

PowerEnJoy

Code Inspection

A.Y 2016/2017

Francesco Tinarelli (matr:806146)

Marco Wenzel (matr:878021)

Versione 1.0

Introduction

Code inspection is the systematic examination (often known as peer re-

view) of computer source code. It is intended to find mistakes overlooked

during the initial development phase, with the aim of improving both the

overall quality of software and the developers' skills.

Document Structure

1. **Front page**: Include at least the project title, the version of the

document, your names and the release date.

2. **Table of content:** Include the table of content of your document.

3. **Classes that were assigned to the group**: <state the namespace

pattern and name of the classes that were assigned to you.>

4. **Functional role of assigned set of classes**: <elaborate on the func-

tional role you have identified for the class cluster that was assigned

to you, also, elaborate on how you managed to understand this role

and provide the necessary evidence, e.g., javadoc, diagrams, etc.>

5. **List of issues found by applying the checklist**: <report the class-

es/code fragments that do not fulfill some points in the check list.

Explain which point is not fulfilled and why.>

6. **Any other problem you have highlighted**: <list here all the parts

of code that you think create or may create a bug and explain why.>

Classes Assigned

* **Name**: EntityDataReader
  + **Location:** /apache-ofbiz-16.11.01/framework/entity/src /main/java/org/apache/ofbiz/entity/config/model
  + **Package:** org.apache.ofbiz.entity.config.model
  + **Modifier:** public-final
  + **Method:** EntityDataReader(String), EntityDataReader(Element), getName, getResourceList
* **Name**: SetOperation
  + **Location**: /apache-ofbiz-16.11.01/framework/minilang/

src/main/java/org/apache/ofbiz/minilang/method/envops

* + **Package:** org.apache.ofbiz.minilang.method.envops
  + **Modifier:** public final
  + **Method:** autoCorrect, SetOperation, exec, toString, <inner class> SetOperationFactory

**Functional Role to assigned class**

**EntityDataReader**

For understanding the functional role of this class, first we need to understand what are nodes and elements.

The node is the smallest datatype in the DOM, is an interface implemented by almost all object.

The element is a type of node, is an interface that extends the node interface, this type of node corresponding to structure in XML pages.

The EntityDataReader class have two constructors:

The first one permits through a string variable to insert a value on the variable: name; and provide an empty list of resource.

The second one permits through an Element to extract all children elements node that they have the same type of node (element node) and with the node name equals to “resource”. After the extraction insert for each element the respective resource object into the resource list.

For control the unexpected results this class has an exception if the EntityDataReader have already a name value or if the content of the parameter: element is empty.

So, this class should be use for capture the content and the data structure of a specific XML file.

**SetOperation**

This class takes an element and first correct the deprecated value with the autocorrect() method, then this class “opens” the attribute searching script or operation and puts them into the variables. This searching for the operation is done by using Scriptlet, flexibleMapAccessor and flexibleStringExpander.

First the class searches the Script form the “from-attribute”, second uses the flexibleMapAccessor on the same attribute ,third search esfrom the “value” attribute and finally uses the default attribute.

The method exec executes the operation that are in the variables. This method return, true if script execution should continue, or false if script execution should stop. For the control of the execution this class use the exception.

