

java中LinkedList集合类实现栈和队列 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/145/2021\\_2022\\_java\\_E4\\_B8\\_ADLin\\_c104\\_145579.htm](https://www.100test.com/kao_ti2020/145/2021_2022_java_E4_B8_ADLin_c104_145579.htm) 栈和队列是两种特殊的线性表，它们的逻辑结构和线性表相同，只是其运算规则较线性表有更多的限制，故又称它们为运算受限的线性表。LinkedList数据结构是一种双向的链式结构，每一个对象除了数据本身外，还有两个引用，分别指向前一个元素和后一个元素，和数组的顺序存储结构（如：ArrayList）相比，插入和删除比较方便，但速度会慢一些。栈的定义 栈（Stack）是限制仅在表的一端进行插入和删除运算的线性表。（1）通常称插入、删除的这一端为栈顶（Top），另一端称为栈底（Bottom）。（2）当表中没有元素时称为空栈。（3）栈为后进先出（Last In First Out）的线性表，简称为LIFO表。栈的修改是按后进先出的原则进行。每次删除（退栈）的总是当前栈中“最新”的元素，即最后插入（进栈）的元素，而最先插入的是被放在栈的底部，要到最后才能删除。实现代码：

```
package com.gc.list.import
java.util.*.public class MyStack { private LinkedList ll=new
LinkedList(). public void push(Object o){ ll.addFirst(o). } public
Object pop(){ return ll.removeFirst(). } public Object peek(){ return
ll.getFirst(). } public boolean empty(){ return ll.isEmpty(). } public
static void main(String[] args){ MyStack ms=new MyStack().
ms.push("zhangsan"). ms.push("lisi"). ms.push("wangwu").
System.out.println(ms.pop()). System.out.println(ms.peek()).
System.out.println(ms.pop()). System.out.println(ms.empty()). }}

```

队列定义 队列（Queue）是只允许在一端进行插入，而在另

一端进行删除的运算受限的线性表 (1) 允许删除的一端称为队头 (Front)。(2) 允许插入的一端称为队尾 (Rear)。(3) 当队列中没有元素时称为空队列。(4) 队列亦称作先进先出 (First In First Out) 的线性表, 简称为FIFO表。实现代码:

```
package com.gc.list.import java.util.*.public class MyQueue {
private LinkedList ll=new LinkedList(). public void put(Object o){
ll.addLast(o). } //使用removeFirst()方法, 返回队列中第一个数据, 然后将它从队列中删除 public Object get(){ return
ll.removeFirst(). } public boolean empty(){ return ll.isEmpty(). }
public static void main(String[] args){ MyQueue mq=new
MyQueue(). mq.put("zhangsan"). mq.put("lisi").
mq.put("wangwu"). System.out.println(mq.get()).
System.out.println(mq.get()). System.out.println(mq.get()).
System.out.println(mq.empty()). }} 100Test 下载频道开通, 各类
考试题目直接下载。详细请访问 www.100test.com
```