

PILA (STACK)

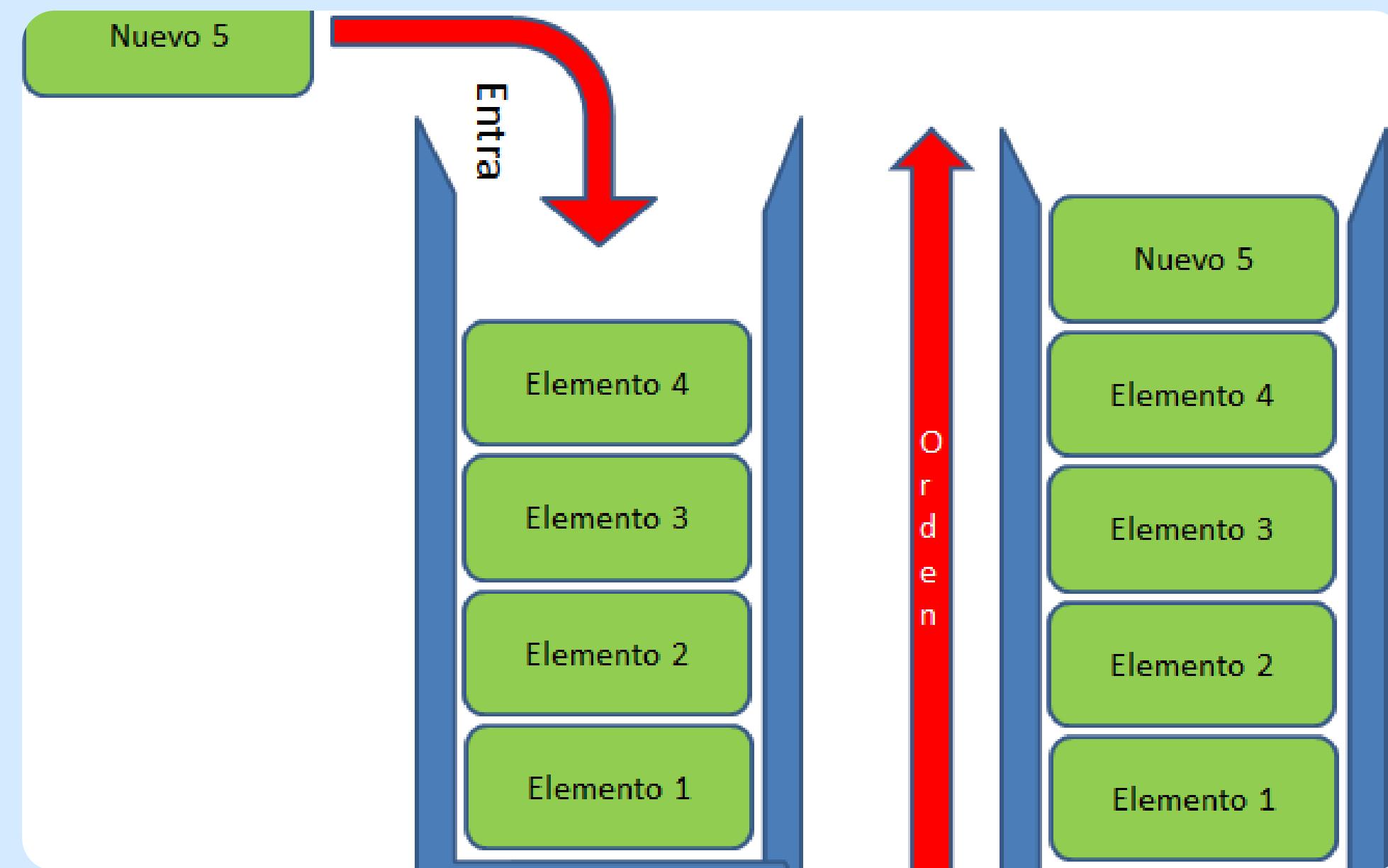
Por Gabriel Castillo Olvera, Ricardo Daniel Infante
Baez y Santiago Avalos Loyda

Explicacion

Gabriel Castillo Olvera

¿Qué es una Pila?

