



2/20/2007





# Opening Discussion

- Do you have any questions about the assignment?
- What type of linked list do you plan to write for your entity list and why?
- Do you have any questions about the reading?





# Code for the Doubly Linked List

- Let's go check on our code for the doubly linked list and see if there are any other methods we really feel like writing.
- Now lets write a main method someplace that actually uses our list and make sure that the methods we have done work the way we want.
- Let's write code in main to print out all the elements of our list in order. What is wrong with this?

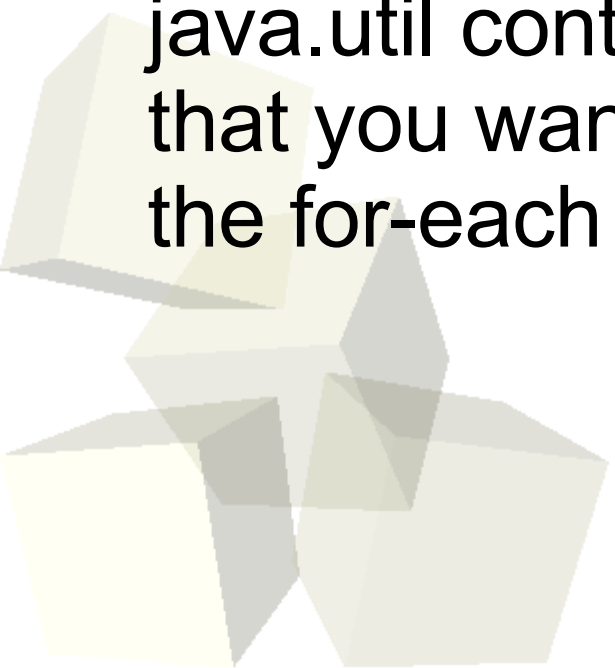


- Many containers, like linked lists, don't have good direct access. With these we need a different way to go through the elements. We need something that has inside knowledge of the class so it can be efficient without exposing the way things work in the class.
- This is what iterators do. Let's go look at the `java.util.Iterator` interface then implement a method in our list that returns an `Iterator`.
- Now let's use that to write a loop.
- In order libraries you will find they use `java.util.enumeration` for the same purpose.



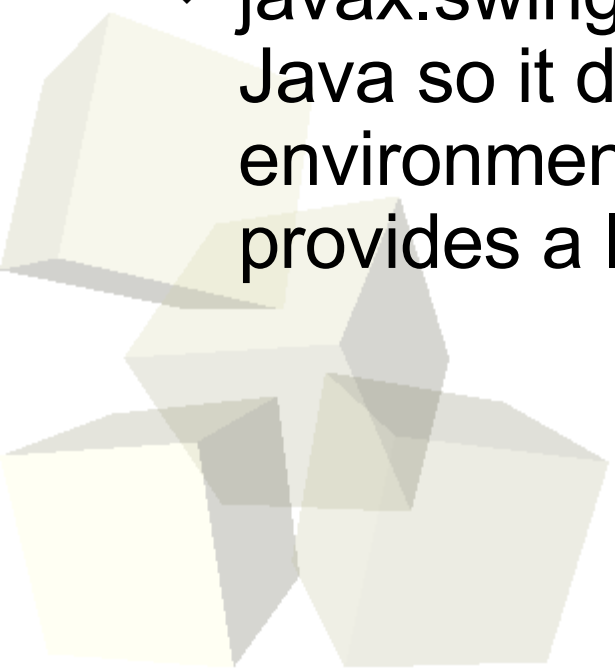
# Iterable and the for-each Loop

- The `java.util.List` interface is a subtype of the `java.lang.Iterable` interface that was added to Java 5.
- `Iterable` has only one method, `iterator` which returns an `Iterator`.
- The for each loop will work with anything of type `Iterable`. This is true for arrays and all the standard `java.util` containers. You can also make any class that you want be `Iterable` so you can use it with the for-each loop.





- One of the greatest benefits of Java is the vast size of the standard libraries and all the functionality they provide.
- GUIs are part of this standard library and, in fact, there are actually two GUI libraries in Java.
  - ♦ `java.awt` is the Abstract Windowing Toolkit and it was the original GUI package in Java.
  - ♦ `javax.swing` was added later and is actually written in Java so it doesn't directly make calls to the windowing environment. Originally this hurt speed, but now it provides a lot of flexibility and power.





# Swing Basics

- We will be using Swing for everything in this class.
- GUIs have a lot of inheritance and Swing stuff is built on AWT stuff.
- Let's go look at some of the things in the javax.swing package.
- All GUI elements in Swing inherit either directly or indirectly from JComponent. (Note that things in Swing often start with J to distinguish them from the AWT counterparts.)
- We need a JFrame generally for the base window and we can put stuff inside of it.
- The components include things like JButton, JList, JComboBox, JRadioButton, JCheckBox, JTree, JTable, etc.



# Layout Managers

- To place components inside of other components we use layout managers. We could position them directly, but this is very inflexible. The layout managers follow fairly simple rules for where things should go and how they change size as windows change size.
- The nesting of components in components (and hence layouts inside of layouts) gives you a lot of power without needing complex layouts.
- Commonly used layouts include GridLayout, FlowLayout, and BorderLayout.
- Swing provides some custom components that do layouts for you.





- As time permits, let's write a GUI.





- What type of GUI do you think we should work on for next class?
- Design #3 is due today and the assignment is due on Thursday.
- The midterm is two weeks from today.

