



PRINCIPIOS SOLID

Antonio Espín Herranz

Contenidos

- Principio de responsabilidad única (Single responsibility principle)
- Principio de abierto/cerrado (Open/closed principle)
- Principio de sustitución de Liskov (Liskov substitution principle)
- Principio de segregación de la interfaz (Interface segregation principle)
- Principio de inversión de la dependencia (Dependency inversion principle)

Introducción

- Los Principios SOLID tienen cinco principios de diseño de clases Orientado a Objetos. Son un conjunto de reglas y mejores prácticas a seguir al diseñar una estructura de clase.
- Estos cinco principios nos ayudan a comprender la necesidad de ciertos patrones de diseño y arquitectura de software en general.
- Los principios SOLID fueron introducidos por primera vez por el famoso científico informático Robert J. Martin (también conocido como el tío Bob) en uno de sus artículos en 2000. Pero el acrónimo SOLID fue introducido más tarde por Michael Feathers.

Principio de responsabilidad única

- El Principio de Responsabilidad Única dice que **una clase debe hacer una cosa y, por lo tanto, debe tener una sola razón para cambiar.**
- Para enunciar este principio más técnicamente: Solo un cambio potencial (lógica de base de datos, lógica de registro, etc.) en la especificación del software debería poder afectar la especificación de la clase.
- Esto significa qué si una clase es un contenedor de datos, como una clase Libro o una clase Estudiante, y tiene algunos campos relacionados con esa entidad, debería cambiar solo cuando cambiamos el modelo de datos.

Principio de abierto/cerrado

- El principio de apertura y cierre exige que **las clases deban estar abiertas a la extensión y cerradas a la modificación.**
- Modificación significa cambiar el código de una clase existente y extensión significa agregar una nueva funcionalidad.
- Entonces, lo que este principio quiere decir es: Deberíamos poder agregar nuevas funciones sin tocar el código existente para la clase. Esto se debe a que cada vez que modificamos el código existente, corremos el riesgo de crear errores potenciales. Por lo tanto, debemos evitar tocar el código de producción probado y confiable (en su mayoría) si es posible.

Principio de abierto/cerrado

- ¿cómo vamos a agregar una nueva funcionalidad sin tocar la clase?, puede preguntarse. Por lo general, se hace con la ayuda de interfaces y clases abstractas.
- A la clase anterior de PRU añadir otra operación de persistencia (guardarEnBD)
- Añadir esta nueva operación implica modificar la clase, con lo cual se viola el principio de abierto y cerrado.

Principio de sustitución de Liskov

- El Principio de Sustitución de **Liskov** establece que las subclases deben ser sustituibles por sus clases base.
- Esto significa que, dado que la clase B es una subclase de la clase A, deberíamos poder pasar un objeto de la clase B a cualquier método que espere un objeto de la clase A y el método no debería dar ningún resultado extraño en ese caso.
- Este es el comportamiento esperado, porque cuando usamos la herencia asumimos que la clase secundaria hereda todo lo que tiene la superclase. La clase secundaria extiende el comportamiento, pero nunca lo reduce.

Principio de segregación de la interfaz

- La segregación significa mantener las cosas separadas, y el Principio de Segregación de Interfaces se trata de separar las interfaces.
- El principio establece que muchas interfaces específicas del cliente son mejores que una interfaz de propósito general. No se debe obligar a los clientes a implementar una función que no necesitan.

Principio de inversión de la dependencia

- El principio de inversión de dependencia establece que nuestras clases deben depender de interfaces o clases abstractas en lugar de clases y funciones concretas.
- "Si el PAC establece el objetivo de la arquitectura OO, el PID establece el mecanismo principal".
- Estos dos principios están realmente relacionados y hemos aplicado este patrón antes mientras discutíamos el Principio Abierto-Cerrado.

Fuentes

- <https://www.freecodecamp.org/espanol/news/los-principios-solid-explicados-en-espanol/>