[39]

Chen, H. M., Chen, W. H., & Lee, C. C. (2018). An automated assessment system for analysis of coding convention violations in Java programming assignments\*. Journal of Information Science and Engineering, 34(5), 1203–1221. <https://doi.org/10.6688/JISE.201809_34(5).0006>

**KEYWORDS AUTOR:**

code quality, coding convention, program assessment, code readability, software

maintainability, software engineering.

**TÍTULO**:

Un sistema de evaluación automatizado para el análisis de violaciones de la Convención de codificación en asignaciones de programación Java.

**PARA ANEXAR A DOCUMENTO:**

**\*\* INICIO \*\*\***

**TAGS**

MEJORES PRACTICAS; CONVENCIONES DE CODIFICACION; JAVA;

Las convenciones de código son un conjunto de reglas de codificación que adoptan algunos desarrolladores de software para escribir código fuente. El beneficio de aplicarlo es que facilitará la lectura y comprensión del mismo en las tareas de mantenimiento. Esas reglas se dividen en las siguientes categorías: estructura del archivo, identación, comentarios, espacios en blanco, nombre de identificadores, declaraciones y prácticas específicas para lenguajes de programación.

TRABAJOS RELACIONADOS

|  |  |  |
| --- | --- | --- |
| AUTOR | DESCRIPCIÓN | DEBILIDADES |
| Li et al | 3 años investigando si los estudiantes adoptan las reglas de codificación para programar. Encontraron que aunque querían aplicarlas en sus tareas, no les alcanzaba el tiempo, por lo que propusieron estrategias para motivarlos, entre ellas, introducir herramientas automáticas que faciliten el cumplimiento de los estándares cuando se aprende a programar. |  |
| In | Propuso realizar procesos de revisión del código en parejas, obteniendo retroalimentación | Los estudiantes pueden engañar y se dificulta en grupos grandes. |
| Wang el al | Un proceso de revisión en parejas mejorado, con mayor número de revisiones y analizando el comportamiento de los participantes. | No hubo compromiso de las partes, por ser un proceso manual que requería de mucho tiempo. |

Herramientas automáticas para la detectar violaciones en convenciones de codificación

|  |  |
| --- | --- |
| **Herramienta** | **Características** |
| CheckStyle | * Para lenguaje Java. * Flexibilidad en la configuración de reglas. * Generación de reportes para examinar resultados. * Fácil integración con ambientes de desarrollo. * Clasifica las convenciones de código en las siguientes categorías: Importaciones, anotaciones, verificación de bloques, diseño de clases, codificación, encabezados, comentarios JavaDoc, métricas, modificadores, convenciones de nombres, Regexp, violaciones de tamaño, espacios en blanco y misceláneos. |
| PMD | * Analizador de código fuente que ayuda a los desarrolladores a encontrar potenciales fallas. * Detecta violaciones de convenciones como llaves, tamaño de código y nombres. * Cubre menos convenciones que CheckStyle. * La curva de aprendizaje para configurar las reglas, el alta. * Usualmente usado como complemento de CheckStyle. |
| SonarQube | * Aplicación web que permite comprender los niveles de calidad desde varios puntos de vista. * Examina la calidad de los programas de forma continua a los largo del tiempo. * Soporta más de 25 lenguajes de programación. * Permite integración con otras herramientas como CheckList y PMD, mejorando los datos reportados y brindando una interfaz amigable con el usuario. |

**Resultados**

Al aplicar el uso de las herramientas, a lo largo de los cursos de programación, se encontraron las siguientes violaciones de codificación más comunes:

|  |  |
| --- | --- |
| **Objeto evaluado** | **Violaciones de codificación más comunes** |
| Programas en lenguaje Java, orientados a objetos | * Identación: el espacio asignado por la tecla tab difiere de un editor a otro, se recomienda mejor usar la barra espaciadora. * Espacios en blanco. |
| 23 Aplicaciones móviles en Java. | * Identación. * Espacios en blanco. * Nombre de identificadores. * Brace (apertura y cierre de llaves). |
|  |  |

Chen et al (2018), desarrollaron la herramienta web ProgEdu, la cual apoya el aprendizaje de los cursos de programación y la revisión de violaciones de codificación en los programas desarrollados por los estudiantes. El ciclo del proceso de aprendizaje es iterativo y consiste en los siguientes pasos:

1. El instructor realiza la asignación de la tarea de programación.
2. El estudiante escribe el código o lo revisa.
3. El sistema analiza el código. El estudiante es retroalimentado y puede volver al paso 2 para realizar los correctivos.
4. Una vez el análisis del código es satisfactorio, pueden consultarse los reportes de todo el histórico en el sistema.

Los instructores y los estudiantes pueden ingresar a ProgEdu, donde podrán consultar su asignación y ver de forma amigable los resultados de la revisión del código y la evolución en el trabajo.

**RETOS**

Ampliar el soporte a otros lenguajes de programación, como C, C++ y Python.

**DATASET**:

**DOCUMENTACIÓN:**

Guías de estilo:

<https://github.com/google/styleguide>

<https://github.com/facebook/jcommon/wiki>

**HERRAMIENTAS**:

<https://checkstyle.sourceforge.io/>

<https://pmd.github.io/>

**\*\* FIN \*\*\***