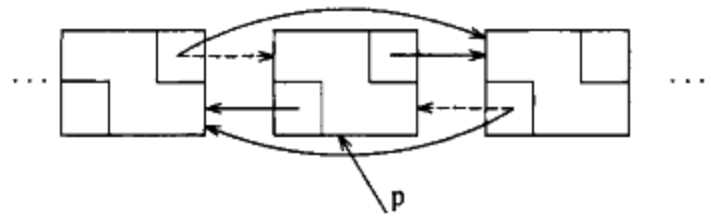

图 3-29 从一个双链表中删除由 p 指定的节点

```
1       /**
2        * Removes the object contained in Node p.
3        * @param p the Node containing the object.
4        * @return the item was removed from the collection.
5        */
6       private AnyType remove( Node<AnyType> p )
7       {
8           p.next.prev = p.prev;
9           p.prev.next = p.next;
10          theSize--;
11          modCount++;
12
13          return p.data;
14      }
```

图 3-30 MyLinkedList 类的 remove 例程

图 3-31 包含前面提到的私有 getNode 方法。如果索引表示该表前半部分的一个节点，那么

```
1       /**
2        * Gets the Node at position idx, which must range from 0 to size( ).
3        * @param idx index of node being obtained.
4        * @return internal node corresponding to idx.
5        * @throws IndexOutOfBoundsException if idx is not between 0 and size().
6        */
7       private Node<AnyType> getNode( int idx )
8       {
9           Node<AnyType> p;
10
11          if( idx < 0 || idx > size( ) )
12              throw new IndexOutOfBoundsException( );
13
14          if( idx < size( ) / 2 )
15          {
16              p = beginMarker.next;
17              for( int i = 0; i < idx; i++ )
18                  p = p.next;
19          }
20          else
21          {
22              p = endMarker;
23              for( int i = size( ); i > idx; i-- )
24                  p = p.prev;
25          }
26
27          return p;
28      }
```

图 3-31 MyLinkedList 类的私有 getNode 例程