Singleton, ejemplos

También conocido como instancia única

Es un patrón creacional

Proporcionar un punto de acceso global a una instancia

Se usa habitualmente en clases que controlan el acceso a un recurso físico único (un teclado por ejemplo) (también se puede analizar el caso de que haya varias impresoras disponibles y una clase que controla la impresión, es importante que sea un único objeto el que exista en el gestor de impresión). O bien cuando un cierto grupo de datos debe estar disponible a todos los demás objetos de la aplicación (por ejemplo una lista de países, a la que varios usuarios tiene acceso, no vale la pena crear una instancia por usuario si la lista siempre va a ser la misma). Su aplicación viene a ser básicamente una variable global

En ingeniería de software, singleton o instancia única es un patrón de diseño que permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón singleton se implementa creando en nuestra clase un método que crea una instancia del objeto solo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con modificadores de acceso como protegido o privado).

La instrumentación del patrón puede ser delicada en programas con múltiples hilos de ejecución. Si dos hilos de ejecución intentan crear la instancia al mismo tiempo y esta no existe todavía, solo uno de ellos debe lograr crear el objeto. La solución clásica para este problema es utilizar exclusión mutua en el método de creación de la clase que implementa el patrón.

Las situaciones más habituales de aplicación de este patrón son aquellas en las que dicha clase controla el acceso a un recurso físico único (como puede ser el ratón o un archivo abierto en modo exclusivo) o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación.

El patrón singleton provee una única instancia global gracias a que:

La propia clase es responsable de crear la única instancia.

Permite el acceso global a dicha instancia mediante un método de clase.

Declara el constructor de clase como privado para que no sea instanciable directamente.

Al estar internamente autoreferenciada, en lenguajes como Java, el recolector de basura no actúa.