

Algoritmos y Estructura de Datos

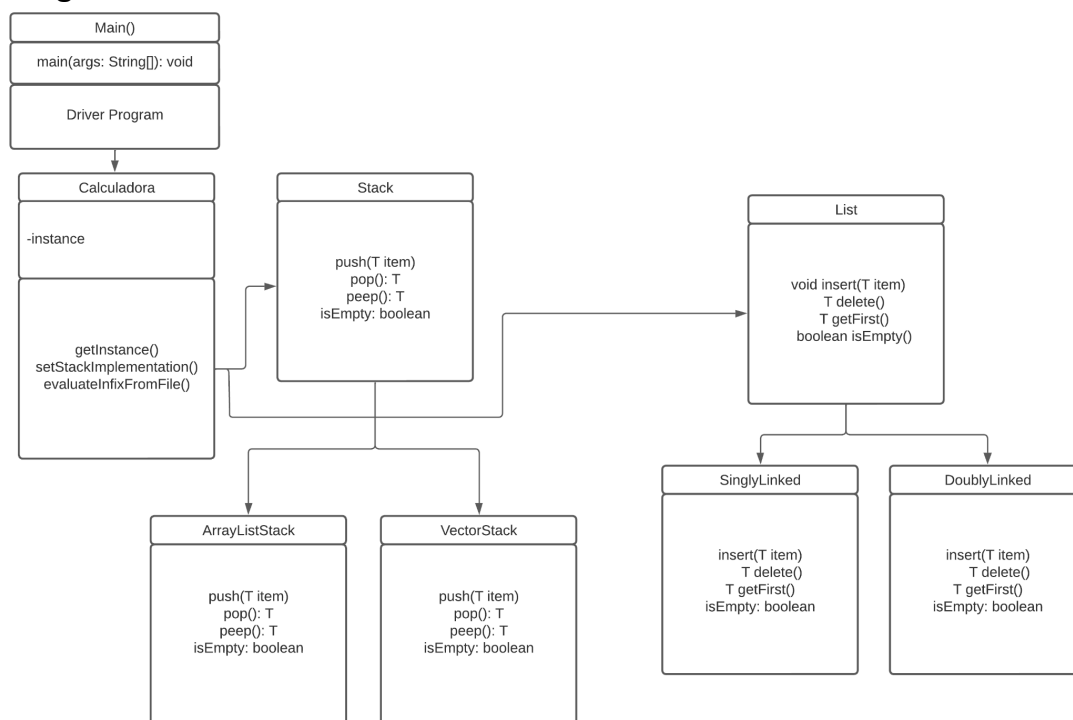
Hoja de Trabajo 4: Infix and Postfix

“Debe de realizar un programa para evaluar expresiones infix (Deben ser convertidas a postfix por el programa y luego ser evaluadas). En un grupo de 2 personas.”

Objetivos:

- Utilización de genéricos.
- Diseño del ADT para pilas (stack), debe tener: 1. Interfaz, 2. Clase Abstracta, 3. Clases de implementación.
- Diseño del ADT para listas, con: 1. Interfaz, 2. Clase Abstracta, 3. Clases de implementación.
- Uso de patrones de diseño: Factory y Singleton.
- Control de versiones del programa.
- Emplear JUnit para casos de prueba.

Diagrama UML:



Repositorio (GitHub):

<https://github.com/adirnnn/HDT-4-Infix-and-Postfix.git>

Pregunta Sobre Uso de SINGLETON:

Revise qué 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?

En cuanto a si el uso del patrón Singleton es adecuado en este programa específico, depende de varios factores, como la complejidad del programa, los requisitos de diseño y las consideraciones de concurrencia. Dado que el programa consiste en una aplicación simple de consola que no necesita administrar múltiples instancias de la Calculadora, el uso del Singleton es apropiado para garantizar que solo haya una instancia de Calculadora en ejecución. Sin embargo, si el programa se volviera más complejo o necesitara admitir múltiples instancias de la Calculadora, podría ser más adecuado utilizar otro enfoque de gestión de instancias.