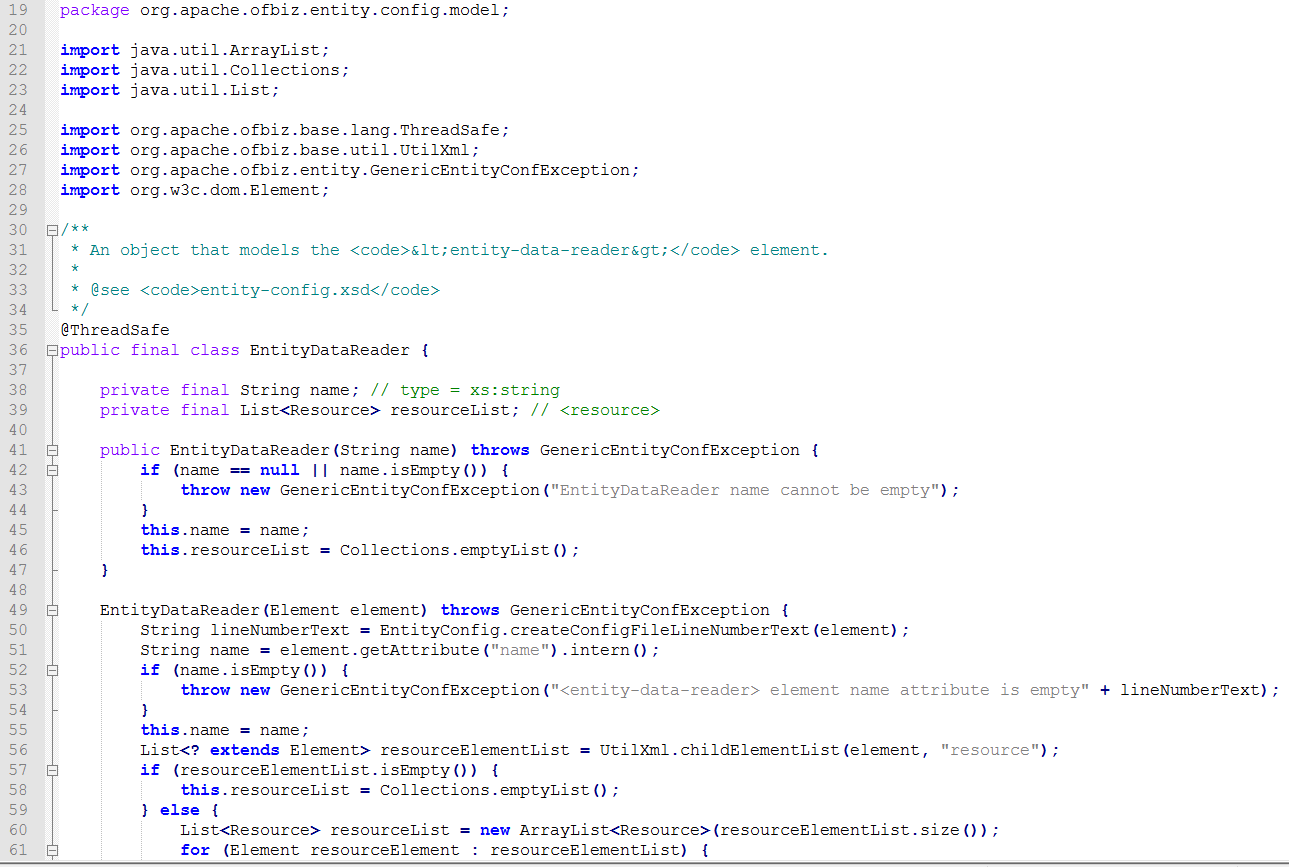
An object SetOperation can be created by the object factory that is an override of the method that is in the MethodContext superclass.

