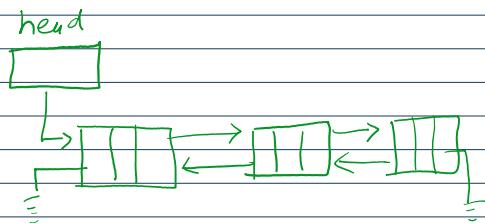


## Linked List

10 February 2025 19:24

1. The underlying data structure is doubly linked list.
2. The insertion is preserved.
3. Duplicate objects are allowed.
4. Heterogeneous objects are allowed.
5. Null insertion is possible.
6. `LinkedList` implements `Serializable` and `Cloneable` interfaces but it will not implement `RandomAccess`.
7. `LinkedList` is the best choice if our frequent operation is insertion or deletion in middle. `LinkedList` is the worst choice if our frequent operation is retrieval operation.



Constructors:-

```
LinkedList l = new LinkedList()  
LinkedList l = new LinkedList(Collection c); => Creates an equivalent  
linked list object for the given collection.
```

Usually we can use linked list to develop stacks and queues. to provide support for this requirement, linked list class defines the following specific methods:

```
void addFirst(Object o);  
void addLast(Object o);  
Object getFirst();  
Object getLast();  
Object removeFirst();  
Object removeLast();
```

Link list → Stack  
→ Queue (Circular Queue)

Difference between `ArrayList` and `LinkedList`?

ArrayList	LinkedList
Add a list is the best choice if our frequent operation is retrieval operation.	Linked list is the best choice if our frequent operation is insertion or deletion in the middle.
<code>ArrayList</code> is the worst choice of our frequent insertion or deletion in the middle, because it internally performs several shift operations.	<code>Link list</code> is the worst choice if our frequent operation is retrieval operation.
In addition, the element will be stored in consecutive memory location, and hence retrieval operation will become easy.	In linked list elements won't be stored in consecutive memory location. And hence, retrieval operation is complex or difficult.