객체 지향 설계 (개인) 프로젝트

# 콘솔 기반 텍스트 편집기 구현

제출일: 2020. 12.12

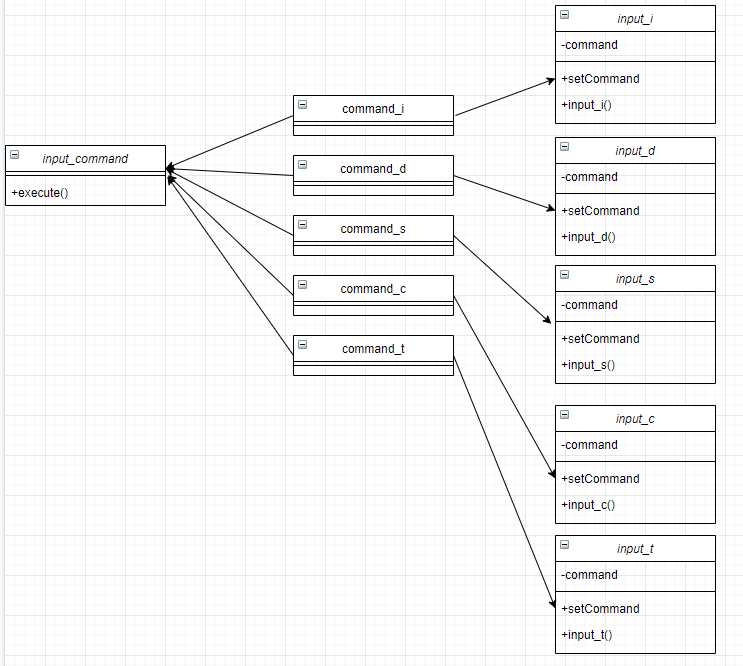
201702078 최대우

1. **프로그램 설명서**
2. **프로그램의 디자인패턴**

* **Command pattern**

실행될 기능을 캡슐화함으로써 주어진 여러 기능을 수행 할 수 있는 재사용성이 높은 클래스를 설계할 때 사용되는 패턴.

term project의 주제는 text 파일 편집기를 만들어 사용자가 입력하는 여러가지 명령에 대해 그에 맞는 기능을 실행하고 수행해야하기 때문에 유용할 것 같아서 command pattern을 골랐습니다.

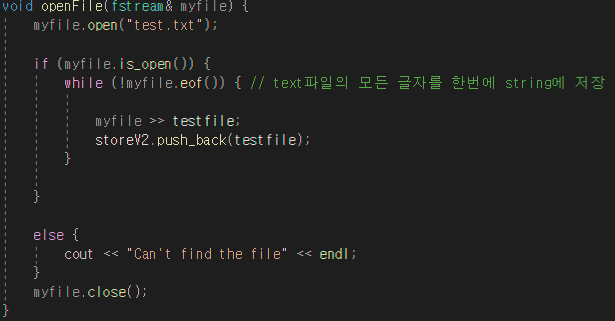
1. **클래스 다이어그램**

약간 이런 식으로 구현하려고 했는데 제대로 구현하지 못한 것 같습니다.

1. **소스 코드**

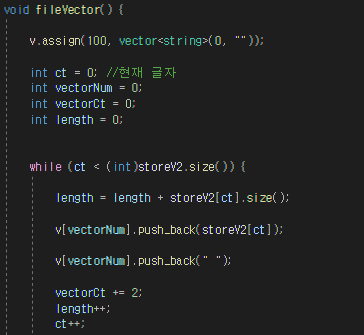
**-주요 함수 설명**

(1) openFile()

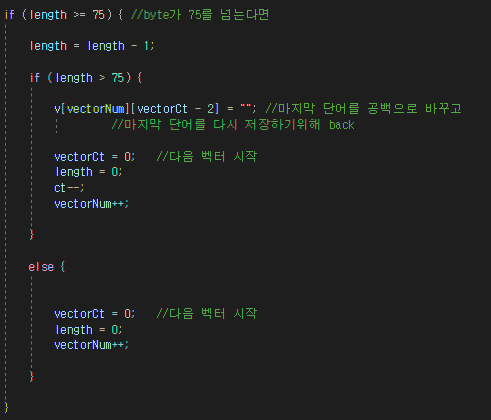


fstream& myfile을 통해서 텍스트 파일을 오픈하는 함수이다. while문 반복을 통해서 myfile의 한 단어를 testfile에 넣고 testfile을 벡터 storeV2의 back에 추가한다.

