

Concepto de Singleton

El patrón Singleton es un patrón de diseño creacional que asegura que una clase tenga una única instancia y proporciona un punto de acceso global a ella. Es útil en situaciones donde tener más de una instancia de una clase podría causar problemas, como en la gestión de configuraciones, conexión a bases de datos, o manejo de recursos compartidos.

Explicación del Ejemplo

En el ejemplo proporcionado, se muestra cómo implementar un Singleton en Java.

1. Clase Singleton:
 - a. Utiliza una variable estática privada `instance` para almacenar la única instancia de la clase.
 - b. El constructor es privado para evitar que se creen instancias adicionales.
 - c. El método `getInstance()` proporciona un punto de acceso global a la instancia única. Si la instancia aún no existe, se crea y se devuelve.
2. Clase Main:
 - a. Muestra cómo se utiliza el método `getInstance()` para obtener la instancia única.
 - b. Se demuestra que, incluso si se intenta obtener la instancia varias veces, siempre se devuelve el mismo objeto.
 - c. Beneficio: El patrón Singleton es útil cuando necesitas controlar el acceso a un recurso compartido o asegurar que solo exista una única instancia de una clase en todo el sistema.