

The following method is a mirror-image:

```
public void printLastToFirst () {
    // Print all the elements in this DLL, in last-to-first order.
    for (DLLNode curr = last; curr != null;
         curr = curr.pred)
        System.out.print(curr.element + " ");
}
```

This would be an instance method of the DLL class. A typical method call would be:

```
zoo2.printFirstToLast();
```

where zoo2 is the DLL object created in Example 4.4. The printed output should be 'bat ant'.

```
public class DLLNode {
    // Each DLLNode object is a DLL node.
    // This node consists of an element (element), a link to its predecessor
    // (pred), and a link to its successor (succ).
    protected Object element;
    protected DLLNode pred, succ;

    protected DLLNode (Object elem,
                        DLLNode pred, DLLNode succ) {
        // Construct a DLL node with element elem, predecessor pred, and
        // successor succ.
        this.element = elem;
        this.pred = pred;
        this.succ = succ;
    }
}
```

Program 4.8 Java class representing DLL nodes.

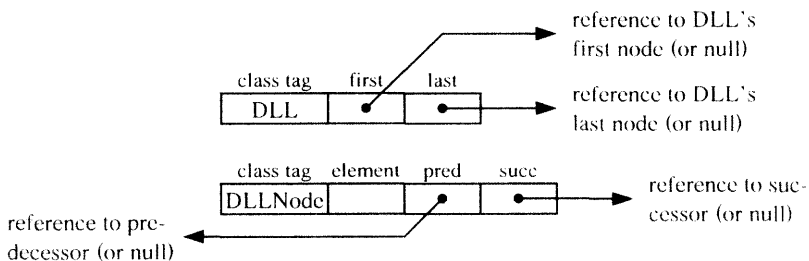


Figure 4.9 Detailed structures of DLL objects and DLLNode objects.