(2) fileVector

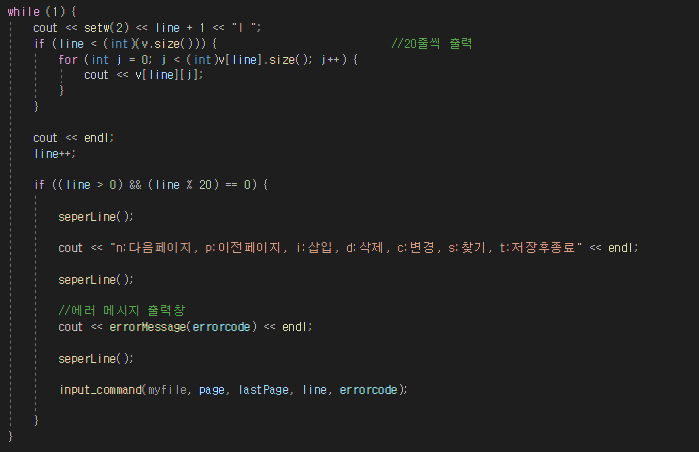


vector v를 통해서 출력하기 위한 line을 저장하는 함수이다. 모든 단어를 꺼낼때까지 반복을 돌린다. 반복문에서 단어의 크기만큼 length에 더해주고 단어와 “ “을 v의 back에 추가해준다. vectorCt와 length에 더해주고 ct를 증가시킨다.



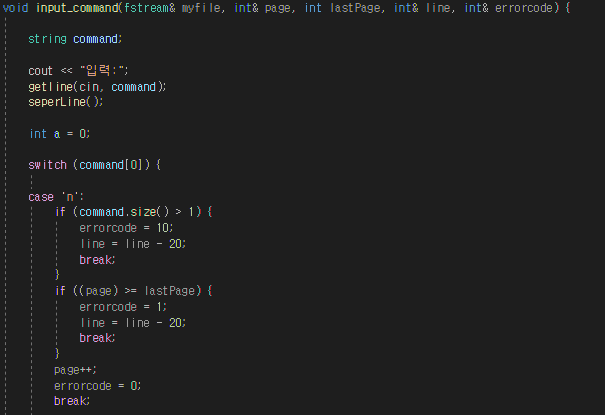
length가 75를 넘는다면 length를 감소시켜 “ “을 빼고 그래도 75가 넘는다면 마지막 단어를 v에서 지우고 다음 줄 벡터를 시작하기 위해 값들을 새로 설정하고, 지워진 단어부터 시작해야기 때문에 ct를 감소시킨다. 그렇지 않다면 줄 벡터를 위한 값들만 새로 설정한다.

(3)print\_txt()



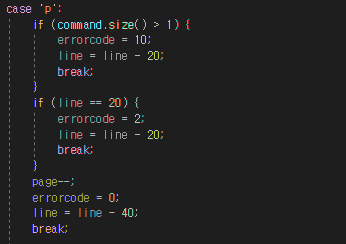
출력문을 위한 print\_txt의 while문이다. 줄벡터를 한 번씩 출력하고, line을 증가시킨다. line이 20줄 단위가 될 때마다 명령어를 입력 받기 위한 출력문을 내보낸다.

(4)input\_commnad()

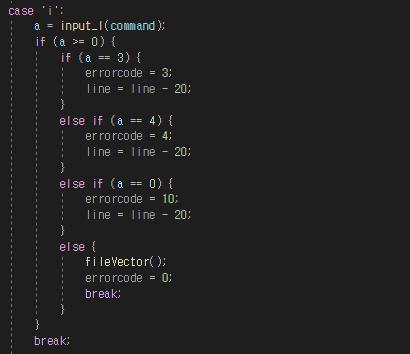


명령어를 입력받고 수행하는 함수이다. command를 입력 받으면 switch 문을 통해 첫 글자를 받아와 분류한다.

