Test Smell Detection Tool for General Fixture Smell

–

Course Name: Software Testing and Quality Assurance
Course Code: SE 605

Submitted by
Noshin Tahsin
BSSE 0914

Submitted To

Abdus Satter

Lecturer

Institute of Information Technology,

University of Dhaka

04.11.2019

1. Overview

This is the user manual for the Test Smell Detection Tool. This Tool can detect the presence of "General Fixture" Test Smell in the test classes of a project.

The general fixture smell occurs if test classes contain broad functionality in the implicit setup, and different tests only access part of the fixture.

2. How to Use the Tool

2.1 Open the project in Eclipse

After extracting the java project, open the project in eclipse.

2.2 Add External Jar File

A Jar file (javaparser-core-3.2.4.jar) is attached in the zip file. It must be added as external jar.

2.3 Run the Project

After running the project, a UI like Figure 1 will be opened.

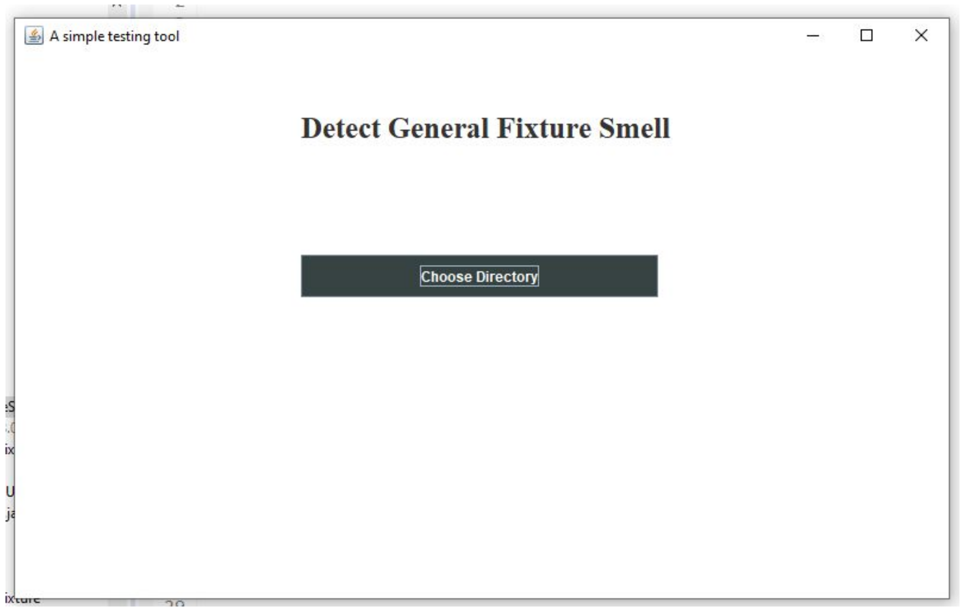
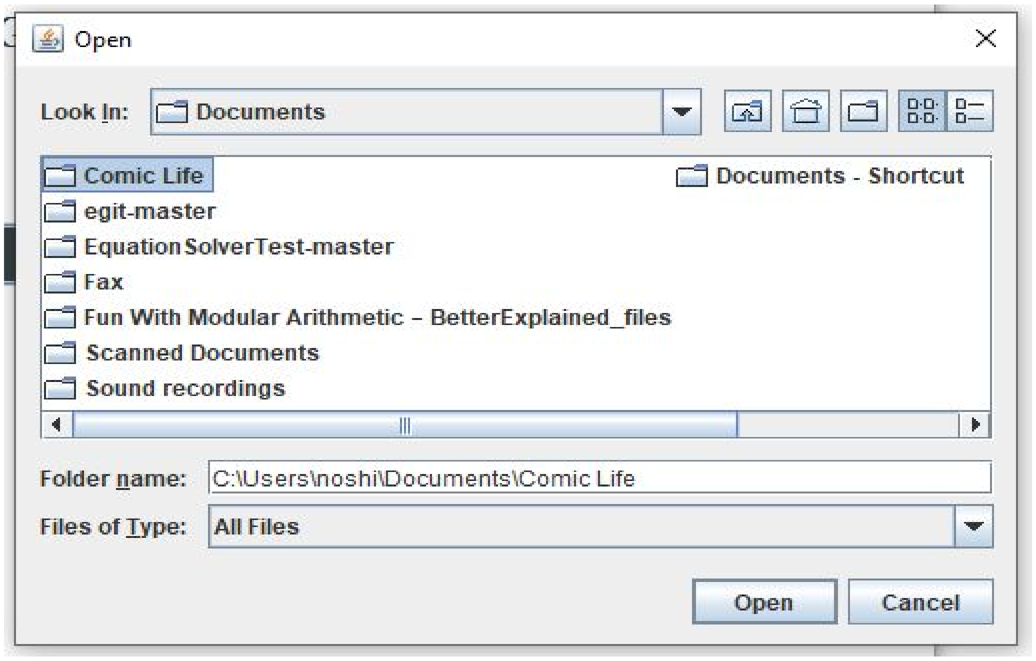


