

5 Difference Between ArrayList And LinkedList In Java With Example

One of the start up [java interview questions](#) on Collections topic is difference between ArrayList and LinkedList , interviewer may also ask to write examples . We already discussed some other basic interview questions like [difference between array and arraylist](#) , [difference between arraylist and vector](#) . In this post difference between arraylist and linkedlist , apart from the differences , we will also discuss the similarities , examples and when to prefer arraylist over linkedlist.

Difference between ArrayList and LinkedList in Java

WHATS HOT

- [Difference between Arraylist and Vector : Core Java Interview Collection Question](#)
- [Best Books for Learning Java](#)

1. Implementation : ArrayList is the resizable array implementation of list interface , while LinkedList is the Doubly-linked list implementation of the list interface.

2. Performance : Performance of ArrayList and LinkedList depends on the type of operation

a. get(int index) or search operation :

ArrayList get(int index) operation runs in constant time i.e $O(1)$ while LinkedList get(int index) operation run time is $O(n)$.

The reason behind ArrayList being faster than LinkedList is that ArrayList uses index based system for its elements as it internally uses array data structure , on the other hand , LinkedList does not provide index based access for its elements as it iterates either from the beginning or end (whichever is closer) to retrieve the node at the specified element index.

b. insert() or add(Object) operation :

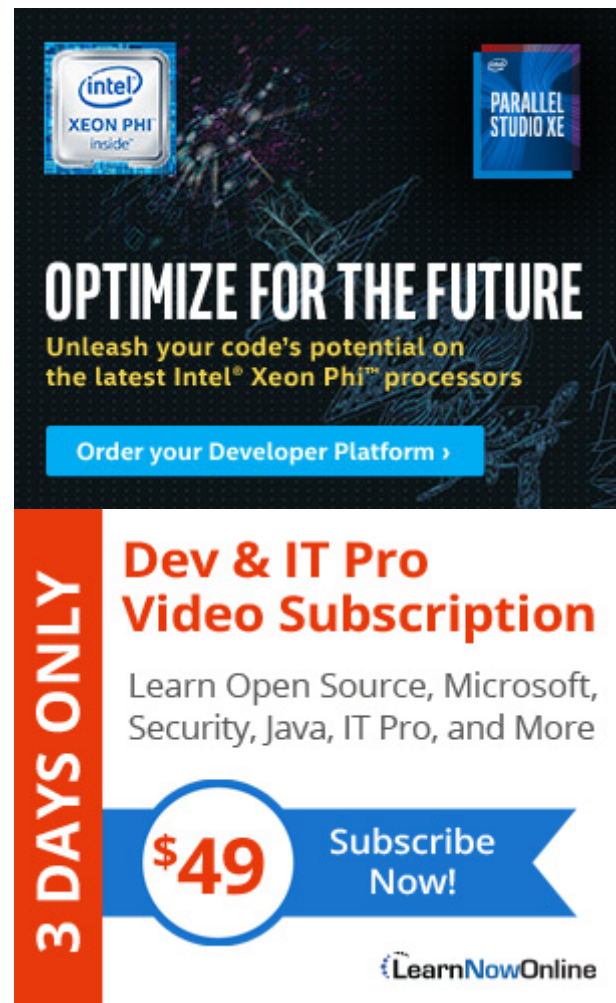
Insertions in LinkedList are generally fast as compare to ArrayList.

In LinkedList adding or insertion is $O(1)$ operation . While in ArrayList, if array is full i.e worst case, there is extra cost of resizing array and copying elements to the new array , which makes runtime of add operation in ArrayList $O(n)$, otherwise it is $O(1)$.

c. remove(int) operation : Remove operation in LinkedList is generally same as ArrayList i.e. $O(n)$.

In LinkedList , there are two overloaded

- Amazon Interview Question : First Non repeated character in String
- Count total number of times each alphabet appears in the string java program code with example
- Java 8 new features : Lambda expressions , optional class , Defender methods with examples



The advertisement features a dark background with a glowing network of nodes and lines. In the top left is the Intel Xeon Phi 'inside' logo. In the top right is the 'PARALLEL STUDIO XE' logo. The main text reads 'OPTIMIZE FOR THE FUTURE' in large white letters, followed by 'Unleash your code's potential on the latest Intel® Xeon Phi™ processors' in yellow. A blue button below says 'Order your Developer Platform >'. On the left, a vertical orange banner says '3 DAYS ONLY'. To the right, 'Dev & IT Pro Video Subscription' is written in red, with 'Learn Open Source, Microsoft, Security, Java, IT Pro, and More' in white below it. A blue ribbon contains '\$49' in white and 'Subscribe Now!' in white. The 'LearnNowOnline' logo is at the bottom right.

remove methods. one is remove() without any parameter which removes the head of the list and runs in constant time $O(1)$.

The other overloaded remove method in LinkedList is remove(int) or remove(Object) which removes the Object or int passed as parameter. This method traverses the LinkedList until it found the Object and unlink it from the original list. Hence this method run time is $O(n)$.

While in ArrayList remove(int) method involves copying elements from old array to new updated array, hence its run time is $O(n)$.

3. Reverse Iterator : LinkedList can be iterated in reverse direction using descendingIterator() while there is no descendingIterator() in ArrayList, so we need to write our own code to iterate over the ArrayList in reverse direction.

4. Initial Capacity : If the constructor is not overloaded, then ArrayList creates an empty list of initial capacity 10, while LinkedList only constructs the empty list without any initial capacity.

