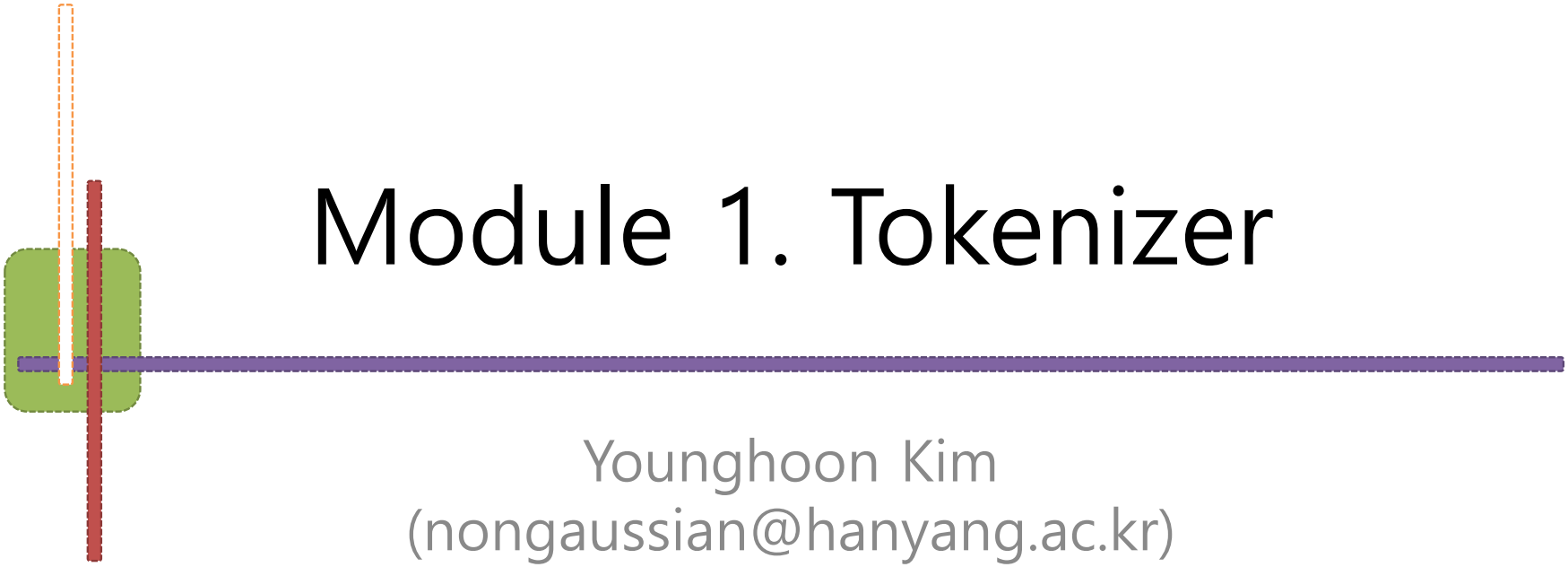


Module 1. Tokenizer

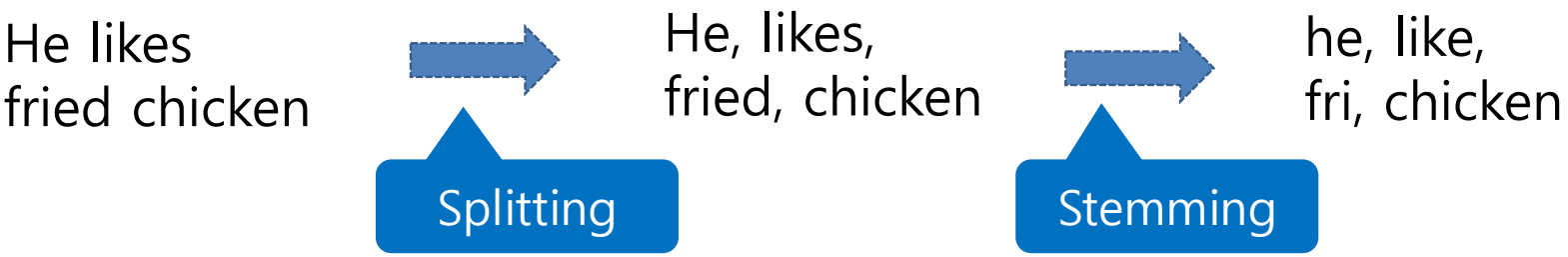
Younghoon Kim
(nongaussian@hanyang.ac.kr)