Figure 1: The UI

2.4 Click the "Choose Directory" Button

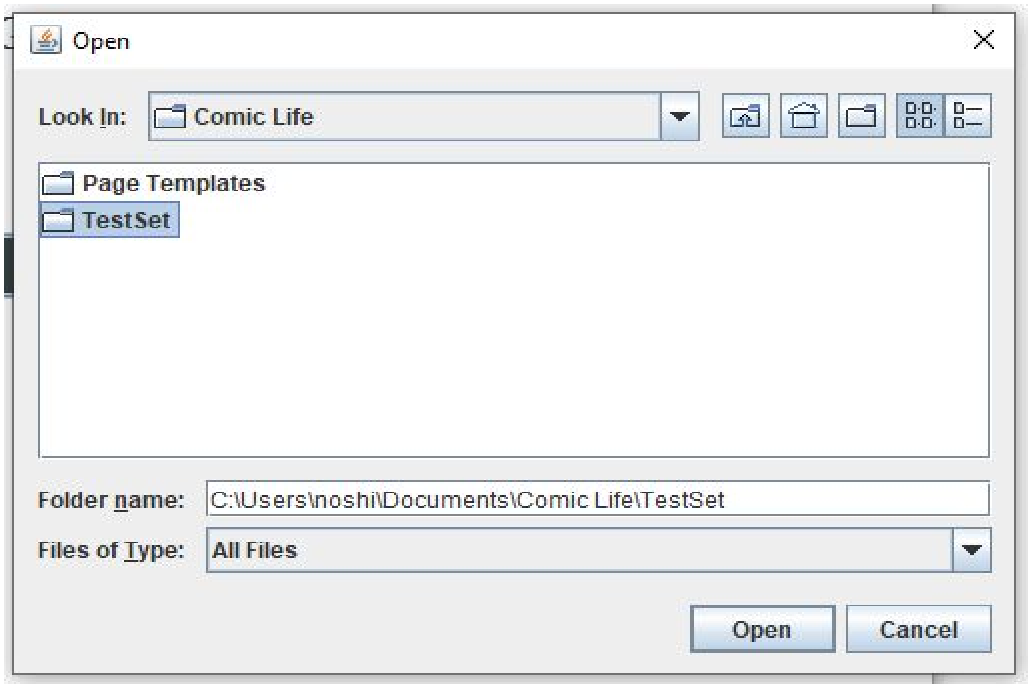
Choose the directory where the source code resides. (Figure 2, Figure 3)



