Stage 1. Tokenization

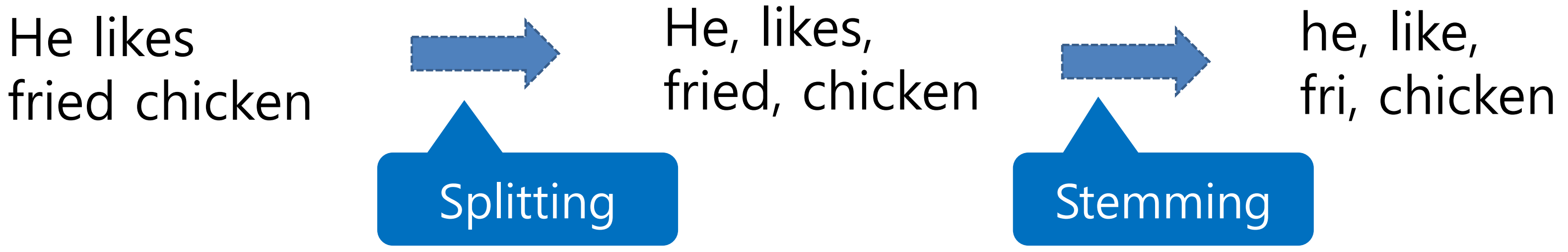
Younghoon Kim
(nongaussian@hanyang.ac.kr)

Note

- The goal of the first stage is
  - To practice how to write and commit/submit your code to Github

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
  - Two maven projects TinySE-submit and TinySE
- TinySE-submit
  - Contains
    - Template codes (`edu.hanyang.submit.TinySETokenizer.java`)
    - JUnit test codes
  - Depend on
    - TinySE framework (<github>/nongaussian/`tinyse`) ← to be updated on every stage
- TinySE
  - Includes
    - Interface files (e.g., Tokenizer.java)
    - Indexer and query processer codes which will complete a search engine by connecting your submissions

Complete Interface in TinySE-submit

- Step 1. Download codes
  - Clone TinySE on local (X)
  - Fork TinySE-submit on your account
  - Clone the TinySE-submit fork on local (Y)
- Step 2. Build TinySE
  - Build package with X & confirm the created jar file
  - Define the dependency on the jar in pom.xml of Y
- Step 3. Write your codes
  - Complete the template codes in Y
  - Run mvn package & mvn test
- Step 4. Submit your module
  - Commit & push into your TinySE-submit fork

Step 1. Download codes

Clone TinySE on local

```
$ git clone https://github.com/nongaussian/TinySE.git
Cloning into 'TinySE'...
remote: Enumerating objects: 583, done.
remote: Total 583 (delta 0), reused 0 (delta 0), pack-reused 583
Receiving objects: 100% (583/583), 1.22 MiB | 209.00 KiB/s, done.
Resolving deltas: 100% (189/189), done.
```

Clone the TinySE-submit fork on local

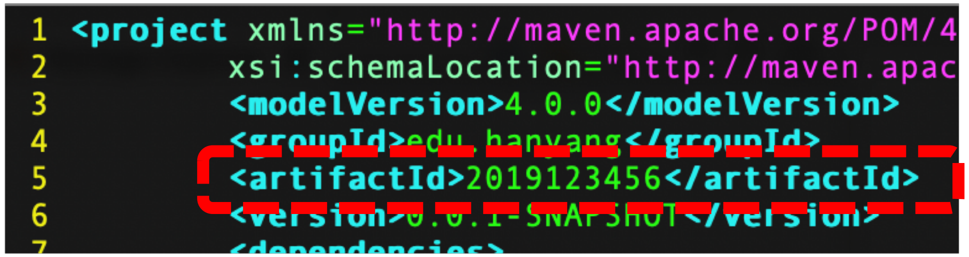
```
$ git clone https://github.com/<your account>/TinySE-submit.git
Cloning into 'TinySE-submit'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 294 (delta 0), reused 3 (delta 0), pack-reused 291
Receiving objects: 100% (294/294), 15.25 MiB | 3.07 MiB/s, done.
Resolving deltas: 100% (102/102), done.
```

