

Hoja de trabajo 4

Ventajas y desventajas de utilizar el patrón singleton

Acceso controlado a una única instancia: garantiza que haya solo una instancia y sea global

Acceso global: puede ser accedida de manera global

Consistencia: asegura un estado consistente en toda la aplicación al usar la misma instancia

Dependencias ocultas: este patrón puede ocultar las dependencias lo que provoca que sea más difícil de leer y mantener el código

Dificultad en pruebas unitarias: Al ser global, los casos de prueba pueden interferir entre sí por compartir la misma instancia.

Ciclo de vida: es más complicado mantener el ciclo de vida de un singleton ya que no es tan directo.

¿Cree que su uso es adecuado en este programa?

Creemos que para este programa no es adecuado utilizar el patrón singleton ya que al ser una única instancia de calculadora esta se vuelve global, el problema con esto es que la calculadora evalúa múltiples expresiones del .txt, al tener una única instancia es probable que los datos de una operación anterior interfieran en el siguiente cálculo si el stack no se limpia cada vez que se corre el programa, también los Junit son más complicados porque el estado global hace que los casos de prueba se puedan afectar entre sí al compartir la misma memoria.

Referencia:

Aswal, A. (2024). *Singleton design pattern (swift). Advantages and disadvantages with examples in swift*. Medium.

<https://medium.com/@amitaswal87/singleton-design-pattern-swift-advantages-and-disadvantages-with-examples-in-swift-c1cc6e2ef3ff>