```
1       private static class Node<AnyType>
2       {
3           public Node( AnyType d, Node<AnyType> p, Node<AnyType> n )
4             { data = d; prev = p; next = n; }
5
6           public AnyType data;
7           public Node<AnyType>   prev;
8           public Node<AnyType>   next;
9       }
```

图 3-25 MyLinkedList 类的嵌套 Node 类

```
1       /**
2        * Change the size of this collection to zero.
3        */
4       public void clear( )
5       {
6           beginMarker = new Node<AnyType>( null, null, null );
7           endMarker = new Node<AnyType>( null, beginMarker, null );
8           beginMarker.next = endMarker;
9
10          theSize = 0;
11          modCount++;
12      }
```

图 3-26 MyLinkedList 类的 clear 例程

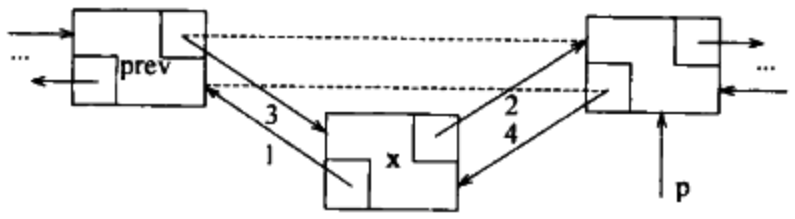

图 3-27 通过获取一个新节点，然后按所指示的顺序改变指针而完成向一个双链表中的插入操作

```
1       /**
2        * Adds an item to this collection, at specified position p.
3        * Items at or after that position are slid one position higher.
4        * @param p Node to add before.
5        * @param x any object.
6        * @throws IndexOutOfBoundsException if idx is not between 0 and size(),.
7        */
8       private void addBefore( Node<AnyType> p, AnyType x )
9       {
10          Node<AnyType> newNode = new Node<AnyType>( x, p.prev, p );
11          newNode.prev.next = newNode;
12          p.prev = newNode;
13          theSize++;
14          modCount++;
15      }
```

图 3-28 MyLinkedList 类的 add 例程