Step 2. Build TinySE

```
$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -------------------------< edu.hanyang:tinyse >-------------------------
[INFO] Building Tiny Search Engine 2018.stage_4.build_1
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ tinyse ---

 …

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ tinyse ---
[INFO] Building jar: /Users/yhkim/git/TinySE/target/tinyse-2019.stage_1.build_1.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.379 s
[INFO] Finished at: 2019-03-20T23:05:43+09:00
[INFO] ------------------------------------------------------------------------
```

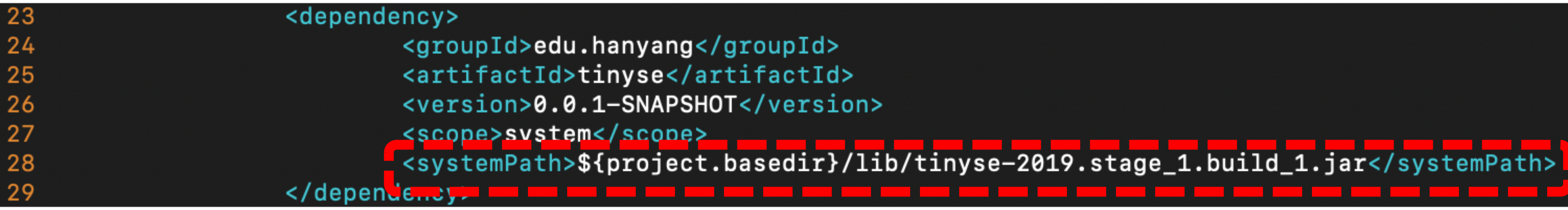
Edit pom.xml of TinySE-submit project

- Change the artifact ID to your student ID in “pom.xml”



```
<project xmlns="http://maven.apache.org/POM/4
        xsi:schemaLocation="http://maven.apac
        <modelVersion>4.0.0</modelVersion>
        <groupId>edu.hanyang</groupId>
        <artifactId>2019123456</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <dependencies>
```

- Copy TinySE jar package file to ${project.basedir}/lib
- Add dependency

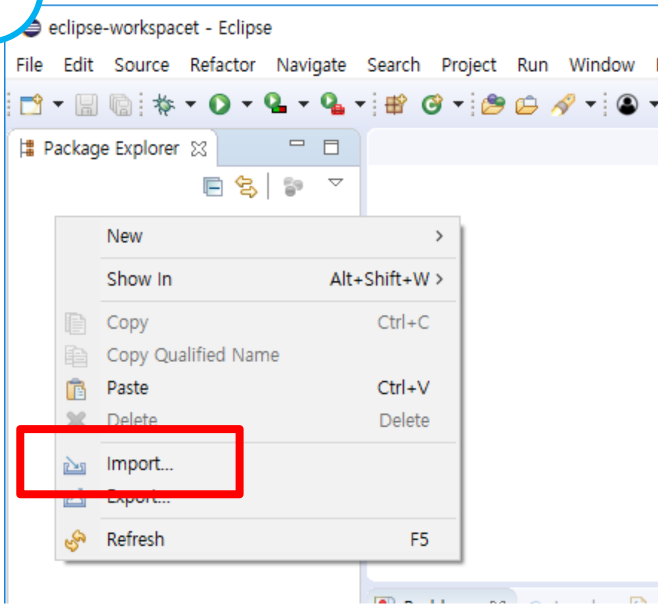


```
		<dependency>
			<groupId>edu.hanyang</groupId>
			<artifactId>tinyse</artifactId>
			<version>0.0.1-SNAPSHOT</version>
			<scope>system</scope>
			<systemPath>${project.basedir}/lib/tinyse-2019.stage_1.build_1.jar</systemPath>
		</dependency>
```

