Caso de Estudio: Calculadora

Aseguramiento de la calidad de Software

*Practica 3´*

Seguiremos con nuestro caso de estudio de Calculadora.

# Inspecciones y revisiones

Como parte de la practica, se enviar´ an los c´ odigos de las primeras clases del proyecto:´

1. Realice las pruebas unitarias para las clases:

1. ArrayIterator*<T >*
2. StackADT*<T >*
3. ArrayStack*<T >*
4. ListADT*<T >*
5. UnorderedListADT*<T >*

Como notaras dos clases son implementaciones de interfaces y tres clases son interfaces. Para realizar´ las inspecciones, puedes realizar pruebas de caja blanca y caja negra. En estos casos, deberas basarte en´ las indicaciones de la practica dos sobre la implementaci´ on. Observa que la implementaci´ on es mediante´ arreglos. Puedes consultar, sitios de internet o libros sobre como se implementan estas estructuras mediante matrices o arreglos.

stackADT y listADT son interfaces de ciertas colecciones.

stackADT que es una interfaz define quien esta al ultimo y quien esta al principio de una pila de elementos.

listADT es una interfaz que define una lista de colecciones que se extiende hasta completar una serie de operaciones en esta.

ArrayStack: Una implementación de Stack API que se basa en un ArrayList en lugar de un Vector, por lo que no está sincronizado para proteger contra el acceso de subprocesos múltiples. Por lo tanto, la implementación funciona más rápido en entornos en los que no necesita preocuparse por la contención de varios subprocesos. El orden de eliminación de un ArrayStack se basa en el orden de inserción: el elemento agregado más recientemente se elimina primero. El orden de iteración no es el mismo que el orden de eliminación. El iterador devuelve elementos de abajo hacia arriba, mientras que el método remove() los elimina de arriba hacia abajo.

arrayIterator: ‎La clase itera sobre una matriz.‎ArrayIterator

1