LinkedList

I've just talked in detail about how the LinkedList, and the ArrayList, are different under the covers.

An ArrayList is implemented on top of an array, but a LinkedList is a doubly linked list.

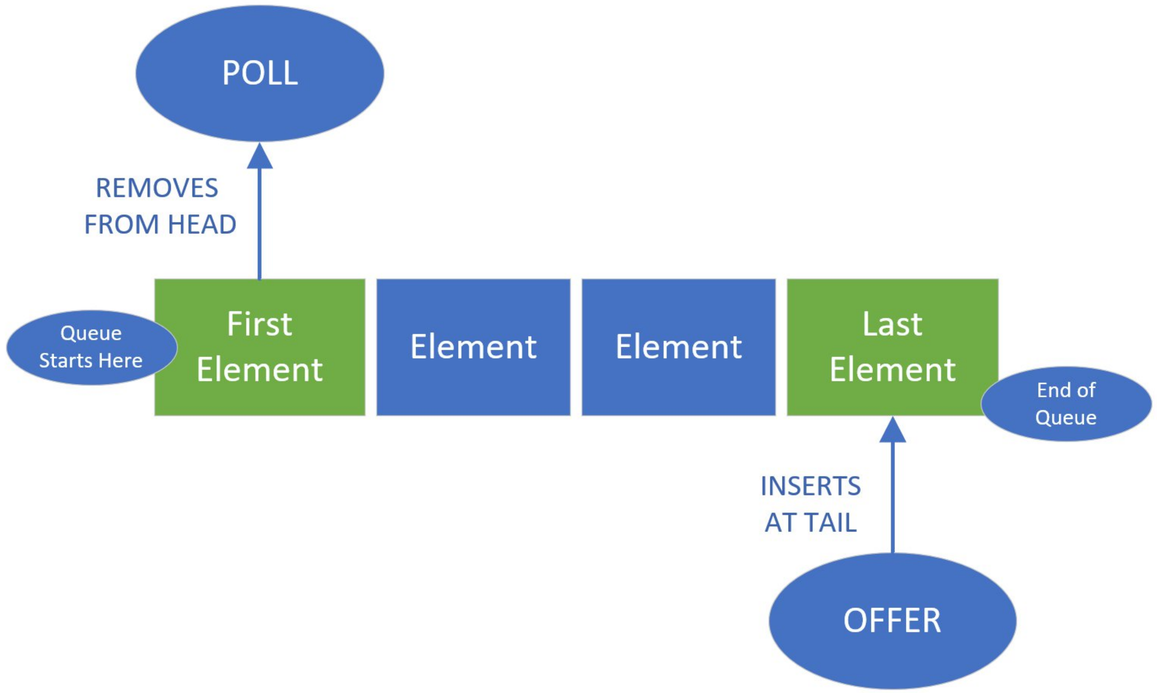
Both implement all of List's methods, but the LinkedList also implements the Queue and Stack methods as well.

A Queue is a First-In, First-Out (FIFO) Data Collection

When you think of a queue, you might think of standing in line.

When you get in a line or a queue, you expect that you'll be processed, in relationship to the first person in line.

We call this a First-in First-out, or FIFO data collection.