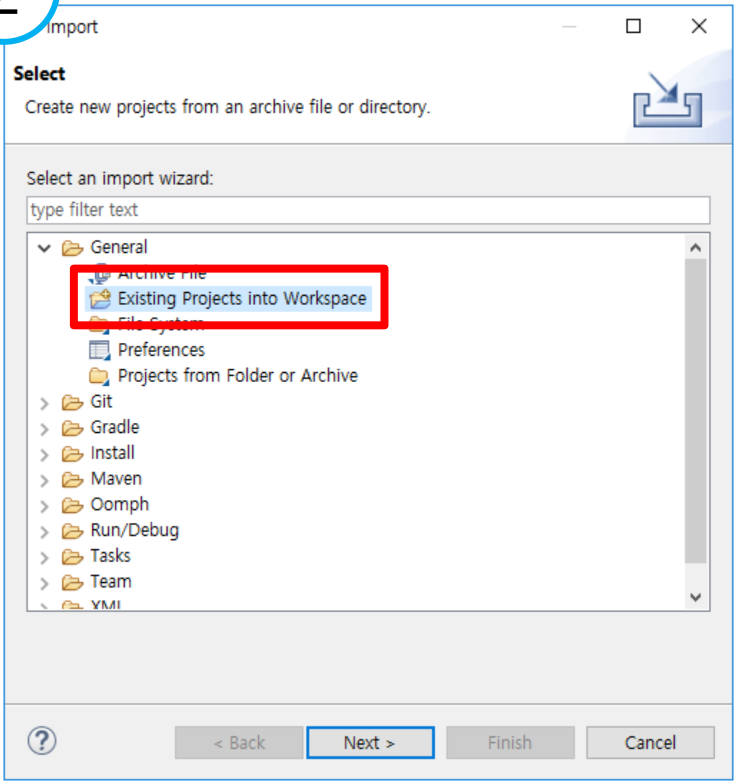
Step 3. Write your codes using Eclipse

- Import the maven project in Eclipse IDE

1



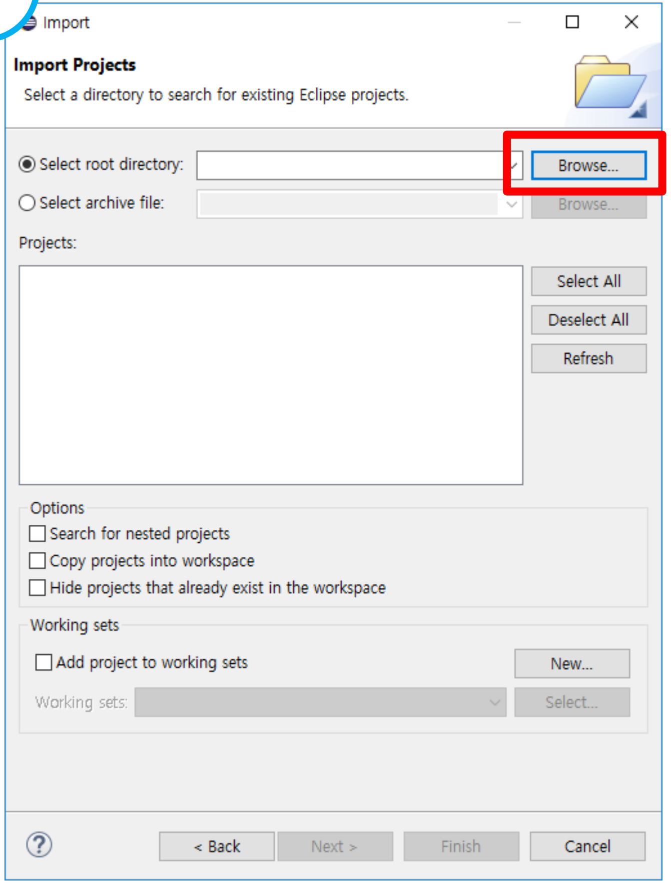
2



To Use Code Template

- Import the project in eclipse IDE

3



Import

Import Projects

Select a directory to search for existing Eclipse projects.

(●) Select root directory: [ ] Browse...

( ) Select archive file: [ ] Browse...

Projects:

