

Singleton

Propósito

Asegura que una clase sólo tiene una instancia y proporciona un punto de acceso a la instancia.

Aplicabilidad

El Singleton se utiliza cuando:

- Debe haber exactamente una instancia de una clase y debe ser accesible a los clientes
- Se debe extender esa única instancia mediante subclasificación, y los clientes deben ser capaces de usar la instancia extendida sin modificar su código

Estructura



Participantes

- Singleton:
 - o Define una operación Instance() que utilizan los clientes para acceder a su instancia única (es una función static en Java o C++, o un método de clase en Smalltalk)
 - o Es el responsable de crear su propia única instancia

Colaboraciones

- Los clientes acceden la instancia del Singleton solamente a través de la operación Instance().

Consecuencias

El patrón Singleton tiene los siguientes beneficios:

1. Acceso controlado a la única instancia. Como la clase Singleton encapsula a su única instancia, tiene un control estricto sobre cómo y cuándo acceden los clientes.
2. Espacio de nombres reducido. El patrón Singleton es una mejora sobre las variables globales. Evita polucionar el espacio de nombres con variables globales que almacenan instancias.
3. Permite el refinamiento y la representación de operaciones. La clase Singleton se puede subclasificar de modo que podría ser sencillo configurar una aplicación con una instancia de la clase extendida. Se puede hacer esa configuración en tiempo de ejecución.
4. Permite un número variable de instancias. El patrón permite un cambio de criterio para permitir más de una instancia. Es más, se puede utilizar la misma solución para

controlar el número de instancias que usa una aplicación. Solo hay que modificar la operación que garantiza el acceso a la instancia del Singleton.