Note

- The goal of Module 1 is to
  - Understand how to implement the given interface
  - Understand the structure of HanyangSE-submit
  - Understand how to define the dependency to the external library using Maven's pom.xml file

Problem Definition

- Given
  - A string (e.g., sentence, article) of type String
- Return
  - A list of terms which is split by whitespaces and stemmed
  - Type: List<String>

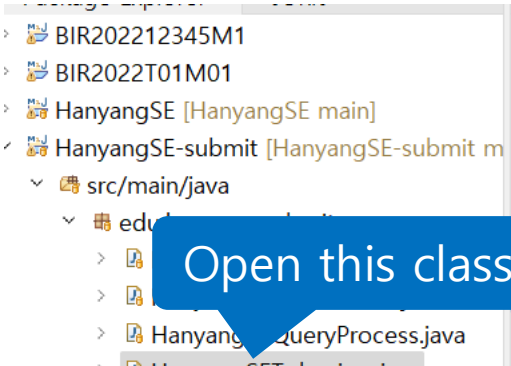


Code Template

- We provide a package of
  - A maven projects HanyangSE-submit
- HanyangSE-submit
  - Contains
    - Template codes (`edu.hanyang.submit.HanyangSETokenizer.java`)
    - JUnit test codes
  - Depends on a package
    - Artifact ID: HanyangSE
    - Group ID: io.github.hyerica-bdml
- HanyangSE
  - Includes
    - Interface class (e.g., Tokenizer.java)

Code Template

- Complete *edu.hanyang.submit.HanyangSETokenizer*



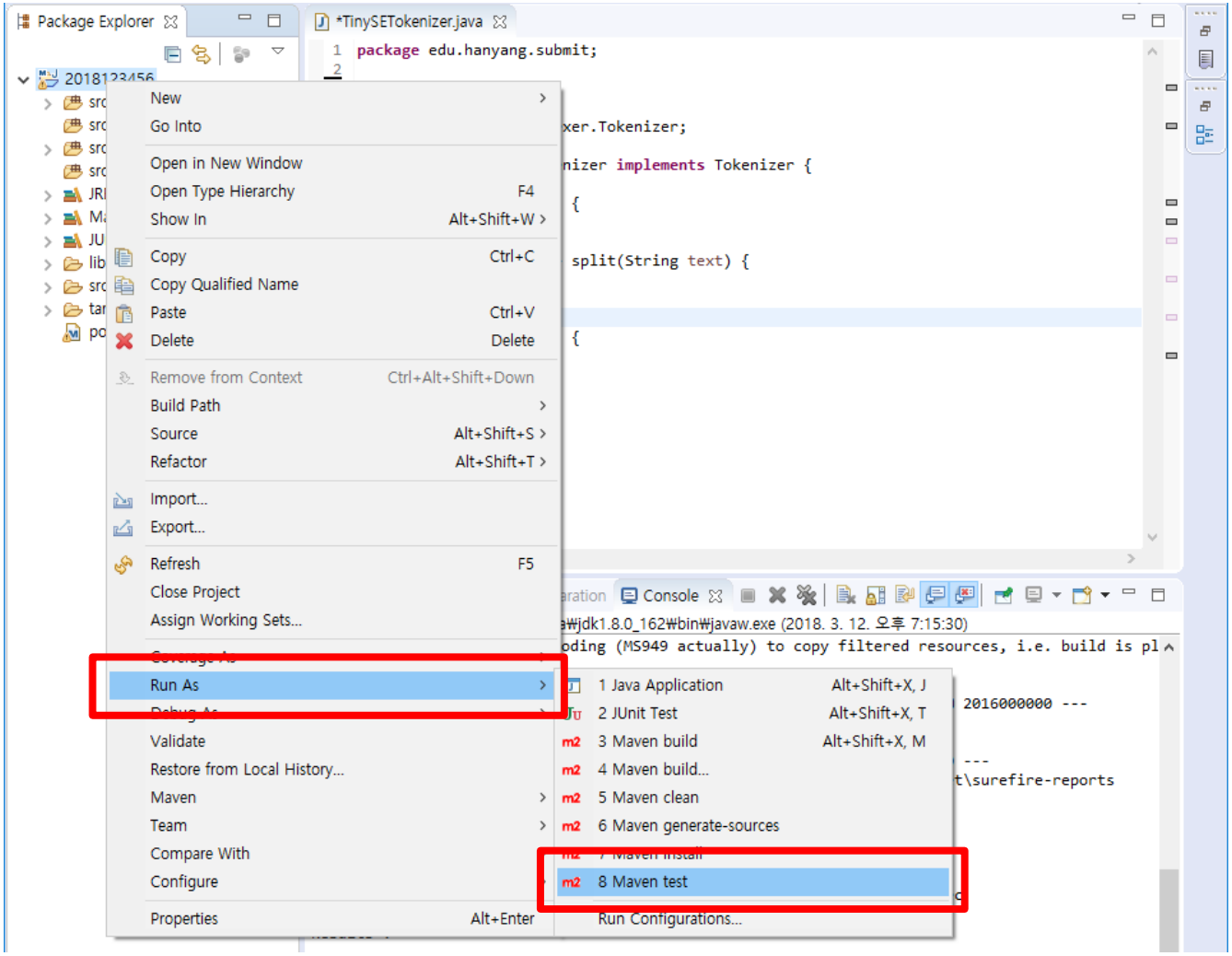

```
6
7
8  public class HanyangSETokenizer implements Tokenizer {
9
10     /**
11      * Tokenizer 객체를 생성하고 준비하는 단계
12      */
13     @Override
14     public void setup() {
15         // TODO: your code here...
16     }
17
18     /**
19      * 입력 문장을 split 및 tokenizer하는 단계
20      * @param str
21      * @return
22      */
23     @Override
24     public List<String> split(String str) {
25         // TODO: your code here...
26         return null;
27     }
28
29     /**
30      * Tokenizer 객체를 모두 닫는 단계
31      */
```

