

LINT REPORT

Noviembre D03 – Diseño y Pruebas II

09/11/2022

Grupo D03

MARTÍNEZ SUÁREZ, DANIEL JESÚS

danmarsua1@alum.us.es

<https://github.com/danmarsua1/Acme-Courses>

ÍNDICE

RESUMEN	1
HISTORIAL DE VERSIONES	2
INTRODUCCIÓN	3
CONTENIDO	4
CONCLUSIONES.....	7
BIBLIOGRAFIA	8

RESUMEN

El objetivo del presente reporte, es mostrar como un proyecto puede “ensuciarse” y ser costoso de mantener, a nivel de código, debido a unas malas prácticas durante la implementación de este.

Se trata de informar sobre los problemas que la herramienta Lint, plugin de Eclipse perteneciente al campo de revisión de calidad del código encargado de revisar malas prácticas (bad smells) dentro de un proyecto y/o artefacto. Posteriormente, estos “bad smells” serán informados, tratados y finalmente corregidos, a lo largo del resto de entregables.

HISTORIAL DE VERSIONES

FECHA	VERSIÓN	DESCRIPCIÓN DE CAMBIOS
09/11/22	V1.0	Redacción del informe al completo

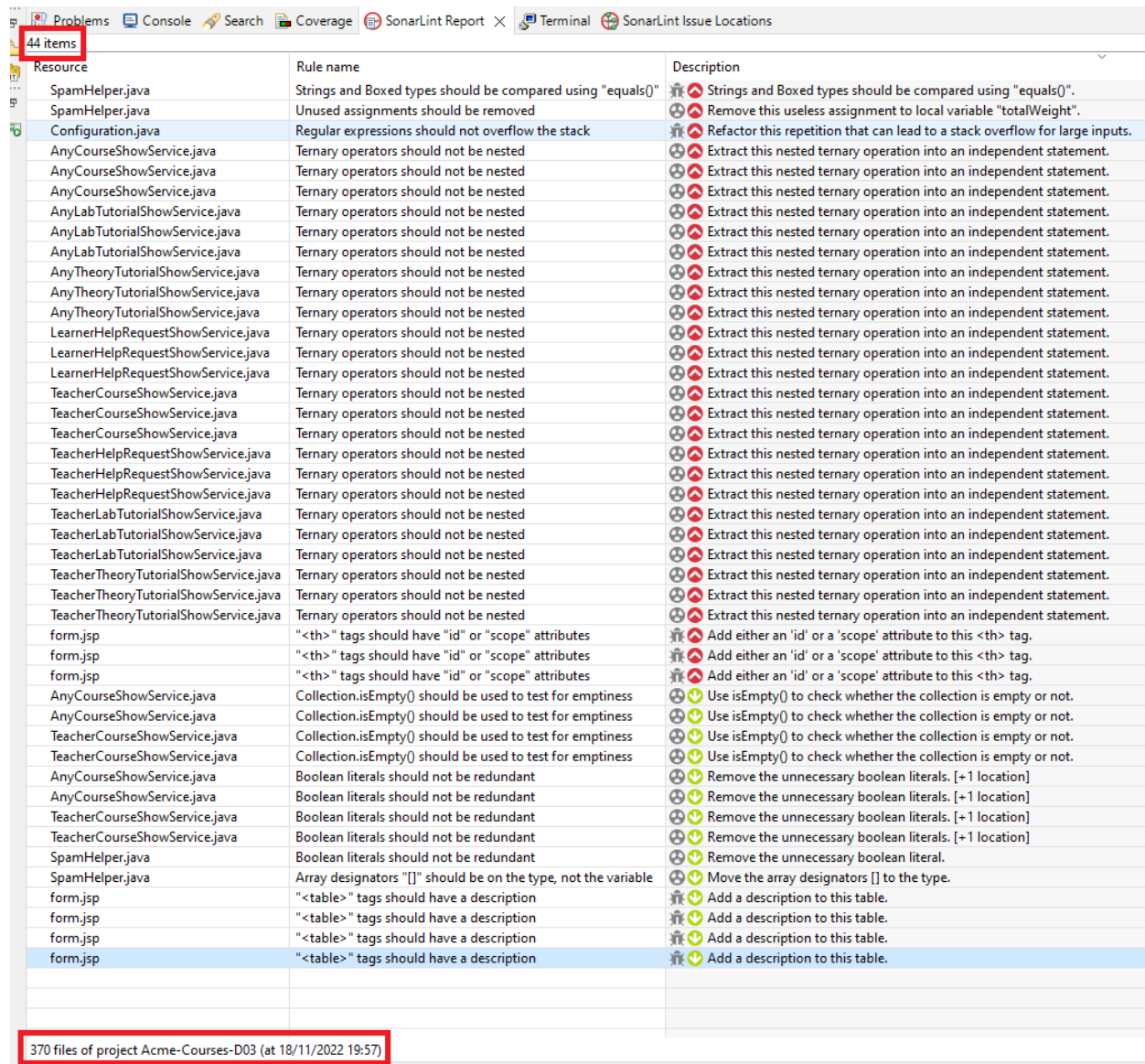
INTRODUCCIÓN

Este documento trata de concienciar, dentro del desarrollo de un proyecto, sobre una correcta implementación del código fuente del mismo, mediante buenas prácticas, y posterior revisión para un desarrollo más seguro, correcto y eficaz.

En este informe, se presentan capturas junto con algunos comentarios, sobre los resultados que proporciona la herramienta Lint de calidad de código antes descrita en el resumen ejecutivo.

CONTENIDO

Se procede a la presentación de los resultados de la herramienta Lint sobre el proyecto perteneciente al entregable D03 de la presente asignatura Diseño y Pruebas II.



Resource	Rule name	Description
SpamHelper.java	Strings and Boxed types should be compared using "equals()"	Strings and Boxed types should be compared using "equals()".
SpamHelper.java	Unused assignments should be removed	Remove this useless assignment to local variable "totalWeight".
Configuration.java	Regular expressions should not overflow the stack	Refactor this repetition that can lead to a stack overflow for large inputs.
AnyCourseShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
AnyCourseShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
AnyCourseShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
AnyLabTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
AnyLabTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
AnyLabTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
AnyTheoryTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
AnyTheoryTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
AnyTheoryTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
LearnerHelpRequestShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
LearnerHelpRequestShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
LearnerHelpRequestShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherCourseShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherCourseShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherCourseShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherHelpRequestShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherHelpRequestShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherHelpRequestShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherHelpRequestShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherLabTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherLabTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherLabTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherTheoryTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherTheoryTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
TeacherTheoryTutorialShowService.java	Ternary operators should not be nested	Extract this nested ternary operation into an independent statement.
form.jsp	"<th>" tags should have "id" or "scope" attributes	Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp	"<th>" tags should have "id" or "scope" attributes	Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp	"<th>" tags should have "id" or "scope" attributes	Add either an 'id' or a 'scope' attribute to this <th> tag.
AnyCourseShowService.java	Collection.isEmpty() should be used to test for emptiness	Use isEmpty() to check whether the collection is empty or not.
AnyCourseShowService.java	Collection.isEmpty() should be used to test for emptiness	Use isEmpty() to check whether the collection is empty or not.
TeacherCourseShowService.java	Collection.isEmpty() should be used to test for emptiness	Use isEmpty() to check whether the collection is empty or not.
TeacherCourseShowService.java	Collection.isEmpty() should be used to test for emptiness	Use isEmpty() to check whether the collection is empty or not.
AnyCourseShowService.java	Boolean literals should not be redundant	Remove the unnecessary boolean literals. [+ 1 location]
AnyCourseShowService.java	Boolean literals should not be redundant	Remove the unnecessary boolean literals. [+ 1 location]
TeacherCourseShowService.java	Boolean literals should not be redundant	Remove the unnecessary boolean literals. [+ 1 location]
TeacherCourseShowService.java	Boolean literals should not be redundant	Remove the unnecessary boolean literals. [+ 1 location]
SpamHelper.java	Boolean literals should not be redundant	Remove the unnecessary boolean literal.
SpamHelper.java	Array designators "[]" should be on the type, not the variable	Move the array designators [] to the type.
form.jsp	"<table>" tags should have a description	Add a description to this table.
form.jsp	"<table>" tags should have a description	Add a description to this table.
form.jsp	"<table>" tags should have a description	Add a description to this table.
form.jsp	"<table>" tags should have a description	Add a description to this table.