5. Memory Overhead : Memory overhead in LinkedList is more as compared to ArrayList as node in LinkedList needs to maintain the addresses of next and previous node. While in ArrayList each index only holds the actual object(data).

Example of ArrayList and LinkedList :



Javahungry

Like Page

2.8K likes

[Subscribe for Our Newsletter](#)

Improve legal document workflow with **auto-redaction, hit highlighting & metadata.**

Download free trial

accusoft

SpreadsheetGear®
Performance Spreadsheet Components

Microsoft Chose SpreadsheetGear

Chris Donohue
MSN Money Program Manager

Download Free Trial

POPULAR POSTS

- [Java Interview Questions](#)
- [Top 50 Java Collections Interview Questions and Answers](#)
- [What Makes You Stand out from the Crowd](#)
- [Threads , Lifecycle Explained with Example](#)

```
import java.util.ArrayList;
import java.util.LinkedList;

public class ArrayListLinkedListExample {

    public static void main(String[] args) {

        ArrayList<String> arrlistobj = new ArrayList<>();
        arrlistobj.add("1. Alive is awesome");
        arrlistobj.add("2. Love yourself");
        System.out.println("ArrayList object output :");

        LinkedList llobj = new LinkedList<>();
        llobj.add("1. Alive is awesome");
        llobj.add("2. Love yourself");
        System.out.println("LinkedList object output :");

    }
}
```

Count number of words in the String with
Example : Java Program Code

Output:

ArrayList object output :[1. Alive is
awesome, 2. Love yourself]
LinkedList object output :[1. Alive is
awesome, 2. Love yourself]

Similarities between ArrayList and LinkedList :

1. Not synchronized : Both ArrayList
and LinkedList are not synchronized ,
and can be made synchronized
explicitly using
Collections.synchronizedList() method.

2. clone() operation : Both ArrayList and LinkedList returns a shallow copy of the original object ,i.e. the elements themselves are not cloned.

3. Iterators : The iterators returned by ArrayList and LinkedList class's iterator and listIterator methods are fail-fast. Fail fast iterators throw ConcurrentModificationException . We have already discussed the **difference between fail-fast and fail-safe iterators**.



4.Insertion Order : As ArrayList and LinkedList are the implementation of List interface,so, they both inherit properties of List . They both preserves the order of the elements in the way they are added to the ArrayList or LinkedList object.

When to Use ArrayList and LinkedList :

In real world applications , you will more frequently use ArrayList than LinkedList. But in a very specific situations LinkedList can be preferred.

1. ArrayList is preferred when there are more `get(int)` or search operations need to be performed as every search operation runtime is $O(1)$.

2. If application requires more `insert(int)`, `delete(int)` operations then the `get(int)` operations then LinkedList is preferred as they do not need to maintain back and forth like arraylist to preserve continues indices.

Recap : Difference between ArrayList and LinkedList in Java

	ArrayList	LinkedList
Implementation	Resizable Array	Doubly-LinkedList
Reverseliterator	No	Yes , <code>descendingIterator()</code>
Initial Capacity	10	Constructs empty list
<code>get(int)</code> operation	Fast	Slow in comparision
<code>add(int)</code> operation	Slow in comparision	Fast
Memory Overhead	No	Yes

References : [Oracle ArrayList docs](http://javahungry.blogspot.com/2015/04/difference-between-arraylist-and-linkedlist-in-java-example.html) ,

Oracle LinkedList docs

So we have covered difference between arraylist and linkedlist in java .In case you have any other doubt please mention in the comments .

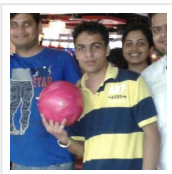
[Like 5](#)[Share](#)[Tweet](#)[Share](#)

0

- [Stun](#)



3



About The Author

Subham Mittal has worked in Oracle for 3 years .

For more java articles ,Click here to

[Subscribe JavaHungry](#)

You might also like

- [Difference Between == and equals method in Java with Example : Core Java Interview Question](#)
- [How to Convert Math Number to Equivalent Readable Word in Java : Code with Example](#)
- [STRING](#)
- [Difference Between String , StringBuilder and StringBuffer Classes with Example : Java](#)

Recommended by

Sponsored Links

Before & After: Parineeti Chopra's Fitness Secrets, Revealed

Food.ndtv.com

Experience Rummy like Never Before & Get a Chance to Win Rs.2 lakhs!

Junglee Rummy

Calculate: Expenses At 60 And Investment Offers For Planning

Big decisions

Hearing aids smaller and more powerful than ever

Soundrise

Experience Hong Kong like Sanjeeda and Aamir, For Free!

DiscoverHongKong.com

Mistry's Mystery Busted!

The Week

Comments

Community

 Login ▾

♥ Recommend

🔗 Share

Sort by Best ▾

Join the discussion...



Shweta Singh • 9 months ago

If the constructor is not overloaded , then ArrayList creates an empty list of initial capacity 10

No,It also have size 0 same as linked-list,I checked it from code.

Can you please give some example for initial capacity 10 of array list

^ | ▾ • Reply • Share ›



swati ➔ Shweta Singh
• 5 months ago

please check difference in size and capacity

^ | v • Reply • Share ›



PowerStar_Pawan → Shweta Singh • 9 months ago

hi shweta singh though it shows size as 0, internally it creates an array of size 10. To check that you can run the program in debug mode and check.

^ | v • Reply • Share ›



Amit Kukadiya • a year ago

thanks good article.

^ | v • Reply • Share ›

[Return to Top ↑](#)

Java Hungry Copyright © 2016 · All Rights Reserved. Designed by studiopress