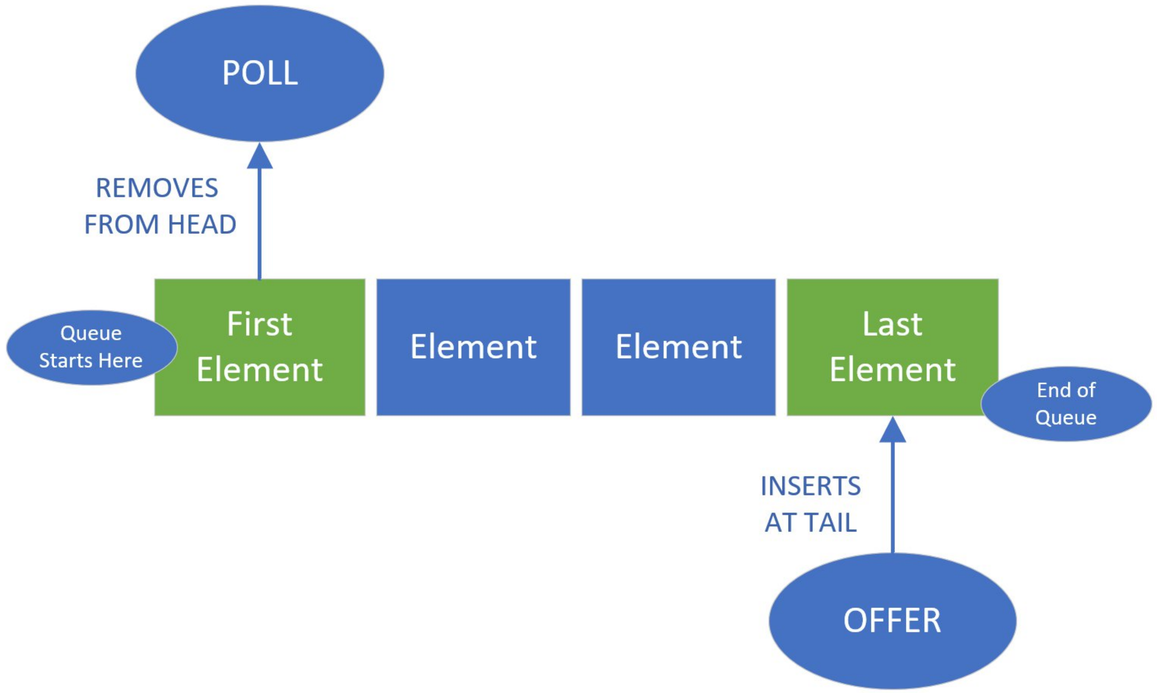


A Queue is a First-In, First-Out (FIFO) Data Collection

If you want to remove an item, you poll the queue, getting the first element or person in the line.

If you want to add an item, you offer it onto the queue, sending it to the back of the line.

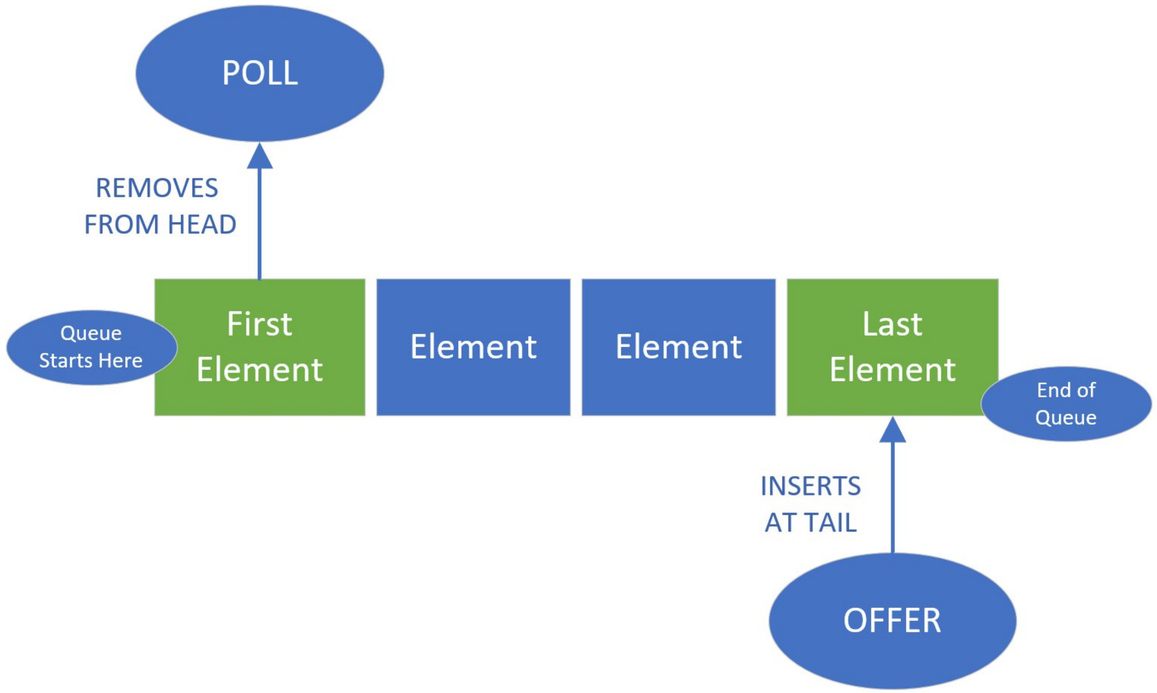
Single-ended queues always process elements from the start of the queue.