Select All | Deselect All | Refresh

Options

☐ Search for nested projects

☐ Copy projects into workspace

☐ Hide projects that already exist in the workspace

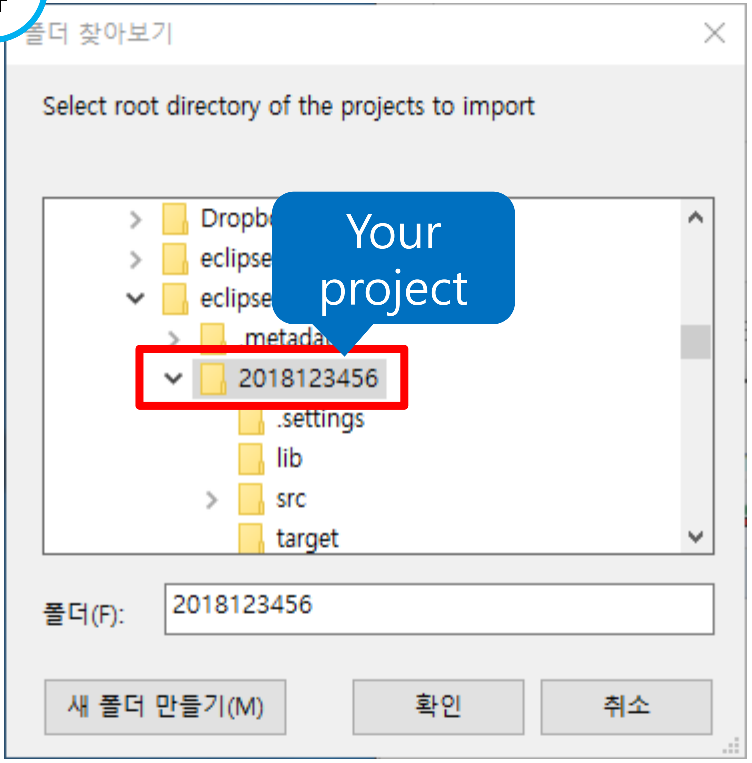
Working sets

☐ Add project to working sets — New...

Working sets: — Select...

< Back | Next > | Finish | Cancel

4



폴더 찾아보기

Select root directory of the projects to import

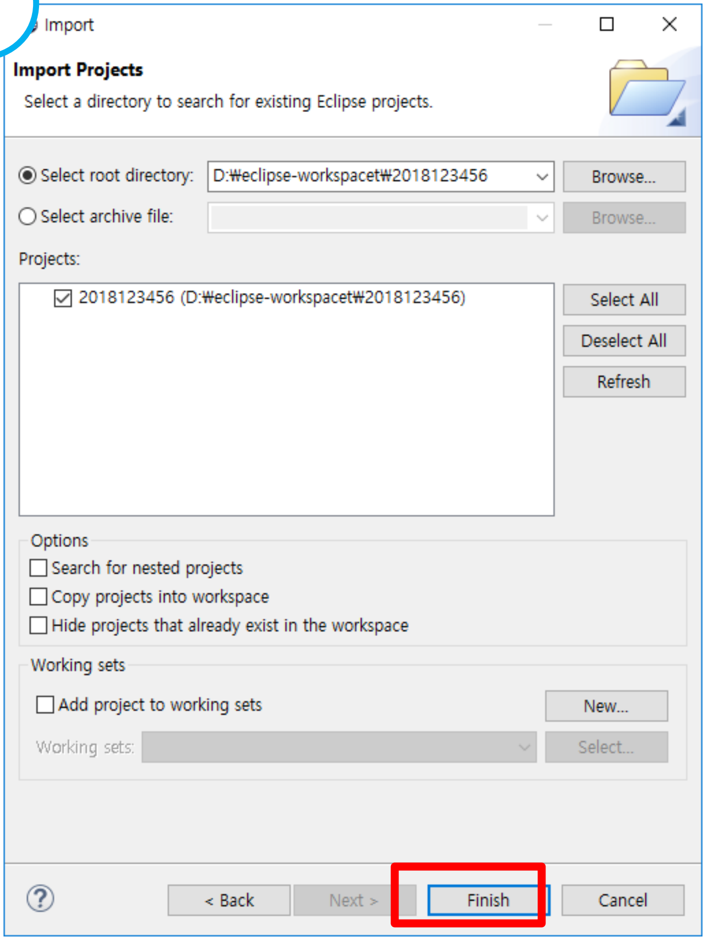
폴더(F): 2018123456

새 폴더 만들기(M) | 확인 | 취소

To Use Code Template

- Import the project in eclipse IDE

5



Import

Import Projects

Select a directory to search for existing Eclipse projects.

Select root directory: D:₩eclipse-workspacet₩2018123456 [Browse...]

Select archive file: [Browse...]

