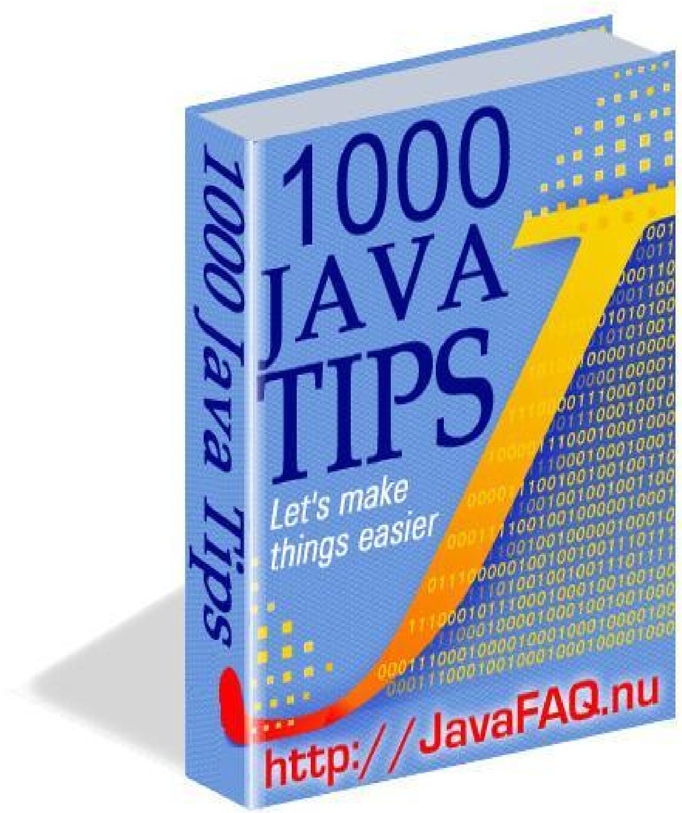


1000 Java Tips

Let’s make things better!

Alexandre Patchine

Reviews

"I used good tips from here when studied Java"

The problem with any language it is not enough to study only commands or API. You need some feeling that you are comfortable, like at home, you know where food where bed. This e-book gave me such feelings to Java. Before that, even getting certified, I felt like stranger in a hotel, but now I am at home with Java.

- Peter, graphic des

"This book has saved me hours of research while coding Java. From the obvious to the obscure, this book answers the real world questions that bring development to a halt. And the authors really know their stuff. A great help."

- Robin Kay from user opinions on CNET about free first edition

"My 5.5 years Java experience can to be improved even more!"

- Net Rambler from Europe from user opinions on CNET about free first edition

"Nice format to learn Java. Excellent e-book for developers who do not have mentors around in the initial stages."

From user opinions on CNET about free first edition

Foreword

Hello dear reader!

"1000 Java Tips" is my collection of tricky questions and answers from my newsletters for last four years and answers I gave to people in my emails.

Let me ask you a few questions:

1. Are you preparing for Java certification exam and want to be sure that you will pass it from first attempt? This book gives you real examples how to use Java.

2. Are you looking for a job as a Java programmer and want to be prepared for the Java interview questions?

3. Are you looking for a good design idea?

You have come to right place!

This e-Book will give you more knowledge, better perspective to programming and computer science especially. It covers wide range of questions.

My "1000 Java Tips" e-book will help you to understand Java much better and be well prepared for your Java Certification Examination.

The e-Book is made in the form of questions and answers. I included also the big section of Advanced Java Tips by Dr. Heinz Kabutz where topics are bigger. They are well written articles! Every article exploring one area and gives you examples.

This e-book is an easy-to-navigate huge collection of Java Tips.

Easy style, right proportion of humor and Java lets you read them at once, in one breath!

Author of “1000 Java Tips” e-book,
Alexandre Patchine.

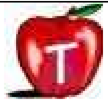
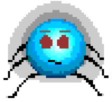
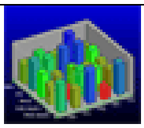
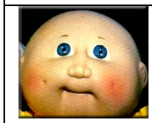
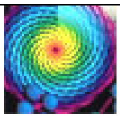
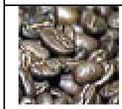
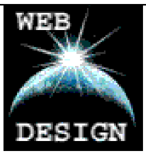
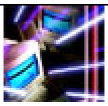
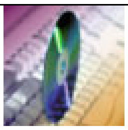
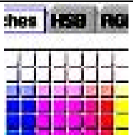
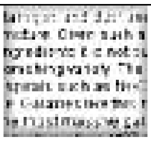
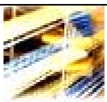
Table of Contents

1. Introduction

è I invite you to visit my site “Java Daily Tips”: http://javafaq.nu . There you can find tons of links, free books, tips and my weekly newsletter! 3000 visitors every day, more than 16000 (by the beginning of 2004) readers of my newsletter can not be wrong!

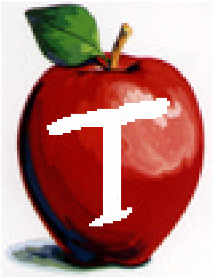
Just click icons below and you will be on my site!

You are welcome!

 Java Applets	 Java Bugs	 Daily Tips	 Databases	 drivers_uti drivers com drive File systems
 Fun, humor	 General Java	 Good To Know!	 Graphics	 Java Tools
 Java Beans	 Java Installation, tuning, options...	 Web development, tech, marketing, security	 Java Mobile Technology	 Networking
 Security	HTTP Server CGI CGI Servlet Servlet Servlet Java VM JSP, Servlets	 Sound, multimedia	 Swing, AWT	 Text processing
		 Threads		

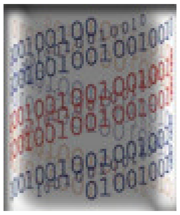
2. Applets

This is DEMO version!



3. Databases & Beans

This is DEMO version!



4. Distributed systems

This is DEMO version!

5. File Systems

This is DEMO version!

6. Graphics, AWT, Swing

7. General Java

è Question: I was looking at java.util.Timer class and there is no possibility to cancel an individual task. I have to cancel whole Timer with all my scheduled tasks... What is solution?

Answer: The solution is in your question :-) you could develop a subclass of Timer that stores a list of all scheduled TimerTask instances. Then retrieve the instance that you want to cancel:

```
TimerTask [] tasks = timer.getScheduledTasks();
tasks[0].cancel();
```

tip based on SUN's forum answer

n Question: Why internationalization often named as i18n?

Answer: The term internationalization is abbreviated as i18n, because there are 18 letters between the first "i" and the last "n."

è Question: I often hear something like this "daemon runs" or "how to make my own daemon"...
Is not it the same daemon and application? They are both executable. What is difference between daemons, services and applications?

Answer: In UNIX terminology, applications that provide services rather than directly communicate with users are called daemons. For example network, printer services and so on...
They do not need any direct interaction with user and run in background, invisible for GUI users.
In Windows the same concept is known as a service.
So, if you can not communicate with application that you develop and it is invisible - you can surely call it daemon or service, until all bugs are fixed :-)

è Question: How can I convert any Java Object into byte array?

Answer: Very elegant way I found on SUN's web site:

```
public static byte[] getBytes(Object obj) throws java.io.IOException{
  ByteArrayOutputStream bos = new ByteArrayOutputStream();
  ObjectOutputStream oos = new ObjectOutputStream(bos);
  oos.writeObject(obj);
  oos.flush();
  oos.close();
  bos.close();
  byte [] data = bos.toByteArray();
  return data;
}
```

è Question: Why does the InputStreamReader class has a read() method that returns an int and not a char? Why we need additional steps for converting?

Answer: When end of stream is reached we always get the value -1. It is not possible with char. How do we know when stream is empty? We can not use EOF, EOL and so on...

n Question: What are direct and non/direct buffers (NIO)?

Answer: A byte buffer is either direct or non-direct. Given a direct byte buffer, the Java virtual machine will make a best effort to perform native I/O operations directly upon it. That is, it will attempt to avoid copying the buffer's content to (or from) an intermediate buffer before (or after) each invocation of one of the underlying operating system's native I/O operations.

A direct byte buffer may be created by invoking the allocateDirect factory method of this class. The buffers returned by this method typically have somewhat higher allocation and deallocation costs than non-direct buffers. The contents of direct buffers may reside outside of the normal garbage-collected heap, and so their impact upon the memory footprint of an application might not be obvious. It is therefore recommended that direct buffers be allocated primarily for large, long-lived buffers that are subject to the underlying system's native I/O operations. In general it is

best to allocate direct buffers only when they yield a measurable gain in program performance.

A direct byte buffer may also be created by mapping a region of a file directly into memory. An implementation of the Java platform may optionally support the creation of direct byte buffers from native code via JNI. If an instance of one of these kinds of buffers refers to an inaccessible region of memory then an attempt to access that region will not change the buffer's content and will cause an unspecified exception to be thrown either at the time of the access or at some later time.

Whether a byte buffer is direct or non-direct may be determined by invoking its isDirect method. This method is provided so that explicit buffer management can be done in performance-critical code.

è Question: I try to call the Runtime.exec() command by passing it an array of strings.

The problem is that I need to use "&" in my command. It runs perfectly on shell prompt but not by Runtime.exec(). Java cuts all everything away after that sign, so my command is incomplete.

Answer: No clear explanation to such behaviour. Use workaround:
write your command into file (shell script) and then run this script. It will work :-)

è Question: I write a batch file dynamically.
First, I collect all data into String value and then write " " as an end of line. But I can not execute this batch file because of this EOL.

Answer: Use system defined line separator,

```
String lineSeparator = System.getProperty("line.separator");
```

è Question: Why Java Hot Spot Compiler has the name - Hot Spot? Part 1.

Answer: The best answer I actually found in "Technical White Paper for the Java HotSpot Virtual Machine, v1.4.1" at SUN web site:
"Most attempts to accelerate Java programming language performance have focused on applying compilation techniques developed for traditional languages. Just-in-time (JIT) compilers are essentially fast traditional compilers that translate the Java technology bytecodes into native machine code on the fly. A JIT running on the end user's machine actually executes the bytecodes and compiles each method the first time it is executed.

Adaptive optimization solves the problems of JIT compilation by taking advantage of an interesting program property. Virtually all programs spend the vast majority of their time executing a minority of their code. Rather than compiling method by method, just in time, the Java HotSpot VM immediately runs the program using an interpreter, and analyzes the code as it runs to detect the critical hot spots in the program. Then it focuses the attention of a global native-code optimizer on the hot spots.

è **Question:** Why Java Hot Spot Compiler has the name - Hot Spot? Part 2.

By avoiding compilation of infrequently executed code (most of the program), the Java HotSpot compiler can devote more attention to the performance-critical parts of the program, without necessarily increasing the overall compilation time. This hot spot monitoring is continued dynamically as the program runs, so that it literally adapts its performance on the fly to the user's needs.

A subtle but important benefit of this approach is that by delaying compilation until after the code has already been executed for a while (measured in machine time, not user time), information can be gathered on the way the code is used, and then utilized to perform more intelligent optimization. As well, the memory footprint is decreased. In addition to collecting information on hot spots in the program, other types of information are gathered, such as data on caller-callee relationships for virtual method invocations. "

è **Question:** Why do I see the attribute in a derivative class? I can not access it anyway!

Answer: You can access protected attributes in an extended class or within of package where this class is.

è **Question:** Are classes implementing Externalizable considered Serializable classes?

Answer: Yes, since public interface Externalizable extends Serializable (according to API)

è **Question:** I tried to send Ctrl+Alt+Del key code to system from my class, but it does not work...

Answer: The problem is that this combination is served differently from all "normal" key combinations on hardware level. When you press those three buttons an interrupt of processor occurs. When an interrupt occur processor looks at its number and executes special program in specific memory location. If you want to use this key combination you need to find proper windows library and use it through JNI. Java does not serve this special case...

è **Question:** The beta release of J2SE 1.5 is scheduled for late 2003. I have heard that there will be some changes in a code writing process that will make our life easier...

Answer: Yes, indeed. The improvements to J2SE being developed in project "Tiger" are intended to make programs clearer, shorter, safer, and easier to develop, without sacrificing compatibility. Java 1.5 will shift the responsibility for writing the boilerplate code from the programmer to the compiler...
Example:

Instead of writing
String title = ((String) words.get(i)).toUppercase();

you will do like this:

```
String title = words.get(i).toUppercase();
```

and it will decrease the number of compile errors. Compiler will try to cast to String without saying it explicitly.

More about all coming changes you can read in this excellent article:
"New Language Features for Ease of Development in the Java 2 Platform, Standard Edition 1.5: A Conversation with Joshua Bloch" on:
http://java.sun.com/features/2003/05/bloch_qa.html

è **Question:** I would like to create a class that can execute JScripts. Does any one know if there is such class?

Answer: Yes, it is Rhino. Rhino is an open-source implementation of JavaScript written entirely in Java. It is typically embedded into Java applications to provide scripting to end users.
Be aware that it has some limitations, Rhino is an implementation of the core language only and doesn't contain objects or methods for manipulating HTML documents.

More read about Rhino here:
http://www.mozilla.org/rhino/

è **Question:** When do I to choose an abstract class over an interface?

Answer: Choosing interfaces and abstract classes is not an either/or proposition...

"Often in a design, you want the base class to present only an interface for its derived classes. That is, you don't want anyone to actually create an object of the base class, only to up cast to it so that its interface can be used. This is accomplished by making that class abstract using the abstract keyword. If anyone tries to make an object of an abstract class, the compiler prevents them. This is a tool to enforce a particular design.

You can also use the abstract keyword to describe a method that hasn't been implemented yet-as a stub indicating "here is an interface function for all types inherited from this class, but at this point I don't have any implementation for it." An abstract method may be created only inside an abstract class. When the class is inherited, that method must be implemented, or the inheriting class becomes abstract as well. Creating an abstract method allows you to put a method in an interface without being forced to provide a possibly meaningless body of code for that method.

The interface keyword takes the concept of an abstract class one step further by preventing any function definitions at all. The interface is a very handy and commonly used tool, as it provides the perfect separation of interface and implementation. In addition, you can combine many interfaces together, if you wish, whereas inheriting from multiple regular classes or abstract classes is not possible."

From "Thinking in Java", http://javafaq.nu/java/book/Chapter01.shtml#Heading29

è Question: What is difference between Hashmap and Hashtable?

Answer: there is almost no difference, since "hashmap is Hash table based implementation of the Map interface" (citation from API). The difference is that Hashtable is synchronized.

è Question: My colleague, he is testing my code, got "Headless Exception". He jokes that is thrown by code which is written by headless people :-) He has shown me log file even. Is he joking with me?

Answer: Probably not :-) I mean that you are Ok although this exception exists in reality.

It is thrown when code that is dependent on hardware like keyboard, display, or mouse is called in an environment that does not support a keyboard, display, or mouse. For example, all modern servers have no display, keyboard and mouse.

è Question: I know that Hashmap implementation is not synchronized. How can I synchronize it?

Answer: Really good answer often can be found in API!
"If multiple threads access this map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally. (A structural modification is any operation that adds or deletes one or more mappings; merely changing the value associated with a key that an instance already contains is not a structural modification.) This is typically accomplished by synchronizing on some object that naturally encapsulates the map. If no such object exists, the map should be "wrapped" using the Collections.synchronizedMap method. This is best done at creation time, to prevent accidental unsynchronized access to the map:

```
Map m = Collections.synchronizedMap(new HashMap(...));
```

The iterators returned by all of this class's "collection view methods" are fail-fast: if the map is structurally modified at any time after the iterator is created, in any way except through the iterator's own remove or add methods, the iterator will throw a ConcurrentModificationException. Thus, in the face of concurrent modification, the iterator fails quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future.

Note that the fail-fast behavior of an iterator cannot be guaranteed as it is, generally speaking, impossible to make any hard guarantees in the presence of unsynchronized concurrent modification. Fail-fast iterators throw ConcurrentModificationException on a best-effort basis. Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs."

From Java API (v 1.4.1)

è Question: When I use Java to write my program under some specific OS, I always set some environment variables to control the application program. For example, CLASSPATH.
What is about other variables that I should be aware of? And how can I set them?

Answer: Java defines own environment which is different from the default one provided by the underlying OS.