‘n’을 입력받으면 n은 ‘n’밖에 없기 때문에 사이즈가 1보다 크면 오류메시지를 출력한다. 현재 페이지가 마지막 페이지면 오류메시지를 출력한다. 오류의 조건에 안 들어가면 페이지를 증가시키고 다음 페이지를 출력 받도록 한다.



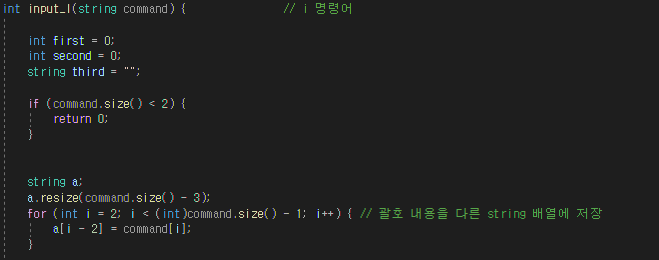
‘p’를 입력받으면 ‘n’와 마찬가지로 사이즈를 조사하고, line이 20과 같으면 첫 번째 페이지라는 말이기 때문에 오류메시지를 출력한다. 문제가 없으면 페이지를 감소하고 line을 40줄을 제거해서 전 페이지를 출력하도록한다.



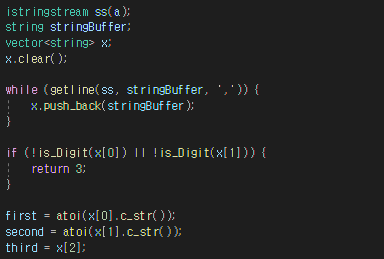
‘i’ 같은 경우 input\_I 함수를 실행해서 오류가 발생할 때 마다 리턴되는 숫자를 가지고 오류를 출력하고 아니면 함수를 통해 편집된 문서를 다시 새로 출력한다.

‘d’ , ‘s’ , ‘c’ , ‘t’ 같은 경우 ‘I’와 마찬가지로 동작한다.

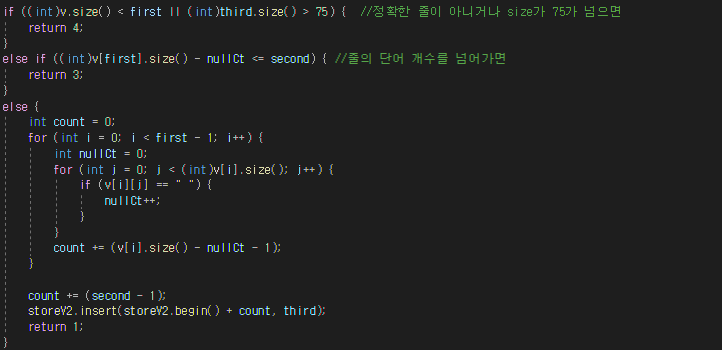
(5) input\_i()



command의 사이즈가 2보다 작으면 오류를 리턴한다. string a 는 command명령어에서 I,(,)를 제외한 크기만큼 사이즈를 할당하고, 괄호 안의 내용을 a에 저장한다.



‘,’로 a의 문자열을 나누기 위해서 벡터를 이용한다. i 명령어에서 첫번째, 두번째는 숫자이기 때문에 검사를 하고 정상이면 각각 변수에 저장한다.

