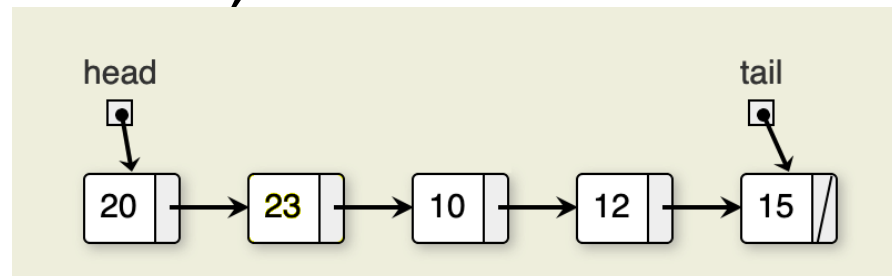


LinkedLists

The List Abstract Data Type

- List: “**ordered**” sequence of data known as elements
- Ordered: each element has a position in the list (like in an array from 0 to the length of the list -1)
- Ordered in this context does **not** mean that the list elements are sorted by value
- A list is said to be **empty** when it contains no elements
- The number of elements currently stored is called the **length** of the list
- The beginning of the list is called the **head**, the end of the list is called the **tail**



Defining the List ADT

This list will contain String object

```
public interface List { // List of String class ADT
    // Remove all contents from the list, so it is once again empty
    public void clear();
    // Insert "it" at the position index in this list.
    // throws IndexOutOfBoundsException - if the index is out of range (index < 0 || index > length)
    public boolean insert(int index, String it);
    // Append "it" at the end of the list
    public boolean append(String it);
    // Removes and return the element at the specified position in this list
    // throws IndexOutOfBoundsException - if the index is out of range (index < 0 || index >= length)
    public String remove(int index);
    // Returns the element at the specified position in this list
    // throws IndexOutOfBoundsException - if the index is out of range (index < 0 || index >= length)
    public String get(int index);
    // Returns true if this list contains the specified element. The empty String otherwise
    public boolean contains(String o);
    // Returns the length / number of elements in this list
    public int size();
    // Returns true if this list is empty
    public boolean isEmpty();
}
```

LinkedList

- Implementation of the List ADT
- Uses Linked Nodes to store data
- Node class is hidden inside the LinkedList class
 - The node class is implemented inside the LinkedList class
 - The Node class is the **inner class** and LinkedList is the **wrapper class**
 - The wrapper class has access to the data fields and methods of the inner class
 - The node class is declared as “private”

Note: An inner class is exactly like classes implemented in their own files.

LinkedList implementation

- Coding