You can define your own "environment" by using

```
java -Dname1=myvalue1 -Dname2=myvalue2
```

è Question: Where can I find sample code for J2EE guidelines, patterns, and code for end-to-end Java applications?

Answer: J2EE Patterns catalogue is set of practical solutions to recurring problems!
J2EE Patterns not only help you build simpler systems that work, but they also help you build beautiful programs
J2EE Patterns catalogue is here:
http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html

è Question: What is date of Java birthday?

Answer: May 23, 1995. John Gage, director of the Science Office for Sun Microsystems, and Marc Andreessen, cofounder and executive vice president at Netscape, stepped onto a stage and announced to the SunWorld audience that Java technology was real, it was official, and it was going to be incorporated into Netscape Navigator, the world's portal to the Internet.

At that time, the entire Java technology team, not yet a division, numbered less than 30 people.

More about early history of Java please read here:
http://java.sun.com/features/1998/05/birthday.html

è Question: Why does not exist "JavaNotFoundException"?
When I deliver my Java program to customers often Java is not installed on theirs machines. Having such exception I could in better form (GUI popup, for example) inform my customers about the problem...

Answer: Is not it obvious that to be able to find the absence of Java with a Java program you need the Java :-)

n Java pitfalls: loop without braces.

Java allows loop without braces. Quite common mistake happens with such code:

```
while (condition)

    statement1; statement2
```

Many think that both statements will be in the body of while loop. Only statement1 is in the body.

Null body loops like this:

```
while (condition); //  infinite loop

   statement;
```

runs forever!!!

è Good to know: In releases prior to 1.4.2, it was possible to create an invalid URL instance such as the following: new URL ("http://www.foo.com:-1/index.html"). Specifying a port number of -1 is not valid. However, in previous releases the -1 was stripped off when sending such URLS with HTTP commands. In 1.4.2, the -1 is left in the URL and this may cause problems for some HTTP servers or proxies. Applications should not create URL instances with a -1 port number. Assuming the -1 refers to the default port in this example, the correct way to create this URL is new URL ("http://www.foo.com/index.html"). This note only applies to the single argument URL constructor.

è Question: I was testing the new J2RE 1.4.2 for compatibility with our programs (written on Java 1.2.2 version) and found that the Installation of the J2RE on Windows has changed, with impact on the supported locales...

Answer: Yes, you are right. this is written in release notes for J2RE 1.4.2:
There now is a single installer, which by default installs a runtime with support for European languages if the Windows host system only supports such languages, or a runtime with support for all languages if the Windows host system supports at least one non-European language. Users can request installation of additional languages in a custom setup. For complete information on which locales and encodings are supported with which installation, see:

the Supported Locales
http://java.sun.com/j2se/1.4.2/docs/guide/intl/locale.doc.html

and Supported Encodings documents
http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html

è Question: Why SUN does not introduce dynamic policies (for security)?

Answer: Support for dynamic policies has been added. In Java 2 SDK releases prior to version 1.4, classes were statically bound with permissions by querying security policy during class loading. The lifetime of this binding was scoped by the lifetime of the class loader. In version 1.4 this binding is now deferred until needed by a security check. The lifetime of the binding is now scoped by the lifetime of the security policy.

è Question: Can I use system provided random numbers generators when I run my Java program on Windows? Not that that provided by SUN?

Answer: The Sun SecureRandom implementation now also makes use of an operating system-provided entropy source on Windows platforms, which can improve the start-up time of cryptographic applications considerably. Edit the

<java.home>/lib/security/java.security to control this feature.

è Question: What are enhancements were added to new javac since 1.4.2?

Answer: The following enhancements and bug fixes have been made to the javac source compiler in the Java 2 SDK.
RFE 4626998 implemented: Added -cp synonym for -classpath option to set classpath.
RFE 4515858 implemented: Added -Xbootclasspath/p: option to prepend to bootclasspath.
RFE 4253402 implemented: Added -Xmaxerrors and -Xmaxwarns to increase the number of errors and warnings printed by javac.

Also optimizations have been made to the core libraries to reduce start-up time. For a small command line application, start-up time has been reduced by roughly thirty percent; for a small Swing application, by fifteen to twenty percent.
In some rare situations, one optimization related to filename canonicalization may cause inconsistent views of the host platform's file system to be briefly visible to Java applications. The optimization caches, for a short time, the results of calls to File.getCanonicalFile() and File.getCanonicalPath(). If a file is moved by another application then the canonicalization result for that file may briefly differ from its true value, although attempts to open and access the file will succeed or fail as before. Applications should not rely on the presence or absence of such inconsistencies. If an application must disable this optimization for correctness purposes then the system property -Dsun.io.useCanonCaches=false may be specified.

è Good to know: Performance testing has shown that the finalize methods in the channel classes impose a significant GC penalty upon server-type programs that must deal with thousands of new open channels per second.

And now in J2SE 1.4.2the finalize methods of the primitive channel classes (SocketChannel, ServerSocketChannel, etc.) have been removed . Additional performance test measurements revealed that the presence of these methods imposed a significant (factor of 100) garbage-collection penalty upon server-type applications that process thousands of new open channels per second. User code that deals in channels should be carefully written to close channels explicitly rather than rely upon the garbage collector to do so.

è Question: If I turn off daylight saving during an installation of Windows XP then Java runtime detects the platform time zone in the GMT offset format (e.g., "GMT+09:00"), not as a time zone ID (e.g., "Asia/Tokyo"). What is fix for that problem?

Answer: To fix this installation problem, take the following steps after the installation (unless you intend to turn off the setting):
1. Open Date/Time in Control Panel.
2. On the Time Zone tab, choose a time zone that observes daylight saving time (e.g., "(GMT-08:00) Pacific Time (US & Canada); Tijuana"), select the "Automatically adjust clock for daylight saving changes" check box, and press the Apply button.
3. Choose your time zone back and press the OK button.

è Question: I am getting strange artefacts when rendering Asian text in a TextArea component. I run Japanese version of Windows NT...

Answer: This problem occurs if you install English version of SW onto non-Latin versions of NT. For example, Visual C++ or Office 97.
The problem is caused when the installation of those programs replaces the Asian DLL Riched32.dll file with the English version of the same file. The problem can be corrected by replacing Riched32.dll with the Asian version.

è Question: If I run my applet with the SUN VM I get exception: "Either "code" or "object" should be specified, but not both." It does not happen with the Microsoft VM. Part 1

Answer: I found good recommendations on Sun's website about how to install latest browsers and avoid possible problems with running applets in different browsers:

"This is a compatibility issue. To avoid the exception with the Sun VM, do not use an attribute named object with the PARAM element in an applet.
For Netscape 7, or Mozilla 1.1 or later, running on Microsoft Windows, the browser will always use Sun's latest JRE—even if you have deselected the "Netscape 6 or later" or "Mozilla 1.1 or later" check box in the Browser tab of the Java Plug-in Control Panel. This is due to the auto-scan features of these browsers. For instance, if Netscape 7 is installed on a Windows machine that has Sun's 1.4.2 JRE installed as the latest version and you deselect "Netscape 6 or later" in the Control Panel, Netscape 7 will still run Sun's 1.4.2 JRE (Java Plug-in).

Using Java Plug-in 1.4.2 in conjunction with Netscape TM 6.x browsers can cause erratic behavior, including closure of the browser window, for applets that invoke applet functions from JavaScript. This problem is due to a bug in the browser, and you can avoid this problem by using Netscape 7.0 browsers.

è Question: If I run my applet with the SUN VM I get exception: "Either "code" or "object" should be specified, but not both." It does not happen with the Microsoft VM. Part 2

The Netscape 7 installer provides three choices for installation: Recommended, Full, and Custom. The Java Plug-in may not work properly in Recommended or Full installations of Netscape 7 browsers. Such Netscape 7 browsers may close when attempting to launch an applet. To avoid this problem, install Netscape 7 using the Custom option, and choose not to install the Quality Feedback Agent.

When installing Java Plug-in, it is necessary to shut down and then restart your web browser. On Microsoft Windows platforms, Netscape 7 has a Quick Launch feature that keeps it running in the background so that it can be launched with minimal start-up time from a Quick Launch icon in the system tray. To completely shut down Netscape 7, you must exit Netscape 7 from its Quick Launch icon in the system tray.
"

This tip is based on:
http://java.sun.com/j2se/1.4.2/relnotes.html#vm

è Good to know: If your system does take in external XML data this security tip is for you!

While XML does not allow recursive entity definitions, it does permit nested entity definitions, which produces the potential for Denial of Service attacks on a server which accepts XML data from external sources. For example, a SOAP document like the following that has extremely deeply nested entity definitions can consume 100% of CPU time and a lot of memory in entity expansions.

```
<?xml version="1.0" encoding ="UTF-8"?>
<!DOCTYPE foobar[
<!ENTITY x100 "foobar">
<!ENTITY x99 "&x100;&x100;">
<!ENTITY x98 "&x99;&x99;">
...
<!ENTITY x2 "&x3;&x3;">
<!ENTITY x1 "&x2;&x2;">
]>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=...>
<SOAP-ENV:Body>
<ns1:aaa xmlns:ns1="urn:aaa" SOAP-ENV:encodingStyle="...">
<foobar xsi:type="xsd:string">&x1;</foobar>
</ns1:aaa>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

A system that doesn't take in external XML data need not be concerned with issue, but one that does can utilize one of the following safeguards to prevent the problem:
New system property to limit entity expansion
The entityExpansionLimit system property lets existing applications constrain the total number of entity expansions without recompiling the code. The parser throws a fatal error once it has reached the entity expansion limit. (By default, no limit is set, because such a constraint would make the XML parser incompatible with the XML 1.0 specification.)

To set the entity expansion limit using the system property, use an option like the following on the java command line: -DentityExpansionLimit=100000

New parser property to disallow DTDs
The application can also set the http://apache.org/xml/features/disallow-doctype-decl parser property to true. A fatal error is then thrown if the incoming XML document contains a DOCTYPE declaration. (The default value for this property is false.) This property is typically useful for SOAP based applications where a SOAP message must not contain a Document Type Declaration.

This tip is based on:
http://java.sun.com/j2se/1.4.2/relnotes.html#jaxp

è Question: In Ukraine first day of week is Monday but not Sunday

By defaults in the calendar object obtained with locale ua_UA getFirstDayOfWeek () function return SUNDAY as a first week day... Java version is 1.3.1_01

Answer: Yes indeed, according to this list:
http://www.pjh2.de/datetime/weeknumber/wnd.php
the first day in Ukraine is Monday. For those who does not know will say that in many countries Sunday or even Saturday is first day, for example in Egypt - Saturday and in Argentina and Brasil - Sunday...

It was a bug since according to ISO 8601 in all European countries a week starts with Monday...
I know that with Hungarian calendar was the same bug... In latest 1.4.1 and beyond the bug is fixed.

è Question: We compile a large number of files and we have strange problem: javac -deprecation doesn't print a warning of a usage in file A.java of a deprecated method in B.java if B.java is being
compiled at the same time as A.java.
It is big problem since we want to know if we use deprecated code...

Answer: Yes the problem presents in all Java versions since 1.1.7 and then in 1.2, 1.3 and 1.4
You can compile file by file, although it is not convenient :-(
I hope this will be corrected soon by Sun!

è Question: How can I completely clean up my Windows XP from Microsoft JVM before I install Sun Microsystems JVM?