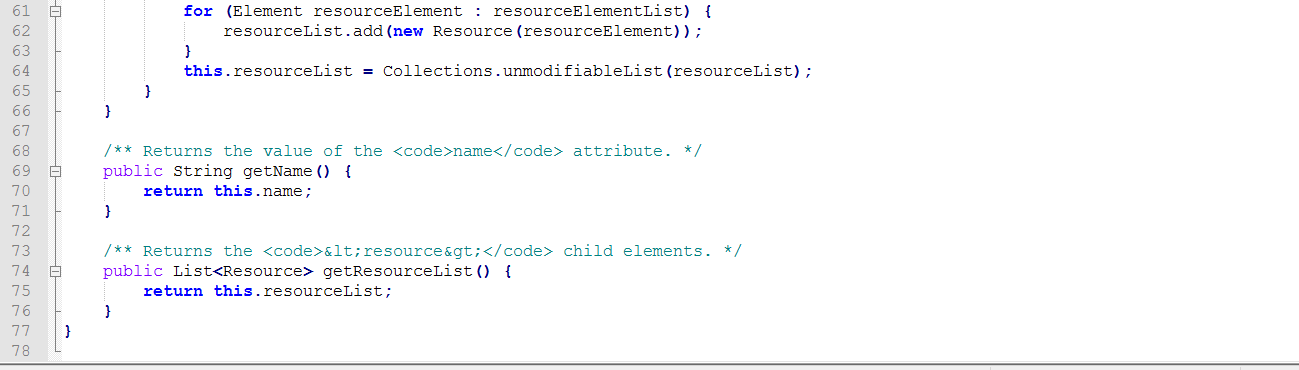
**Analysis Method:**

For analyzing this class, we start from the analysis of the attributes and methods of the modifier, then of the double constructor and of the external method calls.

To understand the intrinsic meaning we used the OfBiz documentation but this work was hard because in this class the Javadoc is missing.

**Code:**





Issues:

1. createConfigFileLineNumberText(element) this method doesn’t have a meaningful name because returns a string that indicates the number of the starting line a more meaningful name can be startingLineNumber(element)

13)Line 53 (110c) Line 56 (92c) Line 60(83c)

18)There are no comments about what the code is doing

19)There are no comments for explaining the reasoning behind the code