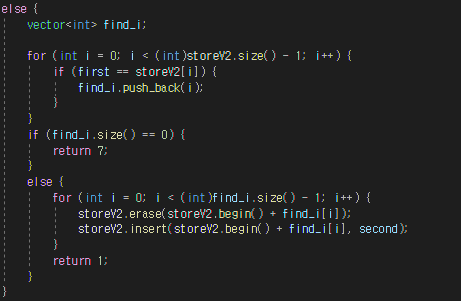


정확한 줄이 아니거나 단어가 75가 넘으면 오류를 리턴, 줄의 단어 개수를 넘기면 오류를 리턴한다. 정상적으로 작동하면 v를 탐색해서 “ “의 개수를 세고 총 단어의 개수에서 그만큼 뺀다. 그 다음 second만큼 더해주고 storeV2벡터의 count에 단어를 삽입한다.

(6) input\_D()

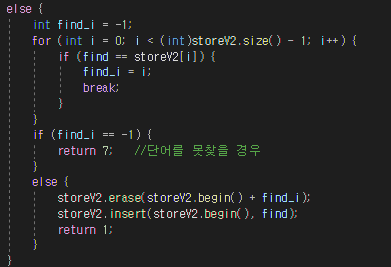
전체적인 부분이 i와 비슷하다.

(7) input\_c()



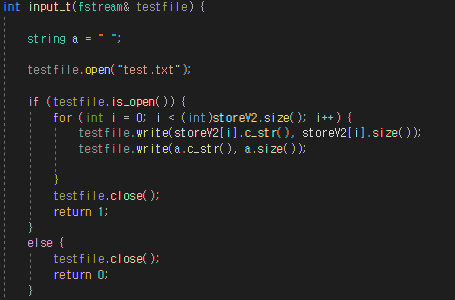
c 같은 경우 모든 단어를 찾아서 바꿔줘야 하기 때문에 처음부터 storeV2를 탐색해서 찾는 단어를 모두 삭제하고 새로운 단어를 삽입한다.

(8) input\_s()



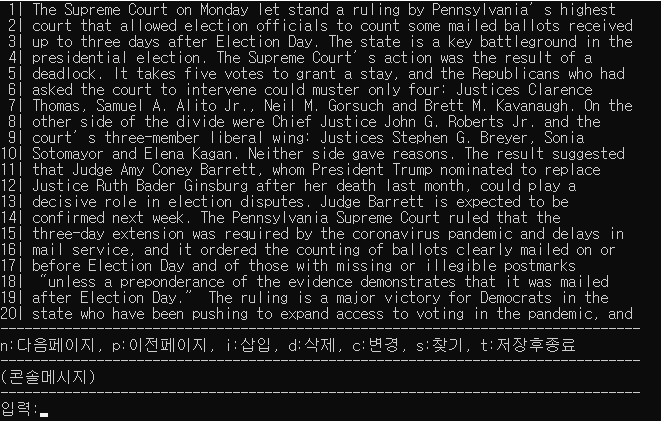
s같은 경우 storeV2를 탐색하는 중 제일 먼저 찾은 i번째 단어를 삭제하고 첫번째 위치에 삽입한다.

