

LinkedList

Add and implement the following functions in LinkedList class. You can't change Element and LinkedListBase class. You can't add any instance variables in LinkedList class.

1. ***void add(Element e)***: Appends the specified element to the end of this list.
2. ***void add(int index, Element element)***: Inserts the specified element at the specified position in this list.
3. ***void addFirst(Element e)***: Inserts the specified element at the beginning of this list.
4. ***void addLast(Element e)***: Appends the specified element to the end of this list.
5. ***void clear()***: Removes all of the elements from this list.
6. ***boolean contains(Element e)***: Returns true if this list contains the specified element.
7. ***Element get(int index)***: Returns the element at the specified position in this list.
8. ***Element getFirst()***: Returns the first element in this list.
9. ***Element getLast()***: Returns the last element in this list.
10. ***int indexOf(Element e)***: Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
11. ***int lastIndexOf(Element e)***: Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.
12. ***Element remove(int index)***: Removes the element at the specified position in this list.
13. ***boolean remove(Element e)***: Removes the first occurrence of the specified element from this list, if it is present.
14. ***Element removeFirst()***: Removes and returns the first element from this list.
15. ***boolean removeFirstOccurrence(Element e)***: Removes the first occurrence of the specified element in this list (when traversing the list from head to tail).
16. ***Element removeLast()***: Removes and returns the last element from this list.
17. ***boolean removeLastOccurrence(Element e)***: Removes the last occurrence of the specified element in this list (when traversing the list from head to tail).
18. ***int size()***: Returns the number of elements in this list.
19. ***boolean isEmpty()***: Returns whether the list is empty or not.
20. ***String toString()***: Returns a string representation of this list, containing the String representation of each element separated by comma(,).