23)There is no Javadoc about this class

27)There is double constructor that controls two variables with the same name; the variable “name” is defined two times but the two variables are not the same and are well controlled (this.name=name)

36) lineNumberText variable is not used if the variable “name” is empty

56)that FOR type is correct but we don’t know the initial value.

**SetOperation**

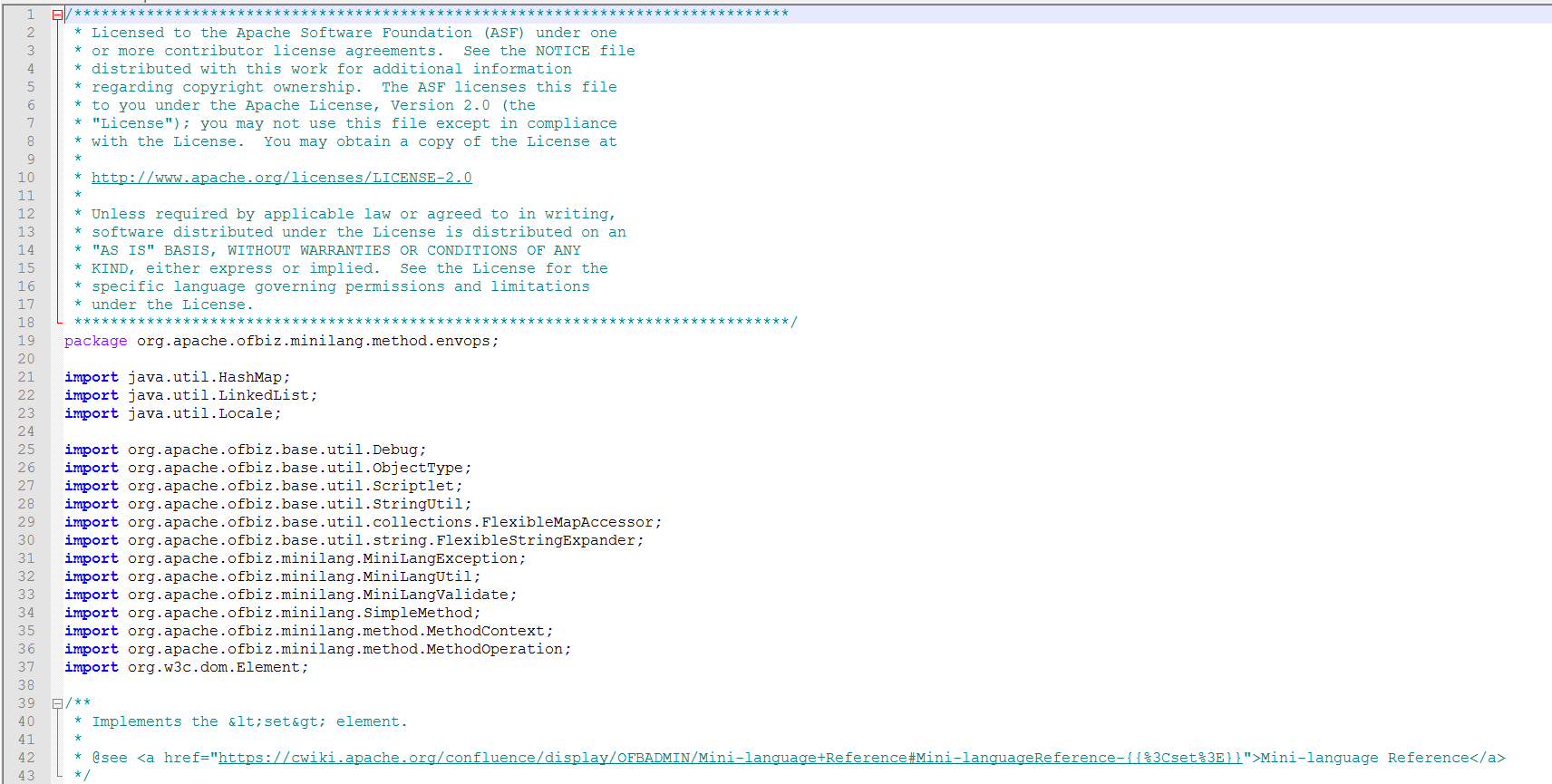
This class takes an element and first correct the deprecated value with the autocorrect () method, then this class “opens” the attribute searching script or operation and puts them into the variables. This searching for the operation is done by using Scriptlet, flexibleMapAccessor and flexibleStringExpander.