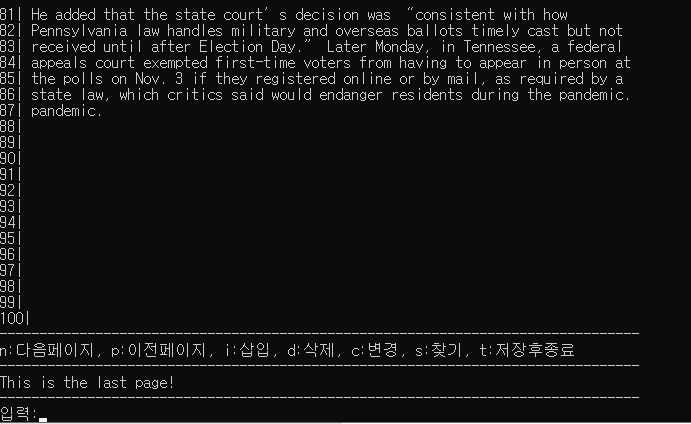
(9) input\_t



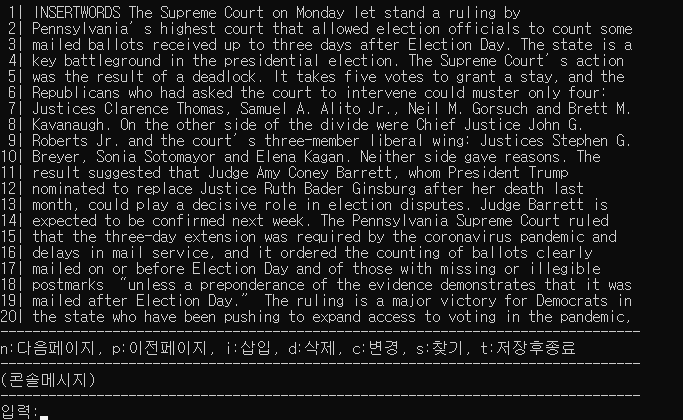
t는 편집된 txt를 저장하고 종료하는 함수이다. test.txt을 열고 편집된 storeV2를 처음부터 끝까지 저장한다. 단어 중간중간 빈칸을 넣어준다.

**4. 결과 화면 캡처**

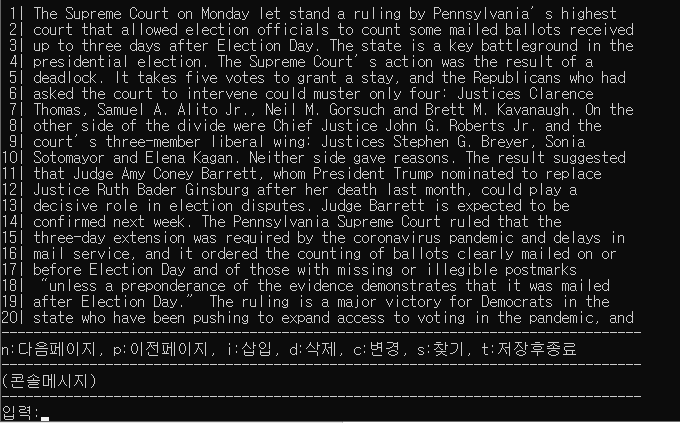




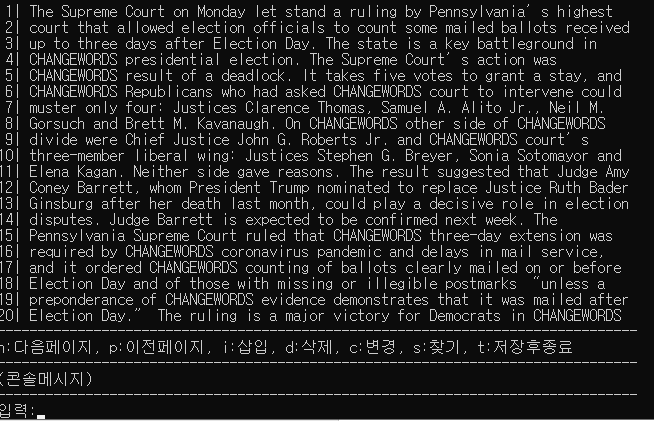
i(1,1,INSERTWORDS)



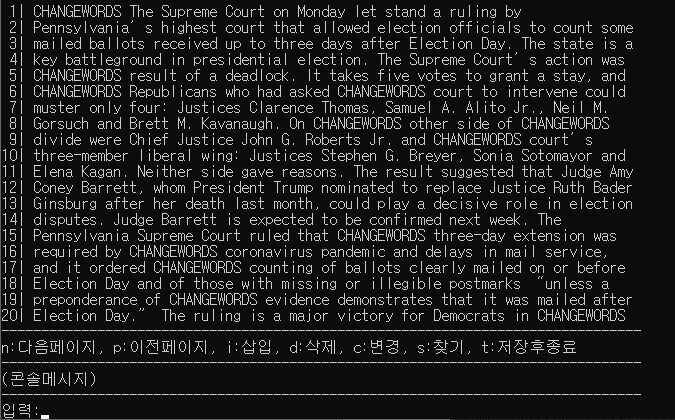
d(1,1)



c(the,CHANGEWORDS)



s(CHANGEWORDS)



t를 입력

