DP2 2023-2024

Lint Report

Acme Software Factory

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

Members:

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

23-04-24

Content Table

Abstract

This report offers a detailed examination of the lint analysis conducted for project deliverable D03 pertaining the group requirements of the project. It presents a thorough evaluation of "bad smells" reported by Sonar Lint, justifying their innocuous nature if appropriate.

Revision Table

Date	Version	Description of the changes	Deliverable
23/04/2024	V1	• Abstract. • Introduction. • Contents: bad smells. • Conclusions.	3

Introduction

In this phase of delivery, the focus was directed towards the implementation of different features, creating multiple bad smells in the process of doing so. The subsequent analysis in this document pertains solely to the lint evaluation of the collective group work. It was done through the analysis of all files modified during the duration of this delivery for both obligatory and supplementary requirements. The lint analysis for the individual students can be consulted in their respective folders.

The document structure is outlined as follows: an introductory section that contextualizes the lint analysis, followed by a comprehensive enumeration of bad smells detected by Sonar Lint within the group work. Finally, a succinct conclusion will encapsulate the report, summarizing key findings.

Contents

Bad Smells

The following bad smells appeared several times in the individual files analyzed for this report. For the sake of simplicity, they have been consolidated into a singular enumeration.

1. Override the "equals" method in this class.
   As we follow the Acme Framework recommendations given in the theory lectures, this bad smell can be considered innocuous.

2. Use concise character class syntax '\\d'.
   This bad smell is innocuous, as it performs the same functionalities as the alternative option proposed and does not hinder the maintainability of the code. The change is only more concise.

3. Rename this package name to match the regular expression '^[a-z_]+(\.[a-z_][a-z0-9_]*)*$'.
   As we follow the structure recommendations given by the Acme Jobs example project, this bad smell can be considered innocuous.

4. Replace this assert with a proper check.
   As we follow the Acme Framework recommendations given in the theory lectures, this bad smell can be considered innocuous.

5. Define a constant instead of duplicating this literal “X” times.
   This bad smell is innocuous by nature, as the duplication of literals does not directly affect the implementation of the features. It could potentially pose a refactoring problem, but as the recommendations and examples given by the subject also duplicate literals, we can safely ignore this.

6. Complete the task associated to this TODO comment.
   This bad smell is innocuous, although it is important to keep track of missing TODO comments in the code to avoid confusion in the development of the features.

7. Catch Exception instead of Throwable.
   Exception extends Throwable, hence it’s equivalent. Despite this, again, Acme Jobs follows this structure, so we can consider it innocuous.

8. Declare “variable” on a separate line.
   This bad smell is inherently innocuous. The only reason why this is a bad smell is because it is considered “hard to read” which is arguable.

9. Refactor this repetition that can lead to a stack overflow for large inputs.
   Although this can be considered a bad smell, it is necessary to implement repetition in this regex pattern.

Conclusions

In conclusion, this lint report document allows us to examine the obstacles faced in analyzing and implementing diverse requirements, as well as a revision of our code to avoid future maintainability problems.

Bibliography

Intentionally blank.