

Slide 1

Administrivia

- Reminder: Midterm next Tuesday. Review sheet on Web. Remember that all significant code we do in class is available via the "sample programs" page. (Go to the javadoc-generated pages and look for links to source code.)
- Reminder: Homework 3 code due today. Accepted without penalty through next Tuesday.
- Homework 4 will be on Web soon, probably tomorrow. Design to be due Thursday, code the following Tuesday.

Slide 2

Lists

- List ADT (review):
 - "Values" are lists of elements.
 - Many operations possible — add element, remove element, search for element, etc., etc.
- Something we often want to do with this and other "container classes" is do something to all elements — idea of "iterator". `java.util.Iterator` defines interface for this.

Lists, Continued

Slide 3

- A really simple unordered list might provide only methods to add an element (don't care where), remove an element (if found), and create an iterator that could be used to "walk through" the list.
- First let's define an interface for this version of lists, and then write some very simple code using `Vectors`. (No, no reasonable person would do it this way, but it's an example of how to write an iterator.)

Lists, Continued

Slide 4

- Now think about how to implement the same interface with a "linked list" as described last time. First, draw pictures . . .
- Then think about what you need to turn the pictures into code. Probably you'll need:
 - Variables (e.g., something to point to the first "node" (little box).
 - Classes-within-the-class (for nodes / little boxes, iterators).
 - Methods for interface.

Minute Essay

- None — quiz.

Slide 5