Answer: You might want to remove the Microsoft JVM, which Microsoft no longer supports, in favour of the more recent Sun Microsystems JVM. To remove the Microsoft JVM, perform the following steps:
1. From the Start menu, select Run.
2. Enter the command
RunDll32 advpack.dll,LaunchINFSection java.inf,UnInstall
to start the uninstall process
3. Click Yes to the confirmation, then select Reboot.
4. After the machine restarts, delete the following items:
the \%systemroot%\java folder
java.pnf from the \%systemroot%\inf folder
jview.exe and wjview.exe from the \%systemroot%\system32 folder
The HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Java VM registry subkey
The HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\AdvancedOptions\JAVA_VM registry subkey (to remove the Microsoft Internet Explorer (IE) options)

Now you can get latest JVM from Sun!

è Question: When does garbage collection performance matter to the user?

Answer: For many applications it doesn't. That is, the application can perform within its specifications in the presence of garbage collection with pauses of modest frequency and duration. An example where this is not the case (when the default collector is used) would be a large application that scales well to large number of threads, processors, sockets, and a large amount of memory.

n Question: Can my program interact with Garbage Collector and get notified about a status of object (going to be GCollected, already collected)?

Answer: You should use java.lang.ref package!

This package provides reference-object classes, which support a limited degree of interaction with the garbage collector. A program may use a reference object to

maintain a reference to some other object in such a way that the latter object may still be reclaimed by the collector. A program may also arrange to be notified some time after the collector has determined that the reachability of a given object has changed.

Going from strongest to weakest, the different levels of reachability reflect the life cycle of an object. They are operationally defined as follows:

> An object is strongly reachable if it can be reached by some thread without traversing any reference objects. A newly-created object is strongly reachable by the thread that created it.
>
> An object is softly reachable if it is not strongly reachable but can be reached by traversing a soft reference.
>
> An object is weakly reachable if it is neither strongly nor softly reachable but can be reached by traversing a weak reference. When the weak references to a weakly-reachable object are cleared, the object becomes eligible for finalization.
>
> An object is phantom reachable if it is neither strongly, softly, nor weakly reachable, it has been finalized, and some phantom reference refers to it.
>
> Finally, an object is unreachable, and therefore eligible for reclamation, when it is not reachable in any of the above ways.

Based on Java API description

è Question: What is young generation in GC (Garbage Collector)?

Answer: In the J2SE platform version 1.4.1 two new garbage collectors were introduced to make a total of four garbage collectors from which to choose.

Beginning with the J2SE platform, version 1.2, the virtual machine incorporated a number of different garbage collection algorithms that are combined using generational collection. While naive garbage collection examines every live object in the heap, generational collection exploits several empirically observed properties of most applications to avoid extra work.

The default collector in HotSpot has two generations: the young generation and the tenured generation. Most allocations are done in the young generation. The young generation is optimized for objects that have a short lifetime relative to the interval between collections. Objects that survive several collections in the young generation are moved to the tenured generation. The young generation is typically smaller and is collected more often. The tenured generation is typically larger and collected less often.

The young generation collector is a copying collector. The young generation is divided into 3 spaces: eden-space, to-space, and from-space. Allocations are done from eden-space and from-space. When those are full a young generation is collection is done. The expectation is that most of the objects are garbage and any surviving objects can be copied to to-space. If there are more surviving objects than

can fit into to-space, the remaining objects are copied into the tenured generation. There is an option to collect the young generation in parallel.

The tenured generation is collected with a mark-sweep-compact collection. There is an option to collect the tenured generation concurrently.

This tip is based on FAQ about Garbage Collection in the HotSpot Java Virtual Machine
http://java.sun.com/docs/hotspot/gc1.4.2/faq.html
and article "Tuning Garbage Collection with the 1.4.2 Java Virtual Machine"
http://java.sun.com/docs/hotspot/gc1.4.2/

è Question: What are additional types of GC do exist in the J2SE platform, version 1.4.2?

Answer:

The throughput collector: this collector uses a parallel version of the young generation collector. It is used if the -XX:+UseParallelGC option is passed on the command line. The tenured generation collector is the same as the default collector.
The concurrent low pause collector: this collector is used if the -XX:+UseConcMarkSweepGC is passed on the command line. The concurrent collector is used to collect the tenured generation and does most of the collection concurrently with the execution of the application. The application is paused for short periods during the collection. A parallel version of the young generation copying collector is used with the concurrent collector if the combination -XX:+UseConcMarkSweepGC -XX:+UseParNewGC is passed on the command line.
This kind is also known as concurrent GC.
The incremental (sometimes called train) low pause collector: this collector is used only if -Xincgc is passed on the command line. By careful bookkeeping, the incremental garbage collector collects just a portion of the tenured generation at each minor collection, trying to spread the large pause of a major collection over many minor collections. However, it is even slower than the default tenured generation collector when considering overall throughput.

Important! -XX:+UseParallelGC should not be used with -XX:+UseConcMarkSweepGC. The argument parsing in the J2SE platform, version 1.4.2 should only allow legal combinations of command line options for garbage collectors, but earlier releases may not detect all illegal combinations and the results for illegal combinations are unpredictable.

Also important to remember: always try default GC before using additional GCs!

è Question: What type of garbage collection does a System.gc() do?

Answer: An explicit request to do a garbage collection does a full collection (both young generation and tenured generation). A full collection is always done with the application paused for the duration of the collection.

è Question: My application runs much slower with 1.3/1.4 when compared with the 1.2 Production release for Solaris. Why?

Answer: Assuming that you are running a lot of bytecode make sure that you are using the correct mode of the virtual machine. For applications where small footprint and fast start up are important, use -client. For applications where overall performance is the most important, use -server. Don't forget that -server or -client must be the first argument to java, -client is the default. If this isn't your problem, read on for more tuning parameters you can try...

from the Java Hotspot Virtual Machine FAQ:
http://java.sun.com/docs/hotspot/PerformanceFAQ.html

è Question: What's the difference between the -client and -server systems (options in JVM)?

Answer: These two systems are different binaries. They are essentially two different compilers (JITs) interfacing to the same runtime system. The client system is optimal for applications which need fast start-up times or small footprints, the server system is optimal for applications where the performance is most important. In general the client system is better on GUIs. Some of the other differences include the compilation policy used, heap defaults, and inlining policy.

from the Java Hotspot Virtual Machine FAQ:
http://java.sun.com/docs/hotspot/PerformanceFAQ.html

è Question: With 1.3.1 we got 4GB heaps on Solaris, why can't I get this to work on Windows?

Answer: It's due to fragmentation of the address space. After Windows has loaded all its stuff, plus the java dlls, the largest available continuous chunk of addresses for the heap is about 1.4-1.6G on Win/NT. Might be smaller on Win/98.

If you really need more space they can try rebasing java.exe and the JDK dlls higher. But that won't buy you much (maybe a couple hundred meg).
from the Java Hotspot Virtual Machine FAQ:
http://java.sun.com/docs/hotspot/PerformanceFAQ.html

è Question: Is it possible to use RSS ("Really Simple Syndication") together with JSP?

Answer: Yes, SUN recently developed RSS Utilities package - a JSP Tag Library to be used by anybody with a basic understanding of RSS, JavaServer Pages, and HTML. The taglib is mostly geared towards non-technical editors of web sites that use RSS for aggregating news content.
It contains a set of custom JSP tags which make up the RSS Utilities Tag library, and a flexible RSS Parser.

Please read more here:
http://developer.java.sun.com/developer/technicalArticles/javaserverpages/rss_utilities/

è Question: I use an old Java Tool from SUN. Today I read that SUN started End Of Life Process for this Java tool.
What does it mean? Does it mean that when the process will be finished my tool will stop to work? What is about customers?

Answer: Everything is Ok! Life continues and the tool will work :-)
During the EOL transition period, all announced products will be supported as per existing customer support agreements. After the EOL transition period, these products will no longer be supported by Sun. Products that have completed the EOL process will be moved to the Archive area here:
http://java.sun.com/products/archive/index.html

è Question: I use the MediaTracker class is a utility class to track the status of a number of media objects. I need to load images over the net. How could I check the progress and know when loading is finished?

Answer: Use an asynchronous update interface for receiving notifications about Image information as the Image is constructed - ImageObserver. The imageUpdate method gets called with infoflag 8 (SOMEBITS) every time when more pixels arrive. And with infoflag 32 (ALLBITS) when loading is complete.

--
alex

è Question: ArrayList vs. LinkedList

Answer: The List interface defines an ordered collection of elements. Available implementations include ArrayList and LinkedList, where both implement the List interface. When your program needs to work with a List, it should not care if it is an ArrayList or a LinkedList, only that whatever class is used provides a known behavior. That behavior is the interface.

If your program frequently provides random access to the data of the list, the ArrayList class offers quick access to individual elements of the list. This quick access comes at a cost of slower operations for adding and removing in the middle of the list.
If this latter behavior is what you desire, than the LinkedList class offers a better alternative. It provides quick sequential access, additions, and deletes, at a cost of slower random access.

The tip based on Sun's newsletter

è Question: How does JVM interact with profilers?

Answer: The Java Virtual Machine has Profiler Interface (JVMPI) in the Java 2 SDK, Standard Edition. This interface is an experimental feature in the Java 2 SDK. The JVMPI is not yet a standard profiling interface. It is intended for tools vendors to develop profilers that work in conjunction with Sun's Java virtual machine implementation.

"The JVMPI is a two-way function call interface between the Java virtual machine and an in-process profiler agent. On one hand, the virtual machine notifies the profiler agent of various events, corresponding to, for example, heap allocation, thread start, etc. On the other hand, the profiler agent issues controls and requests for more information through the JVMPI. For example, the profiler agent can turn on/off a specific event notification, based on the needs of the profiler front-end.

The profiler front-end may or may not run in the same process as the profiler agent.

It may reside in a different process on the same machine, or on a remote machine connected via the network. The JVMPI does not specify a standard wire protocol. Tools vendors may design wire protocols suitable for the needs of different profiler front-ends.

A profiling tool based on JVMPI can obtain a variety of information such as heavy memory allocation sites, CPU usage hot-spots, unnecessary object retention, and monitor contention, for a comprehensive performance analysis.
JVMPI supports partial profiling, i.e. a user can selectively profile an application for certain subsets of the time the virtual machine is up and can also choose to obtain only certain types of profiling information.

In the current version of JVMPI, only one agent per virtual machine can be supported." Details can be found here:
http://java.sun.com/j2se/1.4.1/docs/guide/jvmpi/jvmpi.html

è Question: Could you tell me about chain exceptions?

Answer: It is common for Java code to catch one exception and throw another:

```
try {
...
} catch(YourException e) {
  throw new MyException();
}
```

Unfortunately, the information contained in the "causative exception" (YourException in the example above) is generally lost, which greatly complicates debugging.

The version 1.4 libraries add a unified, general-purpose mechanism to record such information. The pre-existing ad-hoc facilities have been retrofitted to use the new technique. Two new constructors were added to Throwable. Each takes another Throwable as the cause.

```
Throwable(Throwable cause)
Throwable(String message, Throwable cause)
```

Common exceptions classes have had analogous new constructors added. The intended usage is:

```
try {
  lowLevelOp();
} catch (LowLevelException le) {
  throw new HighLevelException(le);
}
```

Legacy exception classes can use a modified form of new idiom through the Throwable initCause(Throwable cause) method, such as:

```
try {
    lowLevelOp();
} catch (LowLevelException le) {
    throw new (HighLevelException)
    HighLevelException().initCause(le);
}
```

The cause information can be retrieved through the method Throwable getCause(). For more information, see:
http://java.sun.com/j2se/1.4/docs/guide/lang/chainedexceptions.html

è **Question:** How can my application know how many CPUs work in a system?

Answer: Use 1.4.x JDK... The method activeProcessorCount() was added to the java.lang.Runtime class.
Sun Microsystems, Inc. 12 of 37
The activeProcessorCount() method returns the number of hardware processors currently available to the VM. Large applications will typically use
this value to calculate the number of worker threads they should create. On some operating systems, such as the Solaris operating environment (OE) and Microsoft Windows NT and 2000, the processor count may change dynamically.

An application will, in general, have to poll to determine when the count changes.

The tip is based on Sun’s "Version 1.4 Core Libraries" description

è **Question:** I read that in the Java (version 1.4.x) reflection implementation has been optimized for speed and it is significantly faster now...?

Answer: The APIs and semantics are the same as in previous J2SE versions. There are three key optimizations in the new implementation:

Caching of results of Class.getFields(), Class.getMethods(), and related routines
Dynamic generation of bytecodes for Method.invoke() and

Constructor.newInstance()

Optimized security checks for reflective calls

Due to these optimizations, reflective method calls are between 20 and 25 times faster, and reflective object creation is between 5 and 6 times faster. Reflection intensive code, such as serialization, has benefited from this additional speed.

The tip is based on Sun’s "Version 1.4 Core Libraries" description

è **Good to know:** JFluid - a profiling tool for the Java programming language that allows you to profile an arbitrary subset of your program, that can be changed on-the-fly, while the program is running. This is a capability not available in any other profiling tool for Java at this time. Since JFluid relies on our unique dynamic bytecode instrumentation mechanism, it can run only on our specially modified HotSpot VM (currently derived from JDK(tm) version 1.4.2).

In JFluid, you can select your application's methods to profile by pointing at a single method. This method is then treated as a call subgraph root. JFluid instruments this method, and also all methods that it calls, transitively. In other words, it dynamically reveals a whole call subgraph. Special bytecode packets are injected into these methods on-the-fly, so that they start to generate profiling events such as method

entry and exit. These events are recorded, and then the obtained performance data is displayed in the standard forms of context call tree (what you see e.g. in OptimizeIt) and sorted accumulated method time list.

Alternatively, you can instrument and profile an arbitrary code fragment within a method. This kind of instrumentation is extremely lightweight and inobtrusive, therefore in this way you can obtain very precise CPU time measurements.

Finally, you can activate and deactivate memory profiling that allows you to see the number and types of all objects allocated by the program, the locations where these objects are allocated, and the call paths at the moment of allocation. We are using a special semi-statistical technique for gathering information about call paths that results in a highly controllable, and usually quite modest, overhead when collecting this kind of information.

To learn more about the JFluid tool, read the documentation page (http://research.sun.com/projects/jfluid/help.html) (this is the same page that is available from within the tool when you hit "Help"). Click here (http://research.sun.com/projects/jfluid/download/) to download JFluid.

Future:
At present JFluid is an experimental technology. The possibility of productizing the JVM changes necessary to run JFluid, as well as the tool itself, is being discussed, but no firm decisions have been taken yet. Note that if the JVM API that is currently used by JFluid becomes a standard, it could then be used by any tool from any vendor, for example for inobtrusive EJB application monitoring, fault injection, and other interesting things.
All info is here:
http://research.sun.com/projects/jfluid/

è Question: What is Signal Chaining in JVM?

Answer: The Java TM 2 SDK, Standard Edition, (J2SDK) v1.4 contains a new signal-chaining facility. Signal-chaining enables the Java Platform to better interoperate with native code that installs its own signal handlers. The facility works on both Solaris and Linux platforms.

The signal-chaining facility was introduced to remedy a problem with signal handling in previous versions of the Java HotSpot VM. Prior to version 1.4, the Java HotSpot VM would not allow application-installed signal handlers for certain signals including, for example, SIGBUS, SIGSEGV, SIGILL, etc, since those signal handlers could conflict with the signal handlers used internally by the Java HotSpot VM.

The signal-chaining facility offers:

A. Support for pre-installed signal handlers when the HotSpot VM is created.
B. Support for signal handler installation after the HotSpot VM is created, inside JNI code or from another native thread.

Pre-installed signal handlers (A) are supported by means of saving existing signal handlers, for signals that are used by the VM, when the VM is first created. Later,

when any of these signals are raised and found not to be targeted at the Java HotSpot VM, the pre-installed handlers are invoked. In other words, pre-installed handlers are "chained" behind the VM handlers for these signals.

more please read here:
http://java.sun.com/j2se/1.4.1/docs/guide/vm/signal-chaining.html

è Question: How can I use the new error-reporting mechanism in JVM?

Answer: When the Java Virtual Machine (JVM) detects a crash in native code such as JNI code written by the developer or when the JVM itself crashes, it will print debug information about the crash. This error message normally will include information such as the function name, library name, source-file name and line number at which the error occurred. (Currently, information about file name and line number are available only available on Microsoft Windows platforms.) For an example of a message emitted by the error handler, see JNI Error example.
The information provided by the new error-reporting mechanism will allow developers to more easily and efficiently debug their applications. If an error message indicates a problem in the JVM code itself, it will allow a developer to submit a more accurate and helpful bug report.

Sometimes the error-reporting mechanism won't be able to determine the information that might be helpful in locating the source of a crash. To get the most out of the error handler, developers should be aware of the following guidelines and limitations.

Compile in debug mode

In some circumstances, the error-reporting mechanism won't be able to determine symbol names. The most common reason for this is that the binary code that crashed was not compiled in debug mode and therefore doesn't have symbol tables. Developers should compile their code in debug mode to ensure that it contains the necessary debug information. In Visual Studio, for example, this means choosing "Debug" rather than "Release" as the project's build mode. When using gcc or cc on Linux or on the Solaris TM operating environment, compile using the -g command-line option.

More limitations are described in the article here:
http://java.sun.com/j2se/1.4.1/docs/guide/vm/error-handling.html

è Question: I understand that the XML is some kind of language to describe documents. Why do we need the XML for data exchange between systems?

Answer: XML is used to aid the exchange of data. It makes it possible to define data in a clear way.
Both the sending and the receiving party will use XML to understand the kind of data that's been sent. By using XML everybody knows that the same interpretation of the data is used.

XML makes communication easy. It's a great tool for transactions between businesses.
But it has much more possibilities.
You can define other languages with XML. A good example is WML (Wireless Markup

Language), the language used in WAP-communications.
WML is just an XML dialect.

Web services depend on the ability of parties to communicate with each other even if they are using different information systems. XML (Extensible Markup Language), a markup language that makes data portable, is a key technology in addressing this need. Enterprises have discovered the benefits of using XML for the integration of data both internally for sharing legacy data among departments and externally for sharing data with other enterprises. As a result, XML is increasingly being used for enterprise integration applications, both in tightly coupled and loosely coupled systems. Because of this data integration ability, XML has become the underpinning for Web-related computing.

è Question: What is an XML registry and JAXR?

Answer: XML registries are a vital component in the implementation of shared data exchanges. Developers looking to express information using XML need support in establishing common lexicons and grammars. This Registry enables the consistent use of XML, both vertically within projects and horizontally across organizations.

JAXR (Java API for XML Registries) enables Java software programmers to use a single, easy-to-use abstraction API to access a variety of XML registries. A unified JAXR information model describes content and metadata within XML registries.

JAXR gives developers the ability to write registry client programs that are portable across different target registries. JAXR also enables value-added capabilities beyond those of the underlying registries.

The current version of the JAXR specification includes detailed bindings between the JAXR information model and both the ebXML Registry and the UDDI version 2 specifications. You can find the latest version of the specification at:
http://java.sun.com/xml/downloads/jaxr.html

è Question: What is meant by "transformation" in JAXP?

Answer: The Extensible Stylesheet Language (XSL) includes both a transformation language and a formatting language. Each of these, naturally enough, is an XML application. The transformation language provides elements that define rules for how one XML document is transformed into another XML document. The transformed XML document may use the markup and DTD of the original document or it may use a completely different set of elements. In particular, it may use the elements defined by the second part of XSL, the formatting objects.

è Question: How do I use JavaBeans components (beans) from a JSP page?

Answer: The JSP specification includes standard tags for bean use and manipulation. The useBean tag creates an instance of a specific JavaBeans class. If the instance already exists, it is retrieved. Otherwise, it is created. The setProperty and getProperty tags let you manipulate properties of the given object. These tags are described in more detail in the JSP specification and tutorial.
tip is from the JSP FAQ: http://java.sun.com/products/jsp/faq.html

è Question: What is Model-View-Controller (MVC) architecture?

Answer: The MVC paradigm (design pattern) is a way of breaking an application, or even just a piece of an application's interface, into three parts: the model, the view, and the controller. MVC was originally developed to map the traditional input, processing, output roles into the GUI realm:

Input --> Processing --> Output
Controller --> Model --> View

The user input, the modelling of the external world, and the visual feedback to the user are separated and handled by model, viewport and controller objects. The controller interprets mouse and keyboard inputs from the user and maps these user actions into commands that are sent to the model and/or viewport to effect the appropriate change. The model manages one or more data elements, responds to queries about its state, and responds to instructions to change state. The viewport manages a rectangular area of the display and is responsible for presenting data to the user through a combination of graphics and text.

Separating responsibilities among model, view, and controller objects reduces code duplication and makes applications easier to maintain. It also makes handling data easier, whether adding new data sources or changing data presentation, because business logic is kept separate from data. It is easier to support new client types, because it is not necessary to change the business logic with the addition of each new type of client.

Source: comp.object newsgroup and Sun’s web site.

è Question: Where can I find brief description for common set of design patterns which I can use in my application (J2EE platform)? Part 1

Answer: look at this list below:

Intercepting filter--This pattern applies to request pre- and post-processing. It applies additional services needed to process a request. For example, an intercepting filter such as a servlet filter may handle all incoming requests to the Web site and provide a central mechanism for authorization.

View helper--A view helper encapsulates the presentation and data access logic portions of a view, thus refining the view and keeping it simpler. Presentation logic concerns formatting data for display on a page, while data access logic involves retrieving data. View helpers are often JSP tags for rendering or representing data and JavaBeans for retrieving data.

Composite view--This pattern makes view presentation more manageable by creating a template to handle common page elements for a view. Often, Web pages contain a combination of dynamic content and static elements, such as a header, footer, logo, background, and so forth. The dynamic portion is particular to a page, but the static elements are the same on every page. The composite view template captures the common features.

Front controller--This pattern provides a centralized controller for managing requests. A front controller receives all incoming client requests, forwards each request to an appropriate request handler, and presents an appropriate response to the client.

è Question: Where can I find brief description for common set of design patterns which I can use in my application (J2EE platform)? Part 2

Value object--This pattern facilitates data exchange between tiers (usually the Web and EJB tiers) by reducing the cost of distributed communication. In one remote call, a single value object can be used to retrieve a set of related data, which then is available locally to the client. See Chapter 5 for more information on value objects.

Session facade--This pattern coordinates operations between cooperating business objects, unifying application functions into a single, simplified interface for presentation to the calling code. It encapsulates and hides the complexity of classes that must cooperate in specific, possibly complex ways, and isolates its callers from business object implementation changes. A session facade, usually implemented as a session bean, hides the interactions of underlying enterprise beans.

Business delegate--This pattern intervenes between a remote business object and its client, adapting the business object's interface to a friendlier interface for the client. It decouples the Web tier presentation logic from the EJB tier by providing a facade or proxy to the EJB tier services. The delegate takes care of lower-level details, such as looking up remote objects and handling remote exceptions, and may perform performance optimizations, such as caching data retrieved from remote objects to reduce the number of remote calls.

Data access object--This pattern abstracts data access logic to specific resources. It separates the interfaces to systems resource from the underlying strategy used to access that resource. By encapsulating data access calls, data access objects facilitate adapting data access to different schemas or database types. See Chapters 5 and 6 for more information on data access objects.

I found this list which was published in "Designing Enterprise Applications with the J2EETM Platform, Second Edition" book and available online here: http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/DEA2eTOC.html

è Question: I am trying to make my code compact. I already "JARed" all files... Is it possible to make the JAR even smaller?

Answer: Yes, it is possible to make a JAR file smaller by using code obfuscators. Most of the modern obfuscators replacing long class/method, variables names (for example: MyFavoriteSistersBlueClass) by shorter, just 2-3 characters...
Use it on final step, before delivery to customer, not for further development, since this procedure makes a code difficult to read and follow... In some cases a size can be reduced up to 30%!!!

è Question: Can you tell me about "100 Percent Pure Java Certification Program ". My project leader read somewhere that our SW should pass this certification. He also says that code for this program can be written only by Java certified programmers. I do not want to certify myself; I believe that I know already Java (working 3 years in the industry).

Answer: You can sleep easily, the Sun's "100 Percent Pure Java Certification Program " is over and is no longer supported by Sun. SUN used the program from

the beginning to make the Java popular and clear (for example to keep away from Microsoft's influence).

è Question: Do you know any free Java editor for the Java?

Answer: jEdit! Giving you an overview I took the info from JEdit home page:
"Overview
jEdit is a mature and well-designed programmer's text editor that has been in development for over 5 years.

While jEdit beats many expensive development tools for features and ease of use, it is released as free software with full source code, provided under the terms of the GNU General Public License.

The core of jEdit is primarily developed by Slava Pestov, and jEdit plugins are being written by a large and diverse team of programmers from around the world.

Some of jEdit's features include:

- Written in Java, so it runs on MacOS X, OS/2, UNIX, VMS and Windows.

- Built-in macro language; extensible plugin architecture. Dozens of macros and plugins are available.

- Plugins can be downloaded and installed from within jEdit using the "plugin manager" feature.

- Auto indent, and syntax highlighting for more than 80 languages.

- Supports a large number of character encodings including UTF8 and Unicode.
- Folding for selectively hiding regions of text.

- Word wrap.

- Highly configurable and customizable.

Every other feature, both basic and advanced, you would expect to find in a text editor. See the Features page for a full list.

See all the features here: http://www.jedit.org/index.php?page=features "
Home page for the project is here: http://www.jedit.org/

è Question: I have found that ZipEntry() method behaves strange: it does not convert filenames from Unicode to platform specifics

Answer: yes, it is known Java bug and discovered in the next version of Java 1.2.1, 1.2.2, 1.3, 1.3.1, 1.4.0_01, 1.4.1..

è Question: In Socket class exists method "getOOBInline" that checks if OOBINLINE is enabled. What is "OOBINLINE" and can I do with it?

Answer: OOB stands for Out-Of-Band Data.

The TCP/IP protocol allows users to send data asynchronously, otherwise known as OUT-OF-BAND data. When using this feature, the operating system interrupts the receiving process if this process has chosen to be notified about out-of-band data. The receiver can grab this input without affecting any information currently queued on the socket.
Therefore, you can use this without interfering with any current activity due to other wire and remote interfaces.

Unfortunately, most implementations of TCP/IP are broken, so use of out-of-band data is limited for safety reasons. You can only reliably send one character at a time. What is about safety? Let's look at this scenario: While a server is working on a task, the server is sensitive to interrupts. An interrupt will cause the server to abort the current task and begin listening for new instructions.
Interrupts can be sent to the server by sending an interrupt ("INT") to the process number of the server with the Unix kill command. The server socket will also raise an interrupt if out of band (OOB) data are received. Thus, the client can send a single arbitrary byte of OOB data to generate an interrupt. So, if someone will send much OOB data to a server it can seriously decrease performance of server...

On one of hacker's web sites I found "OOB Attack Program" that described like this: "This is a good program that exploits the oob attack on port 139 of any Windows machine running MS TCP/IP. Allows for multiple IP addresses to be attacked.
Coded for fun by: Hiro Protagonist "

If you are developing a server that will be exposed to Internet remember that some people "for fun" can take it down because OOBINLINE is enabled.

è Question: I found that in Java API (when looked at v1.4.2) exist "PrivilegedExceptionAction" and "PrivilegedActionException". It takes a while to try to distinguish the meanings of two names. Why naming of classes is done so bad and confuses people. Is not it enough another words?

No answer, just comment: I think it is good sign :-) Java became big enough and has almost all possible methods and classes. We feel the luck of words to describe a wealth of Java :-)

n Question: I was using Java 1.4.x during of development of program. Our customers want to use this version of Java because it causes some compatibility troubles for them... What could I do? I tried to insist on changing of Java but it did not help :-(

è Answer: You can still use newer version of Java, but when you compile use the option "-target". For example:
javac -target 1.3

è Question: Where could I buy gifts to my colleagues (t-shirts, glasses, caps, mugs and so on) with Java logo?

Answer: One place which I know is Java certified :-) since the link I found on Sun's home page) here:
http://www201.ikiosk.com/cgi-shl/index.cgi?shop=33293&date=2002%3A04%3A26%3A10%3A29%3A26

è Question: Where can I find commercial software that is written on Java? We need often some ready components for fast development and ready to pay for that!

Answer: I know one site which is good resource who is looking for different programs and components (beans, applets, libraries) for your software development: http://www.javashareware.com

è Question: Could you point me Java Data Binding Tools (to generate Java code from instances of WXS in order to represent the structures defined therein)?

Answer: Exists a few such tools. They generate Java object models from XML structures and encapsulates the low-level details of working with instances of XML documents and provide an "an alternative to the DOM" for processing XML from Java.

Java XML Binders simplify your XML development by creating Java classes that encapsulate XML parsing and validation and which have methods that map directly to your XML data elements and attributes.

Breeze XML Binder
Breeze is a commercial tool produced by Breeze Factor. It's available as an evaluation copy from http://www.breezefactor.com/.
If you want to use the software and the generate code you have to pay a license fee.

Castor
Castor is an open source data binding framework for Java[tm]. It's basically the shortest path between Java objects, XML documents and SQL tables. Castor provides Java to XML binding, Java to SQL persistence, and then some more.
It is available free of charge in binary or source form at http://www.castor.org. The code generation tool has a command line interface and a programmatic interface.

JAXB Reference Implementation
JAXB is a standard mapping developed by Sun in cooperation with a number of partners. JAXB in itself only specifies the intended behavior of a data binding tool and is not a tool in itself. However, the mapping is accompanied by a reference implementation. The JAXB jars and the reference implementation are both part of the Sun Java Web Services Developer Pack 1.2, which is available for free at http://java.sun.com/xml/jaxb/ .

XGen
XGen is an open source tool produced by Commerce One. The tool has been developed with the Castor code as a basis, but with a substantial number of changes in the mapping and the functionality. The tool is included in the Commerce One Conductor DocSOAP XML Developer's Kit available free of charge at http://www.commerceone.com/developers/docsoapxdk

In this tip were used descriptions from the article: http://www.xml.com/pub/a/2003/09/03/binding.html

In the article you can find test results for the listed above tools please look at this test.

è Question: I often read that Java folk discuss Java products that are not on the Sun’s web site yet. I mean different alpha, beta releases and so on...

Where they are taking all those versions of products from?

Answer: Most of Java products (from SUN I mean) are accessible through so named "Early Access". It is free to all Java Developer Connection(JDC) members. You can also register there and contribute into bug hunting process :-)
The list of products that are "Early Accessible" here:
http://developer.java.sun.com/developer/earlyAccess/index.html

When product is ready SUN moves into public "products" area... This page is very valuable resource, because you can plan your development in advance without awaiting of product to be finally released.
Use of future APIs today!

è Question: I just started in the big Java project and my task is to review Javadoc comments in the current code and estimate how much left to do...
Does exist an "easy" way to do that?

Answer: yes, probably if you will try to use Sun Doc Check Doclet. Although it is experimental code developed originally for internal use at Sun and you will use at your own risk, I advise you to try.
More info from SUN: "The Sun TM Doc Check Doclet is an extension to the Javadoc tool. It runs on source code and reviews documentation comments, generating an HTML report that identifies empty comments and other omissions and irregularities in the documentation comments. The report displays the number of doc errors, categorizes problems according to severity, and suggests text to include. (Those suggestions can then be copied into the code and edited.) It can be instructed to ignore packages and classes using wildcard patterns. An included utility package contains functionality that scans a directory tree to generate a list of packages, as well as pattern-matching classes for filenames."
You can read more about the tool and download it here:
http://java.sun.com/j2se/javadoc/doccheck/docs/index.html

è Question: How could I check if a class is serializable?

Answer: You can use the utility that comes with JDK - serialver.
You can use it in command line or GUI based form:

