**LinkedList** and **ArrayList** are two different implementations of the **List interface**. **LinkedList** **implements** it with **a** **doubly-linked list**. **ArrayList** **implements** it with **a** **dynamically re-sizing array**.

**LinkedList** allows for **constant-time insertions or removals** using iterators, but only **sequential access of elements**. In other words, we can walk the list forward or backward, but finding a position in the list takes time proportional to the size of the list.

**ArrayList** on the other hand allows **fast random read access**, so you can **grab any element in constant time**. But **adding or removing** from anywhere but the end **requires shifting all the latter elements over**, either to make an opening or fill the gap. Also, if we **add more elements than the capacity** of the underlying array, **a new array (1.5 times the size) is allocated**, and the **old array is copied to the new one**.