**CLASE PILA**

Una pila (stack en inglés) es una lista ordinal o estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO (del inglés Last In First Out, último en entrar, primero en salir) que permite almacenar y recuperar datos.

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 des-apilar, pop), que retira el último elemento apilado.

En cada momento sólo se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado, llamado **cima**.

La operación retirar permite la obtención de este elemento, que es retirado de la pila permitiendo el acceso al anterior (apilado con anterioridad), que pasa a ser el último, la nueva **cima.**

Habitualmente, junto a las dos operaciones básicas de apilar y desapilar (push, pop), las pilas puede implementar otra serie de funciones:

* **Crear (constructor):** crea la pila vacía.
* **Tamaño (size):** regresa el número de elementos de la pila.
* **Apilar (push):** añade un elemento a la pila.
* **Desapilar (pop):** lee y retira el elemento superior de la pila. (Lo elimina de la pila)
* **Leer último (top o peek):** lee el elemento superior de la pila sin retirarlo. (no lo elimina de la pila)
* **Vacía (empty):** devuelve cierto si la pila está sin elementos o falso en caso de que contenga alguno.
* **llena (full):** devuelve cierto si la pila está vacía.

Las pilas se pueden implementar por medio de arrays o de listas enlazadas (no vistas en clase). Cuando se introduce un elemento en la pila primero se incrementa el valor de la **cima** y después se inserta el elemento.

Cuando se retira un elemento de la pila, simplemente, se decrementa el valor de la **cima**.

No se puede extraer el elemento **N** si previamente no se ha extraído todos los elementos anteriores a él.

Define una clase llamada **Pila** con las siguientes características:

**ATRIBUTOS**

* Un atributo de tipo array de enteros que almacena los elementos de la pila.
* Un atributo de tipo entero que es la cima de la pila.
* Un atributo de tipo entero que almacena el tamaño de la pila.
* Un atributo de tipo entero que indica cuántos objetos de tipo pila se han creado en el programa.

**MÉTODOS**

* Un constructor que reciba el tamaño de la pila a crear.
* Un método que decremente en una unidad el número de objetos de tipo pila que se han creado en el programa.
* Los métodos propios de un objeto pila.
* Un método que reciba como parámetro un objeto Pila y devuelva otro objeto Pila que será una copia exacta de la recibida como parámetro. Las dos pilas deben ser exactamente iguales. Si necesitas crear un objeto pila auxiliar no debe queda reflejado en el número de objetos Pila que se han creado en el programa.
* Un método que devuelva en forma de String todos los elementos de la Pila (la pila no se debe destruir)

Crea un programa que cree dos objetos de tipo Pila. A la primera se le irán dando sus elementos. En la segunda se copiarán los elementos de la primera.

Realiza control de excepciones donde sea necesario.