Software Engineering II (2020)

Code Smells Detection

1. PEDAGOGICAL OBJECTIVES

This work aims at the acquisition of skills in the area of Software quality, namely, quality in code production. According to SWEBOK 3.0 (Software Engineering Body of Knowledge), software quality is an area of knowledge in software engineering that can refer to: "the desired features of software products, the extent to which a particular software product possesses those features and the processes, tools and techniques that are used to ensure those features".

Although there are several techniques and many detection tools, the subjectivity of code smells is still very large, and there are big differences depending on the approach used in their detection.

The corresponding pedagogical objectives are the following:

- Acquisition of knowledge about various types of code smells and their influence on the quality of the code. Among the 22 code smells in Martin Fowler's initial catalogue, we will use the following: Long Method, God Class, and Feature Envy;
- Know how to identify the 3 types of code smells referred to in the previous point and understand the various detection techniques used;
- Know how to identify in the code the code smells, as well as the gravity of each one of them, based on the quality metrics of the code;
- The domain of an open-source code smells detection tool.

2. INSTRUMENTS

The ECLIPSE tool is an IDE for Java development but supports several other programming languages. It has a set of features that help the programmer to produce higher quality code like refactoring operations.

JDeodorant is a plug-in for Eclipse that identifies problems in the software, known as code smells, and suggests the refactoring operation to solve them. It is developed by a team of researchers from Concordia University - Canada, University of Macedonia - Greece and Alberta University - Canada.

3. EXECUTION OF WORK

3.1. Importing the JAVA project in which you will detect code smells - JASML 0.10

For this work will be used the JAVA application JASML 0.10, which will be object of all the operations it will perform.

JASML 0.10 project import:

a) Download the Jasml 0.10 project, from the address:

https://sourceforge.net/projects/jasml/files/

b) Import the project in eclipse: Menu "file -> Import", Choose "**General" -> "Projects from Folder or Archive**", "next";

c) Click on the "Archive…" button and select the file you downloaded;

d) Select the checkbox of the Java project and finish importing the file.

3.2. Detection of code smells using JDeodorant

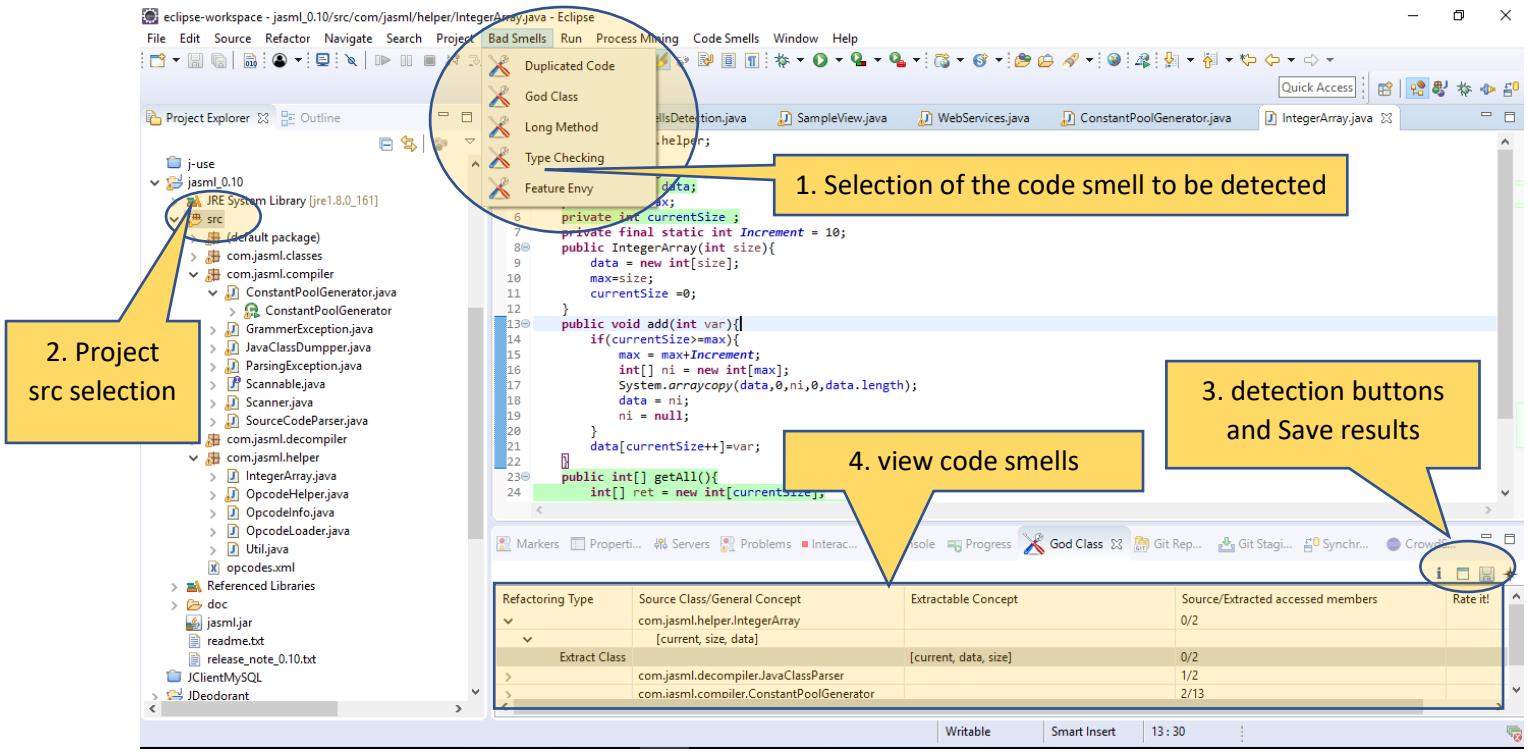
Install JDeodorant (if you have not installed it yet).

