

JAVA基础:语言中链表和双向链表的实现(1) 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\\_E5\\_9F\\_BA\\_E7\\_A1\\_80\\_c104\\_145491.htm](https://www.100test.com/kao_ti2020/145/2021_2022_JAVA_E5_9F_BA_E7_A1_80_c104_145491.htm) 链表是一种重要的数据结构, 在程序设计中占有很重要的地位。C语言和C++语言中是用指针来实现链表结构的, 由于JAVA语言不提供指针, 所以有人认为在JAVA语言中不能实现链表, 其实不然, JAVA语言比C++更容易实现链表结构。JAVA语言中的对象引用实际上是一个指针(本文中的指针均为概念上的意义, 而非语言提供的数据类型), 所以我们可以编写这样的类来实现链表中的结点。

```
class Node { Object data; Node next; // 指向下一个结点 }
```

将数据域定义成Object类是因为Object类是广义超类(所有类的祖先), 任何类对象都可以给其赋值, 增加了代码的通用性。为了使链表可以被访问还需要定义一个表头, 表头必须包含指向第一个结点的指针和指向当前结点的指针。为了便于在链表尾部增加结点, 还可以增加一指向链表尾部的指针, 另外还可以用一个域来表示链表的大小, 当调用者想得到链表的大小时, 不必遍历整个链表, 下图是这种链表的示意图。我们可以用类List来实现链表结构, 用变量Head、Tail、Length、Pointer来实现表头。存储当前结点的指针时有一定的技巧, Pointer并非存储指向当前结点的指针, 而是存储指向它的前趋结点的指针, 当其值为null时表示当前结点是第一个结点。那么我们为什么要这样做呢? 这是因为当我们删除当前结点后仍需保证剩下的结点构成链表, 如果Pointer指向当前结点, 则会给操作带来很大困难。那么如何得到当前结点呢, 我们定义了一个方法cursor(), 返回值是指向当前结点的

指针。类List还定义了一些方法来实现对链表的基本操作，通过运用这些基本操作我们可以对链表进行各种操作。例如reset()方法使第一个结点成为当前结点。insert( Object d )方法在当前结点前插入一个结点，并使其成为当前结点。remove()方法删除当前结点同时返回其内容，并使其后继结点成为当前结点，如果删除的是最后一个结点，则第一个结点变为当前结点。链表类List的源代码如下：

```
import java.io.*; public class List { /* 用变量来实现表头 */ private Node Head=null; private Node Tail=null; private Node Pointer=null; private int Length = 0; public void deleteAll() /* 清空整个链表 */ { Head = null; Tail = null; Pointer = null; Length = 0; } public void reset() /* 链表复位，使第一个结点成为当前结点 */ { Pointer = null; } public boolean isEmpty( ) /* 判断链表是否为空 */ { return( Length == 0 ); } public boolean isEnd() 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
```