A Queue is a First-In, First-Out (FIFO) Data Collection

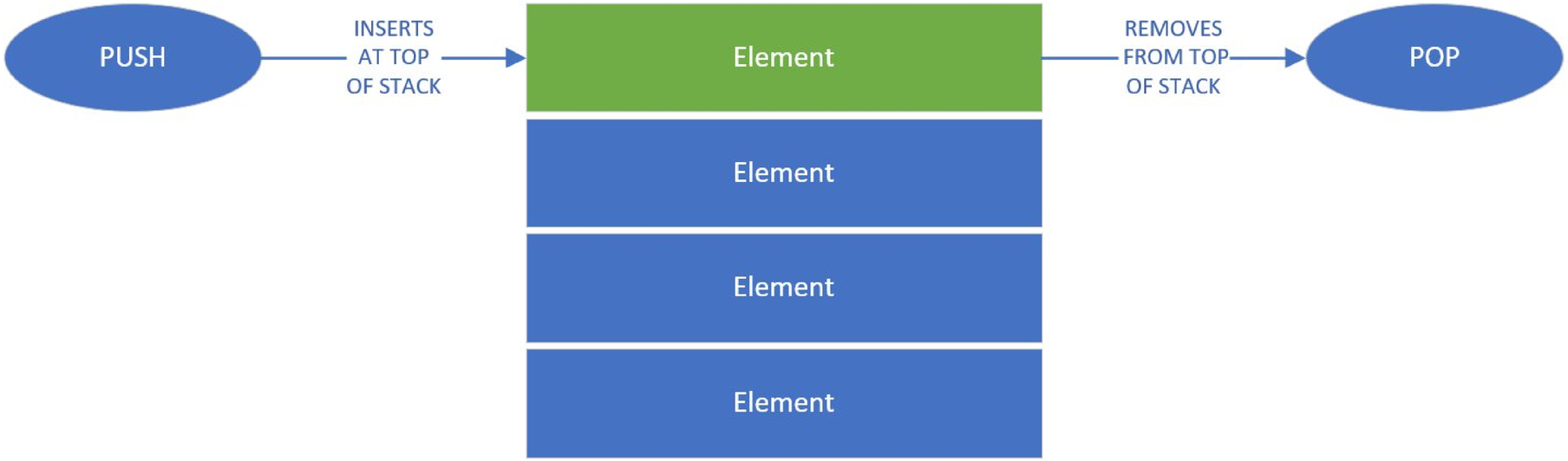
A double-ended queue allows access to both the start and end of the queue.

A LinkedList can be used as a double ended queue.



A Stack a is Last-In, First-Out (LIFO) Data Collection

When you think of a stack, you can think of a vertical pile of elements, one on top of another, as we show on this slide.



When you add an item, you push it onto the stack.

If you want to get an item, you'll take the top item, or pop it from the stack.

We call this a Last-In First-out, or LIFO data collection.

A LinkedList can be used as a stack as well.