Pilas (Stack)

Tuesday, February 5, 2019 5:27 PM

Estructura de datos abstracta (ADT Abstract Data Type).
Last-In, First-Out (LIFO)
Estructura lineal.
Tienen **un solo** extremo (tope o cima) para realizar operaciones.

- Agregar datos (push).
- Quitar elemento (pop).
- Consultar (peek).
- Determinar si está vacía (isEmpty).

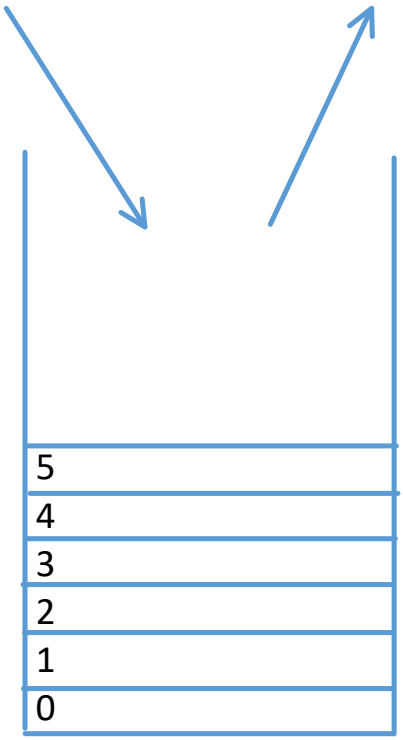
Implementación

- Interfaz

```
Public interface PilaADT<T>{

    Public void push(T dato);

    Public T pop();

    Public T peek();

    Public boolean isEmpty();
}
```

- Clase

```
Public class Pila <T> implements PilaADT<T> {
    Private T[] pila;
    Private int tope;
    Private final int MAX = n;

    Public PilaA(){
```


5
4
3
2
1
0

```
     ,pila = (T[]) new Object[MAX];
     ,tope = -1; //habla del último elemento, por eso no es 0.
}

Public PilaA(int max){
     ,pila = (T[]) new Object[max];
     ,tope = -1;
}

Public boolean isEmpty(){
     Return tope == -1;
}

Public void push(T dato){
     If(tope == pila.length-1) //si está lleno expande
          ,expand();
     ,tope++;
     ,pila[tope] = dato;
}

Private void expande(){
     T[] nuevo = (T[])new Object[pila.length*2];

     For(int i = 0; i <= tope; i++){
          ,nuevo[i] = pila[i]
     }
     ,pila = nuevo; //intercambia la dirección de memoria
}

Public T pop(){
     T res;

     If(isEmpty())
          ,throw new EmptyCollectionException(); //truena y sale
     ,res = pila[tope];
     ,pila[tope] = null;
     ,tope--;
     Return res;
}
```

```
    Public T peek(){
        If(isEmpty())
            ,throw new EmptyCollectionException();
        Return pila[tope]
    }
}
```