

A Java Grab Bag

Phillip Rhodes
Triangle Java User's Group
01-16-2006

Who is this speaker?

Phillip Rhodes

Applications Software Engineer w/ Red Hat, Inc.

email:

prhodes@redhat.com

blog:

<http://www.jroller.com/page/mindcrime>

Speaker Background

NC Native from Wilmington area, moved to Chapel Hill in 2000 to catch the “Dot Com Bubble.”

Have worked for Voice Data Solutions, IBM, Ineo USA and now Red Hat.
Have worked on: Computer Telephony / IVR applications, E-Business applications, Middleware, and Systems Management.

Education: A.G.E. From Brunswick Community College, a few semesters at UNC-Wilmington, A.S. in Computer Programming from Wake Tech, currently finishing up an A.S. In High Performance Computing @ Wake Tech.

In a “past life” was a North Carolina Fire & Rescue Commission certified instructor with qualifications to teach Firefighter I&II certification, Incident Command and LP Gas Firefighting.

Why should I believe you?

1. You shouldn't.
2. don't accept things “on authority”
3. If you haven't benchmarked it yourself, don't believe it.
4. just how slow is instanceof anyway?

Instance of

For at least one scenario:

Instance of operations

- 20
- 200
- 2000
- 200000
- 2000000
- 20000000
- 200000000

Time

- Too small to measure
- Too small to measure
- Too small to measure
- Too small to measure
- Too small to measure
- ~100 milliseconds
- ~1000 milliseconds

“In a static way”

Have you ever seen a warning that looks like:

The static method `sayFoo(String)` from the type `Foo` should be accessed in a static way

Why is this a warning? What is the compiler complaining about and why?

The Rule of Least Surprise

“In user interface design, programming language design, and ergonomics, the principle (or rule) of least astonishment (or surprise) states that, when two elements of an interface conflict or are ambiguous, the behaviour should be that which will least surprise the human user or programmer at the time the conflict arises, because the least surprising behavior will usually be the correct one.”

http://en.wikipedia.org/wiki/Principle_of_least_astonishment

<http://www.faqs.org/docs/artu/ch11s01.html>

Unique IDs in Java

Prior to Java 1.5

`java.rmi.server.UID`

<http://java.sun.com/j2se/1.4.2/docs/api/java/rmi/server/UID.html>

`java.rmi.dgc.VMID`

<http://java.sun.com/j2se/1.4.2/docs/api/java/rmi/dgc/VMID.html>

In Java 1.5+

`java.util.UUID`

<http://java.sun.com/j2se/1.5.0/docs/api/java/util/UUID.html>

Class ThreadLocal

Used to scope a value to a Thread.

Useful for storing things like Transactions or UserIDs.

<http://java.sun.com/j2se/1.5.0/docs/api/java/lang/ThreadLocal.html>

Dynamic Proxies

- Dynamic Proxy facility added in JDK 1.3
- Allow you to assign proxies at runtime
- Can be used as limited form of Aspect-Oriented Programming.
- Transparently add security, transactions, logging, etc. without touching your classes.
- Related to GOF Decorator pattern, but using Dynamic Proxies can eliminate some duplicate code.
- But there is a performance hit.
- How much of a performance hit? Who knows, benchmark it yourself and find out.

Reference classes

- Allow the programmer the ability to interact with the garbage collector (a little)
- Useful when you need a little more control over when things get collected, or need to perform post-collection actions.
- `WeakReference`: when an object has only weak references, it will be garbage collected. Useful for making sure you don't unintentionally retain objects by registering for event notifications, etc.
- `SoftReference`: when an object is only softly reachable, it will be collected before an `OutOfMemoryError` is thrown.
- `WeakHashMap`
- Notification queues / `PhantomReference`, more flexible alternative to finalization.

java.lang.management

- New JMX features in JDK 1.5, for introspecting the virtual machine and obtaining state information.
- Useful for system management / monitoring / alerting. To use a buzzword “autonomic computing” would probably leverage this stuff.
- Great stuff for long-running services / servers. Ability to track memory usage, thread usage, cpu usage, etc.
- With some creative use of this stuff, you can probably build “self-tuning” software that can optimize itself as it runs.

