单链表的特点：

1.灵活，可以根据需求进行储存空间的动态分配，不需跟顺序表一样按照逻辑结构来定义存储结构

2.插入删除效率高

3.空间利用率低，适用于实现动态的线性表。

4.便于扩充

功能和实现就不详细介绍了，因为也比较熟了。

LinkedList

简介：

1.LinkedList 是一个继承于AbstractSequentialList类的双向链表。

2. LinkedList 实现 List 接口，能对它进行队列操作。

3. LinkedList 实现 Deque 接口，即能将LinkedList当作双端队列使用。

4. LinkedList 实现了Cloneable接口，即覆盖了函数clone()，能克隆。

5. LinkedList 实现java.io.Serializable接口，这意味着LinkedList支持序列化，能通过序列化去传输。

AbstractSequentialList 实现了get(int index)、set(int index, E element)、add(int index, E element) 和 remove(int index)这些函数。这些接口都是随机访问List的，LinkedList是双向链表。

特点：

1.LinkedList实际是通过双向链表实现的。他内部含有一个非常重要的内部类：Entry。Entry是双向链表节点对应的数据结构，它的属性有：当前结点的数据、上下节点的指针。

2.LinkedList不存在容量不足的问题。

3.LinkedList的克隆函数，即是将全部元素克隆到一个新的LinkedList对象中。

4. LinkedList实现了Deque，而Deque接口定义了在双端队列两端访问元素的方法。提供插入、移除和检查元素的方法。每种方法都存在两种形式：一种形式在操作失败时抛出异常，另一种形式返回一个特殊值（null 或 false，具体取决于操作）。

第一个元素（头部） 最后一个元素（尾部）

抛出异常 特殊值 抛出异常 特殊值

插入 addFirst(e) offerFirst(e) addLast(e) offerLast(e)

移除 removeFirst() pollFirst() removeLast() pollLast()

检查 getFirst() peekFirst() getLast() peekLast()

5. LinkedList可以作为FIFO(先进先出)的队列.

队列方法 等效方法

add(e) addLast(e)

offer(e) offerLast(e)

remove() removeFirst()

poll() pollFirst()

element() getFirst()

peek() peekFirst()

6. LinkedList可以作为LIFO(后进先出)的栈

栈方法 等效方法

push(e) addFirst(e)

pop() removeFirst()

peek() peekFirst()

7.遍历见之前学习文档