

## LA CLASE Stack DE JAVA

Una pila (stack en inglés) es una estructura de datos de tipo LIFO (del inglés Last In First Out, último en entrar, primero en salir) que permite almacenar y recuperar datos. Se aplica en multitud programas debido a su simplicidad y ordenación implícita en la propia estructura.

Para el manejo de los datos se cuenta con dos operaciones básicas: apilar (push), que coloca un **objeto** en la pila, y su operación inversa, retirar (o desapilar, pop), que retira el último elemento apilado.

Java provee una implementación de esta estructura de datos en la clase `java.util.Stack` que cuenta con los métodos `empty`, `peek`, `pop`, y `push`.

**Ejemplo 1:** programa que permite insertar 5 palabras por teclado en una pila y posteriormente visualiza los mismos

```
import java.util.Scanner;
import java.util.Stack;
public class pilas {
    public static void main(String[] args) {
        Scanner en=new Scanner(System.in);
        Stack<String> pila=new Stack<String>();
        for(int i=0;i<5;i++)
            pila.push(en.next());
        while(!pila.isEmpty())
        {
            System.out.print(pila.pop()+" ");
        }
    }
}
```

**Ejemplo 2:** programa que permite insertar N números enteros por teclado en una pila y posteriormente visualiza los mismos

```
import java.util.Scanner;
import java.util.Stack;
public class Ejemplo {
    public static void main(String arg[]){
        Scanner en=new Scanner(System.in);
        Stack<Integer> pila=new Stack<Integer>();
        int N=en.nextInt();
        for(int i=0;i<N;i++)
            pila.push(en.nextInt());

        while(!pila.isEmpty()){
            System.out.println((int)pila.pop());
        }
    }
}
```

**Ejemplo 3:** programa que permite insertar N caracteres por teclado en una pila y posteriormente visualiza los mismos

```
import java.util.Scanner;
import java.util.Stack;
public class Ejemplo {
    public static void main(String arg[]){
        Scanner en=new Scanner(System.in);
        Stack<Character> pila=new Stack<Character>();
        int N=en.nextInt();
        for(int i=0;i<N;i++){
            pila.push(en.next().charAt(0));

            while(!pila.isEmpty()){
                System.out.println((char)pila.pop());
            }
        }
    }
}
```

**Ejemplo 4:** Programa que permite insertar N datos de Productos en una pila (los datos de cada Producto son Nombre, código, stock). Posteriormente visualice los datos de los productos cuyo stock sea menor a 10.

```
public class Producto {
    private String nombre;
    private int codigo;
    private int stock;
    Producto(String n,int c, int s){
        nombre=n;
        codigo=c;
        stock=s;
    }
    int getStock(){
        return stock;
    }
    void mostrar()
    {
        System.out.println(nombre+" "+codigo+" "+stock);
    }
}
```

```
import java.util.Stack;
import java.util.Scanner;
public class prueba {
    public static void main(String[] args) {
        Scanner en=new Scanner(System.in);
        Stack<Producto> p=new Stack(); //declaracion de la pila
        Producto pro; //pro -> variable de tipo Producto
        System.out.print("inserte cantidad de datos a insertar:");
        int N=en.nextInt(); //cantidad de datos por teclado
        for (int i=0;i<N;i++){
            pro=new Producto(en.next(),en.nextInt(),en.nextInt());
            p.push(pro); //se inserta el producto en la pila
        }
    }
}
```

```
        while(!p.isEmpty()){
            pro=p.pop(); //se asigna a pro el dato de la pila
            int st=pro.getStock(); //se obtiene el stock
            if(st<10)
                pro.mostrar();
        }
    }
}
```