Management MBeans

- ManagementFactory: for accessing Mbeans
- ClassLoadingMXBean
- CompilationMXBean
- GarbageCollectorMXBean
- MemoryManagerMXBean
- MemoryMXBean
- MemoryPoolMXBean
- OperatingSystemMXBean
- RuntimeMXBean
- ThreadMXBean

Monitoring Threads and Memory

- ThreadInfo
- MemoryUsage
- MemoryNotificationInfo
- Provide much more detailed info on the state of the system that was previously available.

Using getObjectSize()

- Part of the new Instrumentation package added in Java 5
- Finally a standard way to determine the size of an object without resorting to hacks using serialization
- A little tricky to configure, will get better in Mustang (Java 6)
- “The only way to access an instance of the Instrumentation interface is for the JVM to be launched in a way that indicates the agent class - see the package specification. The Instrumentation instance is passed to the premain method of the agent class. Once an agent acquires the Instrumentation instance, the agent may call methods on the instance at any time.”
- Say what?!?!
- Unfortunately this is not nearly as useful as it could be. It only returns the memory overhead of an “empty” instance of the class in question. For example, it returns the same value for String no matter how large the String is. Likewise it doesn't consider objects contained in collections.
- To get an accurate representation of the entire object graph, you'll still have to use reflection and recursively navigate the object graph and accumulate the totals.

Random Numbers

- Class `Random`, simpler than `Math.random()` for many cases
- `Collections.shuffle()` to shuffle the elements in a `List`
- `Collections.shuffle()` can take a `Random` (or subclass) as it's source of randomness
- Can be used for picking topics for a Tri-JUG presentation.

ExecutorService

- A little cleaner / simpler way to schedule tasks to run, wait for them to finish and retrieve a result.
- Callable interface has call() method which can return a result, unlike run() method of Runnable interface.
- Use a Future to retrieve the return value.
- For more sophisticated scheduling tasks, look at Quartz.
- <http://www.opensymphony.com/quartz/>

Jakarta HttpClient

- Much more usable than doing HTTP with the low-level classes (URL, URLConnection, etc.) supplied with J2SE.
- Companion of Jakarta Commons-Net, which also APIs for FTP, TFTP, NNTP, Telnet, POP3, etc.
- Use for “screen scraping” web-sites, programmatically submitting forms, building a web-crawler, etc.

FreeTTS

- Open Source Text To Speech engine.
- Embed in your applications to allow them to talk.
- Sven the Drunken Swedish Sailor
- <http://freetts.sourceforge.net/docs/index.php>
- Voice Reco is also available, using Sphinx
- <http://cmusphinx.sourceforge.net/html/cmusphinx.php>

OpenOffice.org + Java

- <http://udk.openoffice.org/>
- <http://framework.openoffice.org/scripting/index.html>
- <http://framework.openoffice.org/scripting/utilities-download.html>
- <http://framework.openoffice.org/scripting/scriptingf1/developer-guide.html>
- <http://framework.openoffice.org/scripting/ScriptingFrameworkChapter.pdf>
- <http://www.ofoforum.org/>
- Since OpenOffice.org 2.0, you can officially script OOo with Java. Some builds of the 1.x series shipped with a beta version of the Scripting Framework.
- As well as Python, Beanshell, C++, Perl, CLI, Javascript and OOo Basic.
- Everything below this line and on next slide, left as an exercise for the reader.
- Imagine integrating OOo into an SOA (easily done with Java's nice web-services support) and using it to create an “executive dashboard.”
- Or something more mundane such as mail-merge to generate letters and mailing labels.