Folder name: C:\Users\noshi\Documents\Comic Life

Files of Type: All Files

Figure 2: Choosing Directory



Folder name: C:\Users\noshi\Documents\Comic Life\TestSet

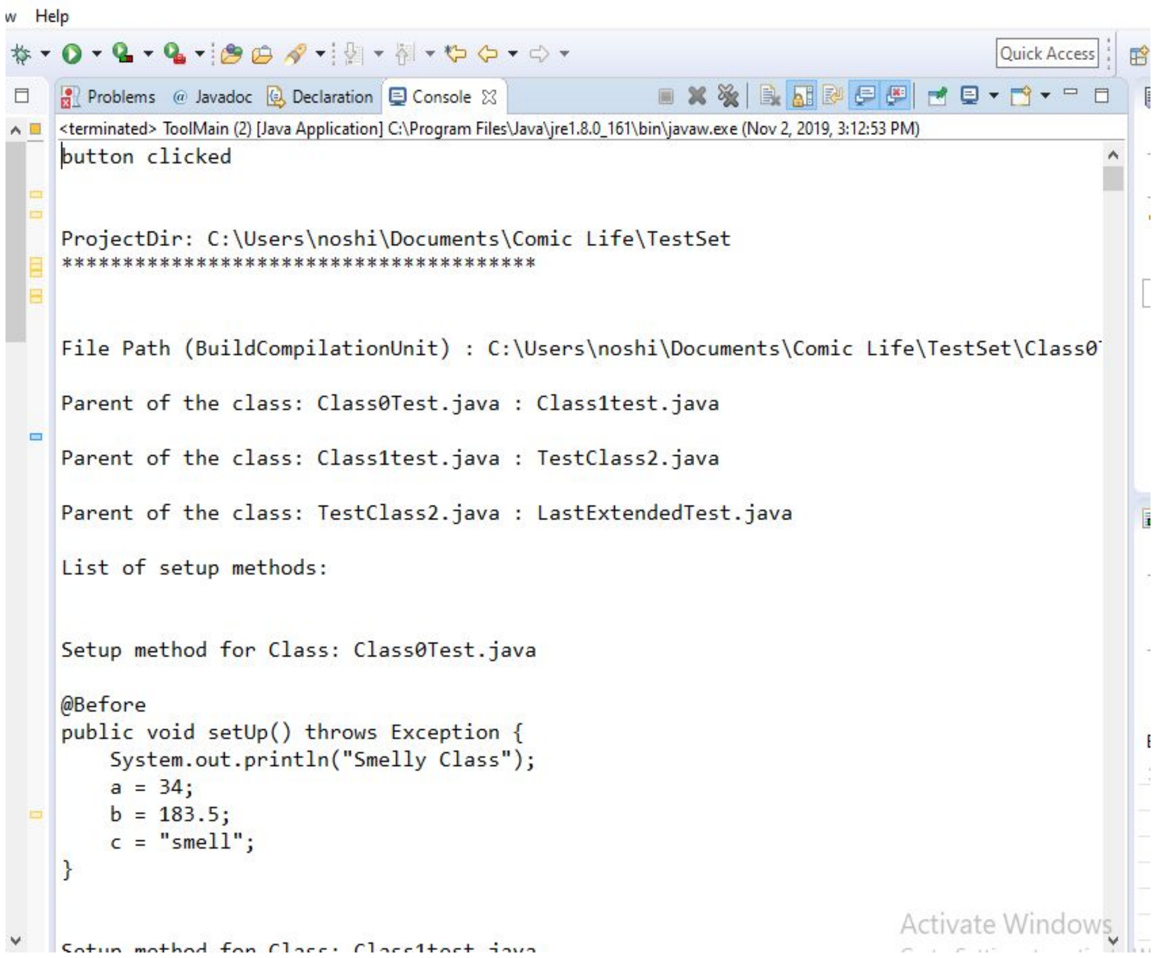
Files of Type: All Files

Figure 3: Choosing Directory

2.5 Choose the Directory where the Test Classes are (The TestCase folder in this case)

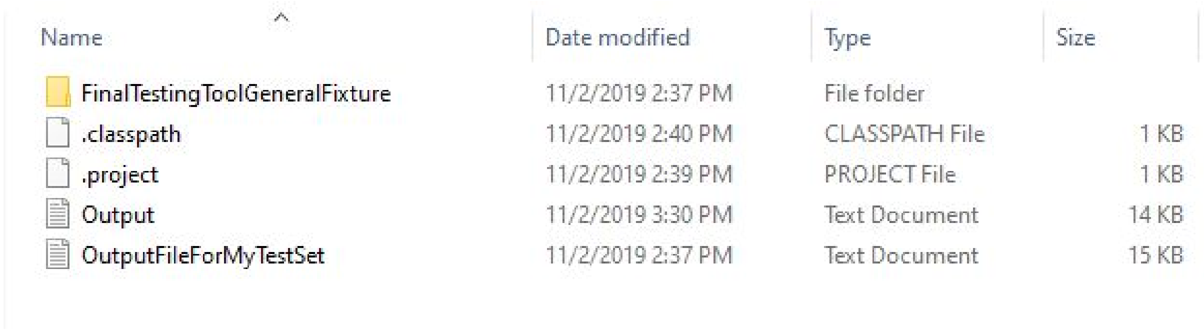
2.6 Get the output

You will get the output in the console (Figure 4) as well as in an output file (Output.txt) residing in your project directory. (Figure 5)



```
<terminated> ToolMain (2) [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (Nov 2, 2019, 3:12:53 PM)
button clicked


ProjectDir: C:\Users\noshi\Documents\Comic Life\TestSet
****************************************



File Path (BuildCompilationUnit) : C:\Users\noshi\Documents\Comic Life\TestSet\Class0

Parent of the class: Class0Test.java : Class1test.java

Parent of the class: Class1test.java : TestClass2.java

Parent of the class: TestClass2.java : LastExtendedTest.java

List of setup methods:


Setup method for Class: Class0Test.java

@Before
public void setUp() throws Exception {
    System.out.println("Smelly Class");
    a = 34;
    b = 183.5;
    c = "smell";
}
```

Figure 4: Output on console



Name	Date modified	Type	Size
FinalTestingToolGeneralFixture	11/2/2019 2:37 PM	File folder	
.classpath	11/2/2019 2:40 PM	CLASSPATH File	1 KB
.project	11/2/2019 2:39 PM	PROJECT File	1 KB
Output	11/2/2019 3:30 PM	Text Document	14 KB
OutputFileForMyTestSet	11/2/2019 2:37 PM	Text Document	15 KB

