1. The time complexity (big O notation) for the "get" operation in linked list depends on the position of the element you want to retrieve. The best case is O(1) and the worst case is O(n)
2. A close-up of a code

   Description automatically generated

nodeRef.getNext

nodeRef.getData();

head

1. ArrayList: Utilizes an internal array, enabling rapid access to elements through index. Appending or removing elements at the end is efficient (O(1)), but operations in the middle involve shifting subsequent elements, resulting in a time complexity of O(n).

LinkedList: Organized as a series of nodes, with each node holding a reference to the next node in the sequence. Adding or removing elements at any position is efficient (O(1)), as it only requires adjusting references. However, accessing elements by index is less efficient (O(n)) due to the need for traversing the list.