

# Finishing Linked Lists

2-22-2012

# Opening Discussion

- Minute essay comments.
  - What to add for IcP #4.
  - Why use singly linked list?
  - Modal dialogs.
  - Temp of majors lab.
  - Wrapping a line of text in HTML.

# Implementing a Singly Linked List

- Let's work together to build an implementation of a singly linked list.
- We will implement the Buffer trait. It has eight abstract methods that we will be forced to implement to get everything else.

# Sentinels

- A sentinel is an extra node in the list that represents the “end” of the list and doesn't store data.
- The purpose of the sentinel is to remove special cases. The next of the sentinel is what we have called head.
- They are most useful in a doubly linked list where the previous of the sentinel is tail.

# Implementing a Doubly Linked List

- Now let's implement `java.util.List` with a doubly linked list with a sentinel. The list will also be circular.
- You should notice that this implementation never has to check for null because no references in the list should ever be null. This simplifies the code significantly. We also implicitly get a head and a tail with no extra work. If you don't have a sentinel you will write a lot of extra checks for nulls and even more to include a tail.

# Minute Essay

- Questions?