Implement this three method

USE *org.apache.lucene.analysis.core.SimpleAnalyzer* and *org.tartarus.snowball.ext.PorterStemmer*

Submission

- Test your code



Submission

- Test your code

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running edu.hanyang.TokenizerTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.071 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------
[INFO] Total time: 1.020 s
[INFO] Finished at: 2018-03-12T19
[INFO] Final Memory: 10M/243M
[INFO] ------------------------
```

Check if your code passes our unit test. Confirm the message with no failures, no errors and no skipped.

External Libraries

- Use SimpleAnalyzer and PotterStemmer in Lucene 7.2.1
  - SimpleAnalyzer is a tokenizer that splits a sentence with whitespaces
  - PotterStemmer is a well-known and simple stemmer for English
  - Dependency on Lucene is already defined in pom.xml
- JavaDoc
  - SimpleAnalyzer: https://lucene.apache.org/core/7_2_1/analyzers-common/index.html?org/apache/lucene/analysis/core/SimpleAnalyzer.html
  - PorterStemmer: https://lucene.apache.org/core/7_2_1/analyzers-common/org/tartarus/snowball/ext/PorterStemmer.html

Dependencies in POM

```
<dependency>
    <groupId>org.apache.lucene</groupId>
    <artifactId>lucene-analyzers-common</artifactId>
    <version>7.2.1</version>
</dependency>
<dependency>
    <groupId>org.apache.lucene</groupId>
    <artifactId>lucene-core</artifactId>
    <version>7.2.1</version>
</dependency>

…

<dependency>
    <groupId>io.github.hyerica-bdml</groupId>
    <artifactId>hanyangse</artifactId>
    <version>1.0</version>
</dependency>
```

Interface

```
public interface Tokenizer {

    /**
     * Initializing a tokenizer
     *
     */
    void setup();

    /**
     * Extracting tokens from a given input string
     *
     * @param strInput string
     * @returnList of tokens
     */
    List<String> split(String str);

    /**
     * Finalizing the tokenizer
     *
     */
    void clean();
}
```

Sample Code

```
public class HanyangSETokenizer implements Tokenizer {
        private Analyzer analyzer = null;
        private PorterStemmer s = null;
        public void setup() {
                analyzer = new SimpleAnalyzer();
                s = new PorterStemmer();
        }
        public List<String> split(String text) {
                List<String> result = new ArrayList<String>();
                try {
                        TokenStream stream = analyzer.tokenStream(null, new StringReader(text));
                        while (stream.incrementToken()) {
                                result.add(stemString(
                                        stream.getAttribute(
                                        CharTermAttribute.class).toString()));
                        }
                        stream.close();
                } catch (IOException e) {
                        throw new RuntimeException(e);
                }
                return result;
        }
        public void clean() { analyzer.close(); }
        private String stemString(String word) { s.setCurrent(word); s.stem(); return s.getCurrent(); }
}
```