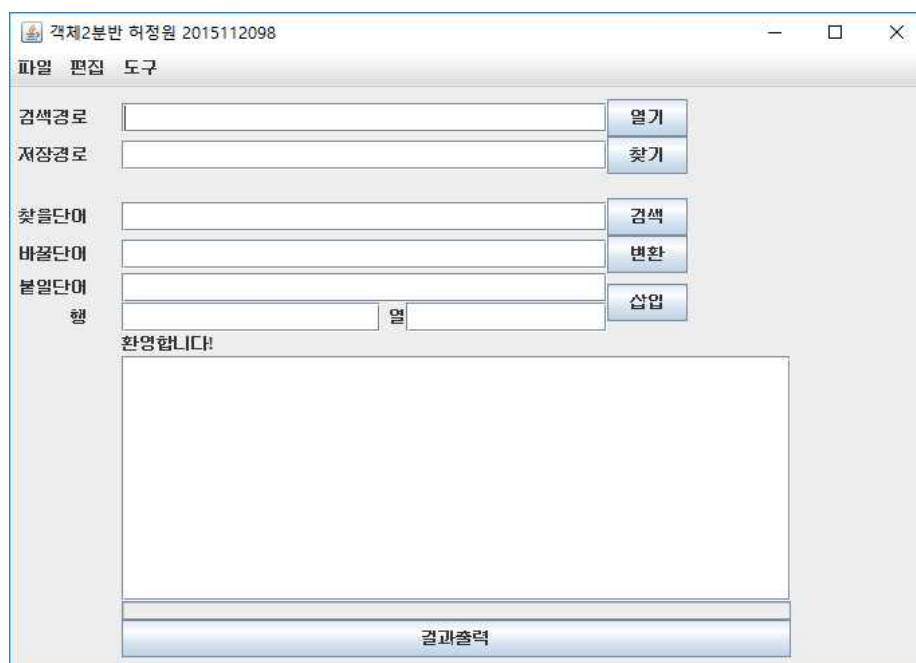
//Method이름 클래스

```
public class Method이름 implements Runnable {  
    //필요한 정보를 저장할 필드 생성  
    ImprovedEditor engine;  
    public MethodReport(ImprovedEditor e, 필요한정보... ){  
        engine=e;  
        ...  
    }  
    //동기화시킬 객체 ImprovedEditor과 기타 필요한 정보 받음  
    public void run() {  
        synchronized(engine) {  
            engine.이름(필요한정보...);  
        }  
    }  
    //engine객체 동기화  
}
```

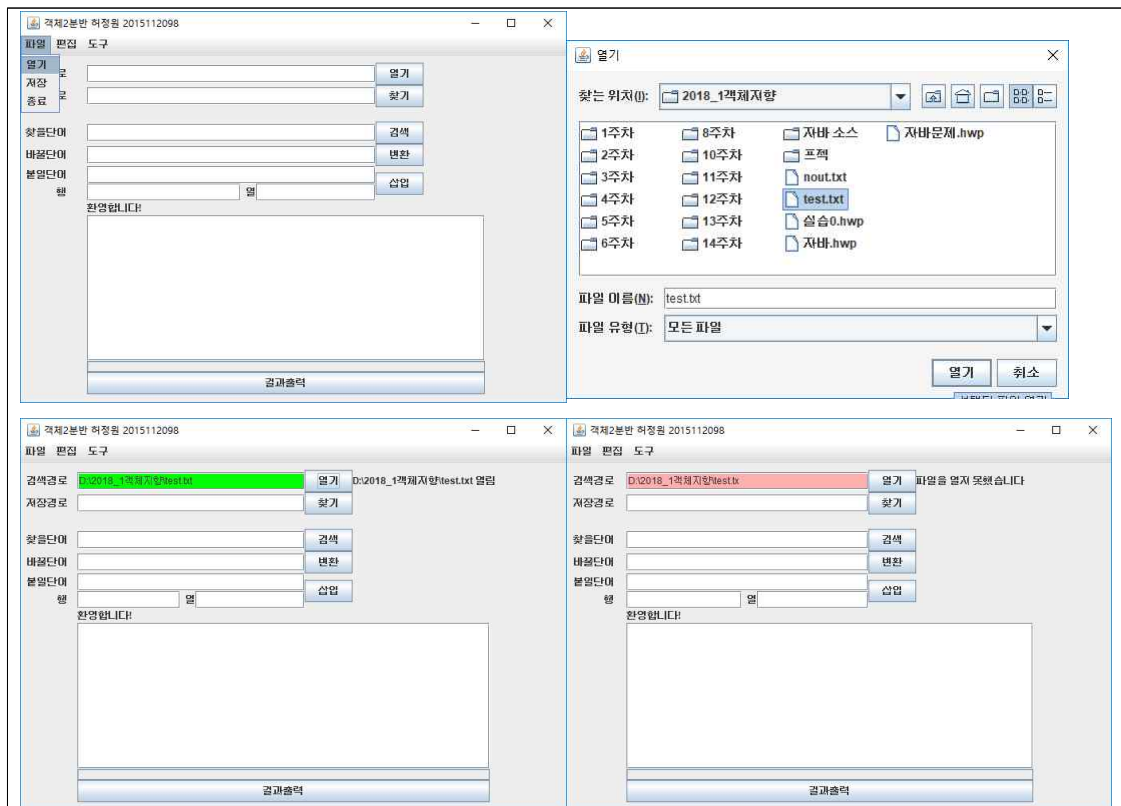
//Frame 클래스 gbAdd 메소드

```
private void gbAdd(Component c, int x, int y, int w, int h, double  
weightx) {  
    gc.gridx = x;  
    gc.gridy = y;  
    gc.gridwidth = w;//넓이  
    gc.gridheight = h;//높이  
    gc.weightx=weightx;  
  
    grid.setConstraints(c, gc);  
    panel_fileArea.add(c);  
}  
//필요 정보를 받아 편리하게 Frame 위치를 정해주는 메소드
```

