

Partiendo del ejemplo Productor-Consumidor, se pide volver a implementar el ejemplo, utilizando ahora una pila de enteros que puede tener un tamaño máximo de 3 elementos para representar el objeto compartido. Esto implica que ahora el productor puede añadir hasta 3 elementos en el objeto compartido sin que sean consumidos.

Una pila (o stack) es una estructura de datos fundamental que sigue el principio de último en entrar, primero en salir (LIFO, por sus siglas en inglés).

En Java, puedes implementar una pila utilizando la clase Stack de la biblioteca de Java o puedes crear tu propia implementación utilizando un ArrayList, un array, o una lista enlazada.

Uso de la Clase Stack en Java

La clase Stack está incluida en el paquete java.util y proporciona una implementación básica de la estructura de datos de pila. Métodos más comunes:

push(E item): Agrega un elemento al tope de la pila.

pop(): Elimina y devuelve el elemento en la parte superior de la pila.

peek(): Devuelve el elemento en la parte superior de la pila sin eliminarlo.

isEmpty(): Verifica si la pila está vacía. Este método devuelve true si la pila no contiene elementos y false si tiene al menos un elemento.

size(): Devuelve el número de elementos en la pila.

Ejemplo de Uso de la Clase Stack

```
import java.util.Stack;
public class PilaEjemplo {
    public static void main(String[] args) {
        Stack<Integer> pila = new Stack<>();
        // Agregar elementos a la pila
        pila.push(1);
        pila.push(2);
        pila.push(3);
        System.out.println("Elementos en la pila: " + pila);
        // Ver el elemento en la parte superior
        int cima = pila.peek();
        System.out.println("Elemento en la parte superior: " + cima);
        // Eliminar elementos de la pila
        while (!pila.isEmpty()) {
            int elemento = pila.pop();
            System.out.println("Elemento eliminado: " + elemento);
        }
        System.out.println("Pila después de eliminar todos los elementos: " +
            pila);
    }
}
```

Implementación Propia de una Pila

Si deseas implementar tu propia pila, puedes hacerlo usando un array o un ArrayList.

```
import java.util.ArrayList;

class Pila<T> {
    private ArrayList<T> elementos;

    public Pila() {
        elementos = new ArrayList<>();
    }

    public void push(T item) {
        elementos.add(item);
    }

    public T pop() {
        if (isEmpty()) {
            throw new RuntimeException("Pila vacía");
        }
        return elementos.remove(elementos.size() - 1);
    }

    public T peek() {
        if (isEmpty()) {
            throw new RuntimeException("Pila vacía");
        }
        return elementos.get(elementos.size() - 1);
    }

    public boolean isEmpty() {
        return elementos.isEmpty();
    }

    public int size() {
        return elementos.size();
    }
}

public class PilaEjemplo {
    public static void main(String[] args) {
        Pila<Integer> pila = new Pila<>();

        pila.push(10);
        pila.push(20);
        pila.push(30);

        System.out.println("Elemento en la parte superior: " + pila.peek());
        System.out.println("Elementos en la pila: " + pila.size());

        while (!pila.isEmpty()) {
            System.out.println("Elemento eliminado: " + pila.pop());
        }
    }
}
```