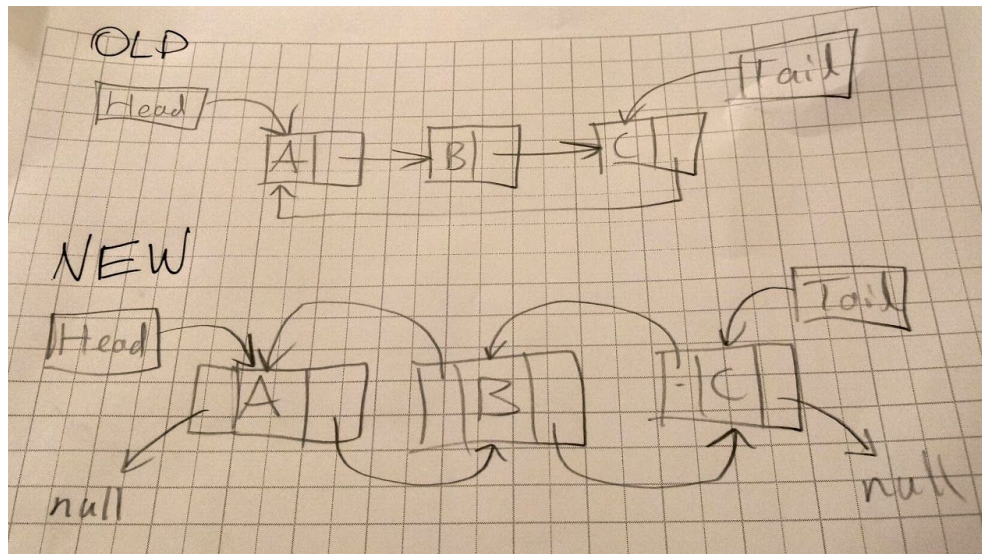


## Final Design of the Linked Lists



### BasicDoubleLinkedList.java

addToFront(T data)

- Create newNode
- newNode.prev = null
- newNode.next = firstNode.next
- firstNode = newNode
- size++

addToEnd(T data)

- Create newNode
- newNode.prev = lastNode.prev
- newNode.next = null
- lastNode = newNode
- size++

getFirst()

- Get firstNode

getLast()

- Get lastNode

getSize()

- Get size

retrieveFirstElement()

- Create tempNode = firstNode
- firstNode = firstNode.next
- firstNode.prev = null

- Return tempNode

retrieveLastElement()

- Create tempNode = lastNode
- lastNode = lastNode.prev
- lastNode.next = null
- Return tempNode

iterator()

- 

remove(T target, Comparator comparator)

- 

### **SortedDoubleLinkedList.java**

extends BasicDoubleLinkedList.java

add(T data)

- Is the list empty?
  - Add the item to the list
- Does it belong first?
  - Add as first
    - newNode.next = firstNode;
    - firstNode = newNode;
- Does it belong last?
  - Add as last
    - newNode.previous = lastNode
    - lastNode = newNode
- If it isn't either, we need to find what spot to put it in
  - While (C.compare(data, searchNode.element) > 0 )
    - searchNode = searchNode.next
  - newNode.previous = searchNode
  - newNode.next = searchNode.next
  - searchNode.next = newNode

addToFront(T data)

- Throws UnsupportedOperationException

addToEnd(T data)

- Throws UnsupportedOperationException

iterator()

- Calls super

remove(T data, Comparator comparator)

- Calls super