//결과

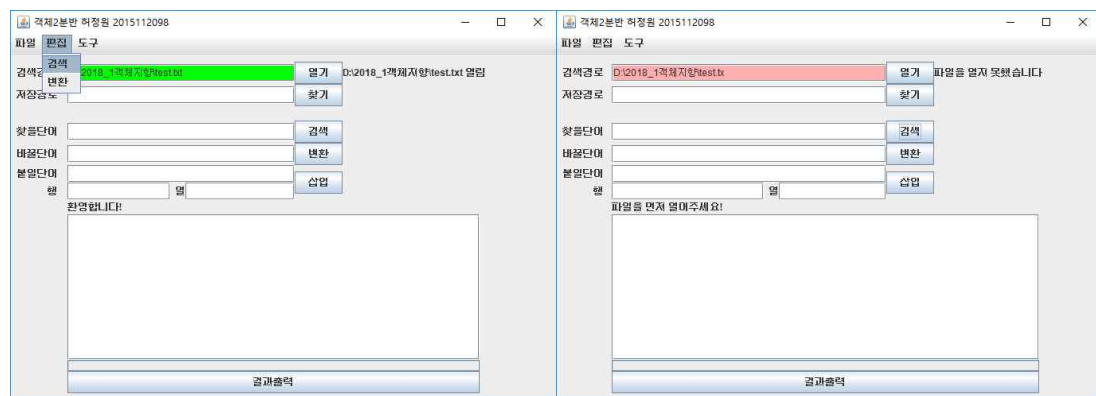


//메뉴바 텍스트 필드 진행상태 스크롤 필드가 구현되어있다.



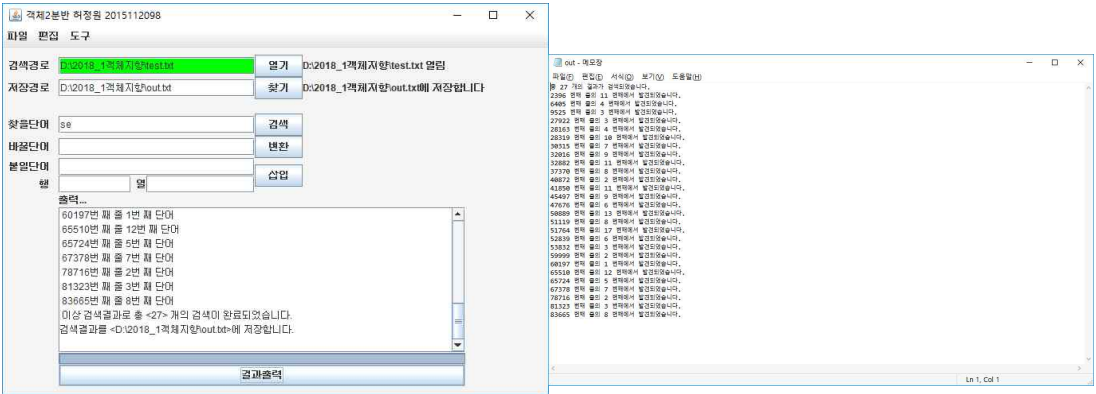
//파일-열기, 열기 버튼, 텍스트 필드에서 enter 입력 시 파일을 연다
 //실패 시 결과를 옆 레이블에서 알려준다

//저장경로 설정도 위와 같다.