```
c:\> serialver -show
```

è Question: I have used quite long a few serializable classes in my application. Now when I added new functionality to one of them I get InvalidClassException exception. Why my class is invalid? It compiles, no problem...

Answer: All serializable classes have unique serialVersionUID. The serialVersionUID is computed using the signature of a stream of bytes that reflect the class definition. The National Institute of Standards and Technology (NIST) Secure Hash Algorithm (SHA-1) is used compute a signature for the stream. The first two 32-bit quantities are used to form a 64-bit hash. A java.lang.DataOutputStream is used to convert primitive data types to a sequence of bytes. The values input to the stream are defined by the Java Virtual Machine (VM) specification for classes.

So, in your case you need manually run serialver to generate a serialVersionUID for the old classes and manually assign them to newer versions. Add the

serialVersionUID to the class as a "static final long".

You are not required to define the serialVersionUID field in every class, but in the cases like yours you may find it useful.

è Question: Can I use my own long value for serialVersionUID in my class?

Answer: If you don't have any previously serialized objects of that class around, you can use any serialVersionUID as you want.

But in case if you have already older versions of your classes and want to keep compatibility you must use serialVersionUID which is obtained from the serialver tool.

è Question: What I should think about when I want secure serialization?

Answer: If you develop some class that provides access to file system, for example, you must know that a file descriptor contains a handle that provides access to an operating system resource. Being able to forge a file descriptor would allow some forms of illegal access, since restoring state is done from a stream.

Easiest way is to use static and static transient fields that contain sensitive information. Transient and static fields are not serialized or deserialized. Marking the field will prevent the state from appearing in the stream and from being restored during deserialization. Since writing and reading (of private fields) cannot be superseded outside of the class, the class's transient fields are safe.

è Question: I evolve my serialized class and do some changes to it. What are changes make next versions of my class incompatible (compatible)?

Part 1.

Answer: Actually it could not be better described (from my point of view) as it done in "Object Serialization Specification" by SUN.

Incompatible Changes

Incompatible changes to classes are those changes for which the guarantee of interoperability cannot be maintained. The incompatible changes that may occur while evolving a class are:

è

Deleting fields - If a field is deleted in a class, the stream written will not contain its value. When the stream is read by an earlier class, the value of the field will be set to the default value because no value is available in the stream. However, this default value may adversely impair the ability of the earlier version to fulfil its contract.

Moving classes up or down the hierarchy - this cannot be allowed since the data in the stream appears in the wrong sequence.

Changing a non-static field to static or a non-transient field to transient - This is equivalent to deleting a field from the class. This version of the class will not write that data to the stream so it will not be available to be read by earlier versions of the

class. As in the deleting a field case above, the field of the earlier version will be initialized to the default value but that can cause the class to fail in unexpected ways.

Changing the declared type of a primitive field - Each version of the class writes the data with its declared type. Earlier versions of the class attempting to read the field will fail because the type of the data in the stream does not match the type of the field.

è Question: I evolve my serialized class and do some changes to it. What are changes make next versions of my class incompatible (compatible)?

Part 2.

Incompatible Changes (continue)
Changing the writeObject or readObject method so that it no longer writes or reads the default field data or changing it so that it attempts to write it or read it when the previous version did not. The default field data must consistently either appear or not appear in the stream.

Changing a class from Serializable to Externalizable or visa-versa is an incompatible change since the stream will contain data that is incompatible with the implementation in the available class.

Compatible Changes
The compatible changes to a class are handled as follows:

è

Adding fields - When the class being reconstituted has a field that does not occur in the stream, that field in the object will be initialized to the default value for its type. If class specific initialization is needed the class may provide a readObject method that can initialize the field to non-default values.

Adding classes - The stream will contain the type hierarchy of each object in the stream. Comparing this hierarchy in the stream with the current class can detect additional classes. Since there is no information in the stream from which to initialize the object, the class's fields will be initialized to the default values.

Removing classes - Comparing the class hierarchy in the stream with that of the current class can detect that a class has been deleted. In this case, the fields and objects corresponding to that class are read from the stream. Primitive fields are discarded but the objects referenced by the deleted class are created since they may be referred to later in the stream. They will be garbage collected when the stream is garbage collected or reset.

Adding writeObject/readObject methods - If the version reading the stream has these methods then readObject is expected, as usual, to read the required data written to the stream by the default serialization. It should call defaultReadObject first before reading any optional data. The writeObject method is expected as usual to call defaultWriteObject to write the required data and then may write optional data.

è Question: I evolve my serialized class and do some changes to it. What are changes make next versions of my class incompatible (compatible)?

Part 3.

Compatible Changes (continue)

è

Removing writeObject/readObject methods - If the class reading the stream does not have these methods, the required data will be read by default serialization and the optional data will be discarded.

Adding java.io.Serializable - This is equivalent to adding types. There will be no values in the stream for this class so is fields will be initialized to default values. The support for subclassing non-serializable classes requires that the class's supertype have a no-arg constructor and the class itself will be initialized to default values. If the no-arg constructor is not available the NotSerializableException is thrown.

Removing java.io.Serializable so that it is no longer Serializable - This is equivalent to removing the class, and it can be dealt with by reading and discarding data for the class.

Changing the access to a field - The access modifiers public, package, protected and private have no effect on the ability of serialization to assign values to the fields.

Changing a field from static to non-static or transient to non-transient - This is equivalent to adding a field to the class. The new field will be written to the stream but earlier classes will ignore the value since serialization will not assign values to static or transient fields.

http://java.sun.com/products/jdk/rmi/doc/serial-spec/version.doc.html#6519

è Question: Is it possible with Java convert HTML documents to PDF ones?

Answer: Yes, with Sun MIF Doclet. The MIF Doclet is experimental code developed for internal use at Sun and released to the developer community for use as-is. It is not a supported product. Use at your own risk. While it was designed to be useful, bugs might not be fixed and features might not be added.

What is the MIF Doclet?

The MIF Doclet is an extension to the Javadoc tool that generates API documentation in MIF (Maker Interchange Format) from declarations and doc comments in source code in the Java TM programming language. It uses exactly the same set of Javadoc tags (such as @return, @param and @see) that the standard Doclet uses. FrameMaker batch processing programs enable these MIF documents to automatically be opened in Adobe FrameMaker and converted to FrameMaker native binary format, PDF or PostScript. Alternatively, you can manually choose "Save As" to save to Microsoft RTF, Microsoft Word or WordPerfect.

The goal has been to create a general, flexible means of conversion from doc comments to MIF that can be easily adapted to your own FrameMaker layout, style definitions and book organization. It tries to replicate much of the content generated by the standard HTML Doclet. However, the page layout has been modified from the standard HTML Doclet to add more layout options and to reduce vertical space, as page count needs to be conserved in print documentation. To obtain denser content, we put subheadings and data on the same line (rather than on separate lines) where

possible, and eliminated repetition, such as by removing the "first sentence" from the index. The features list below covers the myriad options.

Take it from here:
http://java.sun.com/j2se/javadoc/mifdoclet/docs/mifdoclet.html#batchprograms

è Good to know: Browsing the SUN web site past weekend I found that SUN together with Barnes & Noble.com (one of the biggest online stores) offer to Java developers 10% discount on not discounted items! I thought it could be good to inform you about this offer:
It is below:

"July 2003
Sun Developer Network and Barnes & Noble.com have joined forces to offer developers exclusive discounts on the latest Java books, bestsellers, DVDs and Music, all available on the Sun Developer Network online bookstore. Developers will save 10% off the list price of all non-discounted titles. This can add up to big savings for all the hot new Java books that you need. Free shipping is also offered (see site for details) and the discount is applied automatically at checkout. Start saving today!
"
Details are here: http://developers.sun.com/offers/

P.S. I am not an affiliate to SUN (could be good) and it is not commercial announcement, just info for you :-)

è Question: What are the weak keys? And if they are weak why we need to such weak keys? Is not it better do not use weak things for such strong technology like Java? :-)

Answer: This name has used for keys in WeakHashMap. An entry in a WeakHashMap will automatically be removed when its key is no longer in ordinary use. More precisely, the presence of a mapping for a given key will not prevent the key from being discarded by the garbage collector that is, made finalizable, finalized, and then reclaimed. When a key has been discarded its entry is effectively removed from the map, so this class behaves somewhat differently than other Map implementations.

Both null values and the null key are supported. This class has performance characteristics similar to those of the HashMap class, and has the same efficiency parameters of initial capacity and load factor.

è Question: I see that both Math and StrictMath contain methods for performing numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions... What is difference?

Answer: You should use StrictMath to ensure portability of Java programs, the definitions of many of the numeric functions in this package require that they produce the same results as certain published algorithms.
Unlike some of the numeric methods of class StrictMath, all implementations of the equivalent functions of class Math are not defined to return the bit-for-bit same results. This relaxation permits better-performing implementations where strict reproducibility is not required.

By default many of the Math methods simply call the equivalent method in StrictMath for their implementation. Code generators are encouraged to use platform-specific native libraries or microprocessor instructions, where available, to provide higher-performance implementations of Math methods. Such higher-performance implementations still must conform to the specification for Math.

The quality of implementation specifications concern two properties, accuracy of the returned result and monotonicity of the method. Accuracy of the floating-point Math methods is measured in terms of ulps, units in the last place. For a given floating-point format, an ulp of a specific real number value is the difference between the two floating-point values closest to that numerical value. When discussing the accuracy of a method as a whole rather than at a specific argument, the number of ulps cited is for the worst-case error at any argument. If a method always has an error less than 0.5 ulps, the method always returns the floating-point number nearest the exact result; such a method is correctly rounded. A correctly rounded method is generally the best a floating-point approximation can be; however, it is impractical for many floating-point methods to be correctly rounded. Instead, for the Math class, a larger error bound of 1 or 2 ulps is allowed for certain methods. Informally, with a 1 ulp error bound, when the exact result is a representable number the exact result should be returned; otherwise, either of the two floating-point numbers closest to the exact result may be returned.

Besides accuracy at individual arguments, maintaining proper relations between the methods at different arguments is also important. Therefore, methods with more than 0.5 ulp errors are required to be semi-monotonic: whenever the mathematical function is non-decreasing, so is the floating-point approximation, likewise, whenever the mathematical function is non-increasing, so is the floating-point approximation. Not all approximations that have 1 ulp accuracy will automatically meet the monotonicity requirements.

in this tip used API descriptions for Math and StrictMath classes

è Question: I work in Software Development many years and know that exist a lot of good and useful standards, standards "de facto" and simply unwritten rules for exchange of data between different OSs, programs and so on. For example XML... My question is: I know that different languages, platforms, OSs describe quite the same objects... I am wondering if exist any standard, agreement let say, how common objects are described... Part 1

Answer: The data, that you say describe all those common for whole SW industry objects, has own name - metadata. Metadata can be defined as information about data, or simply data about data. In practice, metadata is what most tools, databases, applications and other information processes use to define the structure and meaning of data objects.

One of the key issues limiting data interoperability today is that of incompatible metadata. The problem is that most applications define metadata using their own formats (schemas). This is compounded by the common practice of storing the metadata in different data management systems with different file formats. Without a common model for creating and accessing metadata, developers must hard-code discrete interfaces between applications to allow for data exchange and synchronization, thereby limiting interoperability and increasing the cost of developing and maintaining heterogeneous systems.

Standardizing on XML DTDs, which many industries are attempting to do, as a solution to this problem is insufficient, as DTDs do not have the capability to represent complex, semantically rich, hierarchical metadata.

Fortunately, Java has the solution for this problem - JMI (Java Metadata Interface) The JMI Specification implements a dynamic, platform-neutral infrastructure that enables the creation, storage, access, discovery, and exchange of metadata. JMI is based on the Meta Object Facility (MOF) specification from the Object Management Group (OMG), an industry-endorsed standard for metadata management. The MOF standard consists of a set of basic modelling artefacts described using UML. Models of any kind of metadata (called metamodels) can be built up from these basic building blocks. The MOF specification then defines a generic programming mechanism (using IDL) that allows for the discovery, query, access, and manipulation of these metamodel instances, either at design time or runtime. Thus the semantics of any modelled system can be completely discovered and manipulated.

JMI defines standard Java interfaces to these modelling components, and thus enables platform-independent discovery and access of metadata. These interfaces are generated automatically from the models for any kind of metadata. Additionally, metamodel and metadata interchange via XML is enabled by JMI technology's use of the XML Metadata Interchange (XMI) specification.

è Question: Part 2. What can YOU win by using JMI?

JMI will increase the adoption of standards-based metadata and accelerate the creation of applications and solutions in which there are no barriers to information exchange.
Applications using JMI can use the metadata to dynamically interpret the meaning of information, take action on that information and automate transactions across disparate systems and data sources.
used info from: http://java.sun.com/products/jmi/overview.html

è Question: (Java Bugs) Performance testing has shown that the finalize methods in the channel classes impose a significant GC penalty upon server-type programs that must deal with thousands of new open channels per second... (Java 1.4.1) What is solution?

Answer: The finalize methods of the primitive channel classes (SocketChannel, ServerSocketChannel, etc.) have been removed (4777499). Performance testing revealed that the presence of these methods imposed a significant (factor of 100) garbage-collection penalty upon server-type applications that process thousands of new open channels per second. User code that deals in channels should be carefully written to close channels explicitly rather than rely upon the garbage collector to do so.

Tip based on doc "Enhancements and Changes Java 2 SDK, Standard Edition Version 1.4.2 "

è Question: What are changes for JNDI in Java 1.4.2?

Answer: Bug 4639896 fixed: Java clients are now able to automatically follow referral using ldaps.

The JNDI/LDAP service provider now supports LDAPS URLs for requesting access to an LDAP server using an SSL connection. LDAPS URLs are accepted anywhere LDAP URLs are accepted.

In addition, if the user does not specify a port number in an LDAP URL, the default port is 389 for non-SSL connections and 636 for SSL connections. Prior to the Java 2 SDK, v 1.4.2, the port number defaults to 389 for all types of connections, including SSL.

This change affects applications that use LDAP over SSL without explicitly specifying a port number and expect the default SSL port to be 389. 389 is the standard non-SSL port for LDAP so this configuration should be rare. The workaround is to explicitly specify the port number.

Prior to the Java 2 SDK, v 1.4.2, the DNS service provider for JNDI did not include a trailing dot ('.') when returning a fully qualified domain name. Such a name could be returned within the value of an attribute, or by a call to Context.getNameInNamespace(). With the Java 2 SDK, v 1.4.2, the trailing dot is included.

è Question: Is it possible to get automatically updates of Java to my machine, like let say Norton Antivirus or Windows updates?

Answer: Yes, if you install latest version of Java - 1.4.2_01 and later!
Java Update is a new feature in the SDK/JRE. Its purpose is to provide the latest update of Java in a flexible, user-selectable manner. It works via a scheduler (jusched.exe) that runs as a background process and via the Java Plug-in Control Panel Update tab, which allows users to select the type of update options that they want.
jusched.exe is launched when the user reboots the computer after installing the SDK/JRE. It runs as a background process, completely transparent to the user, but can be viewed in the Process tab of the Windows Task Manager. Should a user not want to have it running in the background, it can be killed via End Task.

The Java Plug-in Control Panel Update tab allows the user to select the options for updating Java. The user can choose Automatic or Manual update. If the user chooses Automatic, then the update schedule (daily, weekly, monthly ...) can be selected via the Advanced button, and notification about updates can be set via the Notify Me: drop-down menu (before an update is downloaded and before it is installed, or only before it is installed). Manual update is performed by pressing the Update Now button. For more detail, see the Java Plug-in Control Panel Update tab.

è Question: The "java" command supports both options for pointing of classpath: -cp and -classpath. But the command "javac" recognizes only -classpath option. Why is such inconsistency?

Answer: Not now, the Java 1.4.2 makes usage of "-classpath" synonym -cp possible.

è Question: I know that in Java exist deprecated Classes, methods, fields and so on. When I look at particular class description I can see that some method is deprecated. Can I get somewhere whole list of deprecated Java stuff?

Answer: Yes! Every version of Java API, if you open for example framed version, contains in the right frame on the very beginning of page link "Deprecated". Just click it and you will be redirected to the page where all deprecated Classes, methods, fields, etc ... are listed.

è Question: What are maintenance releases of Java? And what are functionality releases?

Answer: Maintenance releases (such as 1.4.1, 1.4.2) do not introduce any new language features or APIs, so they maintain source-compatibility with each other.

Functionality releases and major releases (such as 1.3.0, 1.4.0) maintain upwards but not downwards source-compatibility.

è Question: Are all Java versions strongly compatible with previous versions of the Java 2 Platform? I have doubts, since sometime old code does not run on newer Java VMs... Part 1

Answer: Yes, generally speaking Java always strongly compatible with previous versions of the Java 2 Platform. Almost all existing programs should run on latest Java version without modification. However, there are some minor potential incompatibilities that involve rare circumstances and "corner cases" that we are documenting here for completeness.

If you have problems with running your old code on newer Java versions I recommend you check next pages where you can find all Java incompatibilities between Java versions:

The following documents contain information about incompatibilities between adjacent releases.

Incompatibilities in J2SE 1.4.1 (since 1.4.0) - http://java.sun.com/j2se/1.4.1/compatibility.html

Incompatibilities in J2SE 1.4.0 (since 1.3) - http://java.sun.com/j2se/1.4/compatibility.html

Incompatibilities in J2SE 1.3 (since 1.2) - http://java.sun.com/j2se/1.3/compatibility.html

Incompatibilities in J2SE 1.2 (since 1.1) - http://java.sun.com/products/jdk/1.2/compatibility.html

Question: Are all Java versions strongly compatible with previous versions of the Java 2 Platform? I have doubts, since sometime old code does not run on newer Java VMs... Part 2

Incompatibilities in JDK 1.1 (since 1.0) - http://java.sun.com/j2se/index.jsp

Also, see the Java Language Specification Maintenance Page (http://java.sun.com/docs/books/jls/jls-maintenance.html) for a summary of changes that have been made to the specification of the Java programming language since the publication of the Java Language Specification, Second Edition

(http://java.sun.com/docs/books/jls/index.html).

Just one example (Java 1.4.2):
Prior to J2SE 1.4.0, the character encoding scheme for security policy files was unspecified, and the files were read in using the default character encoding. Starting in 1.4.0, the policy files were required to be encoded in UTF-8.

While this allowed a policy file to be used across different locales, it broke existing policy files that contained characters in the default encoding.

In J2SE 1.4.2, a new system property was introduced: sun.security.policy.utf8

If this system property is set to true, the policy file is read in using UTF8 (1.4.0 and 1.4.1 behavior). If the system property is set to false, the policy file is read in using the default encoding (pre-1.4.0 behavior). When the system property is not set (it is NULL), its default value is true.

