Multithreading in Java

4/3/2007

Opening Discussion

- Do you have any questions about the quiz?
- What could go wrong with our tree to mess up performance? It could become unbalanced and it degrades to a linked list.
- Do you have any questions about the assignment?

Coding with java.util.concurrent

- Now that you know more about multithreading and the java.util.concurrent library, I want for us to write some more serious parallel code.
- In fact, I want for us to combine what we have learned about recursion and parallelism to create some code that uses a divide and conquer method to solve a problem, but throws in multithreading to complete it faster.
- How can we go about doing this? How is divide and conquer well suited for parallelization?

Multithreading

- Let's play briefly with our drawing application and see what happens when we try to multithread part of it. Note that multithreading in GUIs can provide some benefits other than real speed by making the interaction smoother.

I/O Streams

- Next class you will work with input and output streams that are part of the java.io package. This is the basic package for doing input and output in Java.
- The package uses significant inheritance with the hierarchies rooted in the InputStream, OutputStream, Reader, and Writer abstract classes. The first two provide I/O based on bytes while the other two use characters.
- These base classes have very little functionality themselves and being abstract they can't even be instantiated.

File Streams

- In order to use streaming you have to be able to instantiate something. One set of classes that you can instantiate is the set of file streams.
- These classes are FileInputStream, FileOutputStream, FileReader, and FileWriter.
- Let's go look at these really quick.

Wrapping Streams

- The file stream classes still don't do much, they just do what their base class does except they are actually attached to a file.
- Being able to just read or write bytes is technically sufficient for any task, but you wouldn't want to write much code that way.
- We gain functionality by “wrapping” stream objects around one another. This is a design pattern called the Decorator.
- Example decorations include buffering, functionality for binary I/O (DataInputStream/DataOutputStream), or formatted printing (PrintWriter).

Minute Essay

- What are some of the potential benefits of the streaming model Java uses for doing I/O? How could inheritance and polymorphism be put to use here?
- There are only six remaining class days.
- This weekend we are hosting a high school programming competition. The competition runs from 10am to 4pm and we will need runners and people to help monitor the rooms where teams are competing.