Going a step further

- Use OOo Java macros to pull data from (DB, SOA, Web, whatever) and generate spiffy reports, charts, etc.
- Publish to a JSR-170 compliant CMS system.
- <http://forge.objectweb.org/projects/exoplatform/>
- <http://incubator.apache.org/jackrabbit/>
- Publish an RSS feed from the CMS system for users to subscribe to notifications of added or modified documents.
- <http://java-source.net/open-source/rss-rdf-tools>
- <http://www.rssowl.org/>
- Use as part of a Workflow Application or Cooperative Document creation process.
- ?? Use your imagination.

Resources: books

- Hardcore Java, Robert Simmons, O'Reilly, ISBN: 0-596-00568-7
- Java Puzzlers: Traps, Pitfalls and Corner Cases, Joshua Bloch & Neal Gafter, Addison-Wesley, ISBN: 0-321-33678-X
- Wicked Cool Java: Code Bits, Open-Source Libraries and Project Ideas, Brian D. Eubanks, No Starch Press, ISBN: 1-59327-061-5
- The Java Developers Almanac 1.4, Volume 1, Patrick Chan, Addison-Wesley, ISBN: 0-201-75280-8
- Java Cookbook, Ian F. Darwin, O'Reilly, ISBN: 0-596-00170-3
- Concurrent Programming in Java, 2nd Ed. Doug Lea, Addison-Wesley, ISBN: 0-201-31009-0
- Jakarta Commons Cookbook, Timothy M. O'Brien, O'Reilly, ISBN: 0-596-00706-X
- Applied Software Engineering using Apache Jakarta Commons, Christian Gross, Charles River Media, ISBN: 1-58450-246-0
- Java Collections, John Zukowski, Apress, ISBN: 1-893115-92-5
- Covert Java: Techniques for Decompiling, Patching and Reverse Engineering, Alex Kalinovsky, Sams, ISBN: 0-672-32638-8
- Better, Faster, Lighter Java, Bruce A. Tate & Justin Gehtland, O'Reilly, ISBN: 0-596-00676-4
- Java 1.5 Tiger: A Developer's Notebook, Brett McLaughlin & David Flanagan, O'Reilly, ISBN: 0-596-00738-8
- Java2 V5.0 Tiger: New Features, Herbert Schildt, Osborne, ISBN: 0-07-225854-3

Resources: websites

- Java Language Specification: <http://java.sun.com/docs/books/jls/>
- Java Virtual Machine Specification: <http://java.sun.com/docs/books/vmspec/>
- <http://www.jcp.org>
- <http://www.java.net>
- <http://www.theserverside.com>
- <http://www.javacrawl.com>
- <http://www.javablogs.com>
- <http://www.javarss.com>
- <http://www.jroller.com>
- <http://www.onjava.com>
- <http://www.javaworld.com>
- <http://www.ibm.com/developerworks/java>
- <http://www.planetjava.org>
- <http://www.trijug.org>

What didn't make it / possible future topics

- Intra-application messaging with JMS
- More of java.util.concurrent
- More Jakarta Commons
- IRC with Java:
 - <http://www.jibble.org/pircbot.php>
 - <http://drone.codehaus.org/>
- Instant Messaging with Java:
 - Ymsg: Yahoo IM protocol
 - <http://jymsg9.sourceforge.net/>
 - Jaimbot: AOL IM protocol
 - <http://jaimbot.sourceforge.net/>
 - Jabber: Open standard IM protocol aka XMPP, used by Google Talk.
 - <http://www.jabber.org/>
- Deeper coverage of the OpenOffice.org scripting stuff
- Creating RSS / Atom feeds
- JSR-94, Rules Engines, Drools
- Javabeans Property Editors
- Mixed Language Programming in Java:
 - Jacl – TCL integration
 - Jython – Python Integration
 - ABCL – Lisp in Java
 - Ciao Prolog – Java interface