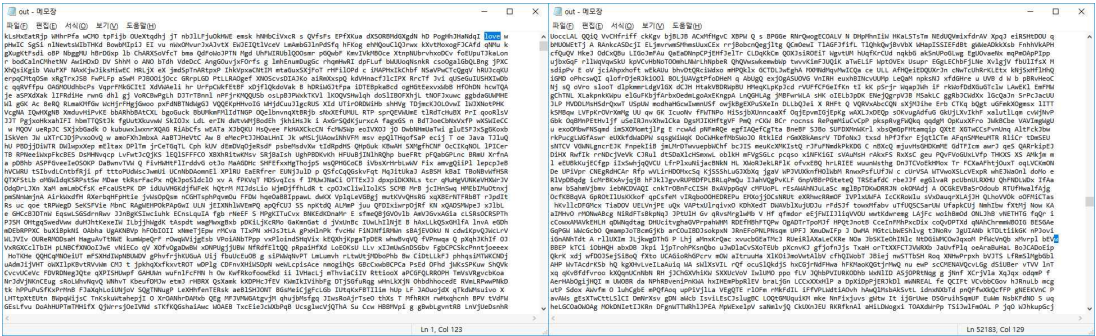
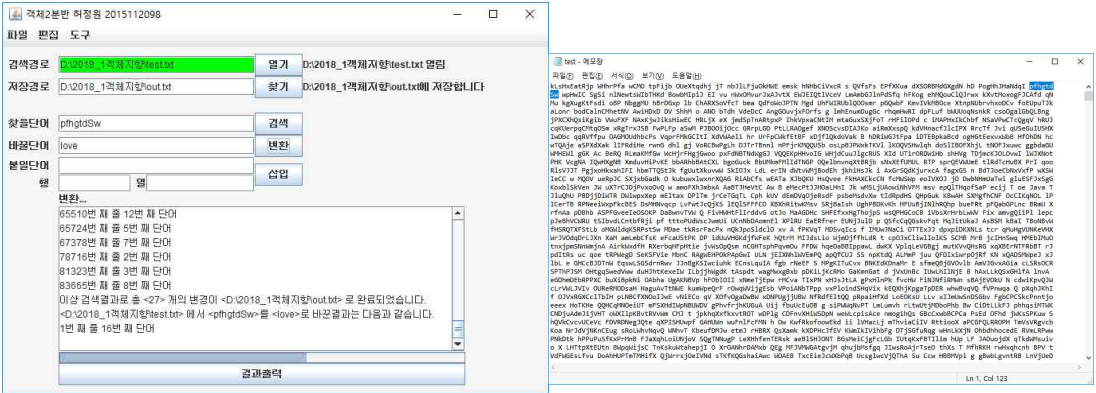


//편집-검색, 검색 버튼, 텍스트 필드에서 enter 입력 시 검색을 수행
 //실패 시 결과를 스크롤 필드 위 레이블에서 알려준다

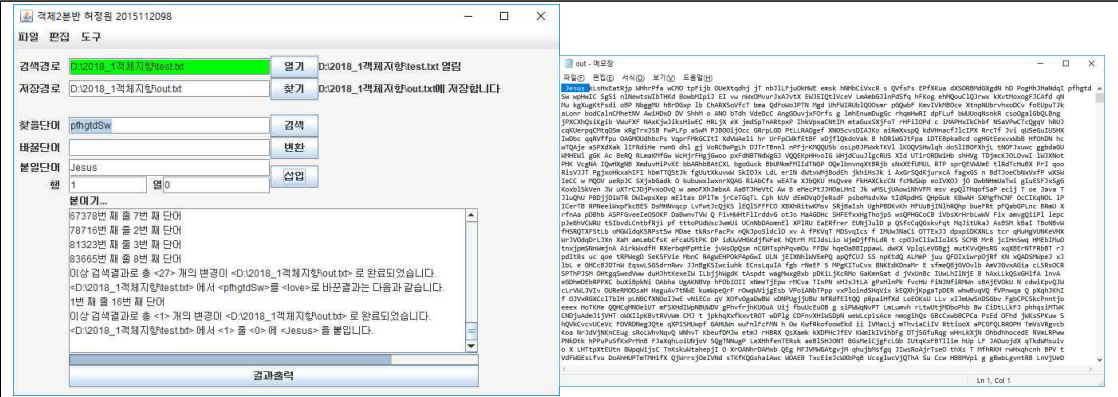
//바꿀 단어, 붙일 단어 설정도 위와 같다.



//단어 검색과 결과 출력 시 검색결과가 저장경로에 출력된 모습



//단어를 바꾼 모습과 문제의 52183줄



//단어를 붙인 모습

//추가로 메뉴바에 파일-종료, 도구-정보 버튼과

//검색경로 지정 시 결과에 따른 textfield 색상

//텍스트 필드에 마우스 선택 시 정보 표시

//예외처리를 구현하였다.

//소감

//한 학기 동안 자바를 배우면서

//객체지향의 개념과 여러 자바에서 정의된 개념을 실습해서 이해할 수 있었다

//실습과 이론의 학습 순서가 달라서 어려움이 있었으나

//유익한 수업이었던 것 같다.

...

//배울 게 많다는 걸 느꼈다

//한 학기 동안 수고하셨습니다.

	최종프로젝트	2015112098 허정원
--	--------	----------------