Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria de Ingeniería, Campus Zacatecas

Programación Orientada a Objetos

Actividad: Investigación

Alumno: Joshua Campos Haro

Profesor: Roberto Oswaldo Cruz Lejía

Ing. Sistemas Computacionales

Zacatecas Zac. 24/10/2019

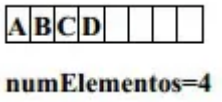
Introducción:

Las listas en Java son variables que permiten almacenar grandes cantidades de datos. Son similares a los Array o a las Matrices.

Ambas (ArrayList y LinkedList) implementan una interfaz genérica, la interfaz List.

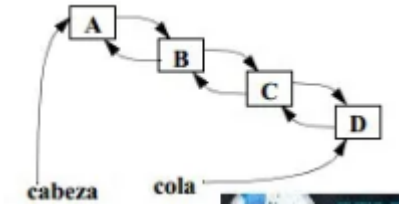
ArrayList: lista implementada con un array.

- Acceso posicional eficiente.
- Inserción y extracción costosas menos en la última posición que es instantánea.
- Cuando se supera el tamaño del array, se crea uno nuevo más grande y se copian en él los elementos del antiguo.



LinkedList: lista doblemente enlazada.

- Acceso posicional costoso
- Inserción y extracción costosas
- Menos en la primera y última posición que es inmediato
- Tamaño ilimitado



ArrayList:

Ventajas	Desventajas
Permite un acceso de lectura aleatorio rápido, para que puedas agarrar cualquier elemento en tiempo constante. Pero agregar o eliminar desde cualquier lugar menos el final requiere desplazar todos los últimos elementos, ya sea para abrir o llenar el espacio.	Un ArrayList no puede contener datos primitivos, sólo Objetos.
Lo implementa con una matriz de redimensionamiento dinámico.	La cantidad de memoria considera la capacidad definida para el ArrayList, aunque no contenga elementos.
Facilidad de Acceso a elementos que se requieran utilizar.	Costos adicionales al añadir o remover elementos

LinkedList:

Ventajas	Desventajas
Permite inserciones o eliminaciones de tiempo constante utilizando iteradores, pero solo acceso secuencial de elementos. En otras palabras, puede recorrer la lista hacia adelante o hacia atrás, pero encontrar un puesto en la lista lleva tiempo proporcional al tamaño de la lista.	Uso de memoria adicional por las referencias a los elementos anterior y siguiente.
LinkedList lo implementa con una lista doblemente vinculada.	El acceso a los elementos depende del tamaño de la lista.
Añadir y remover elementos al final de la lista.	

Métodos:

METODOS ESPECIFICOS DE ARRAYLIST:

Descripción	Interfaz
Añade `o` al final de la lista. Retorna `true`	`boolean add (E o)`
Busca el primer elemento de la lista igual a `o` y lo elimina. Retorna `true` si existe (para encontrar el objeto utiliza su método `equals`)	`boolean remove (E o)`
Reemplaza el elemento de la lista que ocupa la posición `index`, por `element`, y retorna el elemento que estaba ahí	`E set(int index, E element)`
Inserta `element` en la posición `index`, "desplazando" los elementos posteriores	`void add (int index, E element)`
Elimina (y retorna) de la lista el elemento que ocupa la posición `index`, "desplazando" los posteriores	`E remove (int index)`
Hace la lista vacía	`void clear()`

Descripción	Interfaz
Número de elementos de la lista	`int size()`
¿Está vacía?	`boolean isEmpty()`
Retorna true si la lista contiene a `o` al menos una vez	`boolean contains(Object o)`
Retorna el elemento de la lista que ocupa la posición `index`	`E get(int index)`
Búsqueda: retorna el índice de la primera aparición de `o`, o -1 si no existe	`int indexOf(Object o)`
Búsqueda: retorna el índice de la última aparición de `o`, o -1 si no existe	`int lastIndexOf(Object o)`

METODOS ESPECIFICOS DE LINKEDLIST:

Descripción	Interfaz
Retorna el primer elemento	`E getFirst()`
Retorna el último elemento	`E getLast()`
Elimina y retorna el primer elemento	`E removeFirst()`
Elimina y retorna el último elemento	`E removeLast()`
Añade o al principio de la lista	`void addFirst(E o)`
Añade o al final de la lista	`void addLast(E o)`

¿Cuál debo usar?

En la práctica la mayoría de las ocasiones es mejor usar ArrayList porque el tiempo de las operaciones y uso de memoria es menor que en LinkedList, de manera simple: la mejor opción para iniciar a usar listas ArrayList es la mejor opción**.**

Referencias:

http://www.enrique7mc.com/2016/07/diferencia-entre-arraylist-y-linkedlist/

https://es.stackoverflow.com/questions/172954/cuando-es-mejor-usar-linkedlist-y-cuando-arraylist

http://panamahitek.com/el-uso-de-listas-en-java/

https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html

https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html