Figure 5: The output file "Output.txt"

3. Description of the Output file:

3.1 The First line is showing the path of the chosen directory. (Figure 6)

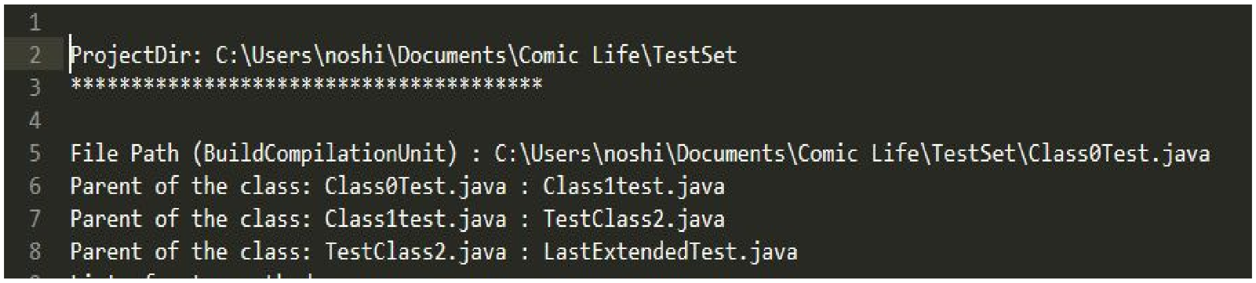
3.2 The File Path specifies the path of the file that is currently being parsed to detect smell. (Figure 6)

3.3 There were a total of 6 classes in my sample test set (Folder: TestSet). The 6 classes are:

a) Class0Test.java,
b) Class1Test.java,
c) LastExtendedtest.java,
d) TestClass2.java,
e) testClassNotExtended.java,
f) TestNoSetup.java.

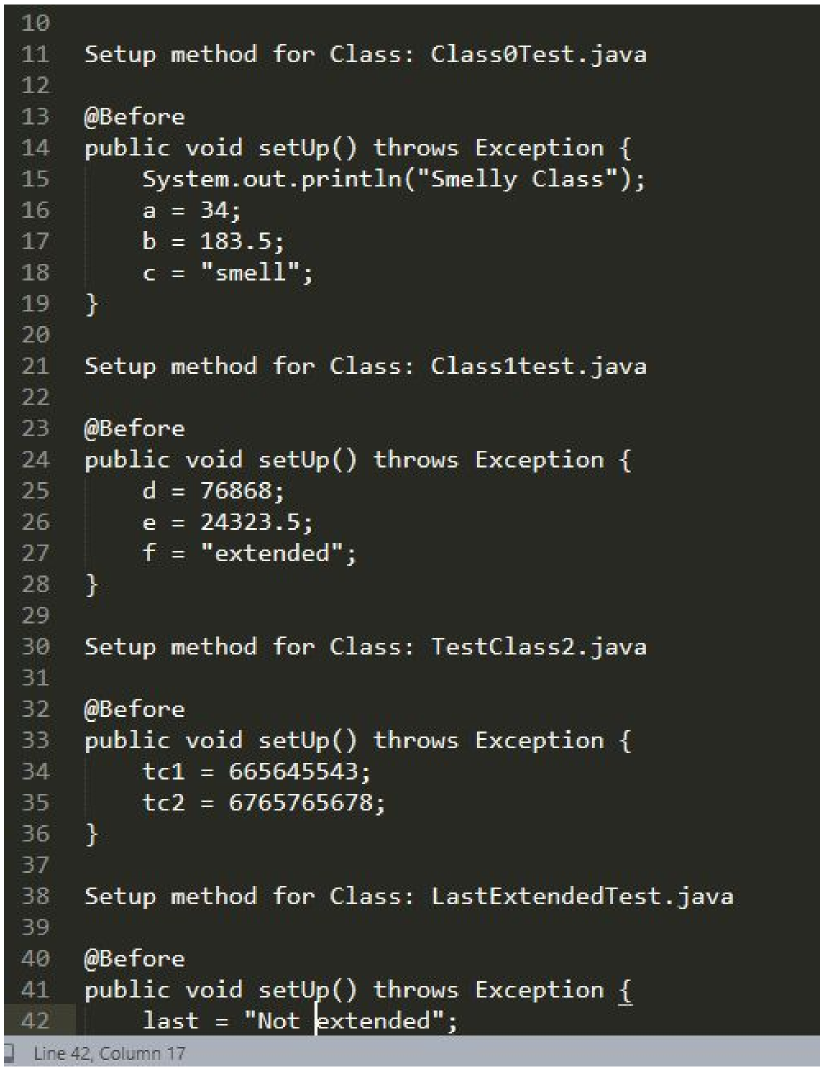
The first class : Class0Test.java will be parsed now. (Figure 6)

