

Servicio Nacional de Aprendizaje SENA

Tecnología en análisis y desarrollo de software ADSO

Número de la ficha

2522002



Nombre de la evidencia

GA7-220501096-AA1-EV02 definir estándares de
codificación de acuerdo con la plataforma de desarrollo elegida

Nombre del aprendiz

José David Mesa Galindo

Jhon Estiwar Rodriguez

Año

2023

Introducción

La presente documentación establece los estándares de codificación en el lenguaje de programación Java, basados en el paradigma orientado a objetos, para promover un código fuente limpio, legible y consistente en todos los proyectos desarrollados en la tienda Minimarket la 5ta. Estos estándares tienen como objetivo mejorar la calidad del código, facilitar el mantenimiento y colaboración entre los desarrolladores, así como fomentar las mejores prácticas de programación orientada a objetos.

Objetivo

El objetivo de este documento es establecer una guía de estándares de codificación en Java, bajo el enfoque del paradigma orientado a objetos, que incluya directrices sobre el nombramiento de variables, la declaración de clases, métodos y otros elementos clave del lenguaje. Estos estándares permitirán mejorar la legibilidad del código, la coherencia entre diferentes partes del sistema y facilitar la comprensión y el mantenimiento a largo plazo en el contexto de la programación orientada a objetos.

Estándares de Codificación para el Paradigma Orientado a Objetos en Java:

Nombramiento de Variables:

1. Las variables deben tener nombres significativos y descriptivos, que reflejen su propósito y función en el contexto de la clase.

- Se utilizará el formato camelCase para nombrar variables (ejemplo: miVariable).
- Evitar el uso de nombres genéricos como "var1", "x", etc.
- Las constantes se nombrarán en mayúsculas con palabras separadas por guiones bajos (ejemplo: MI_CONSTANTE).

2. Declaración de Clases:

- Los nombres de las clases deben ser sustantivos o frases sustantivadas y utilizar el formato UpperCamelCase (ejemplo: MiClase).
- Cada clase debe representar un concepto coherente y tener una única responsabilidad.
- Las clases deben encapsular datos y comportamientos relacionados, siguiendo los principios de cohesión y acoplamiento mínimo.

3. Declaración de Métodos:

- Los nombres de los métodos deben ser verbos o frases verbales y utilizar el formato camelCase (ejemplo: miMetodo).
- Los nombres de los métodos deben ser descriptivos y reflejar claramente la acción que realizan.
- Los métodos deben representar comportamientos específicos de la clase y tener una única responsabilidad.

- Se debe seguir el principio de ocultamiento de información (encapsulamiento) al definir la visibilidad de los métodos.

4. Otros Estándares de Codificación:

- Se utilizarán indentaciones de cuatro espacios para mejorar la legibilidad.
- Cada sentencia estará en una línea separada.
- Se utilizarán comentarios descriptivos para explicar secciones de código complejas y documentar el propósito de las clases, métodos y variables.
- Se evitará el uso de abreviaturas y acrónimos poco claros en los nombres de variables, métodos y clases.

Estándares de Codificación para el Frontend:

1. Nombramiento de Variables:

- Se utilizará el formato lowerCamelCase para nombrar las variables del frontend.
- Los nombres de las variables deben comenzar con minúsculas y las palabras se unen sin espacios, iniciando cada palabra con mayúscula.
Ejemplo: nombrePersona = "Juan".

2. Constantes:

- Los nombres de las constantes en el frontend deben estar escritos completamente en mayúsculas.
- Las palabras se separan utilizando subrayados ("_").
- Todas las constantes se declararán como public static final.

3. Paquetes:

- Los nombres de los paquetes en el frontend se escribirán en minúsculas y sin utilizar caracteres especiales.
- Se recomienda utilizar el dominio de la página web de la organización como el paquete base.
- Dentro del paquete base, se deben organizar los subpaquetes correspondientes.

4. Clases:

- Los nombres de las clases en el frontend serán una mezcla de mayúsculas y minúsculas.
- La primera letra de cada palabra interna en el nombre de la clase debe estar en mayúscula (UpperCamelCase).
- Se deben utilizar nombres de clases simples y descriptivos, evitando abreviaturas y utilizando palabras completas.

5. Métodos:

- Los nombres de los métodos en el frontend deben ser verbos en infinitivo.
- La primera letra del nombre del método debe estar en minúscula, y la primera letra de cada palabra interna debe estar en mayúscula (lowerCamelCase).
- No se deben utilizar caracteres especiales en los nombres de los métodos.

- Los nombres de los métodos deben ser lo suficientemente descriptivos, sin importar su longitud. Ejemplo: registrarEmpleado().

6. Comentarios:

- Se deben utilizar comentarios para documentar cada clase y método según el estándar Javadoc.
- Los comentarios deben explicar la funcionalidad de la clase o método, su uso y cualquier otra información relevante.