

# 자료구조 실습 보고서

숙제명: [제04주] 파일입출력구현

제출일: 2018.03.28

학번/이름: 201404377 / 진승언

이번 과제는 파일입출력을 사용법을 익히고 StringTokenizer를 MyStringTokenizer라는 이름으로 직접 구현해보는 것이었다.

먼저 주석을 보면 알 수 있듯이, MyStringTokenizer클래스의 각각의 변수가 뭘 뜻하는 건지 파악하였다. **String[] words**는 토큰을 저장할 공간이고 **String line**은 텍스트파일에서 받아올 전체문장, **String token**은 토큰을 나눌 기준, **int index**는 words배열의 각각의 값에 접근하는 포인터 역할이라는 것을 알았다.

이렇게 변수의 의미를 알아낸 후, **hasMoreTokens()**와 **nextToken()**을 구현할 수 있었다. **hasMoreTokens()**는 words의 배열의 길이와 토큰을 가리키는 포인터의 값이(int index) 같아지면 배열의 끝을 가리키고 있으므로 다음에 더 불러올게 없다는 의미니까 **false**를 반환하게 하고 아니면 **true**를 반환하게 구현하였다. 그리고 **nextToken()**은 토큰의 값들을 갖고 있는 words의 값을 index를 이용해서 순서대로 반환 시켜줄 수 있게 구현하였다. 다음 거를 계속해서 반환이 가능하게 index에 후위연산자를 사용하였다.

마지막으로 **String[] makeWords(String line, String token)**를 구현하였다. 이 메소드는 토큰을 나누는 기준으로 토큰을 나눠서 배열에 저장해 반환하는 메소드이다. 이 메소드도 처음에 myStringTokenizer클래스를 봤을 때처럼 지역변수의 의미를 먼저 파악했다. **String[] tmp**는 토큰을 저장할 공간이며 반환에 쓰일 배열이고, **int split\_index**는 tmp의 배열의 각각의 값에 접근할 포인터 역할을 하는 변수이고, **int split\_length**는 tmp배열의 크기를 의미하고, **begin\_index**와 **last\_index**는 각각 토큰을 찾았을 때 찾은 토큰의 처음과 끝을 가리키는 인덱스 역할을 한다는 것을 알았다.

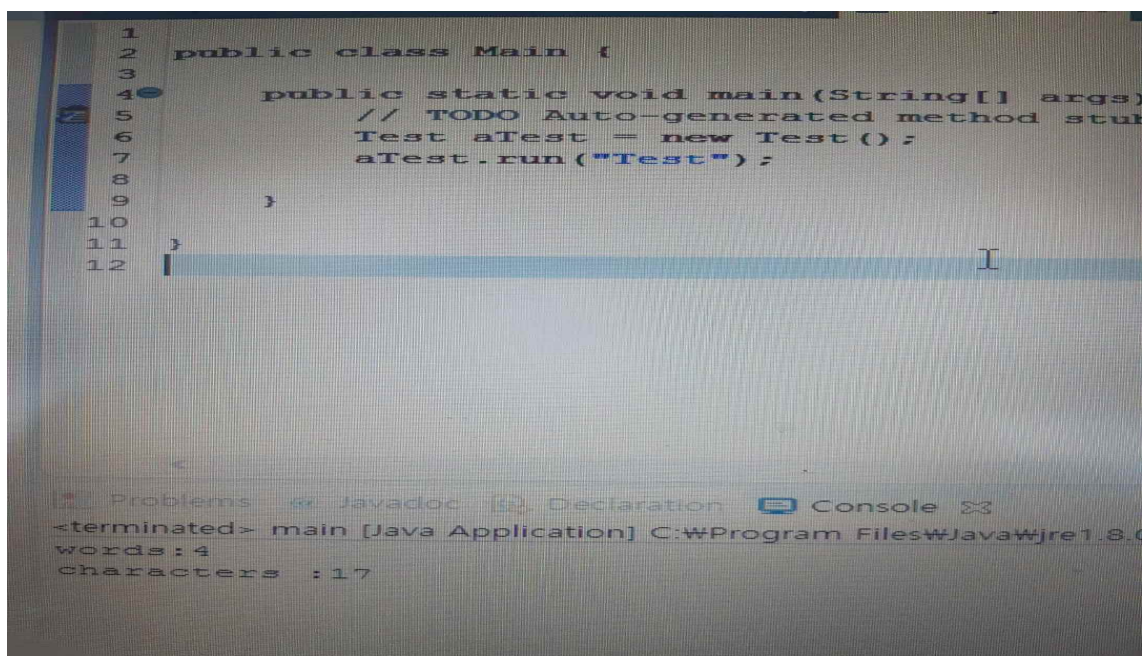
그 후, **else if**부분을 구현하였다. **else if**는 문장이 공백을 만났을

때 시행되는데 공백을 만나기 전의 토큰을 tmp에 **substring**을 이용해서 저장시키면 된다. 그러므로 공백을 만났을 때의 i를 last\_index에 넣어주고, tmp에 토큰을 넣어준다.(다음에 tmp의 다음 배열 칸에 넣을 수 있게 spilt\_index도 후위연산자로 늘려주었다.) 그 후, begin\_index는 공백다음을 가리켜주게 last\_index+1로 바꿔준다.

이렇게 makeWords 메소드는 문장의 토큰을 찾는 작업을 반복하고 저장해서 토큰이 담긴 배열을 반환해준다.

Test클래스에서 이제 입출력을 이용해 텍스트파일을 읽어오고 내가 만든 myStringTokenizer클래스를 생성해서 토큰의 개수와 토큰의 길이를 구하면 된다. 토큰을 불러올 때는 **hasMoreTokens()**을 이용해서 남은 토큰이 있으면 불러오고 토큰을 불러오면 토큰의 개수를 1증가시키고(++words) 불러온 토큰의 길이를 chars에 더해주었다. 그렇게 구한 words와 chars를 출력시켜주었다.

밑은 실행결과이다. 결과가 옳게 나옴을 볼 수 있었다.



```
1 public class Main {
2
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         Test aTest = new Test();
7         aTest.run("Test");
8     }
9
10 }
11
12
```

Problems Javadoc Declaration Console

```
<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0
words: 4
characters : 17
```