



2/22/2007





Opening Discussion

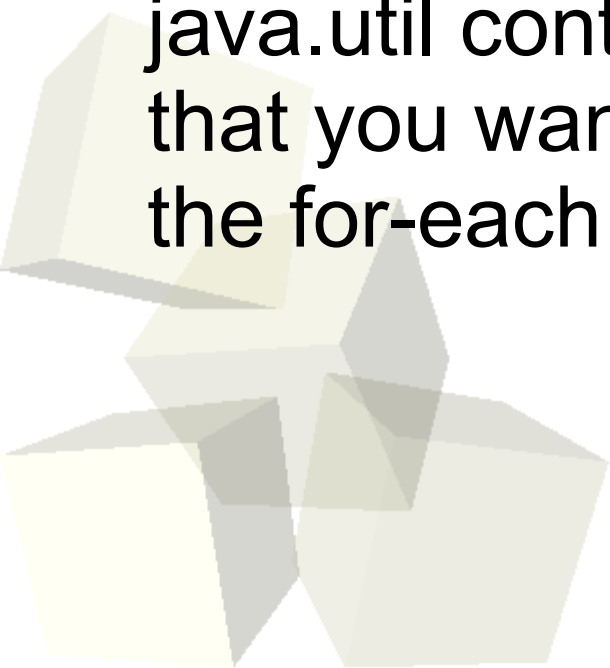
- Let's look at solutions to the interclass problem.
- Do you have any questions about the assignment?





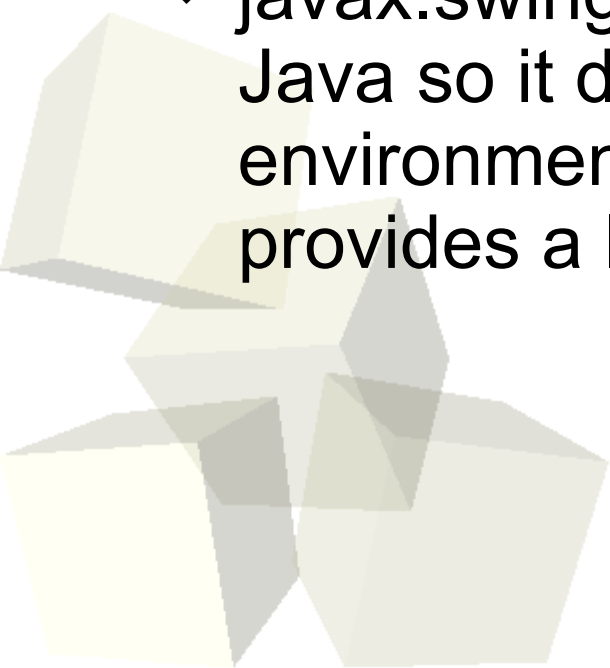
Iterable and the for-each Loop

- 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 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.





- 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.



- Let's write a little GUI. I want a JTree on the left side of the frame with buttons above and/or below that say “Add” and “Remove”, a panel in the middle, and on the right I want a GUI that might be used for setting properties on a rectangle. At the least I'd like to set a width, height, and color.
- Let's give the frame a menu bar. For now we can just add a “File” menu and have an “Exit” selection in it.





- Why should you use layout managers when programming GUIs instead of explicitly setting the positions and sizes of elements?
- Remember to turn in the code for assignment #3 today.
- Interclass Problem – Build a GUI for your RP calculator. It needs to have buttons for the numbers, operators, and other functions. RPC calculators normally have an “Enter” button that pushes the current value onto the stack. Use a JTextField for the display and a JTextArea to show the memory. You can decide exactly how to lay it out.