Method	Operation
`boolean add(Object ob)`	Insert object `ob` at the end of the linked list.
`void add(int pos, Object ob)`	Insert object `ob` at position `pos` after shifting elements at positions following `pos` by one position; throw `IndexOutOfBoundsException` if `pos` is out of range.
`boolean addAll(Collection c)`	Add all the elements from the collection `c` to the end of the linked list; return `true` if the linked list was modified; throw `NullPointerException` if `c` is null.
`boolean addAll(int pos, Collection)`	Add all the elements from the collection `c` at the position `pos` of the linked list after shifting the objects following position `pos`; throw `IndexOutOfBoundsException` if `pos` is out of range, and `NullPointerException` if `c` is null.
`void addFirst(Object ob)`	Insert object `ob` at the beginning of the linked list.
`void addLast(Object ob)`	Insert object `ob` at the end of the linked list; same as `add(ob)`.
`void clear()`	Remove all the objects from the linked list.
`Object clone()`	Return the copy of the linked list without cloning its elements.
`boolean contains(Object ob)`	Return `true` if the linked list contains the object `ob`.
`boolean containsAll (Collection c)`	Return `true` if the linked list contains all of the objects in the collection `c`; throw `NullPointerException` if `c` is null (inherited).
`boolean equals(Object ob)`	Return `true` if the current linked list and object `ob` are equal (inherited).
`Object get(int pos)`	Return the object at position `pos`; throw `IndexOutOfBoundsException` if `pos` is out of range.
`Object getFirst()`	Return the first object in the linked list; throw `NoSuchElementException` if the linked list is empty.
`Object getLast()`	Return the first object in the linked list; throw `NoSuchElementException` if the linked list is empty.
`int hashCode()`	Return the hash code for the linked list (inherited).
`int indexOf(Object ob)`	Return the position of the first occurrence of object `ob` in the linked list; return –1 if `ob` is not found.
`boolean isEmpty()`	Return `true` if the linked list contains no elements, `false` otherwise (inherited).
`Iterator iterator()`	Generate and return an iterator for the linked list (inherited).
`int lastIndexOf(Object ob)`	Return the position of the last occurrence of object `ob` in the linked list; return –1 if `ob` is not found.
`LinkedList()`	Create an empty linked list.
`LinkedList(Collection c)`	Create a linked list with copies of elements from collection `c`; throw `NullPointerException` if `c` is null.
`ListIterator listIterator()`	Generate and return a list iterator for the linked list initialized to position 0 (inherited).

`ListIterator listIterator(int n)`	Generate and return a list iterator for the linked list initialized to position `n`; throw `IndexOutOfBoundsException` if `n` is out of range.
`boolean remove(Object ob)`	Remove the first occurrence of `ob` in the linked list and return `true` if `ob` was in the linked list.
`Object remove(int pos)`	Remove the object at position `pos`; throw `IndexOutOfBoundsException` if `pos` is out of range.
`boolean removeAll(Collection c)`	Remove from the linked list all the objects contained in collection `col`; return `true` if any element was removed; throw `NullPointerException` if `c` is null (inherited).
`Object removeFirst()`	Remove and return the first object on the linked list; throw `NoSuchElementException` if the linked list is empty.
`Object removeLast()`	Remove and return the last object on the linked list; throw `NoSuchElementException` if the linked list is empty.
`void removeRange(int first, int last)`	Remove from the linked list all the objects from position `first` to position `last`–1 (inherited).
`boolean retainAll(Collection c)`	Remove from the linked list all objects that are not in the collection `c`; return `true` if any object was removed; throw `NullPointerException` if `c` is null (inherited).
`Object set(int pos, Object ob)`	Assign object `ob` to position `pos` and return the object that occupied this position before the assignment; throw `IndexOutOfBoundsException` if `pos` is out of range.
`int size()`	Return the number of objects in the linked list.
`List subList(int first, int last)`	Return the sublist of the linked list (not its copy) containing elements from `first` to `last`–1; throw `IndexOutOfBoundsException` if either `first` or `last` and `IllegalArgumentException` if `last < first` (inherited).
`Object[] toArray()`	Copy all objects from the linked list to a newly created array and return the array.
`Object[] toArray(Object a[])`	Copy all objects from the linked list to the array `a` if `a` is large enough or to a newly created array and return the array; throw `ArrayStoreException` if type of `a` is not a supertype of the type of every element in the linked list and `NullPointerException` if `a` is null.
`String toString()`	Return a string representation of the linked list that contains the string representation of all the objects.