Fernando Figueroa 14175
José Andrés Arenas 14470
Sebastián Arriola 11463
Universidad del Valle de Guatemala
Estructuras de Datos y Algoritmos
19/2/2018

Acerca del Patrón de Diseño Singleton

Ventajas

- El patrón es una mejora sobre las variables globales.
- Si se tiene un recurso del cual solo puede existir una única instancia, este patrón resuelve este problema.
- No se reservan nombres para las variables globales, ahora solo existen instancias.
- Controla el acceso a la instancia única, la clase encapsula la única instancia.
- Inicialización “Lazy” o tardía, la cual provee la ventaja de que en el caso que no se requiera una instancia, nunca será creada.

Desventajas

- Debido a que ellos mismos controlan su creación, violan el Principio de Única Responsabilidad, lo cual puede llevar a confusiones para otros desarrolladores.
- En el momento de que la complejidad de nuestro programa crezca, este patrón puede causar problemas, debido a que el objeto mismo controla su creación.
- Al usar este patrón, se debe prohibir extender la clase que implemente el patrón, debido a que las subclases estarían creando nuevas instancias, lo cual viola totalmente el patrón.

El Patrón Singleton en la Hoja de Trabajo # 4

Como ventajas podemos mencionar que el usuario no debe preocuparse por instanciar una calculadora, sino que al momento de querer usarla, solamente solicita una instancia de la misma, con lo cual el usuario queda totalmente ajeno a la creación de la calculadora.

Como desventajas podemos mencionar que una vez definido el Stack que utiliza la calculadora, no se puede cambiar. Este caso tendría que tomarse en consideración si se quisiera tener un comportamiento en el cual el Stack se pudiera cambiar en tiempo de ejecución. Creemos que las desventajas que introduce este patrón son relacionadas a la complejidad en el código, no es nada que no se pueda resolver, pero requiere de una planeación más cuidadosa para prevenir desastres en el futuro.