Projects:

☑ 2018123456 (D:₩eclipse-workspacet₩2018123456)

[Select All] [Deselect All] [Refresh]

Options

☐ Search for nested projects
☐ Copy projects into workspace
☐ Hide projects that already exist in the workspace

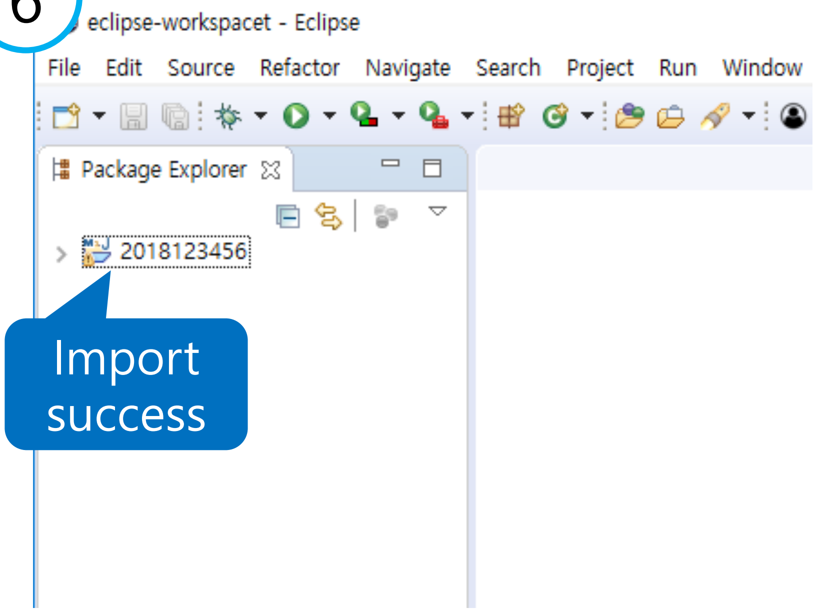
Working sets

☐ Add project to working sets [New...]

Working sets: [Select...]

[< Back] [Next >] [Finish] [Cancel]

