



Instituto Politécnico Nacional
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE
INGENIERÍA
CAMPUS ZACATECAS

Programación Orientada a Objetos
Investigación -. LinkedList Vs ArrayList

Kassandra Rivera Carrillo
2019670148

Roberto Cruz Leija

Ing. Sistemas Computacionales

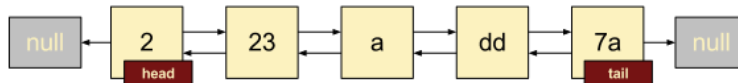
24 Octubre de 2019
2CM1

LinkedList Vs ArrayList

LinkedList y ArrayList son dos diferentes implementaciones de la interfaz List. LinkedList usa internamente una lista doblemente ligada, mientras que ArrayList usa un arreglo que cambia de tamaño dinámicamente.

Array vs. Linked List

Linked List



Array



LinkedList permite **eliminar e insertar elementos en tiempo constante** usando iteradores, pero el acceso es secuencial por lo que encontrar un elemento toma un tiempo proporcional al tamaño de la lista.

Normalmente la complejidad de esa operación promedio sería $O(n/2)$ sin embargo usar una lista doblemente ligada el recorrido puede ocurrir desde el principio o el final de la lista por lo tanto resulta en $O(n/4)$.

Por otro lado ArrayList ofrece **acceso en tiempo constante** $O(1)$, pero si quieres añadir o remover un elemento en cualquier posición que no sea la última es necesario mover elementos. Además si el arreglo ya está lleno es necesario crear uno nuevo con mayor capacidad y copiar los elementos existentes.

Tanto ArrayList como LinkedList tienen más métodos, en las tablas están algunos que podemos resaltar.

Complejidad

LinkedList

Operación	Complejidad Promedio
Get	$O(n/4)$

Operación	Complejidad Promedio
add(E element)	O(1)
add(int index, E element)	O(n/4)
remove(int index)	O(n/4)
Iterator.remove()	O(1)
ListIterator.add(E element)	O(1)

ArrayList

Operación	Complejidad Promedio
Get	O(1)
add(E element)	O(1)
add(int index, E element)	O(n/2)
remove(int index)	O(n/2)
Iterator.remove()	O(n/2)
remove(int index)	O(n/2)

Estos métodos son aplicables **a las clases genéricas** de Java LinkedList y ArrayList sin embargo estos tipos no son exclusivos de Java, es posible encontrar implementaciones en otros lenguajes de programación, por ejemplo en C# existen LinkedList y List.

Ventajas y desventajas

LinkedList

Ventajas	Desventajas
Añadir y remover elementos con un iterador	Uso de memoria adicional por las referencias a los elementos anterior y siguiente
Añadir y remover elementos al final de la lista	El acceso a los elementos depende del tamaño de la lista

ArrayList

Ventajas	Desventajas
Añadir elementos	Costos adicionales al añadir o remover elementos
Acceso a elementos	La cantidad de memoria considera la capacidad definida para el ArrayList, aunque no contenga elementos

[1]

Tanto ArrayList como LinkedList implementan la interfaz **List** y se encuentran en el paquete java.util y son parte de lo que se conoce como java collection framework.

La principal diferencia es un tema de implementación:

- ArrayList implementa la lista con arreglos linkeados, lo que la hace muy eficiente para recorrer pero ineficiente para insertar o eliminar elementos que se no encuentren en el final.
- LinkedList es una lista de elementos linkeados (en realidad es una lista doblemente enlazada) lo que la hace lenta para recorrer pero sumamente eficiente para insertar o eliminar elementos.

[2]

Interface list

Es importante mencionar la interface List ya que de aquí parten tanto la LinkedList como la ArrayList. Esta interface es la encargada de agrupar una colección de elementos en forma de lista, es decir, uno detrás de otro. En una lista los elementos pueden ser accedidos por un índice que indica la posición del elemento en la colección.

List

Esta interfaz también conocida como “secuencia” normalmente acepta elementos repetidos o duplicados, y al igual que los arrays es lo que se llama “basada en 0”. Esto quiere decir que el primer elemento no es el que está en la posición “1”, sino en la posición “0”.

Esta interfaz proporciona debido a su uso un iterador especial (la interfaz Iterator e Iterable las hemos podido conocer en anteriores entregas) llamada ListIterator. Este iterador permite además de los métodos definidos por cualquier iterador:

Que estos métodos son.-

-
- hasNext,
 - next
 - remove
-

Métodos para inserción de elementos y reemplazo, acceso bidireccional para recorrer la lista y un método proporcionado para obtener un iterador empezando en una posición específica de la lista.

Debido a la gran variedad y tipo de listas que puede haber con distintas características como permitir que contengan o no elementos null, o que tengan restricciones en los tipos de sus elementos, hay una gran cantidad de clases que implementan esta interfaz. Entre las más utilizadas:

- ArrayList.
- LinkedList.
- Stack.
- Vector.

[3]

ArrayList

java.util

Class ArrayList<E>

java.lang.Object

java.util.AbstractCollection<E>

java.util.AbstractList<E>

java.util.ArrayList<E>

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess

Direct Known Subclasses:

AttributeList, RoleList, RoleUnresolvedList

Constructors

Constructor and Description

`ArrayList()`

Constructs an empty list with an initial capacity of ten.

`ArrayList(Collection<? extends E> c)`

Constructs a list containing the elements of the specified collection, in the order they are returned by the collection's iterator.

`ArrayList(int initialCapacity)`

Constructs an empty list with the specified initial capacity.

LinkedList

java.util

Class LinkedList<E>

java.lang.Object

java.util.AbstractCollection<E>

java.util.AbstractList<E>

java.util.AbstractSequentialList<E>

java.util.LinkedList<E>

Type Parameters:

E - the type of elements held in this collection

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, Deque<E>, List<E>, Queue<E>

Constructors

Constructor and Description

`LinkedList()`

Constructs an empty list.

`LinkedList(Collection<? extends E> c)`

Constructs a list containing the elements of the specified collection, in the order they are returned by the collection's iterator.

Bibliografia

[1] Munjia Enrique, "Diferencias entre ArrayList y LinkedList" 19 de Julio, 2016 (Sitio Web) <http://www.enrique7mc.com/2016/07/diferencia-entre-arraylist-y-linkedlist/>

[2] Goette Emanuel "Diferencia entre ArrayList y LinkedList en Java" 1 de enero de 2018 <https://emanuelpeg.blogspot.com/2018/01/diferencia-entre-arraylist-y-linkedlist.html>

Link sugeridos:

- <https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html>
- <https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html>

[3] Derechos reservados, "Interface List del api java. Clases ArrayList, LinkedList, Stack, Vector. Ejemplo con ArrayList (CU00920C)" (Doc Web) https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=603:interface-list-del-api-java-clases-arraylist-linkedlist-stack-vector-ejemplo-con-arraylist-cu00920c&catid=58&Itemid=180

[4] <https://docs.oracle.com/javase/7/docs/api/>