

Revise que ventajas / desventajas hay al utilizar el patrón Singleton en general, ya que su comportamiento es muy similar a una variable global. ¿Cree que su uso es adecuado en este programa? Conteste en un documento PDF.

En cierto sentido es cierto que se comporta similar a una variable global, sin embargo, se debe de recordar que el patrón singleton sirve para instanciar objetos, no es una variable como tal. Además, es muy útil debido a que permite ahorrar memoria al momento de manejar dado que se crea un objeto útil que se utilizará para todo el programa. Lo probé con el Hashcode y es cierto, es solo 1 objeto instanciado.

Así mismo, quisiera decir que en este programa su uso es adecuado pero no se siente (observa) tanto ya que se vería todo su potencial si el usuario ingresara continuamente operaciones. En ese caso se usaría constantemente 1 sola calculadora para cada operación lo que facilita mucho el ahorro de memoria, probablemente esto permite que el proceso se realice con mayor versatilidad que si se tuviese que crear 1 calculadora por cada operación. Este proceso aunque no es un singleton si lo utilicé hasta cierto punto al momento de convertir el stack a postfix o de operar el stack dado que se utilizaban los mismos stacks para realizar estas operaciones, se debe destacar que estos stacks luego de realizar una determinada operación regresaban a vacío por lo que no ocupan mucha memoria y mantenían la clase del patrón factory.

Se debe de destacar que dicho patrón es el más sencillo de ver sin embargo, si se crean muchos factories se puede caer en el uso de demasiados recursos. ¿Por qué? Porque imaginense que creamos 3 stacks temporales por cada método, y fueran 7 métodos para hacer una determinada acción completa. Eso significa que el garbage collector debe de eliminar 21 cosas y al menos debe de pasar 7 veces, eso es una barbaridad.