a) For code smells detection, select the code smell to detect from the ***bad smells*** menu, select the ***src*** folder in the Jasml 0.10 project and use the smell code view buttons to do the detection (see figure 1);

b) You must save the detection result to the ***save results*** button;

c) Double click on the table row of the smell code view to automatically view the code with the smell code. Analyze the code to ensure it clearly identifies the smell code. You can consult the JDeodorant manual on its page: https://users.encs.concordia.ca/~nikolaos/jdeodorant/index.php?option=com_content&view=article&id=45;

Figure 1 - JDeodorant



4. EDUCATIONAL GOALS

In this work it is our goal to classify code smells, that is, to indicate if a class or method is or not code smells.

5. GENERAL DESCRIPTION

In this work we will perform the detection of code smells in an open-source JAVA project, using JDeodorant or manually through the code metrics. It is intended validate a set of code smells.

6. EXECUTION OF WORK

6.1. Code Smell manual classification

Import the original JAVA JASML 0.10 application. The application cannot have changes to the original code.

For each of the 3 code smells (Long Method, God Class, and Feature Envy):

a) Import the Jasml 0.10 application. The application cannot have changes to the code;
b) Do the detection of code smells through Jdeodorant, that is, select the code smell to detect from the **bad smells** menu, select **src** folder for God Class and Feature Envy **or package for the Long Method**, and use the view buttons of the code smell to do the detection;
c) Save the detection result to the save results button. Save the file with the application name plus the code smell and the word "classification", example for the code smell God class: **jasml_0.10_GodClass_classification**;
d) Classify the code smells detected by JDeodorant, saying whether or not you agree with the detection of JDeodorant, to do so:
    I. Open the file you just saved with the detection result;
    II. Analyze the detected smell codes and write in the file, in the respective smell code, the word "**YES**" - if you agree that it is a smell code or "**No**" - if you do not agree that it is a smell code.
    III. Classify all the methods or classes (according to the code smell) existing in the chosen package;

    If you wish, you can perform code smells detection by directly analyzing the code and, based on its metrics, validate if there are code smells or not. Register your classification in the text file, writing the name of the class or method and its classification, according to the previous point.
e) Don't forget to save the results file;
f) Go back to point a) and repeat the process for the 3 code smells.