3.4 Class0Test.java extends Class1Test.java, Class1Test.java extends TestClass2.java and TestClass2.java extends LastExtendedtest.java. This is shown in the output file through line 6-8. (Figure 6)



```
1
2 ProjectDir: C:\Users\noshi\Documents\Comic Life\TestSet
3 ***************************************
4
5 File Path (BuildCompilationUnit) : C:\Users\noshi\Documents\Comic Life\TestSet\Class0Test.java
6 Parent of the class: Class0Test.java : Class1test.java
7 Parent of the class: Class1test.java : TestClass2.java
8 Parent of the class: TestClass2.java : LastExtendedTest.java
```

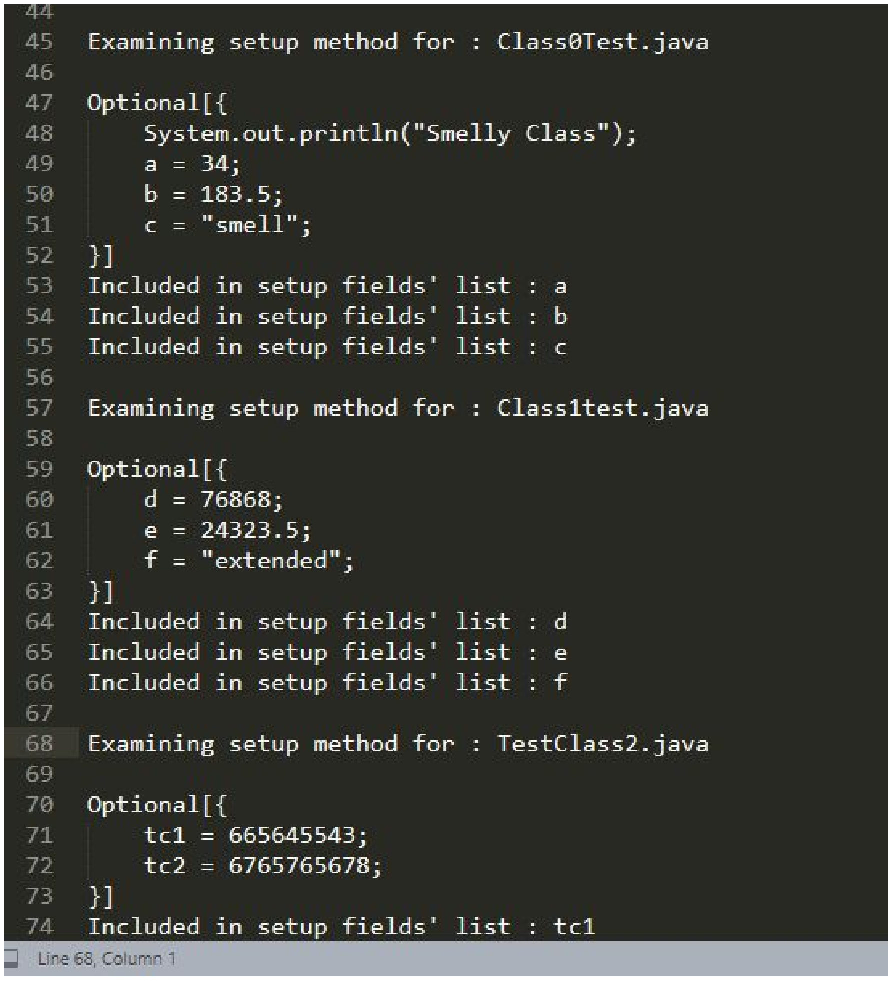
Figure 6: Output.txt



```
10
11  Setup method for Class: Class0Test.java
12
13  @Before
14  public void setUp() throws Exception {
15      System.out.println("Smelly Class");
16      a = 34;
17      b = 183.5;
18      c = "smell";
19  }
20
21  Setup method for Class: Class1test.java
22
23  @Before
24  public void setUp() throws Exception {
25      d = 76868;
26      e = 24323.5;
27      f = "extended";
28  }
29
30  Setup method for Class: TestClass2.java
31
32  @Before
33  public void setUp() throws Exception {
34      tc1 = 665645543;
35      tc2 = 6765765678;
36  }
37
38  Setup method for Class: LastExtendedTest.java
39
40  @Before
41  public void setUp() throws Exception {
42      last = "Not extended";
```

