

CSSE 220 Day 30

Linked List Implementation

Check out *MyLinkedListReal* project from SVN

Announcements

- ▶ Hulbert/Cook lecture 10:50 Hatfield Hall
- ▶ Minesweeper due at 8:05 AM today
 - If you plan to use a late day, please fill out the survey by noon Thursday
 - So I can begin grading the ones that are done.
- ▶ Complete the Minesweeper team evaluation survey on ANGEL
- ▶ Markov due at 11:59 Friday.
- ▶ If you have submitted items that have not been graded, please tell me today.

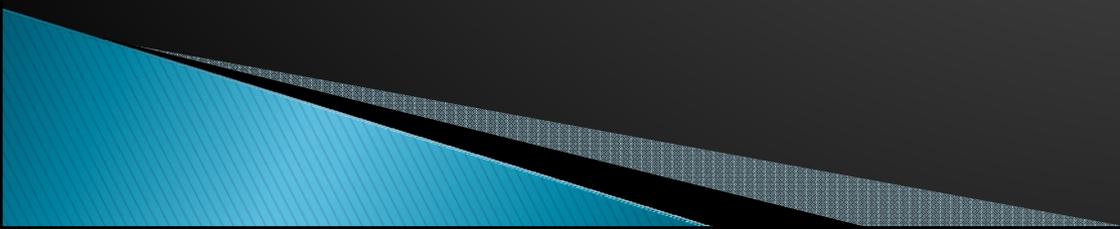
**Final
Exam
Wed
6 PM
O269**

**review
(Q &A)
session
Tuesday
4:00 PM
O-201**

Course Evaluations (Banner Web)

- ▶ What I especially want to hear about:
 - Have you learned a lot?
 - Have you acquired new skills/understanding/confidence?
 - What things about the course/instruction/instructor enhanced/hindered your learning?
 - Your suggestions for improvement
 - Is the textbook readable/helpful?
- ▶ What I already know:
 - The workload is very heavy. It's the nature of the learning programming!
 - "Boot camp" doesn't feel good at every moment. But it develops skills and stamina that can last a lifetime. Those who get through it are usually proud of what they have accomplished.

Questions



What's an iterator?

- ▶ More specifically, what is a `java.util.Iterator`?
 - It's an interface:
 - **interface `java.util.Iterator<E>`**
 - with the following methods:

<code>boolean</code>	<code>hasNext ()</code> Returns <code>true</code> if the iteration has more elements.
<code>E</code>	<code>next ()</code> Returns the next element in the iteration.
<code>void</code>	<code>remove ()</code> Removes from the underlying collection the last element returned by the iterator (optional operation).

An extension, `ListIterator`, adds:

<code>boolean</code>	<code>hasPrevious ()</code> Returns <code>true</code> if this list iterator has more elements when traversing the list in the reverse direction.
<code>int</code>	<code>nextIndex ()</code> Returns the index of the element that would be returned by a subsequent call to <code>next</code> .
<code>Object</code>	<code>previous ()</code> Returns the previous element in the list.
<code>int</code>	<code>previousIndex ()</code> Returns the index of the element that would be returned by a subsequent call to <code>previous</code> .
<code>void</code>	<code>set (Object o)</code> Replaces the last element returned by <code>next</code> or <code>previous</code> with the specified element (optional operation).

Implement LinkedListIterator

- ▶ Live coding together.

Doubly-linked lists

- ▶ Add another field to the `ListNode` class
 - A pointer to the previous node.

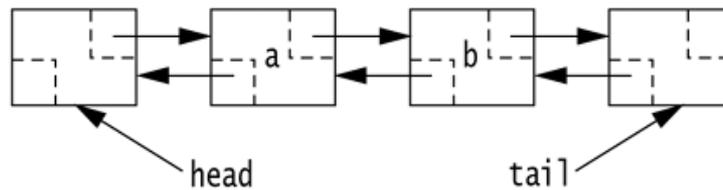


figure 17.15

A doubly linked list

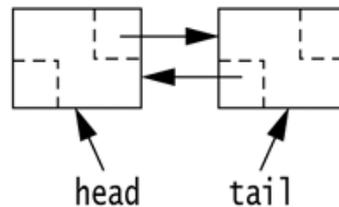
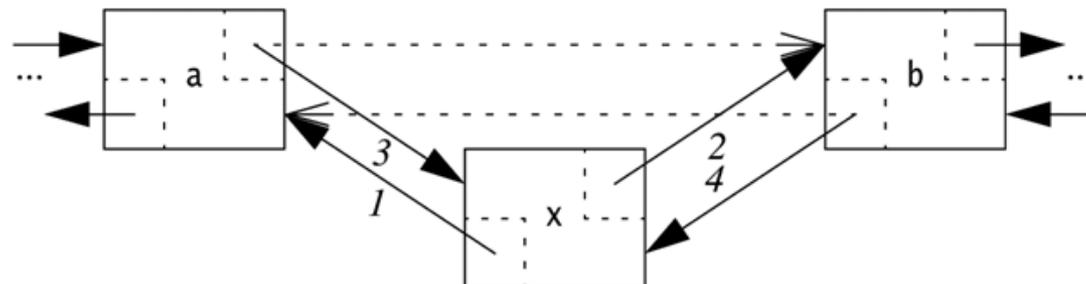


figure 17.16

An empty doubly linked list

figure 17.17

Insertion in a doubly linked list by getting new node and then changing pointers in the order indicated



Doubly-linked list operations

- ▶ `DListNode current<T>;`
- ▶ Write code to remove `current` node from its list.
- ▶ Write code to add a new node containing `x` after `current`.

Markov work time

» or Hulbert/Cook lecture