

CONTROVERSIAS Y LIMITACIONES DE LOMBOK

1. Incapacidad de detectar los constructores de una superclase.

Esto significa que si una superclase no tiene un constructor por defecto, ninguna subclase puede usar la anotación `@Data` sin escribir explícitamente un constructor para hacer uso del constructor de superclase disponible.

CONTROVERSIAS Y LIMITACIONES DE LOMBOK

2. Las anotaciones fueron pensadas para información meta.

Las anotaciones no deben usarse si al eliminarlas el código deja de compilar. Esto es justo lo que sucede con las anotaciones de Lombok. De estas anotaciones surgen nuevos métodos que están destinados a ser utilizados en el propio código de la aplicación. Además, para el correcto funcionamiento requiere soporte por el IDE.

CONTROVERSIAS Y LIMITACIONES DE LOMBOK

3. Código privativo con brujería.

El proyecto Lombok hace uso de un API no públicas para lograr la "brujería". Esto significa que existe el riesgo de que Project Lombok se rompa con las versiones posteriores de IDE o JDK.

CONTROVERSIAS Y LIMITACIONES DE LOMBOK

4. @SneakyThrows permite lanzar excepciones fantasma.

Discordia entre los que apoyan que las excepciones se deben mostrar o no.

RESUMEN (1/3)

1. Es una herramienta de ayuda al desarrollador
2. Elimina código repetitivo
3. Permite crear clases de limpias, concisas y fáciles de mantener.

RESUMEN (2/3)

1. En IntelliJ no es muy estable
2. Existe el riesgo de rotura con las actualizaciones de IDE y JDK
3. Anotaciones ¿para información meta?

RESUMEN (3/3)

La pregunta es

¿El Proyecto Lombok puede proporcionar más valor que costo?

REFERENCIAS

1. Web

- <https://projectlombok.org>

2. Configuración

- <https://projectlombok.org/setup/overview>

3. Instalación

- <https://projectlombok.org/mavenrepo/index.html>

4. API

- <https://projectlombok.org/api/index.html>

5. Google groups

- <https://groups.google.com/g/project-lombok>