Figure 7: Output.txt

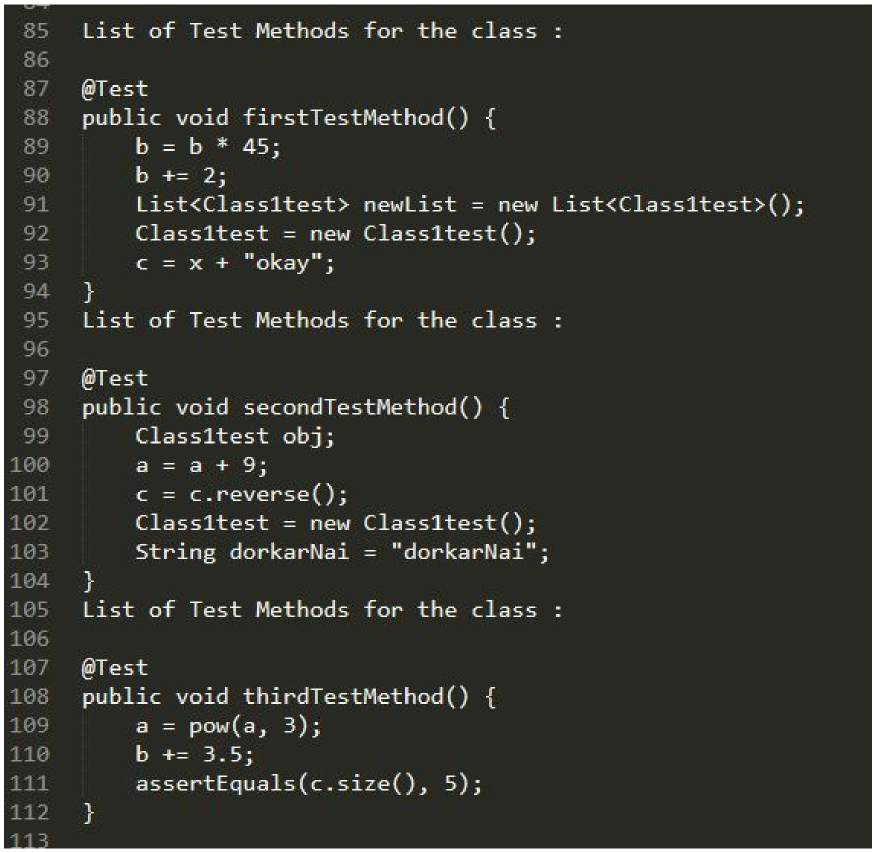
From line 11, the setup methods of Class0Test.java and the classes it has extended is printed. (Figure 7)



```
44
45  Examining setup method for : Class0Test.java
46
47  Optional[{
48      System.out.println("Smelly Class");
49      a = 34;
50      b = 183.5;
51      c = "smell";
52  }]
53  Included in setup fields' list : a
54  Included in setup fields' list : b
55  Included in setup fields' list : c
56
57  Examining setup method for : Class1test.java
58
59  Optional[{
60      d = 76868;
61      e = 24323.5;
62      f = "extended";
63  }]
64  Included in setup fields' list : d
65  Included in setup fields' list : e
66  Included in setup fields' list : f
67
68  Examining setup method for : TestClass2.java
69
70  Optional[{
71      tc1 = 665645543;
72      tc2 = 6765765678;
73  }]
74  Included in setup fields' list : tc1
```

