DP2 2023-2024

Lint report D03

# Acme Software Factory

Repository: https://github.com/rafcasceb/Acme-SF-D03

## Student #1:

- Castillo Cebolla, Rafael rafcasceb@alum.us.es

## Other members:

- Flores de Francisco, Daniel danflode@alum.us.es
- Heras Pérez, Raúl rauherper@alum.us.es
- Mellado Díaz, Luis luimeldia@alum.us.es
- Vento Conesa, Adriana adrvencon@alum.us.es

GROUP C1.049

Version 1.0

22-04-24

## Content Table

Executive summary ........................................ 3
Revision Table ........................................ 4
Introduction ........................................ 5
Contents ........................................ 6
Override the "equals" method in this class.: ........................................ 6
Use concise character class syntax '\\d' instead of '[0-9]'. ........................................ 6
Replace this assert with a proper check. ........................................ 6
Rename this package name to match the regular expression '^[a-z_]+(\.[a-z_][a-z0-9_]*)*$'. ........................................ 6
Replace the type specification in this constructor call with the diamond operator ("<>"). .. 6
Extract this nested code block into a method. ........................................ 6
Define a constant instead of duplicating this literal X Y times. ........................................ 7
Conclusions ........................................ 8
Bibliography ........................................ 9

## Executive summary

This report will offer a listing with the bad smells reported by Sonar's Lint regarding your project and a justification on why they are innocuous.

We will follow a precise but accessible approach aiming to promote comprehension and assure a good final product.

## Revision Table

| Date | Version | Description of the changes | Sprint |
|---|---|---|---|
| 22/04/2024 | 1.0 | • All | 3 |

## Introduction

All files I have created as Student 1 for my individual deliveries will be analyzed using the Sonar’s Lint plug-in for Eclipse.

We have analyzed all the contents of the Java packages “entities.projects”, “features.manager”, “features.any.project” and “features.authenticated.manager” and, besides that, the Manager role and the ManagerDashboard form.

The populator files Project, UserStory, ProjectUserStory and Manager have also been analyzed, as well as the views files (forms.jsp, list.jsp and i18n files) under the folders “views/manager”, “views/any/project” and “views/authenticated/manager”.

## Contents

### Override the "equals" method in this class.:

A common bad smell for most of our Java entities. There is no need to correct anything since this is not used in our implementation and we have commented it in class.

### Use concise character class syntax '\\d' instead of '[0-9]'.

This bad smell refers to the pattern of the code attribute of the Project entity. There is no need to correct anything since the implemented code is a perfectly valid approach for ranges of Latin numbers in regular expressions. It is arguably more comprehensible than the recommended approach.

### Replace this assert with a proper check.

This bad smell refers to the code *assert object != null;* . There is no need to correct anything since this is the recommended implementation by the professors and the framework.

### Rename this package name to match the regular expression '^[a-z_]+(\.[a-z_][a-z0-9_]*)*$'.

Package names follow the framework standard, the sample project AcmeJobs and what was taught by the professors. Hence, there is no need to correct anything.

### Replace the type specification in this constructor call with the diamond operator ("<>").

This bad smell refers to the type inference feature of Java, with which types don't always need to be introduced in collections since they can be inferred. In this case, we have removed the type declaration in the mentioned collection.

### Extract this nested code block into a method.

This bad smell which appeared three times, made reference to aesthetical code blocks, which don't have a particular functionality in the Java language and are only used for readability. Since they were a common practice in the sample project AcmeJobs and are a common practice in the industry, there is no need to correct anything.

## Define a constant instead of duplicating this literal X Y times.

This code smell refers to a literal X is used multiple, Y, times along a class. In our case, I don’t consider three and four times enough to declare a variable, especially when these literals are used together with others that are only used once, and using variables for some and strings for others would harm the visual coherence and maintainability.

## Conclusions

Seven different bad smells have been detected by Sonar’s Lint. However, six of them were minor details caused by following the model taught in class, which justifies the current implementation. The remaining one was another readability good practice change which has been solved after its detection.

## Bibliography

Intentionally blank.