Java20-2 Lab1

32191826 박주희

WordProcessor 클래스가 manager와 같은 역할을 한다.

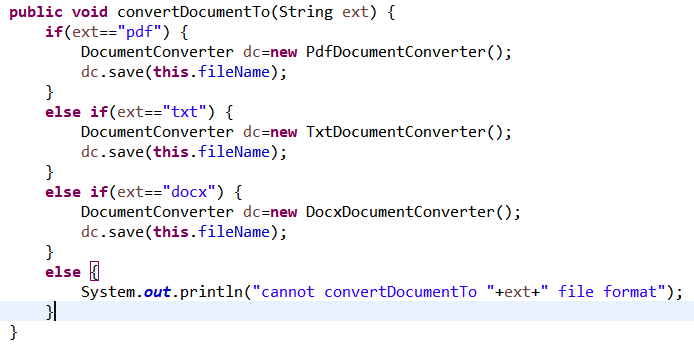
DocumentConverter를 상속하는 PdfDocumentConverter, TxtDocumentConverter, DocxDocumentConverter는 각각 pdf, txt, docx형의 파일을 convert하는 클래스이다.

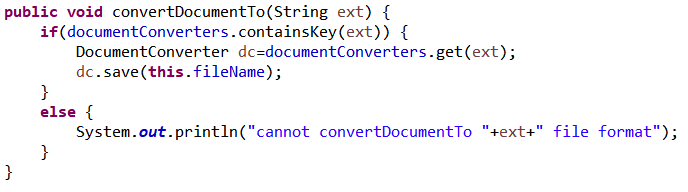
IspellChecker 인터페이스와 이를 상속하는 EnglishSpellChecker 클래스가 있다.

\*추상클래스와 인터페이스의 차이점: 인터페이스는 모든 메소드가 추상메소드인 경우

대부분의 클래스와 메소드를 구현하는데 큰 어려움은 없었다.

하지만, WordProcessor 클래스의 convertDocumentTo() 메소드에서는 고민을 해보았다.

수정 전 코드:

수정 후 코드:

수정 전에는 addDocumentConverter() 메소드처럼 조건문을 이용했었다. 하지만 생각해보니, 저렇게 조건문을 이용하여 코드를 짜게 되면 HashMap을 아예 사용하지 않게된다는 것을 깨달았다. 그리고 교수님이 HeshMap을 쓰라고 하신 이유는 있을 것이라고 생각하여 다시 고민해보았다.

수정 후에는 ext를 key로 포함하는지를 체크한 뒤, 포함한다면 dc에 documentConverters map중에서 ext를 key로 가지는 Document Converter를 저장하였고 save 메소드를 호출하였다. 포함하지 않는다면 convert를 할 수 없다는 메세지를 출력하도록 하였다.

수정을 하고나니 코드 길이도 절반으로 줄어들었고 HashMap을 만들어 놓고 쓰지 않아 쓰레기 정보로 남기는 것 또한 방지 할 수 있었다.

My code: 입력한 파일형을 key로 갖는 요소를 삭제한 후 fileName.남아있는 요소의 파일형을 모두 출력하도록 하였다.



결과 화면 창:

WordProcessor의 객체의 fileName을 doc1.docx에서 .docx를 잘라낸 doc1로 저장

txt, pdf, docx로 각각 convert한 결과 출력

wps형은 존재하지 않으므로 오류 메시지 출력

docx형을 삭제한 나머지 출력

