**Java LinkedList**

* Java LinkedList is an implementation of the List and Deque interfaces.

C:\Users\hitendra.chaudhari\Downloads\linkedlist.png

* Internally, it is an implemented using Doubly Linked List Data Structure.

C:\Users\hitendra.chaudhari\Downloads\doubly-linked-list.png

* It supports duplicate elements.
* It stores or maintains it’s elements in Insertion order.
* We can add any number of null elements.
* It is not synchronised that means it is not Thread safe.
* We can create a synchronised LinkedList using Collections.synchronizedList() method.
* In Java applications, we can use it as a List, stack or queue.
* It does not implement RandomAccess interface. So we can access elements in sequential order only. It does not support accessing elements randomly.
* When we try to access an element from a LinkedList, searching that element starts from the beginning or end of the LinkedList based on where that elements is available.
* In Java LinkedList class, manipulation is fast because no shifting needs to be occurred.
* The iterators returned by this class's iterator and listIterator methods are *fail-fast*
* LinkedList  only constructs the empty list without any initial capacity.
* We can use ListIterator to iterate LinkedList elements.
* From Java SE 8 on-wards, we can convert a LinkedList into a Stream and vice-versa.

Constructors

|  |  |
| --- | --- |
| Constructor | Description |
| LinkedList() | It is used to construct an empty list. |
| LinkedList(Collection c) | It is used to construct a list containing the  elements of the specified collection, in the order  they are returned by the collection's iterator. |