Figure 8: Output.txt

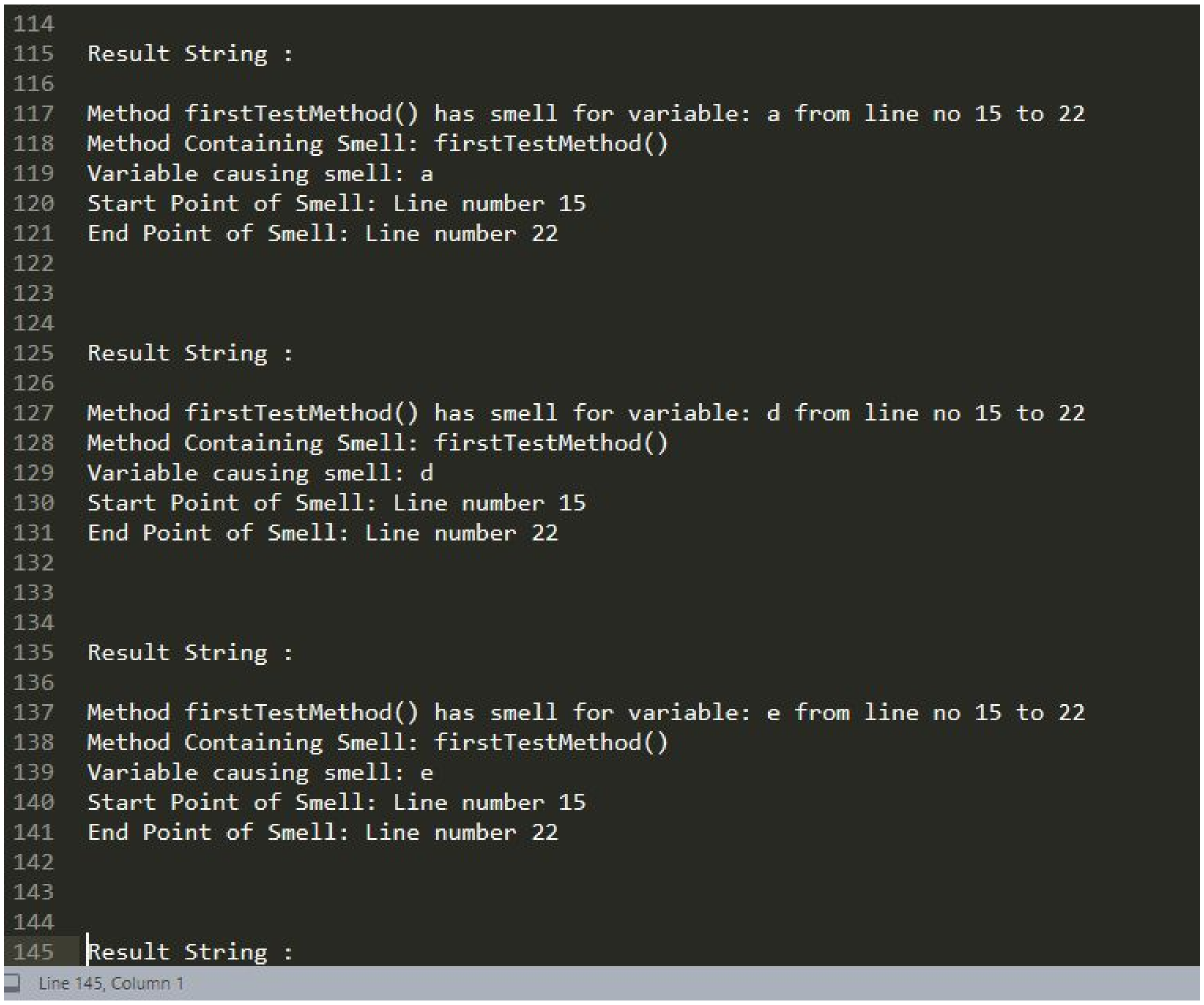
From line 45, the setup methods are examined and the setup fields are added in a list. (Figure 8)



```
 85  List of Test Methods for the class :
 86
 87  @Test
 88  public void firstTestMethod() {
 89      b = b * 45;
 90      b += 2;
 91      List<Class1test> newList = new List<Class1test>();
 92      Class1test = new Class1test();
 93      c = x + "okay";
 94  }
 95  List of Test Methods for the class :
 96
 97  @Test
 98  public void secondTestMethod() {
 99      Class1test obj;
100      a = a + 9;
101      c = c.reverse();
102      Class1test = new Class1test();
103      String dorkarNai = "dorkarNai";
104  }
105  List of Test Methods for the class :
106
107  @Test
108  public void thirdTestMethod() {
109      a = pow(a, 3);
110      b += 3.5;
111      assertEquals(c.size(), 5);
112  }
113
```

