Program 2:

Linked list with Search and delete operations

Finding and Deleting Specified Links
Our next example program adds methods to search a linked list for a data item with a specified key value and to delete an item with a specified key value. The complete linkList2.java program is shown in

```
// linkList2.java
// demonstrates linked list
// to run this program: C>java LinkList2App
////////////////////////////////////////////////////////////////
class Link
{
public int iData; // data item (key)
public double dData; // data item
public Link next; // next link in list
// -------------------------------------------------------------
public Link(int id, double dd) // constructor
{
iData = id;
dData = dd;
}
// -------------------------------------------------------------
public void displayLink() // display ourself
{
System.out.print("{" + iData + ", " + dData + "} ");
}
} // end class Link
Finding and Deleting Specified Links 193
////////////////////////////////////////////////////////////////
class LinkList
{
private Link first; // ref to first link on list
// -------------------------------------------------------------
public LinkList() // constructor
```

```
{
first = null; // no links on list yet
}
// -------------------------------------------------------------
public void insertFirst(int id, double dd)
{ // make new link
Link newLink = new Link(id, dd);
newLink.next = first; // it points to old first link
first = newLink; // now first points to this
}
// -------------------------------------------------------------
public Link find(int key) // find link with given key
{ // (assumes non-empty list)
Link current = first; // start at 'first'
while(current.iData != key) // while no match,
{
if(current.next == null) // if end of list,
return null; // didn't find it
else // not end of list,
current = current.next; // go to next link
}
return current; // found it
}
// -------------------------------------------------------------
public Link delete(int key) // delete link with given key
{ // (assumes non-empty list)
Link current = first; // search for link
Link previous = first;
while(current.iData != key)
{
if(current.next == null)
return null; // didn't find it
else
{
previous = current; // go to next link
current = current.next;
}
} // found it
```

```
if(current == first) // if first link,
first = first.next; // change first
else // otherwise,
previous.next = current.next; // bypass it
return current;
}
// --------------------------------------------------------------
public void displayList() // display the list
{
System.out.print("List (first-->last): ");
Link current = first; // start at beginning of list
while(current != null) // until end of list,
{
current.displayLink(); // print data
current = current.next; // move to next link
}
System.out.println("");
}
// --------------------------------------------------------------
} // end class LinkList
////////////////////////////////////////////////////////////////
class LinkList2App
{
public static void main(String[] args)
{
LinkList theList = new LinkList(); // make list
theList.insertFirst(22, 2.99); // insert 4 items
theList.insertFirst(44, 4.99);
theList.insertFirst(66, 6.99);
theList.insertFirst(88, 8.99);
theList.displayList(); // display list
Link f = theList.find(44); // find item
if( f != null)
System.out.println("Found link with key " + f.iData);
else
System.out.println("Can't find link");
Link d = theList.delete(66); // delete item
if( d != null )
```

```
System.out.println("Deleted link with key " + d.iData);
else
System.out.println("Can't delete link");
theList.displayList(); // display list
} // end main()
} // end class LinkList2App
//////////////////////////////////////////////////////////////
```

The main() routine makes a list, inserts four items, and displays the resulting list. It then searches for the item with key 44, deletes the item with key 66, and displays the list again. Here's the output:

```
List (first-->last): {88, 8.99} {66, 6.99} {44, 4.99} {22, 2.99}
Found link with key 44
Deleted link with key 66
List (first-->last): {88, 8.99} {44, 4.99} {22, 2.99}
```