6



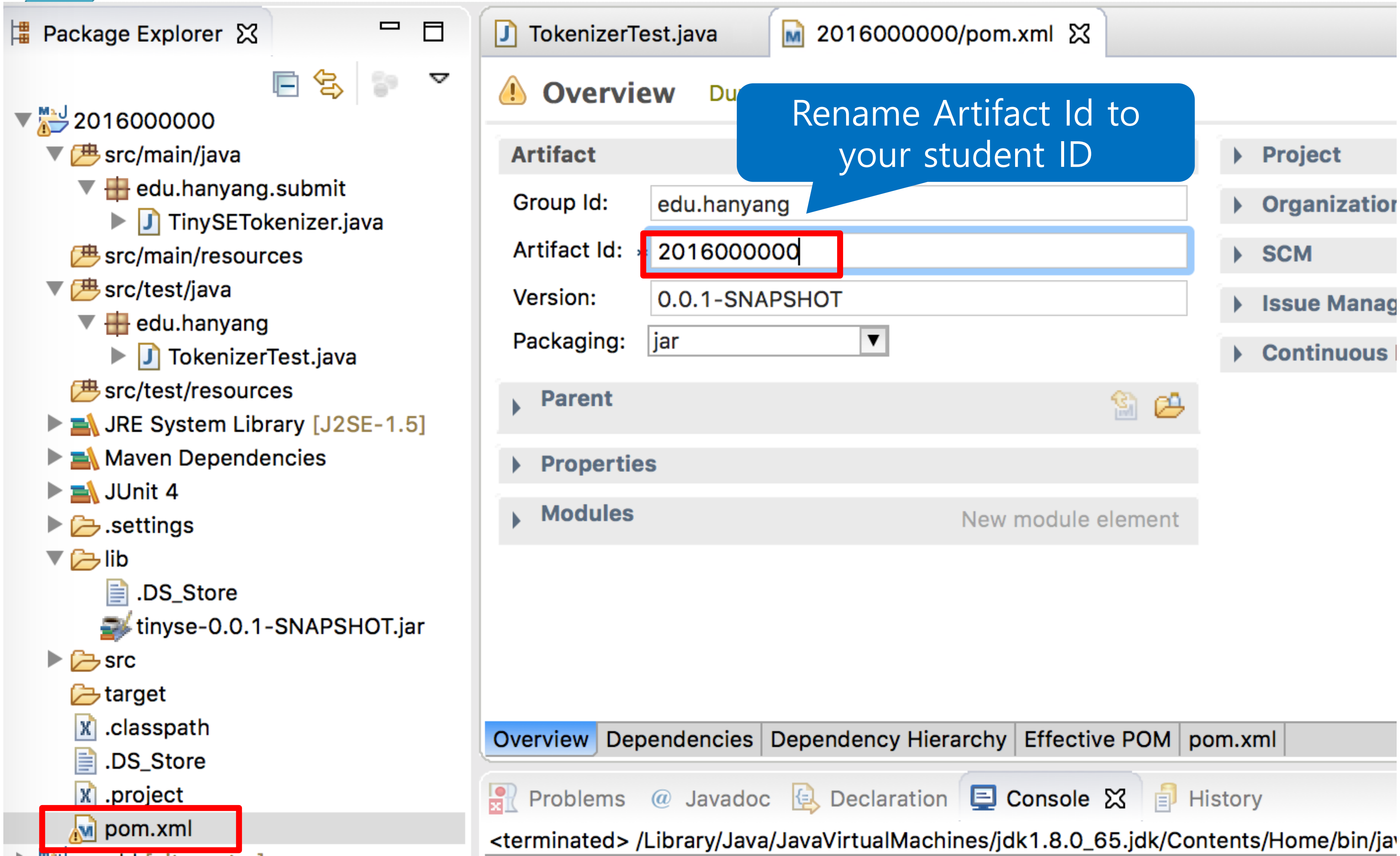
eclipse-workspacet - Eclipse

File Edit Source Refactor Navigate Search Project Run Window

Package Explorer

2018123456

To Use Code Template



Package Explorer

- 2016000000
  - src/main/java
    - edu.hanyang.submit
      - TinySETokenizer.java
  - src/main/resources
  - src/test/java
    - edu.hanyang
      - TokenizerTest.java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - JUnit 4
  - .settings
  - lib
    - .DS_Store
    - tinyse-0.0.1-SNAPSHOT.jar
  - src
  - target
  - .classpath
  - .DS_Store
  - .project
  - pom.xml

