

Ejercicios Tipos Abstractos de Datos

Pila

Implementa un tipo de datos **pila** que permita almacenar enteros sobre un array. La pila debe tener un número máximo de elementos el cuál no se podrá rebasar. Utiliza una constante para almacenar dicho límite.

Las funciones/operaciones que debes implementar son:

Valor de retorno	función	Descripción
Array	create	Crea una pila con el array ya instanciado con el número de elementos máximos.
Boolean	isEmpty(stack)	Devuelve true o false en función de si la pila está vacía.
Boolean	isFull(stack)	Devuelve true o false en función de si la pila está llena.
Number	size(stack)	Devuelve el número de elementos de la pila.
Number	push(stack,elem)	Añade un nuevo elemento a la pila. Devuelve el tamaño de la pila una vez añadido.
Number	pop(stack)	Devuelve el último elemento de la pila, y lo devuelve para que sea consumido.
Number	peek(stack)	Igual que el anterior pero no elimina el elemento de la pila.
String	toString(stack)	Devuelve la pila en formato cadena. El delimitador de elementos será “-”.
Number	search(stack,elem)	Devuelve la posición del elemento indicado. Si el elemento no está en la pila devuelve -1.
Number	capacity(stack)	Devuelve el máximo número de elementos que podemos tener en la pila.
-	clear(stack)	Vacía la pila.
Number	firstElement(stack)	Devuelve el primer elemento de la pila
Number	lastElement(stack)	Devuelve el último elemento de la pila

Tabla 1 Operaciones pila

En cuanto a la gestión de errores. Ten en cuenta las excepciones que debería lanzar cada función.

Función	Excepción
push(stack,elem)	- El elemento no es un Number - La pila está llena
pop(stack)	- La pila está vacía
peek(stack)	- La pila está vacía
search(stack,elem)	- El elemento no es un Number
firstElement(stack)	- La pila está vacía
lastElement(stack)	- La pila está vacía

Tabla 2 Excepciones funciones de pila

Cola

Implementa un tipo de datos **cola** que permita almacenar enteros sobre un array. La cola debe tener un número máximo de elementos el cuál no se podrá rebasar. Utiliza una constante para almacenar dicho límite.

Las funciones/operaciones que debes implementar son:

Valor de retorno	función	Descripción
Array	create	Crea una cola con el array ya instanciado con el número de elementos máximos.
Boolean	isEmpty(queue)	Devuelve true o false en función de si la cola está vacía.
Boolean	isFull(queue)	Devuelve true o false en función de si la cola está llena.
Number	size(queue)	Devuelve el número de elementos de la cola.
Number	add(queue,elem)	Añade un nuevo elemento a la cola. Devuelve el tamaño de la cola una vez añadido.
Number	poll(queue)	Devuelve el primer elemento de la cola sacándolo de ésta.
Number	peek(queue)	Igual que el anterior pero no elimina el elemento de la cola.
String	toString(queue)	Devuelve la cola en formato cadena. El delimitador de elementos será “-”.
Number	search(queue,elem)	Devuelve la posición del elemento indicado. Si el elemento no está en la cola devuelve -1.
Number	capacity(queue)	Devuelve el máximo número de elementos que podemos tener en la cola.
-	clear(queue)	Vacía la cola.
Number	firstElement(queue)	Devuelve el primer elemento de la cola
Number	lastElement(queue)	Devuelve el último elemento de la cola

Tabla 3 Operaciones cola

En cuanto a la gestión de errores. Ten en cuenta las excepciones que debería lanzar cada función.

Función	Excepción
add(queue,elem)	- El elemento no es un Number - La cola está llena
poll(queue)	- La cola está vacía
peek(queue)	- La cola está vacía
search(queue,elem)	- El elemento no es un Number
firstElement(queue)	- La cola está vacía
lastElement(queue)	- La cola está vacía

Tabla 4 Excepciones funciones de cola

Lista

Implementa un tipo de datos **lista** que permita almacenar enteros sobre un array. La lista debe tener un número máximo de elementos el cuál no se podrá rebasar. Utiliza una constante para almacenar dicho límite.

Las funciones/operaciones que debes implementar son:

Valor de retorno	función	Descripción
Array	create	Crea una lista con el array ya instanciado con el número de elementos máximos.
Boolean	isEmpty(list)	Devuelve true o false en función de si la lista está vacía.
Boolean	isFull(list)	Devuelve true o false en función de si la lista está llena.
Number	size(list)	Devuelve el número de elementos de la lista.
Number	add(list,elem)	Añade un nuevo elemento al final de la lista. Devuelve el tamaño de la lista una vez añadido.
Number	addAt(list,elem,index)	Añade un nuevo elemento en la posición especificada en la lista. Devuelve el tamaño de la lista una vez añadido.
Number	get(list,index)	Devuelve el elemento de la lista de la posición indicada.
String	toString(list)	Devuelve la cola en formato cadena. El delimitador de elementos será “-”.
Number	indexOf(list,elem)	Devuelve la posición del elemento indicado. Si el elemento no está en la lista devuelve -1.
Number	lastIndexOf(list,elem)	Devuelve la posición del elemento indicado comenzando por el final. Si el elemento no está en la lista devuelve -1.
Number	capacity(list)	Devuelve el máximo número de elementos que podemos tener en la lista.
-	clear(list)	Vacía la lista.
Number	firstElement(list)	Devuelve el primer elemento de la lista
Number	lastElement(list)	Devuelve el último elemento de la lista
Number	remove(list,index)	Elimina el elemento de la posición indicada. Devuelve el elemento borrado.
Boolean	removeElement(list,elem)	Elimina el elemento indicado de la lista. Devuelve true si se ha podido borrar el elemento, false en caso contrario.
Number	set(list,elem,index)	Reemplaza el elemento de la lista indicado por el índice. Devuelve el elemento que estaba anteriormente en la lista.

Tabla 5 Operaciones lista

En cuanto a la gestión de errores. Ten en cuenta las excepciones que debería lanzar cada función.

Función	Excepción
---------	-----------

add(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Number - La lista está llena
addAt(list,elem,index)	<ul style="list-style-type: none"> - El elemento no es un Number - La lista está llena - El índice está fuera de los límites de la lista
get(list,index)	<ul style="list-style-type: none"> - El índice está fuera de los límites de la lista
indexOf(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Number
lastIndexOf(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Number
firstElement(list)	<ul style="list-style-type: none"> - La lista está vacía
lastElement(list)	<ul style="list-style-type: none"> - La lista está vacía
remove(list,index)	<ul style="list-style-type: none"> - El índice está fuera de los límites de la lista
removeElement(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Number
set(list,elem,index)	<ul style="list-style-type: none"> - El elemento no es un Number - El índice está fuera de los límites de la lista

Tabla 6 Excepciones funciones de lista

Lista ordenada

Implementa un tipo de datos **lista ordenada** que permita almacenar enteros sobre un array.

Las características de este tipo de datos son las mismas que para el tipo de datos lista, salvo que los elementos tienen una relación de orden entre sí, la cual será una ordenación ascendente de los mismos.

Las funciones/operaciones serán las mismas que para la lista, aunque con algunas excepciones. Las siguientes funciones no tendrían sentido dada la relación de orden ya que podríamos romper dicha relación.

- addAt
- lastIndexOf
- set

El resto de funciones quedarían como siguen. Las funciones en negrita son las que han cambiado su funcionalidad.

Valor de retorno	función	Descripción
Array	create	Crea una lista con el array ya instanciado con el número de elementos máximos.
Boolean	isEmpty(list)	Devuelve true o false en función de si la lista está vacía.
Boolean	isFull(list)	Devuelve true o false en función de si la lista está llena.
Number	size(list)	Devuelve el número de elementos de la lista.
Number	add(list,elem)	Añade un nuevo elemento a la lista manteniendo la relación de orden. Devuelve el tamaño de la lista una vez añadido.
Number	get(list,index)	Devuelve el elemento de la lista de la posición indicada.
String	toString(list)	Devuelve la cola en formato cadena. El delimitador de elementos será "-".

Number	indexOf(list,elem)	Devuelve la posición del elemento indicado. Si el elemento no está en la lista devuelve -1.
Number	lastIndexOf(list,elem)	Devuelve la posición del elemento indicado comenzando por el final. Si el elemento no está en la lista devuelve -1.
Number	capacity(list)	Devuelve el máximo número de elementos que podemos tener en la lista.
-	clear(list)	Vacía la lista.
Number	firstElement(list)	Devuelve el primer elemento de la lista
Number	lastElement(list)	Devuelve el último elemento de la lista
Number	remove(list,index)	Elimina el elemento de la posición indicada. Devuelve el elemento borrado.
Boolean	removeElement(list,elem)	Elimina el elemento indicado de la lista. Devuelve true si se ha podido borrar el elemento, false en caso contrario.

Tabla 7 Operaciones lista ordenada

En cuanto a la gestión de errores quedaría de la siguiente manera.

Función	Excepción
add(list,elem)	- El elemento no es un Number - La lista está llena
get(list,index)	- El índice está fuera de los límites de la lista
indexOf(list,elem)	- El elemento no es un Number
firstElement(list)	- La lista está vacía
lastElement(list)	- La lista está vacía
remove(list,index)	- El índice está fuera de los límites de la lista
removeElement(list,elem)	- El elemento no es un Number

Tabla 8 Excepciones funciones de lista ordenada

Al tener una relación de orden entre los elementos, se puede mejorar la eficiencia de algunas funciones. Las funciones que habría que reimplementar son:

- indexOf
- removeElement