Networking with RMI and Course Review

12-10-2002

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment?

Difficulties in Networking

- The code that we did last time didn't have this problem, but we could add it in fairly easily. Most networked programs can send lots of different types of messages to the other program. Having the server figure out what message the client is sending can be a significant task.
- What would we need to do to have our earlier program be able to send and receive either "data" of object formats?

Remote Method Invocation

- Java has built in a system that makes communication across the network even easier than it was with java.net, though unlike java.net, it can't be used with programs written in other languages.
- With RMI you can have references to "remote objects". These are objects that are on another computer but you interact with them as if they were local.

Remote Interfaces

- The entity you get for an object remotely is actually not the object itself, but an interface that object implements. The interface must extend java.rmi.Remote.
- The implementation class should also extend java.rmi.server.UnicastRemoteObject.
- Note that this implies that the implementation can have more functionality than the remote interface does.
- All remote methods can throw java.rmi.RemoteException.

Passing

- Different types of objects are passed differently
  - Remote objects - any object that extends java.rmi.Remote is passed as a remote object. You get a skeleton that does network communication.
  - Serializable objects - If an object is not Remote it must be serializable and then it is passed by value.
  - Primitives - No pass by reference of primitives

Registering and Lookup

- Once you have a remote object, it can pass you others, but getting the first one takes a different approach.
- An object can register itself with the RMI registry using the rebind method of java.rmi.Naming. (You have to start a local registry with rmiregistry first.)
- Objects can get a remote reference to a registered object using the lookup method of java.rmi.Naming.

Compiling

- Once you have written your code you first compile it with a normal Java compiler. After that you have to do another step to create stub and skeleton classes that do most of the work behind RMI. You to this is the RMI compiler, rmic. Just run rmic specifying the name of the implementation class.

Code

- Let's look at some code written to do networking using RMI. If time allows, we'll augment our code from last time to use RMI for passing widgets around and we will also look at a little chat program that I wrote last year using RMI.

Course Recap

- We looked at a number of different topics in this course.
  - Object-orientation and how it is done in Java.
  - Inclusion polymorphism.
  - Immutability and how it helps in SE.
  - Arrays and the processing of them.
  - Basic data structures
    - Stacks, queues, linked lists, priority queues, binary trees, heaps.
  - Recursion and using it to solve problems.

Details of Java

- You also learned about how to do some different tasks in Java with the Java API.
  - GUI building and graphics
  - Exceptions for error handling
  - Threads
  - Streams
  - Networking
- Combined, these allow you to create interesting an powerful programs.

Course Objectives

- More important than the material is if the course met its objectives.
  - Did this course make you think?
  - Has this course enabled you to think about things in new ways?
  - Do you feel you have a better understanding of the object model of computing and how to use it to solve problems?
- Are you motivated to do more? You know how to do interesting things in Java, find a pet project and practice your skills.

Minute Essay

- Write on a piece of paper when you think the best time for a review session would be, or when you absolutely can't make it to a review session.
- It's time to do course evaluations. I hope you had a great semester and I'll see you at the final.