线性表

接下来的四章是我们在编程过程中经常遇到的一些数据结构，包含了线性表，栈，队列，树和二叉树以及各种排序算法。认真学习，反复琢磨。
本笔记来自于李刚《Java程序员的基本修养》

- 单链表
- 双链表
- 循环链表
- Java中的线性表实现

1: 什么是线性表

在Java中，数据之前相互存在一定的关系。且数据元素之前基本的关系不外乎下面四种情况：

- 数据元素不存在关系，只是单纯的“同属于一个集合”；
- 数据元素之间存在一对一的关系，即线性关系；
- 数据元素之间存在一对多的关系，即树状关系；
- 数据元素之间存在多对多的关系，即图状关系；

根据存储逻辑的不同，分为：

- 顺序存储
- 链表存储

线性表是包含着一系列具有相同性质元素的优先序列，这些元素不仅仅限于基本数据类型，可能还包含着复合类型。线性表的基本操作包括：

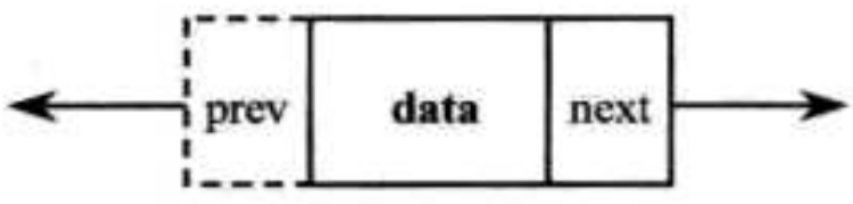
- 构造一个空表
- 返回线性表长度
- 在表头或者表中的位置添加一个新元素，表长度+1
- 在表尾或者表中任意位置删除一个元素，表长度-1；
- 判断表是不是空表
- 清空线性表
- 根据元素下表查询元素值或者根据元素值查找元素在表中的位置，如果不存在返回-1；

1.1: 线性表的顺序存储结构以及实现

在上面我们提到，对于线性表的实现，有两种方法，**顺序存储以及链式存储。**先来看看比较简单的顺序存储。
顺序存储的数据元素之间的物理关系（在内存中的地址）和逻辑关系（线性表中的相对顺序）是一致的。通常在Java源码中，使用数组来维护一个线性表，这样充分利用了数组高效查询的作用，但是对于插入和删除就比较复杂，需要整体移动数组。**值得注意的还有，因为线性表的顺序存储是用数组维护，因此在插入元素过程中，线性表的长度不能大于数组长度，否则就要对数组进行扩容，然后再进行插入。**

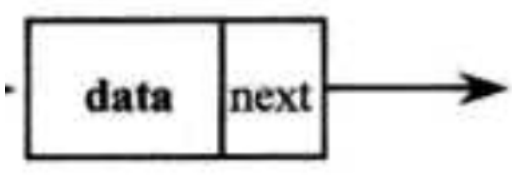
1.2: 线性表对链式存储以及实现

链式存储的基本结构是一个链式节点，如下图（这是一个双向链表节点表示）：

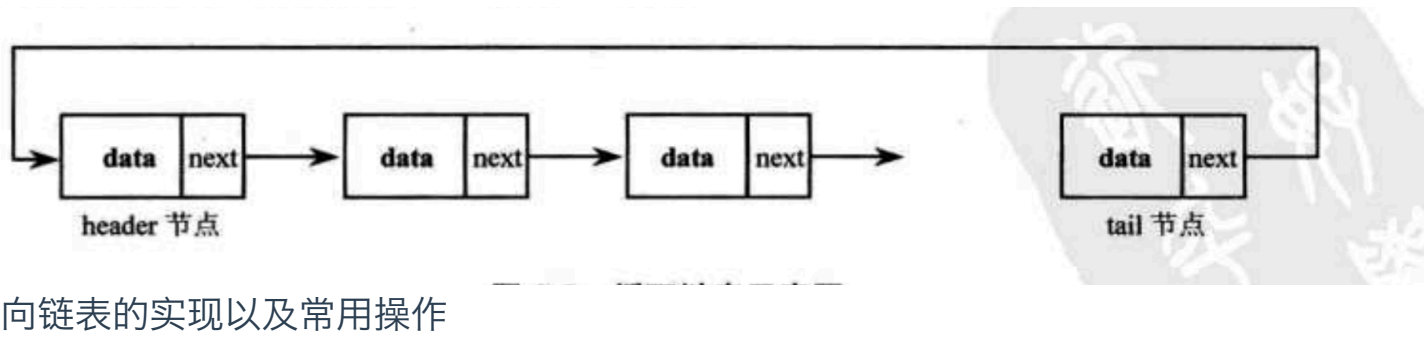


其中prev指向当前节点前一个节点地址，next指向当前节点的下一个节点地址。链式存储的物理关系和逻辑关系并不一致，这样可以充分利用计算机内存。其优点是插入元素和删除元素比较方便，但是对于查询就比较复杂。同时因为每个节点存储的数据较多，因此对于内存的消耗也大于顺序存储。
空链表就是表头为null的节点。

- 单链表的实现以及常用操作
  单链表也是用一个节点来进行构建一个单链表，单链表中只存放当前节点数据和下一个节点地址。如下图：

  单链表的构建一般采用尾部连接法，也就是每一个新节点都插入在表的最后一个位置，这样就保证了插入顺序和访问顺序的一致性。
  单链表常见的操作有：
  - 插入：根据index来获取即将插入节点的位置，然后通过链表之间的链接完成插入，在练习中多次见到，应该不难理解
  - 删除：和插入一样，通过链表之间的链接操作完成删除，被删除的节点会被垃圾回收机制回收
  - 访问：根据index访问数据，遍历来实现。
- 循环列表的实现以及常用操作
  循环列表和单链表的底层实现机制都一样，只不过对于单链表来说，尾部的next指向的地址是null，但是在循环链表里面，尾部的节点的next指向的地址是链表头部的地址，这样就构成了一个循环链表。 看下图：

- 双向链表的实现以及常用操作
  和单链表一样，底层是用一个节点来维护的，只不过这个节点不仅仅保存当前节点值，还有前一个节点的地址和后一个节点的地址。
  双向链表的常用操作：
  - 插入：根据index来获取即将插入的节点位置，和单链表不同的是，双向链表的插入维护需要对前后的链接进行操作；
  - 删除：和插入一样，获取删除节点的位置后，解除节点的链接关系，被删除的节点会被垃圾回收机制回收
  - 访问：根据index来遍历双向链表访问数据

1.3: 线性表的分析

以上我们看了线性表的两种实现方式，这两种实现方式的优缺点是什么？什么时候用顺序存储实现的表，什么时候用链式存储的表？

性能比较	顺序表	链式表
空间性能	因为顺序存储的是用数组维护的，因此总会存在有开辟的数组空间没有被用到	链式存储会充分利用计算机内存，但是由于节点要保存的信息较多，也会丧失一部分的空间性能
时间性能	因为是用数组维护的，因此访问速度快，插入和删除速度较慢	读取较慢，但是插入和删除快

因此从上面到分析可以看到，如果我们的操作主要集中在访问，那么顺序存储多占优势，可是如果我们的操作大多数在插入和删除，那么链式存储的实现占优势。
在Java中List接口就是这里的线性表，而且Java也提供了两种我们常用的实现类。一个是ArrayList，一个是LinkedList。其中ArrayList底层是由数组维护，因此适合用于访问；LinkedList是用链式存储，适用于插入和删除频繁的操作。