Chapter 6 and 7 HW

6.1. Add a constructor to the class LList that creates a list from a given array of objects. Consider at least two different ways to implement such a constructor. Which way does the least amount of work?

6.3. Suppose that you want an operation for the ADT list that adds an array of items to the end of the list. The header of the method could be as follows:

```
public void addAll(T[] items)
```

Write an implementation of this method for the class LList.

6.4. Suppose that you want an operation for the ADT list that returns the position of a given object in the list. The header of the method could be as follows:

```
public int getPosition(T anObject)
```

Write an implementation of this method for the class LList.

6.5. Implement an equals method for the class LList that returns true when the entries in one list equal the entries in a second list.

6.6. Suppose that a list contains Comparable objects. Implement a method that returns a new list of items that are less than some given item. The header of the method could be as follows:

```
public LList<T> getAllLessThan(Comparable<T> anObject)
```

Write an implementation of this method for the class LList. Make sure that your method does not affect the state of the original list.

6.7. In a **doubly linked chain**, each node can reference the previous node as well as the next node. Figure 6-17 shows a doubly linked chain that has both a head reference and a tail reference. Write a class to represent a node in a doubly linked chain. Write the class as an inner class of a class that implements the ADT list. You can omit set and get methods.

7.1. Suppose that you want an operation for the ADT list that removes the first occurrence of a given object from the list. The header of the method could be as follows:

```
public boolean remove(T anObject)
```

The method returns true if the list contained anObject and the object was removed. Write an implementation of this method for the class LList.

7.3. Suppose that you want an operation for the ADT list that moves the first item in the list to the end of the list. The header of the method could be as follows:

```
public void moveToEnd()
```

Write an implementation of this method for the class LList.

7.5. Suppose that a list contains Comparable objects. Implement the following methods for the class LList:

```
public T getMin() // Returns the smallest object in the list.

public T removeMin() // Removes and returns the smallest object in the list.
```