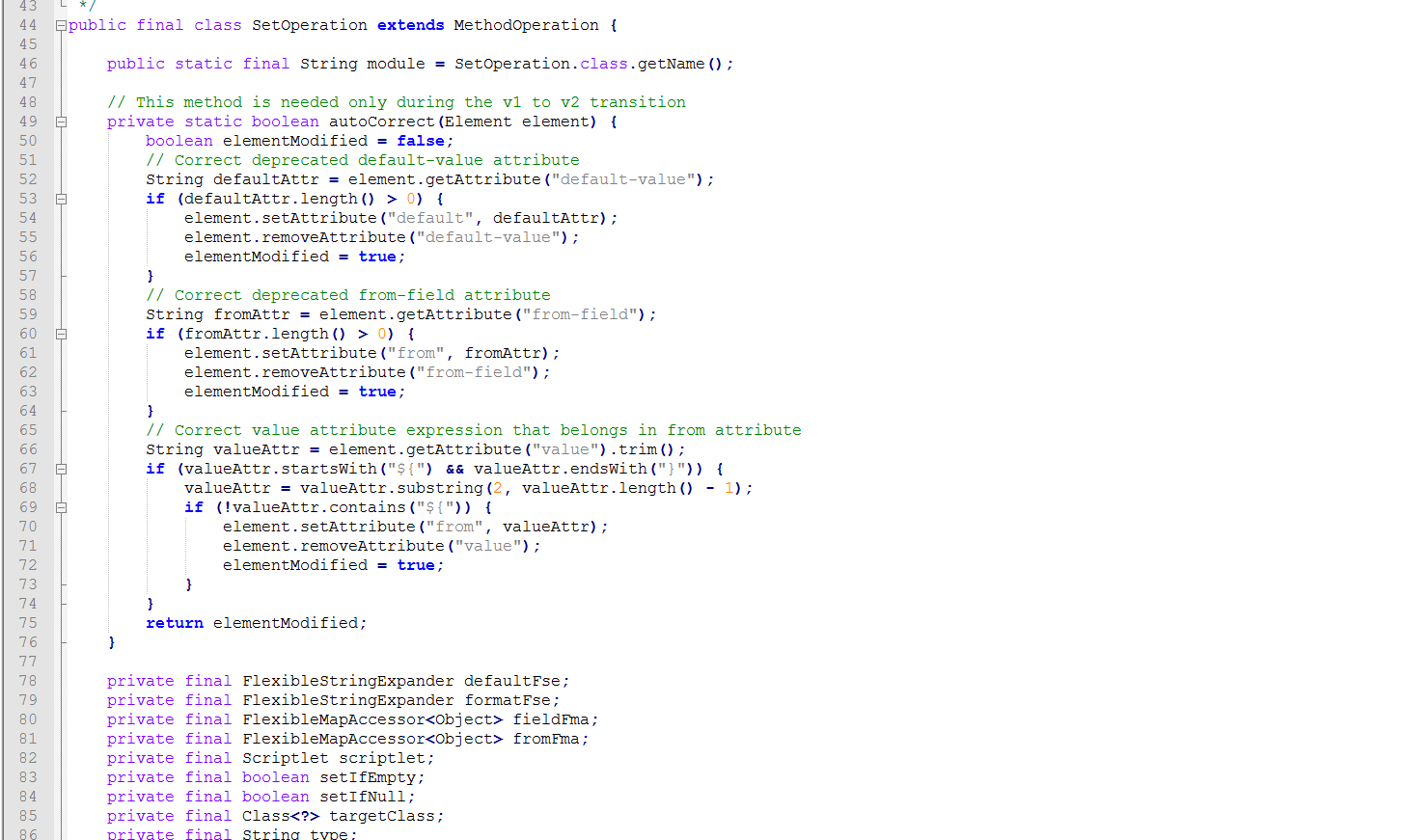
First the class searches the Script form the “from-attribute”, second uses the flexibleMapAccessor on the same attribute, third search esfrom the “value” attribute and finally uses the default attribute.