Figure 9: Output.txt

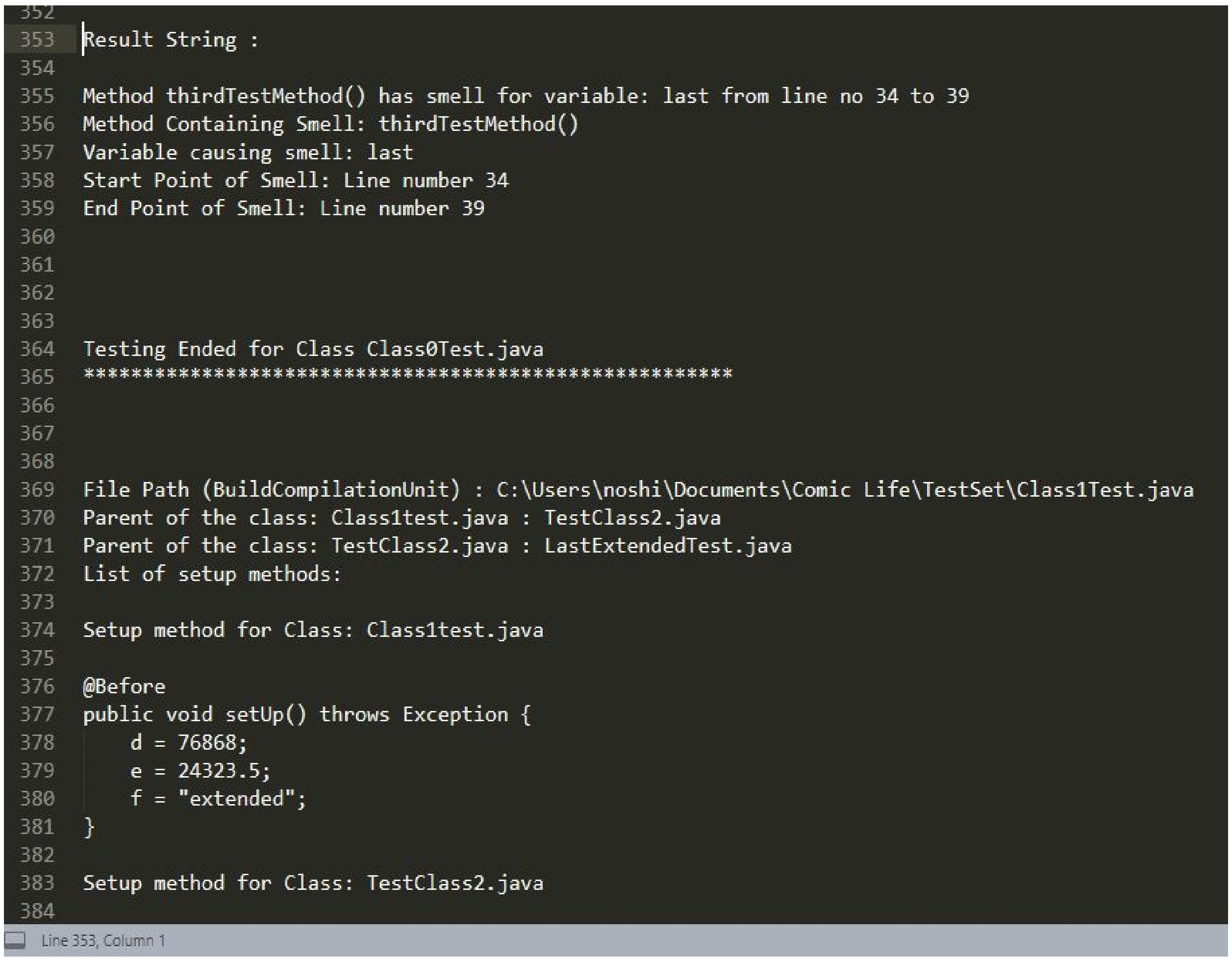
From line 85, the test methods to be examined for now are printed, (Test methods of class Class0Test.java). (Figure 9)



```

Result String :

Method firstTestMethod() has smell for variable: a from line no 15 to 22
Method Containing Smell: firstTestMethod()
Variable causing smell: a
Start Point of Smell: Line number 15
End Point of Smell: Line number 22



Result String :

Method firstTestMethod() has smell for variable: d from line no 15 to 22
Method Containing Smell: firstTestMethod()
Variable causing smell: d
Start Point of Smell: Line number 15
End Point of Smell: Line number 22



Result String :

Method firstTestMethod() has smell for variable: e from line no 15 to 22
Method Containing Smell: firstTestMethod()
Variable causing smell: e
Start Point of Smell: Line number 15
End Point of Smell: Line number 22



Result String :
```

Figure 10: Output.txt

The result string shows the method containing smell, the field causing the smell, starting point and ending point of that method. (Figure 10)



```
352
353 Result String :
354
355 Method thirdTestMethod() has smell for variable: last from line no 34 to 39
356 Method Containing Smell: thirdTestMethod()
357 Variable causing smell: last
358 Start Point of Smell: Line number 34
359 End Point of Smell: Line number 39
360
361
362
363
364 Testing Ended for Class Class0Test.java
365 *******************************************************
366
367
368
369 File Path (BuildCompilationUnit) : C:\Users\noshi\Documents\Comic Life\TestSet\Class1Test.java
370 Parent of the class: Class1test.java : TestClass2.java
371 Parent of the class: TestClass2.java : LastExtendedTest.java
372 List of setup methods:
373
374 Setup method for Class: Class1test.java
375
376 @Before
377 public void setUp() throws Exception {
378     d = 76868;
379     e = 24323.5;
380     f = "extended";
381 }
382
383 Setup method for Class: TestClass2.java
384
```

Figure 11: Output.txt

Line 364 shows that testing has ended for the class.

The path of the next test class of the chosen folder is then sent to parse.

4. Resources:

The code snippet for exploring the project directory is taken from Federico Tomassetti's Blog: https://tomassetti.me/getting-started-with-javaparser-analyzing-java-code-programmatically/