

```
1 public class MyLinkedList<AnyType> implements Iterable<AnyType>
2 {
3     private static class Node<AnyType>
4     { /* Figure 3.26 */ }
5
6     public MyLinkedList( )
7     { clear( ); }
8
9     public void clear( )
10    { /* Figure 3.26 */ }
11    public int size( )
12    { return theSize; }
13    public boolean isEmpty( )
14    { return size( ) == 0; }
15
16    public boolean add( AnyType x )
17    { add( size( ), x ); return true; }
18    public void add( int idx, AnyType x )
19    { addBefore( getNode( idx ), x ); }
20    public AnyType get( int idx )
21    { return getNode( idx ).data; }
22    public AnyType set( int idx, AnyType newVal )
23    {
24        Node<AnyType> p = getNode( idx );
25        AnyType oldVal = p.data;
26        p.data = newVal;
27        return oldVal;
28    }
29    public AnyType remove( int idx )
30    { return remove( getNode( idx ) ); }
31
32    private void addBefore( Node<AnyType> p, AnyType x )
33    { /* Figure 3.28 */ }
34    private AnyType remove( Node<AnyType> p )
35    { /* Figure 3.30 */ }
36    private Node<AnyType> getNode( int idx )
37    { /* Figure 3.31 */ }
38
39    public java.util.Iterator<AnyType> iterator( )
40    { return new LinkedListIterator( ); }
41    private class LinkedListIterator implements java.util.Iterator<AnyType>
42    { /* Figure 3.32 */ }
43
44    private int theSize;
45    private int modCount = 0;
46    private Node<AnyType> beginMarker;
47    private Node<AnyType> endMarker;
48 }
```

图 3-24 MyLinkedList 类