链滴

LinkedList

作者：helly

原文链接：https://ld246.com/article/1510288492006

来源网站：链滴

java.util.LinkedList

一、特点

1、允许元素为空;

2、允许元素重复;

3、元素有序;

4、非线程安全。

二、源码

```
public class LinkedList<E> extends AbstractSequantialList<E> implements List<E>, Deque<E>
 cloneable, java.io.Serializable {

 /*
  * **************************节点***********************************
  */
 private static class Node<E> {
   E item;
   Node<E> prev;
   Node<E> next;

   Node(Node<E> prev, E element, Node<E> next) {
     this.prev = prev;
     this.item = element;
     this.next = next;
   }
 }

 /*
  * **************************字段***********************************
  */
 transient int size;
 transient Node<E> first;
 transient Node<E> last;

 /*
  * **************************构造器***********************************
  */
 public LinkedList() {}

 public LinkedList(Collection<? extends E> c) {
   this();
   addAll(c);
 }

 /*
  * **************************增***********************************
  */

  // ********************实现List接口的方法******************************

  // 在链表尾部添加一个元素
```

```
public boolean add(E e) {
  linkLast(e);
  return true;
}

void llinkLast(E e) {
  Node<E> l = last;
  Node<E> newNode = new Node(l, e, null);

  last = newNode;
  if (l == null) {
    first = newNode;
  } else {
    l.next = newNode;
  }
  size++;
  modCount++;
}

// 在指定的位置插入一个元素
public void add(int index, E e) {
  checkPositionIndex(index); // 范围是 0-size，如果是size相当于在尾部添加元素

  if (index == size) {
    linkLast(e);
  } else {
    linkBefore(e, node(index));
  }
}

void linkBefore(E e, Node<E> succ) {
  final Node<E> prev = succ.prev;
  final Node<E> newNode = new Node(prev, e, succ);

  succ.prev = newNode;
  if (prev == null) {
    first = newNode; // !!!
  } else {
    prev.next = newNode;
  }
  size++;
  modCount++;
}

// 在链表尾部添加一个集合
public boolean addAll(Collection<? extends E> c) {
  return addAll(size, c);
}

// 在指定的位置插入一个集合
public boolean addAll(int index, Collection<? extends E>) {
  checkPositionIndex(index);

  Object[] a = c.toArray();
```

```
    int numNew = a.length;
    if (length == 0) {
      return false;
    }

    Node<E> pred;
    Node<E> succ;
    if (index == size) {
      pred = last;
      succ = null;
    } else {
      succ = node(index);
      pred = succ.prev;
    }

    for (Object o : a) {
      @SuppressWarnings("unchecked")
      E e = (E) o;
      Node<E> newNode = new Node(pred, e, null);

      if (pred == null) {
        first = newNode;
      } else {
        pred.next = newNode;
      }
      pred = newNode;
    }

    if (succ == null) {
      last = pred;
    } else {
      pred.next = succ;
      succ.prev = pred;
    }

    size += numNew;
    modCount++;
    return true;
  }

  // ********************实现Deque接口的方法*******************************

  // 在链表头部插入一个元素
  public  void  addFirst(E e)

  // 在链表尾部添加一个元素
  public  void  addLast(E e)

  // 在链表尾部添加一个元素（返回的是boolean）
  public  boolean  offer(E e)

  // 在链表头部插入一个元素（返回的是boolean）
  public  boolean  offerFirst(E e)
```

```
// 在链表尾部添加一个元素（返回的是boolean）
public boolean offerLast(E e)

/*
 * ************************删*************************************
 */


/*
 * ************************查*************************************
 */

 // 获取指定位置元素
 public E get(int index) {
   checkElementIndex(index);
   return node(index).item;
 }


/*
 * ************************改*************************************
 */

/*
 * ************************辅助方法*********************************
 */

 // 检查下标是否越界
 private void checkPositionIndex(int index) {
   if (!isPositionIndex(index)) {
     throw new IndexOutOfBoundsException(OutOfBoundsMsg(index));
   }
 }

 // 检查下标是否存在


 // 获取指定下标的节点
 Node<E> node(int index) {
  if (index < (size >> 1)) {
    Node<E> x = first;
    for (int i = 0; i < index; i++) {
      x = x.next;
    }
  } else {
    Node<E> x = last;
    for (int i = size - 1; i > index; i--) {
      x = x.prev;
    }
  }
}

/*
 * ************************迭代器*********************************
```

```
    */
}
```

LinkedList不是线程安全的，如果想使LinkedList变成线程安全的，可以使用如下方式：

```
List list=Collections.synchronizedList(new LinkedList(...));
```

参考：

LinkedList源码

http://www.importnew.com/25023.html

http://blog.csdn.net/qq_19431333/article/details/54572876