

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
Facultad de Ciencias



Estructuras de Datos

Práctica 10: Stack

Profesora:

Amparo López Gaona

Ayudante: Adrián Aguilera Moreno

Ayudante de Laboratorio: Kevin Jair Torres Valencia

Objetivo

El objetivo de esta práctica es que los estudiantes comprendan y utilicen la estructura de datos pila (Stack) del Java Collection Framework (JCF) para evaluar expresiones matemáticas en notación postfija (notación polaca inversa) y prefija (notación polaca).

Introducción

El Java Collections Framework (JCF) es una biblioteca en Java que proporciona una arquitectura unificada para manipular y gestionar colecciones de datos. Incluye interfaces, implementaciones y algoritmos que facilitan la manipulación de estructuras de datos como listas, conjuntos, colas y mapas. Dentro del JCF, una de las estructuras de datos más utilizadas es la pila (Stack), que sigue el principio **LIFO** (*Last In, First Out*), es decir, el último elemento en entrar es el primero en salir.

La clase `Stack<E>` de Java extiende la clase `Vector<E>` e incluye varios métodos útiles, entre ellos:

- `push(E item)`: Inserta un elemento en la cima de la pila.
- `pop()`: Extrae y devuelve el elemento en la cima de la pila.
- `peek()`: Devuelve el elemento de la cima de la pila sin eliminarlo.
- `empty()`: Verifica si la pila está vacía.
- `search(Object o)`: Busca un elemento en la pila y devuelve su posición (1 es la parte superior de la pila).

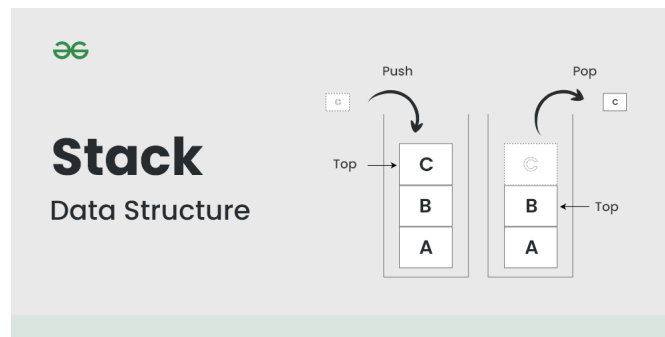


Figure 1: Representación de una Pila

Desarrollo

En matemáticas y ciencias de la computación, las expresiones pueden representarse en diferentes notaciones. Dos notaciones comunes son:

- **Notación Postfija (Notación Polaca Inversa - RPN):** Los operandos aparecen antes que el operador. No requiere paréntesis para definir el orden de las operaciones.

Por ejemplo: $3\ 4\ +$, lo que es equivalente a $3 + 4$

- **Notación Prefija (Notación Polaca):** El operador aparece antes que los operandos. Por ejemplo: $+ 3\ 4$, lo que es lo que es equivalente a $3 + 4$

El problema consiste en que implementen los métodos de evaluación para expresiones en notación postfija y prefija, utilizando la estructura de datos **Stack** en Java. Para ello:

- Se les proporciona las clases de **Main**, **Operaciones**, **Postfijo** y **Prefijo**. En las últimas dos, contienen los métodos `evaluar(String expresion)`, los cuales deben ser completados.
- Completen la implementación del método `evaluar` en ambas clases, asegurándose de utilizar la pila correctamente para evaluar la expresión.
- Utiliza la clase **Operaciones** para manejar los operadores y realizar las operaciones matemáticas.
- Prueba tu implementación utilizando la clase **Main**, ingresando expresiones en ambas notaciones y verificando que los resultados sean correctos.

Como recomendación para realizar la implementación del método `evaluar` en ambas clases **Postfijo** y **Prefijo** es:

- Divide la expresión en tokens utilizando `Operaciones.tokenizarExpresion(expresion)`
- Recorre los tokens:
 - Si el token es un número, agrégalo a la pila.
 - Si el token es un operador, extrae los operandos necesarios de la pila, realiza la operación y almacena el resultado en la pila.
- Al final, la pila debe contener solo un elemento, que será el resultado de la expresión.

Formato de Entrega

1. Las prácticas se entregarán en parejas.
2. NO incluir los archivos .class dentro de la carpeta.
3. Los archivos de código fuente deben estar documentados.
4. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier práctica plagiada total o parcialmente será penalizada con cero para los involucrados.
5. La práctica se debe subir al Github Classroom correspondiente.
6. La entrega en classroom debe contener el link HTTPS y SSH de su repositorio y es lo único que se debe entregar.
7. El horario y día de entrega se acordará en la clase de laboratorio y no deberá sobrepasar 2 clases de laboratorio.