Una pila es una estructura de datos lineal que sigue el principio LIFO (Last In, First Out), es decir, el último elemento en entrar es el primero en salir.



Operaciones principales

1. **push(elemento)** - Insertar

Agrega un nuevo elemento en la parte superior de la pila.

Ejemplo:

Si la pila es [1, 2, 3] y haces push(4), la pila queda así: [1, 2, 3, 4].

2. **pop()** - Eliminar

Elimina y devuelve el elemento superior de la pila.

Ejemplo:

Si la pila es [1, 2, 3], hacer pop() devolverá 3, y la pila quedará [1, 2].

3. **peek() o top()** - Consultar el tope

Devuelve el elemento que está en la cima sin eliminarlo.

Ejemplo:

Si la pila es [1, 2, 3], peek() devolverá 3 y la pila no cambia.

Operaciones principales

4. isEmpty() - Verificar si está vacía

Devuelve true si la pila no tiene elementos, y false en caso contrario.

Ejemplo:

Una pila [] devuelve true al hacer isEmpty().

5. size() - Obtener el tamaño

Devuelve el número de elementos que hay actualmente en la pila.

Ejemplo:

Una pila [1, 2, 3, 4] tiene un size() de 4.

Ventajas

1. Simplicidad de implementación

La pila es una estructura simple de entender y programar. Su comportamiento LIFO facilita la gestión de datos en muchos algoritmos.

2. Gestión eficiente de datos temporales

Es ideal para almacenar datos que deben procesarse en orden inverso, como en algoritmos recursivos o retrocesos (backtracking).

3. Soporte natural para la recursión

En lenguajes de programación, la pila de llamadas se usa para manejar funciones recursivas, lo que permite volver al punto correcto tras cada llamada.

Usos comunes

1. Gestión de llamadas a funciones (pila de llamadas)

Cuando una función llama a otra, se guarda el estado actual (como variables y dirección de retorno) en una pila. Al terminar una función, se retoma la ejecución desde donde se dejó.

2. Deshacer y rehacer acciones (Undo/Redo)

Aplicaciones como editores de texto o gráficos usan pilas para registrar acciones del usuario. Permiten deshacer (pop) o rehacer (push) fácilmente.

3. Evaluación de expresiones

Se utilizan pilas para evaluar expresiones matemáticas en notación postfija (Polaca inversa) o convertir entre notaciones (infix a postfix/prefix).

4. Recorrido de estructuras como árboles y grafos

El algoritmo DFS (Depth-First Search) usa una pila para explorar nodos en profundidad antes de retroceder.

Implementación en código

Santiago Avalos Loyda

```
public class Stack { no usages
    private int maxSize; 3 usages
    private int[] stackArray; 4 usages
    private int top; 6 usages

    public Stack(int size) { no usages
        maxSize = size;
        stackArray = new int[maxSize];
        top = -1;
    }
    public void push(int value) { no usages
        if (top < maxSize - 1) {
            stackArray[++top] = value;
        } else {
            System.out.println("Stack Overflow");
        }
    }
    public int pop() { no usages
        if (!isEmpty()) {
            return stackArray[top--];
        } else {
            System.out.println("Stack Underflow");
            return -1;
        }
    }
    public int peek() { no usages
        return stackArray[top];
    }
    public boolean isEmpty() { 1 usage
        return (top == -1);
    }
}
```

Resolución de un ejercicio de LeetCode

Ricardo Daniel Infante Baez

Problema a resolver

682. Baseball Game

Easy

Topics

Companies

You are keeping the scores for a baseball game with strange rules. At the beginning of the game, you start with an empty record.

You are given a list of strings `operations`, where `operations[i]` is the i^{th} operation you must apply to the record and is one of the following:

- An integer `x`.
 - Record a new score of `x`.
- `'+'`.
 - Record a new score that is the sum of the previous two scores.
- `'D'`.
 - Record a new score that is the double of the previous score.
- `'C'`.
 - Invalidate the previous score, removing it from the record.

Return *the sum of all the scores on the record after applying all the operations*.

The test cases are generated such that the answer and all intermediate calculations fit in a **32-bit** integer and that all operations are valid.

iGracias!