Lab 4

Goals for this Lab

1. Write a generic linked list class.

Task Description

Your goal for this lab is to write a generic LinkedList class similar to the one in the given in the lecture notes on linked lists. You will test this class using JUnit.

The LinkedList Class

The public constructors and methods required for the LinkedList class are listed here. The type E is the generic type of an element of the list.

LinkedList()	Empty constructor for instantiating the LinkedList.
int size()	Return the size (number of items) in this LinkedList.
boolean isEmpty()	Return true if this LinkedList has no items. (This is the same as the size equal to zero.) Return false if the size is greater than zero.
void add(E value)	Add the given element, value, to the end of the list.
void add(int index, E value)	Add the given element, value, to the list at the given index. After this operation is complete, get(index) will return value. This operation is only valid for 0 <= index <= size(). Throws IndexOutOfBoundsException if index < 0 or index > size().
E get(int index)	Return the element of the list at the given

	index. his operation is only valid for 0 <= index < size(). This operation does not modify the list. Throws IndexOutOfBoundsException if index < 0 or index >= size().
E remove()	Remove and return the first element (element number zero) from the list. This operation is only valid for non-empty (size() > 0) lists. Throws IndexOutOfBoundsException if the list is empty.
E remove(int index)	Remove and return the element with the given index from the list. This operation is only valid for 0 <= index < size(). After this operation, all elements that had an index greater than index (as determined by get()) will have their index reduced by one. Throws IndexOutOfBoundsException if index < 0 or index >= size().

Requirements

1. Your class must be named LinkedList.

2. Your class must provide the methods listed above for constructing, accessing, and manipulating LinkedList objects for a generic type.

3. Other than for testing purposes, your LinkedList class should do no input or output.

4. Your package must enable the provided tests.

Testing

A suite of JUnit tests have been provided to allow you to insure that your LinkedList implementation is correct. You can check your progress by noting the number of passing and failing tests. As you implement more of the class more tests will pass, until all tests are passing.

The provided tests are sufficient to show insure the LinkedList class is implemented correctly. You are free to add more tests as you feel are necessary, however none of the existing tests should be modified or removed.

Notes

Below is a suggested order of completion:

1. Get the program to compile.
2. Do the constructor, size and isEmpty methods.
3. Do the one parameter add method.
4. Do the get method.
5. Do the two parameter add method.
6. Do the no-parameter remove method.
7. Do the one-parameter remove method.

An empty LinkedList class has been provided with a number of TODO comments which indicate the methods to be implemented.

A Node class has been provided. This class provides the data structure to hold the data and the actual links of the list. This class should not need to be modified for this lab, however the LinkedList class will make use of it for most operations.

You may find it useful to implement recursion in this lab for some of the operations, however you will not be penalized for not doing so.