The method exec executes the operation that are in the variables. This method returns, true if script execution should continue, or false if script execution should stop. For the control of the execution this class use the exception.

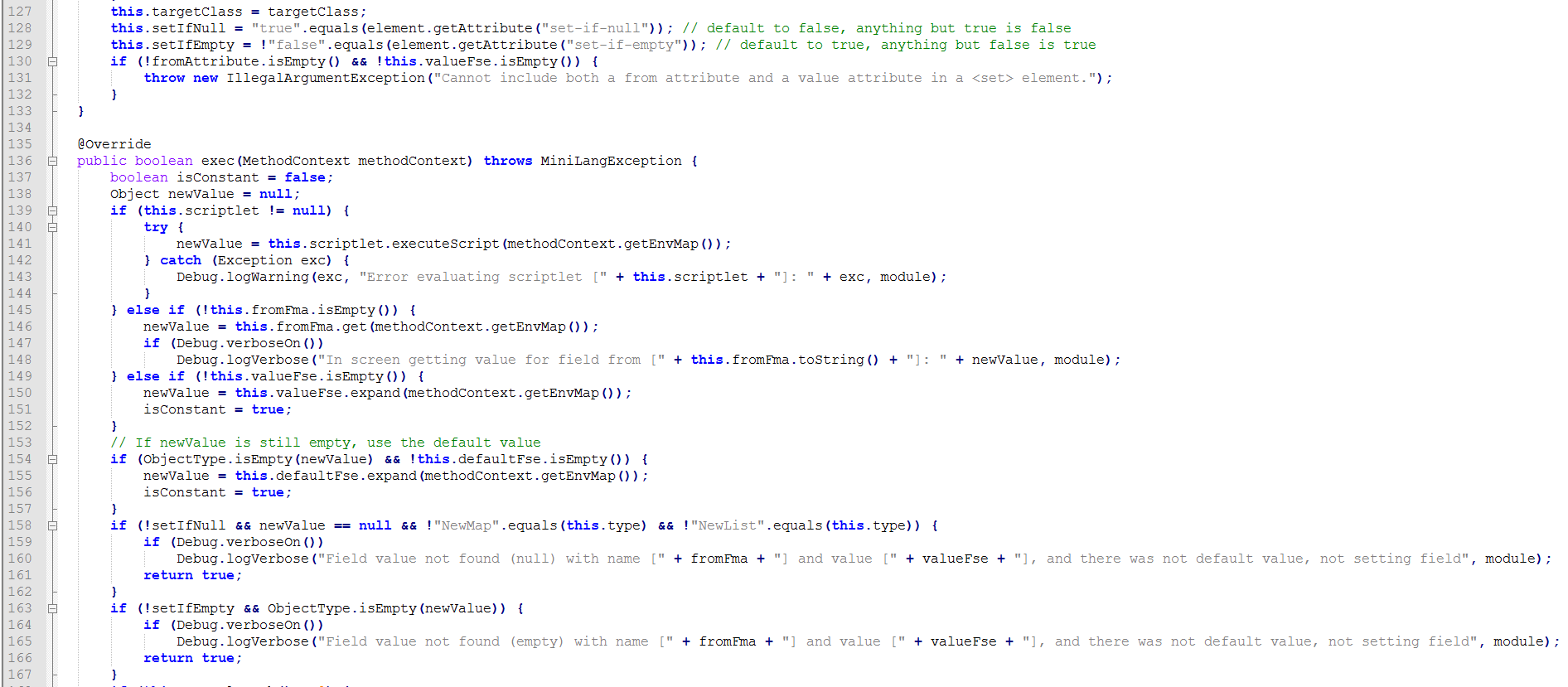
An object SetOperation can be created by the object factory that is an override of the method that is in the MethodContext superclass.

**Code:**

****

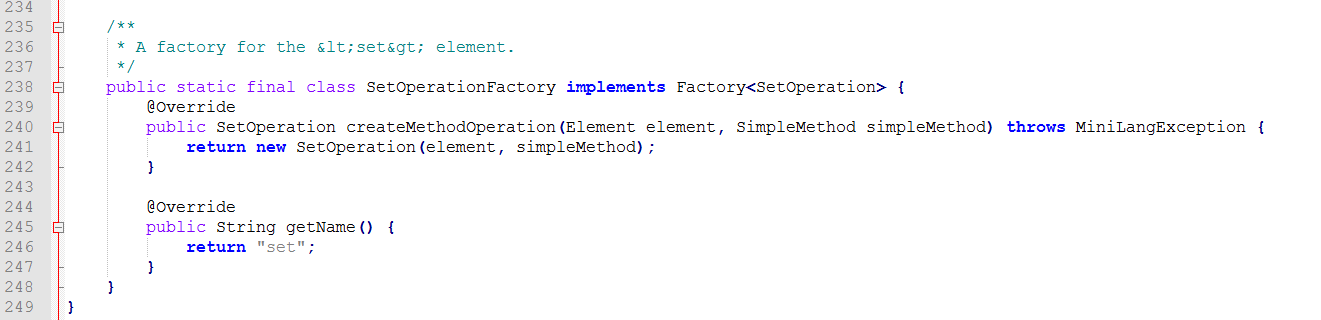












**Issues:**

1) SetOperation it’s a meaningless name, it doesn’t explain what the class does.

11) line 147-159-164-197 one statement without graph braces

13) Line 94, line 128, line 129, line 143, line 148, line 160, line 165, line 187, line 190

14)Line 94(172c) Line 160(166c) Line 165(168c) Line 187(123c) Line 190(182c)

18) The comment at the beginning of the class is useless, the internet page that the URL of this document see is moved to another page.

19) All the comment out of code are without the date.

23) there are no Javadoc about this class

25) AutoCorrect method came first than private final variables.

33) Line 107: variable stringAttribute;

40) line 42, line 158, line 180.