370 files of project Acme-Courses-D03 (at 18/11/2022 19:57)

Así pues, tras analizar por completo el proyecto perteneciente a la entrega D03, el reporte arrojado por la herramienta Lint presenta un total de **370** ficheros analizados, de los cuales se han detectado **44 "bad smells"** en distintas partes del código.

Adicionalmente, la herramienta también cuenta con un sistema de evaluación de impacto que, el código fuente desarrollado, puede llegar a tener dentro del proyecto, ya sea en seguridad, en rendimiento, o simplemente afectando la legibilidad y/o correcta comprensión del mismo.

Los distintos niveles de impacto que se pueden apreciar en el anterior reporte son solo 2, denotados con dos colores distintos:

- Rojo: Severidad alta.
- Verde: Severidad baja.

A continuación, se procede a desglosar las categorías en las que se podrían contener los errores reportados:

BAD SMELL	DESCRIPCIÓN	SEVERIDAD	COMENTARIOS
Strings types should be compared using "equals ()".	Reporta: es necesario usar la comparación entre Strings mediante el método "equals ()".	ALTA	Se acometerá antes del siguiente reporte de Lint.
Remove useless assignment to local variable.	Reporta: es necesario descartar la inicialización de una variable local.	ALTA	Se acometerá antes del siguiente reporte de Lint.
Refactor repetition that can lead to a stack overflow for large inputs.	Reporta: es necesario refactorizar la repetición de grupos dentro de un patrón regex, con el fin de evitar problemas de memoria por inputs de texto demasiado extensos.	ALTA	NO se acometerá debido a la irrelevancia para con este entregable. Infinidad de usos de este campo sin haberse tenido ningún problema de memoria.

Extract nested ternary operation into an independent statement.	Reporta: es necesario transformar operadores ternarios anidados en sentencias independientes.	ALTA	Se acometerá antes del siguiente reporte de Lint.
Add 'id' or 'scope' attribute to '<th>' tag.	Reporta: es recomendable añadir atributos como 'id' o 'scope' a las etiquetas '<th>'.	BAJA	NO se acometerá su resolución al considerarse irrelevante.
Use isEmpty () to check whether the collection is empty or not.	Reporta: es recomendable emplear el método "isEmpty ()" para comprobar si una colección está vacía o no.	BAJA	Se acometerá antes del siguiente reporte de Lint.
Remove the unnecessary boolean literals.	Reporta: es recomendable descartar literales booleanos innecesarios.	BAJA	Se acometerá antes del siguiente reporte de Lint.
Move the array designators '[]' to the type.	Reporta: es recomendable mover el indicador de array al tipado.	BAJA	Se acometerá antes del siguiente reporte de Lint.
Add a description to a table.	Reporta: es recomendable añadir una descripción a las tablas.	BAJA	NO se acometerá su resolución al considerarse irrelevante.

CONCLUSIONES

Gracias a la elaboración de este informe, el alumno ha sido consciente de que velar por mantener unas buenas prácticas y un correcto funcionamiento del código del proyecto, durante el desarrollo del mismo, es esencial para la consecución de los objetivos propuestos en la asignatura.

Por ello, de ahora en adelante, se hará especial hincapié en este procedimiento antes de la conclusión de los siguientes entregables que restan.

BIBLIOGRAFIA

“Intentionally blank”