TokenizerTest.java | 2016000000/pom.xml

Overview

Rename Artifact Id to your student ID

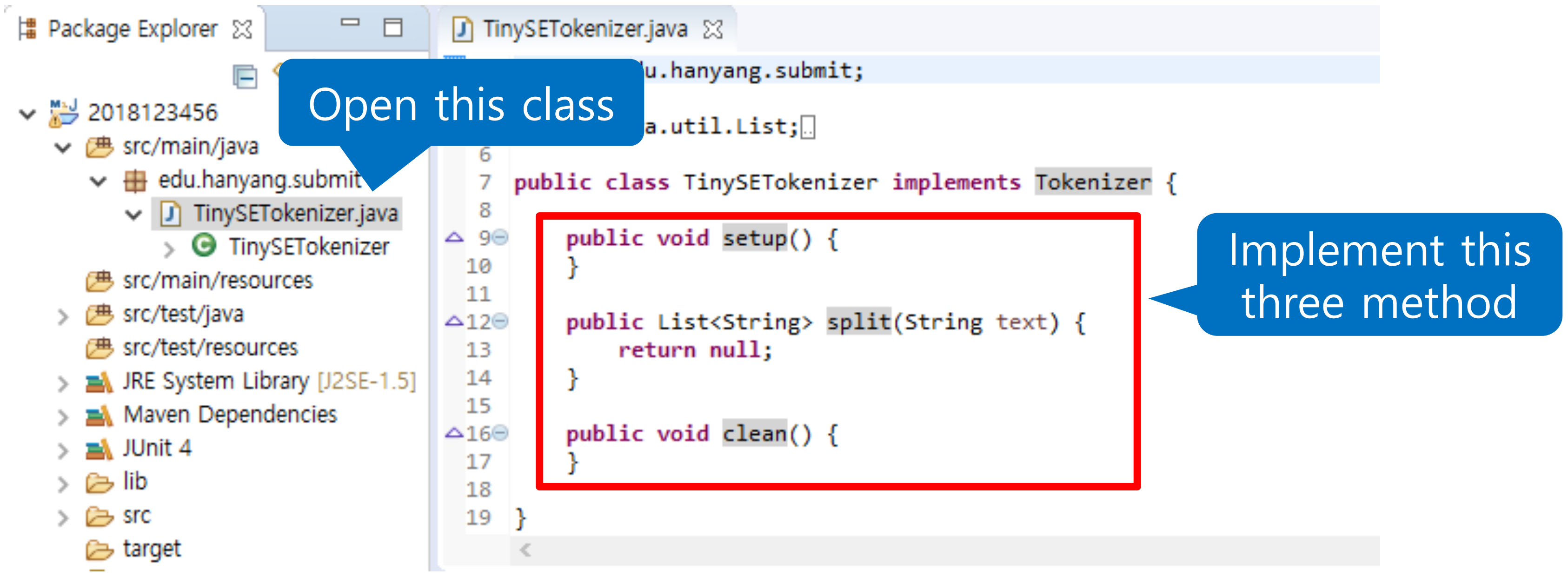
Artifact

Group Id:	edu.hanyang
Artifact Id:	2016000000
Version:	0.0.1-SNAPSHOT
Packaging:	jar

```
<terminated> /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/ja
```

To Use Code Template

- Complete *edu.hanyang.submit.TinySETokenizer*

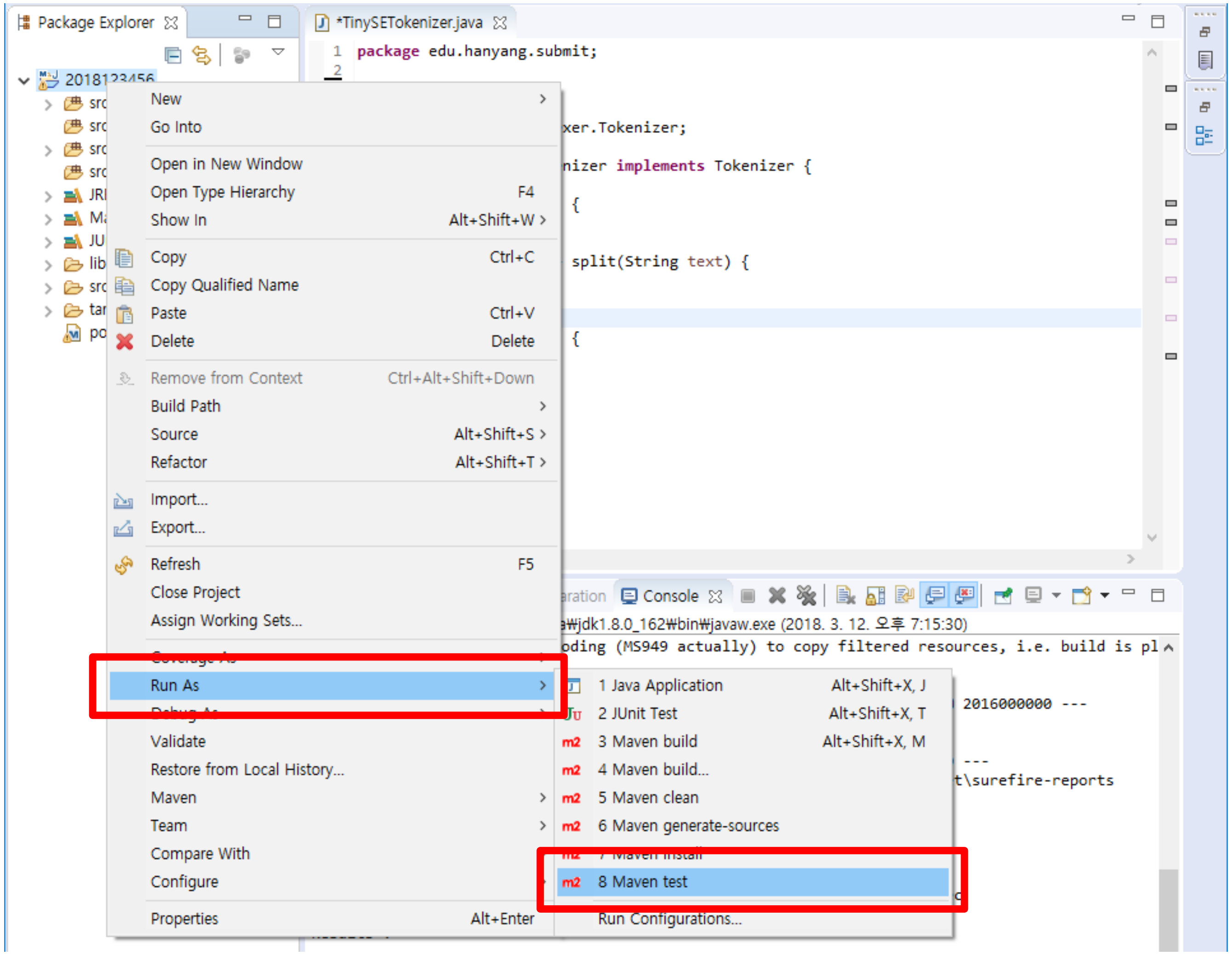


```
u.hanyang.submit;

a.util.List;

public class TinySETokenizer implements Tokenizer {

    public void setup() {
    }

    public List<String> split(String text) {
        return null;
    }

    public void clean() {
    }

}
```

