DP2 2022
Lint Report

Acme Recipes

Enlace al repositorio

Miembros:

- Ballestero López, Jesús (jesballop@alum.us.es)
- Botello Romero, Francisco (frabotrom@alum.us.es)
- León Valderrama, Juan José (jualeoval@alum.us.es)
- Moreno Pérez, Juan Carlos (juamorper2@alum.us.es)
- Serrano Mena, Antonio Roberto (antsermen@alum.us.es)
- Suárez García, Antonio José (antsuagar@alum.us.es)

Tutor: Manuel Jesús Jiménez Navarro

GRUPO S02

Versión 1.0

26/08/2022

Índice

Resumen ejecutivo

Este documento detalla los resultados obtenidos tras el análisis del código con la herramienta SonarLint. Para ello, se muestran capturas de los resultados y se especifica la solución que hemos proporcionado para eliminar por completo los malos olores de nuestro código.

Hemos concluido que el trabajo realizado para deshacernos de los malos olores no nos ha supuesto una gran carga de trabajo ya que tenían fácil arreglo, y los que hemos dejado son un par de detalles menores que solamente dificultarían el trabajo y la comprensión de la aplicación o directamente van en contra del funcionamiento que buscamos implementar. De todas formas, Eclipse cuenta con herramientas que facilitan la corrección de este tipo de detalles (así como replace all).

Consideramos que, aunque esta tarea no nos ha supuesto mucho tiempo ni esfuerzo, habría sido una buena práctica el haber llevado un seguimiento periódico de los malos olores del código, pues podría haberse dado el caso de vernos obligados a corregir varias clases en una ventana de tiempo muy reducida.

Historial de versiones

Fecha	Versión	Descripción	Sprint
23/08/2022	0.1	- Creación del documento	4
25/08/2022	0.2	- Introducción - Resumen ejecutivo - Contenidos - Conclusión	4
26/08/2022	1	- Actualización de contenidos - Cumplimentación del documento	4

Introducción

En este documento se reflejan los resultados obtenidos tras hacer un análisis del código con la herramienta SonarLint. Se van a mostrar capturas de los malos olores detectados por la herramienta, así como una explicación de los cambios que hemos realizado para solucionarlos. Para ello, hemos organizado este documento en distintas secciones descritas a continuación:

Resultados obtenidos. En esta sección se muestran capturas del análisis de nuestro código proporcionado por SonarLint.

Cambios realizados. Se explican los cambios realizados tras el análisis del código, así como una captura final para mostrar la solución a los malos olores.

Conclusiones. Se reflejan las conclusiones que hemos obtenido del análisis del código con SonarLint y se hace un balance de la metodología que hemos seguido para la detección y eliminación de malos olores.

Bibliografía. Esta sección está vacía ya que no hemos recurrido a ningún tipo de bibliografía para la redacción de este documento.

Finalmente, el documento posee la siguiente estructura: portada, índice, resumen ejecutivo, introducción, contenidos (divididos en las secciones Resultados obtenidos y Cambios realizados), conclusiones y bibliografía.

Resultados obtenidos

Al ejecutar *SonarLint* obtuvimos 11 malos olores.

11 items

Resource	Date	Description
AdministratorDashboardShowService.java		Convert this Map to an EnumMap.
AdministratorSystemConfigurationUpdateService.java		Use "Arrays.copyOf", "Arrays.asList", "Collections.addAll" or "System.arraycopy" instead.
ChefThingCreateService.java	few seconds ago	Call "Optional#isPresent()" before accessing the value.
SystemConfiguration.java	3 days ago	Simplify this regular expression to reduce its complexity from 21 to the 20 allowed. [+19 locations]
SystemConfiguration.java	3 days ago	Simplify this regular expression to reduce its complexity from 21 to the 20 allowed. [+19 locations]
Unit.java	few seconds ago	Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$'.
Unit.java	few seconds ago	Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$'.
Unit.java	few seconds ago	Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$'.
Unit.java	few seconds ago	Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$'.
Unit.java	few seconds ago	Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$'.
Unit.java	few seconds ago	Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$'.

Cambios realizados

Primero, se solucionaron los malos olores que ya se pasaron por alto en el anterior entregable relacionados con el fichero *Unit.java*. El error era sencillo de solucionar: los datos del *Enum* estaban escritos en minúscula en lugar de en mayúscula. Una vez corregido esto se revisaron todos los ficheros *.csv*, tanto los de datos como los de test, para realizar el mismo cambio en los ficheros que aparecieran las unidades. Luego se solucionó el único bug del proyecto, en el cual se rodeó de un bloque try/catch la declaración del valor de la variable *Chef* del fichero *ChefThingCreateService.java.*

Los malos olores recogidos en la siguiente captura de pantalla son los que hemos decidido dejar. El primero es un cambio completamente innecesario el cual hemos decidido de ignorar debido a la cercanía de la fecha de entrega y a que no afecta en lo más mínimo a la funcionalidad. El segundo es algo que complicaría más el código, ya que *SonarLint* propone hacer una copia de una lista, a pesar de lo que simplemente se hace en esa línea es añadir un elemento a una lista. Finalmente los dos últimos malos olores debíamos ignorarlos, a pesar de que *SonarLint* los marque en rojo. Corresponden a las expresiones regulares de los términos de spam tanto en inglés como en español. *SonarLint* nos advierte de que la complejidad es demasiado alta, pero debe de ser así para cubrir los distintos casos en los que el *SpamDetector* detectaría spam, por ejemplo: sexo&sexo.

Resource	Date	Description
AdministratorDashboardShowService.java		Convert this Map to an EnumMap.
AdministratorSystemConfigurationUpdateService.java		Use "Arrays.copyOf", "Arrays.asList", "Collections.addAll" or "System.arraycopy" instead.
SystemConfiguration.java	1 day ago	Simplify this regular expression to reduce its complexity from 21 to the 20 allowed. [+19 locations]
SystemConfiguration.java	1 day ago	Simplify this regular expression to reduce its complexity from 21 to the 20 allowed. [+19 locations]

Conclusiones

Tras haber analizado el código con *SonarLint* hemos concluido que debimos solucionar los errores relacionados al fichero *Unit.java* en el anterior entregable para así no tener que modificar tantos archivos *.csv.*

Nuestra metodología para la detección de malos olores ha sido ejecutar *SonarLint* sobre la rama *develop* una vez que todas las features ya habían sido implementadas. Debimos haber realizado un seguimiento más constante de los malos olores del proyecto, pero gracias al uso de buenas técnicas por parte de los integrantes, los malos olores eran pocos y de fácil solución.

Bibliografía

No procede.