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

To Use Code Template

- 6. Test your code

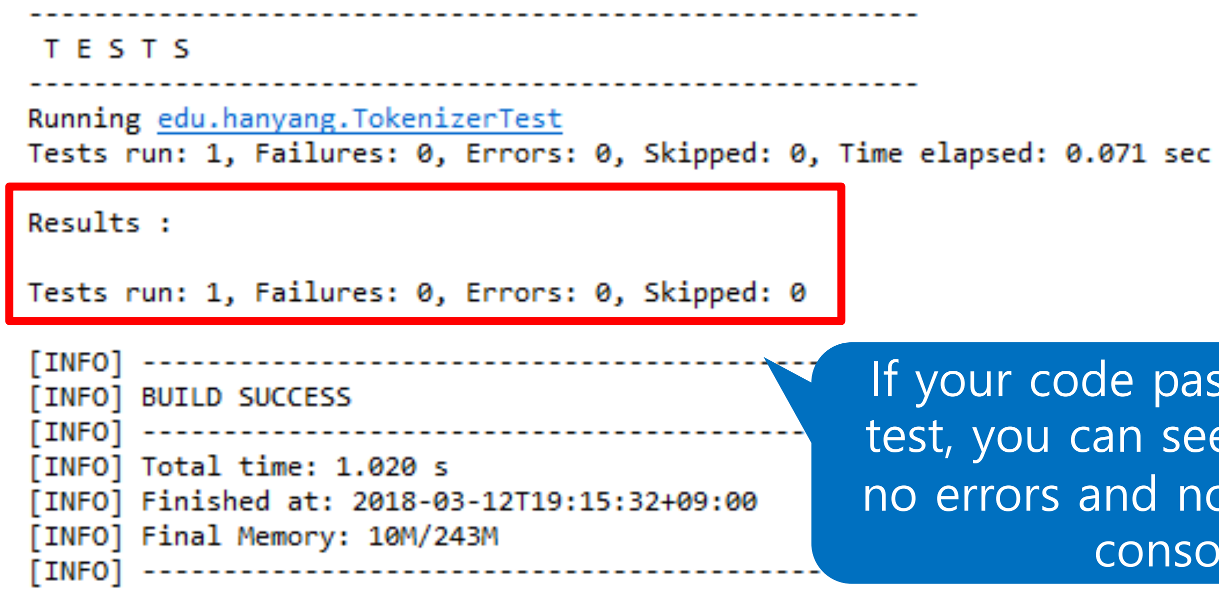


```
package edu.hanyang.submit;
...xer.Tokenizer;
...nizer implements Tokenizer {
    {
 split(String text) {
    {
```

```
...₩jdk1.8.0_162₩bin₩javaw.exe (2018. 3. 12. 오후 7:15:30)
...oding (MS949 actually) to copy filtered resources, i.e. build is pl
... 2016000000 ---
... ---
...t\surefire-reports
```

To Use Code Template

- 6. 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: 2018-03-12T19:15:32+09:00
[INFO] Final Memory: 10M/243M
[INFO] ------------------------------------------------
```

If your code passes our unit test, you can see no failures, no errors and no skipped on console

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

Submission

- How to submit
  - Push the change into your repository
- Due
  - Mar. 28 (Thu) 11:59pm