è Question: We are going to use Java on Linux. Are there any precautions should be taken against possible "well known" problems?

Answer: Unfortunately you should... Different releases of Java have different troubles...

For example:

A glibc 2.2 bug in versions 2.2.4 and earlier can cause the virtual machine to hang on exit when there is only one remaining active thread. This problem can affect Linux platforms such as Red Hat 7.0, 7.1, and 7.2 that have glibc versions 2.2 to 2.2.4. The problem is fixed in glibc 2.2.5.

I think that many programs can have "only one remaining active thread".
Every Java release usually lists "known" problems. For latest one, 1.4.2 it is here:

http://java.sun.com/j2se/1.4.2/install-linux.html#troubleshooting

It is recommendable to read the whole page since it contains very important info for those who first time starts with Java on Linux:

System Requirements
Java 2 SDK Installation Instructions
Installation of Self-Extracting Binary
Installation of RPM File
Java Plug-in Browser Registration Instructions
Java Web Start Installation Notes
Troubleshooting
Known Problems on Non-Supported Platforms

è Question: In Java API I found ACL and AclEntry interfaces, but do not understand what ACL is exactly used for...

Answer: ACL come from Solaris and was used to increase security level for file access. Beside user and groups in one of latest versions of Solaris (I think Solaris 7 or 8) was added ACL feature.

ACLs enable you to define file or folder permissions for the owner, owner's group, others, and specific users and groups, and default permissions for each of these categories. You can set up only one ACL per file or folder. An ACL consists of ACL entries. Each entry has a user type associated with it, much as basic permissions have Owner, Group, or Other associated with them.

If you want to grant file or folder access to a particular user or group, use the User or Group type ACL entry, respectively. You must specify the name of the user or group when you create the ACL entry. For example, you can use an ACL to grant a friend read permission on your resume, while protecting it from being read by the rest of the world (besides yourself).

ACL-enabled files and folders have a mask defined whose default permissions are the group permissions for the file or folder. The mask is the maximum allowable permissions granted to any user on all ACL entries and for Group basic permissions. It does not restrict Owner or Other basic permissions. For example, if a file's mask is read-only, then you cannot create an ACL with write or execute permission for a user without changing the mask value.

Use the mask as a quick way to limit permissions for users and groups.
More information can be found on http://docs.sun.com

I do not know if other UNIX’s also uses ACL concept. Could be good if someone informed me about it :-)

è Question: Often when I see the name of some Java technology, it followed by brackets with "JSR" and some number. For example: "J2ME Web Services version 1.0, (JSR-172) "
What does it mean - JSR-172?

Answer: JSRs are Java Specification Requests (JSRs). JSRs are the actual descriptions of proposed and final specifications for the Java platform. At any one time there are numerous JSRs moving through the review and approval process in JCP (Java Community Process).

You can see a list of all JSRs, by visiting the Java Specification Requests list (http://jcp.org/en/jsr/all). Using the links near the top of the page, you can sort the list by JSR ID number, Title, and the Spec Lead Company Name.

The JCP (Java Community Process) holds the responsibility for the development of Java technology. As an open, inclusive organization of active members and non-member public input, it primarily guides the development and approval of Java technical specifications. Anyone can join the JCP and have a part in its process, and you don't even have to join to contribute as a public participant.

The work of the Java Community under the JCP's procedures helps to ensure Java technology's standard of stability and cross-platform compatibility, enabling it to operate on hundreds of millions of devices, from desktop computers to consumer electronics to industrial robots.

Just as important, the JCP continually grows the platform's specification portfolio to meet the emerging technology needs of developers and organizations globally that depend on Java technology.

Like to get involved? Anyone can join the JCP and have a part in its process, and you don't even have to join to contribute as a public participant. Here are the pages that will help you become a member of the JCP:
http://jcp.org/en/introduction/overview

You can ask me: Why do I need this formal stuff? You can get most fresh information about coming Java technologies and plan your future developments. If you know that some technology, that you need, comes next month then you can concentrate on another things. And use it when it comes, instead of trying to develop your own solution...

è Question: Java 1.4.1 introduced two new garbage collectors. You described shortly them, in one of your tips. Could you give more info on when and which GC is preferable to use? Part 1

Answer: In the J2SE platform version 1.4.1 two new garbage collectors were introduced to make a total of four garbage collectors from which to choose. In J2SE platform version 1.4.2 the performance of the new collectors have been improved through algorithm optimizations and bug fixes, along with documentation updates to educate developers and administrators on what collectors to choose and when.

Throughput Collector
The throughput collector uses a parallel version of the young generation collector. It is used by passing the -XX:+UseParallelGC on the command line. The tenured generation collector is the same as the default collector. Use the throughput collector when you want to improve the performance of your application with larger numbers of processors.

In the default collector garbage collection is done by one thread, and therefore garbage collection adds to the serial execution time of the application. The throughput collector uses multiple threads to execute a minor collection and so reduces the serial execution time of the application.

Concurrent Low Pause Collector
The concurrent collector is used to collect the tenured generation and does most of the collection concurrently with the execution of the application. The concurrent collector employs a separate collector thread that consumes CPU cycles during application execution, this allows the application to be paused for only short periods of time during the collection but could lower overall throughput. It is used by passing the -XX:+UseConcMarkSweepGC on the command line.

Use the concurrent collector if your application would benefit from shorter garbage collector pauses and can afford to share processor resources with the garbage collector when the application is running. Typically applications which have a relatively large set of long-lived data (a large tenured generation), and run on machines with two or more processors tend to benefit from the use of this collector. However, this collector should be considered for any application with a low pause time requirement. Optimal results have been observed for interactive applications with tenured generations of a modest size on a single processor.

è Question: Java 1.4.1 introduced two new garbage collectors. You described shortly them, in one of your tips. Could you give more info on when and which GC is preferable to use? Part 2

Also exist "AggressiveHeap" option to be used on Servers, to improve performance. It was originally intended for machines with large amounts of memory and a large number of CPUs, but in the J2SE platform, version 1.4.1 and later it has shown itself to be useful even on four processor machines. The physical memory on the machines must be at least 256MB before AggresiveHeap can be used.
Detailed recommendations (with tests, graphs, etc.) can be found here:
http://java.sun.com/docs/hotspot/gc1.4.2/index.html

è Question: It is very difficult to tune Garbage Collection for large systems (much memory, many CPUs) properly from first run and without experience. It can take much time to find most optimized configurations. Does exist any kind intelligent GCing which is based on some history of application run, so say self tuning?

Answer: Yes, if you use adaptive sizing, which is actually enabled by default. We just do not know it :-)
A feature available with the throughput collector in the J2SE platform, version 1.4.1 and later releases is the use of adaptive sizing (-XX:+UseAdaptiveSizePolicy), which is on by default. Adaptive sizing keeps statistics about garbage collection times, allocation rates, and the free space in the heap after a collection. These statistics are used to make decisions regarding changes to the sizes of the young generation and tenured generation so as to best fit the behavior of the application. Use the command line option -verbose:gc to see the resulting sizes of the heap.

è Question: Does name "Concurrent Low Pause Collector" (garbage collector) mean that this collector runs very fast in very short time when application paused shortly (low paused)?

Answer: Quite the opposite! "Low Pause" means that GC runs almost always and GC has short pauses. This kind of GC is supposed to share computer resources with the application and run concurrently. Optimal results have been observed for interactive applications.

è Question: What is floating garbage and what should I do with it? Or not?

Answer: Since garbage collector and your application run concurrently on the machine, some part of objects that were alive in the beginning of GCing may become dead when GCing finishes. You should not worry about it. Although the amount of floating garbage is ~20% from all garbage (quite much) it will be collected by garbage collector at the next pass.

è Question: Why do we need wrappers like Integer, Boolean for int, boolean in Java?

Answer: Because wrappers let us treat primitives as objects. For example, you can not add an int to Vector, but you can the instance of Integer.

P.S. It is very short answer. If you know more please send us your explanation and we will publish it!

è Question: I read that JDK 1.x has new features...
I am not quite sure about the stability and performance. Does it mean that 1.x is not suitable for development now?

Answer: Absolutely not. When you are first starting out on a new project, especially big one it is a good idea to work with the most recent version of something. By the time you are ready to ship, the version of software that you use will have had the bugs shaken out of it. And you can include it into advantages of your product saying that it is built on latest software platform :-)

è Question: Is there any way of getting the memory requirements of objects at run-time?

Answer: JDK 1.1 provides a -prof option which among other things prints the bytes per instance and total instances for every class loaded in the VM at termination.

Java 2 provides a heap profiler which gives a detailed map of heap usage at termination.
http://developer.java.sun.com/developer/onlineTraining/Programming/JDCBook/perf3.html

è Question: I'm just getting into Java and have a simple Question:
What is J2EE and how is it different from Java2?'

Answer: Java2 is the name of the current specification of the Java language, the JVM
and all the other things that make up the basics of Java.

Java2 is a specification. It is implemented in the product SDK 1.3 now and soon will be available in 1.4 version (beta now).

Java has 3 subsections:

1. J2EE - Enterprise Edition - for server applications and other back-end...

2. J2SE - Standard edition - for desktop use...

3. J2ME - Micro edition - runs in cell phones, PDAs, and other resource and memory constrained devices, on the order of a 16 bit RISC processor and 128-512k RAM.

The 3 sections listed above are not products - they are specs, and define the tools available, language semantics, standard APIs and optional APIs.

J2ME is a subset of the Java2 spec.
J2EE is a superset of J2SE (+ extra APIs, + functionality).
The difference between J2EE and J2SE is J2EE contains extra APIs
A J2EE application is built using the J2SE tools, the
difference is the extra APIs.

è Question: When do I need to use reflection?

Answer: Reflection is a feature in the Java programming language. It allows an executing Java program to examine or "introspect" upon itself, and manipulate internal properties of the program. For example, it's possible for a Java class to obtain the names of all its members and display them.

The ability to examine and manipulate a Java class from within itself may not sound like very much, but in other programming languages this feature simply doesn't exist. For example, there is no way in a Pascal, C, or C++ program to obtain information about the functions defined within that program.

The reflection API represents, or reflects, the classes, interfaces, and objects in the current Java Virtual Machine. You'll want to use the reflection API if you are writing development tools such as debuggers, class browsers, and GUI builders. With the reflection API you can:

Determine the class of an object.

Get information about a class's modifiers, fields, methods, constructors, and superclasses.

Find out what constants and method declarations belong to an interface.

Create an instance of a class whose name is not known until runtime.

Get and set the value of an object's field, even if the field name is unknown to your program until runtime.

Invoke a method on an object, even if the method is not known until runtime.

Create a new array, whose size and component types is not known until runtime, and then modify the array's components.

Read more about reflection here:
http://developer.java.sun.com/developer/technicalArticles/ALT/Reflection/
http://java.sun.com/docs/books/tutorial/reflect/index.html

è Question: Properties class inherited from Hashtable...
Does it mean that I can use "non-string" keys and values?

Answer: Not really. Why?
I think best answer for this question can be found in API:

"The Properties class represents a persistent set of properties. The Properties can be saved to a stream or loaded from a stream. Each key and its corresponding value in the property list is a string.

A property list can contain another property list as its "defaults"; this second property list is searched if the property key is not found in the original property list. Because Properties inherits from Hashtable, the put and putAll methods can be applied to a Properties object. Their use is strongly discouraged as they allow the

caller to insert entries whose keys or values are not Strings. The setProperty method should be used instead. If the store or save method is called on a "compromised" Properties object that contains a non-String key or value, the call will fail.

When saving properties to a stream or loading them from a stream, the ISO 8859-1 character encoding is used. For characters that cannot be directly represented in this encoding, Unicode escapes are used; however, only a single 'u' character is allowed in an escape sequence. The native2ascii tool can be used to convert property files to and from other character encodings."

- **Question:** Why do we need to use "Preferences" if we already have "Properties" and JNDI in Java?

  Answer: Prior to the introduction of the Preferences API, developers could choose to manage preference and configuration data in an ad hoc fashion, by using the Properties API or the JNDI API as described below.

  Often, preference and configuration data was stored in properties files, accessed through the java.util.Properties API. However, there are no standards as to where such files should reside on disk, or what they should be called. Using this mechanism, it is extremely difficult to backup a user's preference data, or transfer it from one machine to another. As the number of applications increases, the possibility of file name conflicts increases. Also, this mechanism is of no help on platforms that lack a local disk, or where it is desirable that the data be stored in an external data store (such as an enterprise-wide LDAP directory service).

  Less frequently, developers stored user preference and configuration data in a directory service, accessed through the Java Naming and Directory Interface (JNDI) API. Unlike the Properties API, JNDI allows the use of arbitrary data stores (back-end neutrality). While JNDI is extremely powerful, it is also rather large, consisting of 5 packages and 83 classes. JNDI provides no policy as to where in the directory name space the preference data should be stored, or in which name space.

  Neither Properties nor JNDI provide a simple, ubiquitous, back-end neutral preferences management facility. The Preferences API does provide such a facility, combining the simplicity of Properties with the back-end neutrality of JNDI. It provides sufficient built-in policy to prevent name clashes, foster consistency, and encourage robustness in the face of inaccessibility of the backing data store.

  From SUN's Preferences API description

- Question: How does this Preferences API relate to Properties API?

  Answer: It is intended to replace most common uses of Properties, rectifying many of its deficiencies, while retaining its light weight. When using Properties, the programmer must explicitly specify a pathname for each properties file, but there is no standard location or naming convention. Properties files are "brittle", as they are hand-editable but easily corrupted by careless editing. Support for non-string data

types in properties is non-existent. Properties cannot easily be used with a persistence mechanism other than the file system. In sum, the Properties facility does not scale.

From Preferences Design FAQ

- Question: How does Preferences API relate to JNDI?

  Answer: Like JNDI, it provides back-end neutral access to persistent key-value data. JNDI, however, is far more powerful, and correspondingly heavyweight. JNDI is appropriate for enterprise applications that need its power. Preferences API is intended as a simple, ubiquitous, back-end neutral preferences-management facility, enabling any Java application to easily tailor its behavior to user preferences and maintain small amounts of state from run to run.

  From Preferences Design FAQ

- Question: Preferences: Why do all of the get methods require the caller to pass in a default?

  Answer: This forces the application authors to provide reasonable default values, so that applications have a reasonable chance of running even if the repository is unavailable.

  From Preferences Design FAQ

- Question: Why doesn't this API provide stronger guarantees concerning concurrent access by multiple VMs? Similarly, why doesn't the API allow multiple Preferences updates to be combined into a single "transaction", with all or nothing semantics?

  Answer: Preferences: While the the API does provide rudimentary persistent data storage, it is not intended as a substitute for a database. It is critical that it be possible to implement this API atop standard preference/configuration repositories, most of which do not provide database-like guarantees and functionality. Such repositories have proven adequate for the purposes for which this API is intended.

  From Preferences Design FAQ

- Question: Preferences: Why doesn't this API contain methods to read and write arbitrary serializable objects?

  Answer: Serialized objects are somewhat fragile: if the version of the program that reads such a property differs from the version that wrote it, the object may not deserialize properly (or at all). It is not impossible to store serialized objects using this API, but we do not encourage it, and have not provided a convenience method.

  From Preferences Design FAQ

- Question: Why is Preferences an abstract class rather than an interface?

Answer: It was decided that the ability to add new methods in an upward compatible fashion outweighed the disadvantage that Preferences cannot be used as a "mixin" (That is to say, arbitrary classes cannot also be made to serve as Preferences objects.) Also, this obviates the need for a separate class for the static methods. (Interfaces cannot contain static methods.)

From Preferences Design FAQ

n Question: Can I create different preferences for different users and at the same time have some "common", OS based preferences that are the same for all?

Answer: There are two separate trees of preference nodes, one for user preferences and one for system preferences. Each user has a separate user preference tree, and all users in a given system share the same system preference tree. The precise description of "user" and "system" will vary from implementation to implementation. Typical information stored in the user preference tree might include font choice, colour choice, or preferred window location and size for a particular application. Typical information stored in the system preference tree might include installation configuration data for an application.

From Preferences API

n Question: Are methods that modify preferences asynchronous or synchronous?

Answer: All of the methods that modify preferences data are permitted to operate asynchronously; they may return immediately, and changes will eventually propagate to the persistent backing store with an implementation-dependent delay. The flush method may be used to synchronously force updates to the backing store. Normal termination of the Java Virtual Machine will not result in the loss of pending updates -- an explicit flush invocation is not required upon termination to ensure that pending updates are made persistent.

From Preferences API

è Question: What does mean term "peer" and what can I do with it if I use getPeer() in Component?

Answer: You should not use it now since it is deprecated.
Thee peer is a system dependant component. All "heavyweight" graphical components have a respective peer that is responsible for communication between OS and Java. The peer level is different for different platforms but it makes possible to have "write once run anywhere" principle.
Swing almost does not use heavyweight components and you should not use it as well until you are going to adapt Java to new OS or write your own component...

è Question: I have developed an application. At present the application is in English Language.
Now I need to convert this site to another languages but don't know how to do it.
Can anybody tell me the way to do it... I do not want to rewrite every time my code...

Part 1.

Answer: The best way is to use Resource Bundles. Keep different language versions in resource bundle file. Your program can load it from the resource bundle that is appropriate for the current user's locale.

This allows you to write programs that can:
be easily localized, or translated, into different languages handle multiple locales at once be easily modified later to support even more locales.

For example, the base name of a family of resource bundles might be "MyResources". The family should have a default resource bundle which simply has the same name as its family - "MyResources" - and will be used as the bundle of last resort if a specific locale is not supported. The family can then provide as many locale-specific members as needed, for example a German one named "MyResources_de".

Each resource bundle in a family contains the same items, but the items have been translated for the locale represented by that resource bundle. For example, both "MyResources" and "MyResources_de" may have a String that's used on a button for canceling operations. In "MyResources" the String may contain "Cancel" and in "MyResources_de" it may contain "Abbrechen".

If there are different resources for different countries, you can make specializations: for example, "MyResources_de_CH" contains objects for the German language (de) in Switzerland (CH). If you want to only modify some of the resources in the specialization, you can do so.

- Question: I have developed an application. At present the application is in English Language.

Part 2.
When your program needs a locale-specific object, it loads the ResourceBundle class using the getBundle method:

```
ResourceBundle myResources =
                  ResourceBundle.getBundle("MyResources", currentLocale);
```

Resource bundles contain key/value pairs. The keys uniquely identify a locale-specific object in the bundle. Here's an example of a ListResourceBundle that contains two key/value pairs:

```
public class MyResources extends ListResourceBundle {
   public Object[][] getContents() {
      return contents;
  }
  static final Object[][] contents = {
      // LOCALIZE THIS
      {"OkKey", "OK"},
      {"CancelKey", "Cancel"},
      // END OF MATERIAL TO LOCALIZE
   };
}
```

Keys are always Strings. In this example, the keys are "OkKey" and "CancelKey". In the above example, the values are also Strings--"OK" and "Cancel"--but they don't have to be. The values can be any type of object.

You retrieve an object from resource bundle using the appropriate getter method. Because "OkKey" and "CancelKey" are both strings, you would use getString to retrieve them:

```
button1 = new Button(myResources.getString("OkKey"));
button2 = new Button(myResources.getString("CancelKey"));
```

The getter methods all require the key as an argument and return the object if found. If the object is not found, the getter method throws a MissingResourceException.
Besides getString, ResourceBundle also provides a method for getting string arrays, getStringArray, as well as a generic getObject method for any other type of object.

n Question: I have developed an application. At present the application is in English Language.

Part 3

When using getObject, you'll have to cast the result to the appropriate type. For example:

```
int[] myIntegers = (int[]) myResources.getObject("intList");
```

The Java 2 platform provides two subclasses of ResourceBundle, ListResourceBundle and PropertyResourceBundle, that provide a fairly simple way to create resources. As you saw briefly in a previous example, ListResourceBundle manages its resource as a List of key/value pairs. PropertyResourceBundle uses a properties file to manage its resources.

If ListResourceBundle or PropertyResourceBundle do not suit your needs, you can write your own ResourceBundle subclass. Your subclasses must override two methods: handleGetObject and getKeys().

è Question: I have developed an application. At present the application is in English Language.
Now I need to convert this site to another languages but don't know how to do it. Can anybody tell me the way to do it... I do not want to rewrite every time my code...

Part 4

Answer: The following is a very simple example of a ResourceBundle subclass, MyResources that manages two resources (for a larger number of resources you would probably use a Hashtable). Notice that you don't need to supply a value if a "parent-level" ResourceBundle handles the same key with the same value (as for the okKey below).

Example:

```
// default (English language, United States)
public class MyResources extends ResourceBundle {
  public Object handleGetObject(String key) {
    if (key.equals("okKey")) return "Ok";
    if (key.equals("cancelKey")) return "Cancel";
    return null;
  }
}

// German language
public class MyResources_de extends MyResources {
  public Object handleGetObject(String key) {
    // don't need okKey, since parent level handles it.
    if (key.equals("cancelKey")) return "Abbrechen";
    return null;
  }
}
```

You do not have to restrict yourself to using a single family of ResourceBundles. For example, you could have a set of bundles for exception messages, ExceptionResources (ExceptionResources_fr, ExceptionResources_de, ...), and one for widgets, WidgetResource (WidgetResources_fr, WidgetResources_de, ...); breaking up the resources however you like.

This tip is based on API description.

è Question: Can anyone tell me why the Java Vector class is called a "Vector"? I come from the engineering world, so I am trying to make some sense out of the name.

Answer: "Vector" is a common name for an array of one dimension. I think that it was designed to be accessed in only one direction - from start to end.
Of course, you can write code to iterate backwards...
But the enumeration it goes you don't have a getPrev() method, only a getNext():

```
for (Enumeration e = v.elements() ; e.hasMoreElements() ;) {
  System.out.println(e.nextElement());
}
```

So a Vector has dimension (number of elements) and direction (in terms of access).
Are you agreeing?

è Question: How is a multidimensional array stored in Java? Is it different from C?

Answer: Java stores multidimensional arrays as single dimension arrays which consist themselves of more single dimensional arrays.

è Question: Why do we need to use shift operator? Can you give some examples?

Answer:

1. Make faster integer division/multiplication operations:

```
4839534 * 4
```

can be done like this:

```
4839534 << 2
```

or

```
543894 / 2
```

can be done like this:

```
543894 >> 1
```

Shift operations much more faster than multiplication for most of processors.

2. Reassembling byte streams to int values
3. For accelerating operations with graphics since Red, Green and Blue colors coded by separate bytes.
4. Packing small numbers into one single long...

è Question: For example I use some collection and method put() which has return type Object... I do not need it and do not use it in my program.
Does this Object occupy memory space?

Answer: If it is not referenced - it will be garbage collected. And it will stay in memory in opposite case.
The time of garbage collection is not known in advance - it is up to GC (Garbage Collector) to decide when it can be done.

è Question: Why I can not compile it:

```
byte aa = 2;
aa = - - aa;
```

I think the construction: - -aa is equivalent to -(-aa)...

Answer: The Java programming language supports various arithmetic operators for all floating-point and integer numbers. These operators are + (addition), - (subtraction), * (multiplication), / (division), and % (modulo).

In your example aa is a byte variable. You can use --aa instead.

Interesting fact:

probably you know that the Class name you can get like this:

```
MyClass m = new MyClass();
System.out.println(m.getClass());
```

Another not documented way is:

```
MyClass m = new MyClass();
System.out.println(MyClass.class);
```

The interesting fact is that you will not find "class" field in Class or Object!

è Question: What does mean "internalization" for me when I design my program and where should I go if I need to change the language in some program?

Answer: I think the best Answer can be found on SUN's site :-):

"Internationalization is the process of designing an application so that it can be adapted to various languages and regions without engineering changes. Sometimes the term internationalization is abbreviated as i18n, because there are 18 letters between the first "i" and the last "n."

An internationalized program has the following characteristics:

- With the addition of localized data, the same executable can run worldwide.

- Textual elements, such as status messages and the GUI component labels are not hard coded in the program. Instead they are stored outside the source code and retrieved dynamically.

- Support for new languages does not require recompilation.

- Culturally-dependent data, such as dates and currencies, appear in formats that conform to the end user's region and language.

- It can be localized quickly.

- Localization is the process of adapting software for a specific region or language by adding locale-specific components and translating text. The term localization is often abbreviated as l10n, because there are 10 letters between the "l" and the "n."

Usually, the most time-consuming portion of the localization phase is the translation of text. Other types of data, such as sounds and images, may require localization if they are culturally sensitive. Localizers also verify that the formatting of dates, numbers, and currencies conforms to local requirements."

Please check a quick example and checklist (what you need to do...):

http://java.sun.com/docs/books/tutorial/i18n/intro/index.html

è Question: My program creates a huge number of objects dynamically and even more inside each of them...

One thread does some processing and when the user calls a method on the object, it will kill the thread.

After that the object has to be killed so that it can be garbage collected...

How do I do it?

Answer: You do not "kill" objects, just stop just stop keeping any references to them and garbage collector will do the rest... :-)

è Question: Can I force finalization using System.runFinalization()?

Answer: No, in Java API is written:

"Calling this method suggests that the Java Virtual Machine expend effort toward running the finalize methods of objects that have been found to be discarded but whose finalize methods have not yet been run. When control returns from the method call, the Java Virtual Machine has made a best effort to complete all outstanding finalizations. "

It means that you just suggest the JVM that it might be a good idea to do finalization...

Like garbage collection, you cannot force it.

è Question: I read that an object can be strongly, weakly or even phantom (!!!!) reachable...???

Answer: I found the definitions in API. I think it is clear enough to be cited here:

Going from strongest to weakest, the different levels of reachability reflect the life cycle of an object. They are operationally defined as follows:

An object is strongly reachable if it can be reached by some thread without traversing any reference objects. A newly-created object is strongly reachable by the thread that created it.

An object is softly reachable if it is not strongly reachable but can be reached by traversing a soft reference.

An object is weakly reachable if it is neither strongly nor softly reachable but can be reached by traversing a weak reference.

When the weak references to a weakly-reachable object are cleared, the object becomes eligible for finalization.

An object is phantom reachable if it is neither strongly, softly, nor weakly reachable, it has been finalized, and some phantom reference refers to it.

Finally, an object is unreachable, and therefore eligible for reclamation, when it is not reachable in any of the above ways.

About phantom reference please read more in API (java.lang.ref.PhantomReference)

è Question: I use logging in my application... The problem is that the problem is that I am opening the PrintWriter stream only once and keep it open until my application

is stopped. Open PrintWriter stream does not let to open logs in text editor and users can not read logs... Appears a error message like this: "Can not open file, it used by another application..."

Is it possible to solve the problem?

Answer: It depends on OS type. For example in Windows NT you can use command "type":

type log.txt > log_copy.txt and then read log_copy.txt file in any text editor...

In UNIX use command "cat":

cat log.txt > log_copy.txt

è Question: I know that exist many Java compilers, not only "javac" from SUN :-)

Part 1
Could you list some of them and describe advantages of them over SUN's version?

Answer: I list some of them here with short descriptions... Most of them are extending Java with additional features, keywords, functions and so on..

1. Java espresso is a Java 1.0 compliant compiler being developed at Boston University:
http://types.bu.edu/Espresso/JavaEspresso.html

The objectives of this implementation were to understand the issues involved in the development of a modern object-oriented compiler for a real language, to study the Java Virtual Machine (JVM) and its runtime system

2. The GNU Compiler for the Java: http://gcc.gnu.org/java/
GCJ is a portable, optimizing, ahead-of-time compiler for the Java Programming Language. It can compile:
Java source code directly to native machine code,
Java source code to Java bytecode (class files),
and Java bytecode to native machine code.

3. The Jikes compiler:
http://oss.software.ibm.com/developerworks/opensource/jikes/
Jikes has four advantages that make it a valuable contribution to the Java community:

- Open source. Jikes is OSI Certified Open Source Software. OSI Certified is a certification mark of the Open Source Initiative.

- Strictly Java compatible. Jikes strives to adhere to both the Java Language Specification and the Java Virtual Machine Specification as tightly as possible, and does not support subsets, supersets, or other variations of the language. The FAQ describes some of the side effects of this strict language conformance.

- High performance. Jikes is a high performance compiler, making it ideal for use with larger projects.

- Dependency analysis. Jikes performs a dependency analysis on your code that provides two very useful features: Incremental builds and makefile generation.

--> to be continued in our next newsletter -->

è Question: I know that exist many Java compilers, not only "javac" from SUN :-)

Part 2.

Could you list some of them and describe advantages of them over SUN's version?

4. The JUMP compiler:

http://ourworld.compuserve.com/homepages/DeHoeffner/jump.htm

JUMP extends the capabilities of JAVAC in the following ways:

It is about 10 - 30 times faster than JAVAC.
The output format can be changed to adapt JUMP to your favourite development environment.
It contains a fully functional disassembler (option -p) that can be used as a replacement for the JAVAP disassembler that ships with Suns JDK.
It extends the JAVA language with several powerful features well known from C++. These features include class templates, operator overloading and default parameters. They are well accepted in the C++ community and have been proofed to leverage productivity of the developers.

5. The KOPI compiler: http://www.dms.at/kopi/index.html

The Kopi Project is a Java software project from DMS, providing a development environment for database applications using Java, JDBC and JFC/Swing.

6. The Pizza Compiler: http://pizzacompiler.sourceforge.net/

The Pizza language is an extension to Java with three new features:
- Generics (a.k.a. Parametric polymorphism)
- Function pointers (a.k.a. First-class functions)
- Class cases and pattern matching (a.k.a. Algebraic types)

7. The Timber compiler: http://www.pds.twi.tudelft.nl/timber/index.html

The Timber compiler has been developed to demonstrate, and experiment with, compilation techniques for parallel and distributed systems.

The compiler implements most of Java, plus a set of additional language constructs for scientific programming. This set of extensions is called Spar. We only fundamentally deviate from the Java language specification on the following points:

The compiler is a whole-program static compiler from Java/Spar source code to machine code, so JVM bytecode is not supported. In particular, dynamic class loading is not supported.

Java threads are not supported. Instead, we provide our own language constructs for parallel programming.

The Spar language extensions introduce a number of additional keywords.
--
John

è Question: I am writing an application for small memory system. To make it reliable I would like to monitor memory size in my program and try to decrease program activity to enable Garbage Collector do its job before all memory will be consumed.

When enough memory will be available program must continue the job.

How can I do that?

Answer: You idea is excellent to design long time running program for an extreme environment :-)

Use Runtime.totalMemory() and Runtime.freeMemory()

è Question: How can I store and retrieve serialized objects to and from a file? I have a .txt file and two String objects say,

String company = "Company",

String company_name = "Comp_name",

How do I that?

Answer: I found one example how to write a serialized object to a file, tmp. The file tmp will be in the same directory as the program:

```
import java.io.*;
import java.util.*;

// Writing an object.
// Serialize today's date to a file.

public class WriteData implements Serializable {
  public static void main (String[] args) throws IOException {
    FileOutputStream f = new FileOutputStream("tmp");
    ObjectOutputStream s = new ObjectOutputStream(f);
    String company = "My Good Company";
    s.writeObject(company);
    s.flush();
  }
}
```

Here is an example of how to read a serialized object from the file "tmp":

```
import java.io.*;
import java.util.*;
import java.lang.*;

// Reading an Object from a stream.
// Deserialize a String and date from a file.

public class ReadData implements Serializable {
  public static void main (String[] args)
       throws IOException,ClassNotFoundException {
     FileInputStream in = new FileInputStream("tmp");
     ObjectInputStream s = new ObjectInputStream(in);
     String company = "";
     String company_name = "";
     company = (String)s.readObject();
     System.out.println(company + " ");
  }
}
```

è Question: Do you know how can I display special symbols like "copyright" & "trademark"???
What are theirs codes?

Answer: Please take a look at:
Code Charts - http://www.unicode.org/charts/
or
Unicode Character Name Index
http://www.unicode.org/charts/charindex.html

è Question: Why is it better to write own close() method than rely to finalize()?

Answer: Finalize() called by the garbage collector on an object when garbage collection determines that there are no more references to the object. A subclass overrides the finalize method to dispose of system resources or to perform other cleanup.

There are two things bad with finalize:
1. Garbage collection might not ever run...
2. Finalize order is not predictable when you have multiple objects

The more correct approach is to write close() methods for each object.
Also if some exception can be thrown inside of method that calls close() method use finally. It fits best for solving such problems.
No matter how try block exits - either normally or exceptionally - you will be able to run finally statement.

è Question: If a class inherited from an interface, can super() be used in method of the class? An interface has no concrete method... Very strange...
How can super() be used?'

Answer: Super() method appeals to the constructor of parent class that implements interface, not to "parent" interface itself.
If there was no parent class then java.lang.Object will be automatically extended and the default constructor from the Object class will be called.

è Question: Why do we need wrappers like Integer, Boolean for int, boolean in Java?

Answer: Because wrappers let us treat primitives as objects. For example, you can not add an int to Vector, but you can the instance of Integer.

P.S. It is very short answer. If you know more please send us your explanation and we will publish it!

è Question: If there was a "setvalue()" method in Float class, it would have been useful to easily create code that modify parameters passed on methods.
So any good reason why there are no setvalue methods?

Answer: Once a Float instance is created you can not change its value because it is immutable by definition. The rest of wrappers are also immutable, for example Integer, Boolean and so on.
Why? It has two main advantages...
Nobody can change the instance and cause bugs in program.
Also they are thread safe: if nobody can change them why we need to protect them?
If you need to have mutable wrappers you can probably write them and use for your own needs. But you can not do anything with classes from third party companies that use standard implementation.

è Question: When I running Runtime.exec(...), I get a Process object.
I want serialize it to a file to be able to reconstruct it later if required... Is it possible?

Answer: no, it is not possible.

è Question: My application uses just one class from big jar file. Does the entire jar file get loaded into memory?

Answer: No. Classloader indexing once whole content of jar file and loads the list of content into memory. The necessary class will be loaded when it is called, in so called "lazy" fashion...

è Question: Why is thin client has name "thin" in J2EE?

Answer: J2EE applications do not perform most of functions such as executing complex logic, querying databases on a client side. Instead, these operations are executed on J2EE server by enterprise or Web beans. It has advantages in terms of

security, performance, reliability and services. Main function of a thin client is to provide some interface to users, letting to interact with J2EE application on a server side.

è Question: What is difference between the for(;;) and while(true)?

Answer: There is no difference! They both are infinite loops... Loop forever - Perpetum Mobile (Latin) :-)

è Question: I just started to work in big company... Before I worked with software design myself and developed small programs. Now I hear such terms like "business layer", "presentation layer"... Has it something to do with Java? Please clarify.

Answer: It has more to do with applications structure and used for design of large software products. It helps to divide application logic into big logically independent blocks (layers) that interacts by interfaces. It lets to develop big software products without knowledge of all details.

We will describe here three layers model:

1. Presentation layer - responsible for content displaying to user and provides interface for user - application interaction. It is client in client - server application.

2. Business layer - main logic of an application is implemented here: all required calculations, data validations and so on. Usually runs on server side in client - server application.

3. Data layer - responsible for data management. This layer provides business layer with all data that are processed there and then presented to user by presentation layer.

Layering of an application lets to change different parts without significant efforts. For example if data layer operates with a SQL database it can be switched to Oracle database. Well planned and structured application will let do it easy.

è Question: Java has no sizeof() like C++... Why?

Answer: In C and C++ sizeof() tells the number of allocated bytes for data items. Different machines and OSs can allocate different number of bytes for the same data. For example integer can be stored in 16 or 32 bits. That's why important to know for programmer this information.
Java is much better in this situation since it keeps the same size on all machines. The portability is a part of language and you should not worry about this - just know it!
--
Sasha

è Question: I'm trying to write a script to package my class files into a jar file. I can use "jar cf a.jar temp" and it takes all files in the temp directory. Unfortunately it also includes all source java files. Is it possible to just include the *.class files but also recourse all directories?

Answer: Compile with the -d option that lets to specify a directory where your class files must be placed.

è Question: I've been using the java comm API to allow me to build a small java interface to an old datalogger box that connects to a computer's serial port. The program runs and returns data to the interface. I have no problem receiving data from the datalogger.

However, the program that runs on the datalogger needs to be reset with a "cbreak," that is, the equivalent of sending it ^C (control-C).
If I use a terminal program to send the cbreak (by typing ctrl-c), it works fine, resetting the datalogger. What I can't figure out is how to get java to send the cbreak, specifically, what do I need to write that the SerialPort's Outstream? Is it a special character? I've tried sending the string "^C" but that doesn't work.

I assume it must be some Unicode thing that I am not competent enough to deal with.

Answer: No, in fact control-c has the numerical value 3. Just send a byte with a numerical value of 3.
--
Paul Lutus
www.arachnoid.com

è Question: I want to download Java 1.4. I looked on the page for compatibility and did not find anything about Windows XP.

Has anyone installed 1.4 on XP? I want to be sure before doing this...

Answer: Yes, it seems working well. Also you can find Java plugin for Windows XP here:
http://java.sun.com/getjava/download.html

è Question: I am confused:
In javadoc for public abstract class OutputStream:

"public abstract void write(int b) throws IOExceptionWrites the specified byte to this output stream."

But why in the method we have "int b"?
Was not possible to have public abstract void write(byte b) ?
In the same description later: "The 24 high-order bits of b are ignored."
Why need to use int and ignore 3/4 of that afterwards...

Has it relation to memory management efficiency on 32-bits CPUs and OSs?
What's about 64-bit CPUs like Itanium...
Will SUN change int to 64-bits size?

Or does SUN believe that in future bytes will be 32 bits long? :-)

Answer: Equivalent read() returns an int also. This is to enable a return value of -1 in which is not possible if use byte.
Byte is signed and it has the value range: -128 to 127. And even if the sign is not so important there will not be any possibility to distinguish between -1 and 255!
And last point: byte can not be used directly for mathematical operations, it has to be converted to int anyway.

è Question: Why do not distinguish methods based on their return value?
I mean on class names and arguments list? I think, it could be obvious from declaration:

```
void aa(){}
int aa(){}
```

to use it later in a such way:

```
int bbb = aa();
```

compiler could easily distinguish which function must be used here - int, not void...

Answer: Yes, in this situation compiler easily will find out which kind of function must be used. But quite often a method can be called directly - just to do something, not to return some value. The return value is not important here:

```
...
aa();
...
```

In this case compiler can not find out which method must be used. And other programmer reading your program will be confused as well.

è Question: I read that Java creates a default constructor for every class if I do not do it myself. Default - without arguments.
I wrote small program where I defined my constructor with two arguments. But in one place I still need an empty constructor (without arguments).
But compiler complains that it can not find it!

Answer: Java creates a default constructor if you have no constructors at all. If you create even one with arguments a default constructor will not be created and it is your responsibility to add it into a program.

è Question: How can I be sure that my package has a unique name?

Answer: By convention, the first part of the package name is the Internet domain name of the class creator, reversed. Since the Internet domain names are unique

you should follow this agreement. Do not use the package name that already match some domain name that you are not owner of. Find free name and register it. You can find cheap registration now on Internet, ~$10/year! For example, http://www.godaddy.com registers for $8.95 only.

è Question: Some packages use capital letters in name and some low case? I read they must be low case...

Answer: Before Java 2 (1.2) the com, org, edu, etc. extensions were capitalized on Java packages. This was changed in Java 2. Now package name is in low case.

è Question: Quite often I see that programmers use full name of class in a program. For example like this:

```
java.util.Vector vect = new Vector();
```

I do not like such long lines, they are not so nice!
Is not it easier to use the "import" in the beginning of program:

```
import java.util.*;
```

and then: Vector vect = new Vector(); ?

Answer: By using full class name in a program a programmer avoids names collision situation - a case when you could have your own Vector class in your package:

```
com.mypack.utilities.*
```

For example you have:

```
import com.mypack.utilities.*;
import java.util.*;
```

Compiler will not understand which Vector class you refer in the line below:

```
Vector vect = new Vector();
```

The usage of full class name makes the situation clear.

è Question: I use a third party package and I need to have an access to protected methods of one class - the member of that package. I asked developers of the package to change it for me, but they answered that it will take month until the release will be ready (they are occupied with another job now). What can I do? I can not wait with development and need to continue my job.

Answer: Exists one legal way and two illegal :-(
Please read license agreement to the package and make sure that you are not going to do it illegally.

1. Just extend the class and you will get access to all protected methods.

2. Take that class, decompile it and change protected to public. Best tool I know is DJ Java Decompiler with graphical user interface.
http://members.fortunecity.com/neshkov/dj.html

3. Add your own class into the package. Write a method that creates an instance of that class where protected method is and call it. Since protected methods are accessible for package members - you can access it now as well from within of your new class.

Please be sure that you are not breaking a license agreement or contact developers and ask a permission to do that.

è Question: Is JavaScript the same as Java?

Answer: NO! An Amazingly large number of people, including many web designers, don't understand the difference between Java and JavaScript. Briefly it can be summed up as follows:

Java was developed by Sun Microsystems. Java is a full-fledged object-oriented programming language. It can be used to create standalone applications and applet. Applets are downloaded as separate files to your browser alongside an HTML document, and provide an infinite variety of added functionality to the Web site you are visiting. The displayed results of applets can appear to be embedded in an HTML page (e.g., the scrolling banner message that is so common on Java-enhanced sites), but the Java code arrives as a separate file.

JavaScript on the other hand was developed by Netscape, is a smaller and simpler scripting language that does not create applets or standalone applications. In its most common form today, JavaScript resides inside HTML documents, and can provide levels of interactivity far beyond typically flat HTML pages -- without the need for server-based CGI (Common Gateway Interface) programs.

Some server software, such as Netscape's SuiteSpot, lets web application developers write CGI programs in a server-side version of JavaScript. Both client-side and server-side JavaScript share the same core JavaScript language, but each side deals with different kinds of objects. Client-side objects are predominantly the components of an HTML web page (e.g., forms, text boxes, buttons). Server-side objects are those that facilitate the handling of requests that come from clients, as well as connectivity to databases.

è Question: Is Java open source as distributed by Sun, i.e., all the modules including JVMs?
If not, is anyone else doing an open source implementation?

Answer: Java is not open source project. Though you can get the full source code under a Sun license.
http://kaffe.org does open source implementation

I read there: "Kaffe is a clean room, open source implementation of a Java virtual machine and class libraries. It is also a fun project that was started by Tim Wilkinson and was made successful by the contributions of numerous people from all over the

world.

But Kaffe is not finished yet! You can help by developing new and missing functionality, porting Kaffe to new platforms, and testing your Java applications under Kaffe.

Kaffe mostly complies with JDK 1.1.4, except for a few missing parts.
Parts of it are already JDK 1.2 (Java 2) compatible."

http://gcc.gnu.org/java/ - The GNU Compiler for the Java tm Programming Language

è Question: I will be thankful if anyone tells me why JVM is called virtual machine.

Answer: JVM is called a virtual machine because there is no real hardware which interprets the byte code. If you have done any assembly programming for any microprocessor/microcontroller you will able to understand this. A microprocessor has built-in instruction set to interpret the assembly code. Similarly the JVM is similar to a microprocessor in the sense it has its own instruction set but it implemented in software. That is why it is called a virtual machine!

è Question: I read that JDK 1.x has new features...
I am not quite sure about the stability and performance. Does it mean that 1.x is not suitable for development now?

Answer: Absolutely not. When you are first starting out on a new project, especially big one it is a good idea to work with the most recent version of something. By the time you are ready to ship, the version of software that you use will have had the bugs shaken out of it. And you can include it into advantages of your product saying that it is built on latest software platform :-)

è Question: Do anyone know the difference between java and C#.

Answer: They are different languages. Java has been around for about five years. C# has not been publicly released yet. One is written by Sun Microsystems, one my Microsoft. They are fairly similar languages with C# having a few extra bits added on to it.
--
Phil

C# bytecodes can be compiled to native exe files just as Java bytecodes can be. But C# is expected to be more closely tied to the Windows operating system and standard interfaces that are part and parcel of Windows. Writing a native compiler that collects all these interfaces and combines them into a unified whole that can run on ANY operating system may require compiling proprietary windows components which Microsoft will make sure is hard to do and against its licensing policies. So you can expect to see native compilers that compile for Windows platforms but not to other operating systems.
--
alankarmisra

è Question: I read PHP 4 times faster than JSP. Why then do we need JSP?

Answer: These tools fill somewhat different niches and seldom directly compete. PHP is good for situations where your page can interact more or less directly with a database, without the need for complex business logic. Its strength is that it can be used to build pages VERY quickly. And, as you note, they run very quickly as well.

The JSP/Servlet model is more geared toward distributed n-tier applications where there is at least logical, and possibly physical, separation of model, view, and controller functions. It is more complex than PHP, but also more scalable and well-written Java apps may be a great deal more maintainable because of the separation of logical tiers.

They're both nice tools, and I use both, but I don't think either one is going to kill the other anytime soon.
--
Joe

è Question: My question is: is JSP as powerful as Servlet? I heard that JSP will eventually compile into Servlet class file.
One thing can be done by Servlet, can it be done by JSP too? In terms of http.

Answer: Everything a Servlet does can be done in JSP and vice versa. Good programming practice (you will see some articles over the last year in JavaPro) dictates to combine Servlets and JSP in any significant web application.

JSP should be mainly HTML (or XML, or WML or whateverML) with little Java inside. Servlets should be Java with few or not at all lines like this:

```
out.println( "<html>" );
out.printlb( "<body>" );
```

This creates a more or less clean separation between presentation (JSP) and business logic (Servlet).
Java beans also have a role in this. I strongly recommend the JavaPro articles or whatever text on the MVC model you can find.

--
eugene aresteanu

è Question: I do not understand the difference between clone () and new (). Why shouldn't I just call new ()?

Can apply new () to cloned object?

Answer: The difference is that clone() creates the object that possibly has changed fields (after start up) and this object is willing to be cloned by implementation Cloneable interface.

New() method can be applied without objects "permission" and create a new instance with fields just initialized and not changed. They will be changed later during runtime. Clone () "mirroring" this modified object.

That's why you NEVER should call new() on cloned object. You can destroy the clone...

è Question: Doesn't Java have C's preprocessor functions? At least #define's? If not, the only way to define constants is with ‘final’? So... no macros at all?

Isn't there, then, a separate "preprocessor" utility (at least), if I want to define things like

```
   #define proc public void
```

or so?

Answer: The answer to that is either "No" or "No, thank God". Code gets read many, many more times than it gets written. You should be constantly making things easy for the developers that have to maintain this after you've finished it (after all, it's probably going to be you).

That's way Sun pushes its Java coding standards so hard.

Believe it or not, the only reason that I don't do much C++ work boils down to three things that C++ has that makes its code hideous- the preprocessor, typedef, and operator overloading. All three of those are wholly unnecessary and mung C++ code six ways to Sunday.

When C# was announced, I was excited because I'd heard its preprocessor was more restrictive. It is, in that it doesn't permit macros. It still lets you do all sorts of whacky "conditional compiling" stuff, though, that makes code unreadable again.
--
Cuplan

è Question: I'm just starting to learn Java on my own. Should I first learn AWT or should I jump directly into the Swing of things?

Answer: Will you want to code applets that are easy for anyone to run in their browser? If so, you'll probably have to go with the AWT for now.

The AWT isn't so bad, but Swing makes a lot of things much easier, so if you want to ship native-code applications or suchlike then I'd go with Swing. I still use the AWT, but I find myself having to code a lot of 'standard' things myself. Swing make things easier but IE doesn't support it.
--
Mark

è Question: I can't manipulate inodes on my Linux box ... in fact I can't even get real info about a file! Java is a bad hack and is for kids who aren't sharp enough to do C++.

Answer: Think of Java in the same terms as COBOL and VB, and you've got the right idea. Start thinking of it as a replacement for C++ and you're on the wrong track.

Don't expect this portable language to be a tool for low-level coding with hooks into the OS and hardware internals. It just wasn't designed for that. It's an excellent *applications* language, not a *systems* language like C or assembler.

On the other hand, if any pesky Java programmers tell you that C++ is dead and that Java can do everything C++ does, and more, you may howl with laugher and

tell them to eat their JVM.
--
David Ehrens

è Question: How do we exchange data between Java and JavaScript and vice-versa?

Answer: Public variable and methods of Java Applet are visible to a HTML document. So using JavaScript you can access the public variables and public functions.

The syntax is:
var some_var = document.appletname.variable_name

With this you will get the value of the variable variable_name in your JavaScript variable some_var.

è Question: Constructors and methods: are they the same?

I need a little help here...I have been teaching that constructors are not methods. This is for several reasons, but mainly because JLS says "constructors are not members" and members are "classes, interfaces, fields, and methods."

So, now the rest of the staff is ganging up on me and making life a little nasty. They quote Deitel and Deitel, and Core Java (which references "constructor methods") and who knows how many other books.
The one we are teaching in is loaded with so many errors that even though it calls constructors methods NOBODY will quote it as an authority.

How can so many people call constructors methods if they aren't.

Okay. Are they or aren't they? I holding to the definition that they are not unless it is so common to call them that, that I will have to change. Comments?

Answer: If you go by the JLS (Java Language Specification) and the Java API (and you should) , then no, constructors are not methods. Consider that Class.getMethods () returns an array of Method instances and Class.getConstructors () returns an array of Constructor instances, and Constructor and Method or not interchangeable (one is not derived from the other), but both implement the Member interface. Seems to me that Java is going out of its way to differentiate them.

Besides, the mechanics of constructors are so different from the mechanics of methods, there seems to be no value to considering one a member of the set of the other.
Now, as far as teaching the language goes:
Methods:
+ return types
+ called by name
+ executed multiple times

Constructors:
+ super(...) or this(...) as first instructor (often implicit)
- everything else

I very much do not like trying to lump the two concepts together, especially in introductory courses. Conceptually they are very, very different things.

A constructor is a unique element (even if there are several). It has the name of the class, its declaration is different, and it doesn't have the same syntax as a method. It plays a unique role. You can call a method from a constructor, but you cannot call a constructor from a method.

I say constructors are distinct from methods, and for students, blurring the distinction will lead to problems.
--
by Chuck McCorvey, Chris Wolfe,
-- Paul Lutus, www.arachnoid.com

è Question: I the see method getClass() in java.lang.Object. Do we need it? I know all my classes.

Answer: Exactly. If you know - you do not need it. But if you do not know then it helps you. For example if you get some object and would like to instantiate it:

```
Object createNewInstOf(Object obj) {
  return obj.getClass().newInstance();
}
```

--
Igor

è Question: I know that a default constructor is being defined from line 6 to 9 (see below). But I don't quite understand the syntax: this(blah, blah). Surely I know that "this" refers to the current object. But what on earth does "this (blah, blah, blah)" mean?

Would you please help explain in what kind of situation we need to use this kind of statement?

```
public class ThreadPool implements Runnable {
  private final int DEFAULT_MINIMUM_SIZE=5;
  private final int DEFAULT_MAXIMUM_SIZE=10;
  private final int DEFAULT_RELEASE_DELAY=10*1000;
  ...
  public ThreadPool(){
    this(DEFAULT_MINIMUM_SIZE, DEFAULT_MAXIMUM_SIZE,
        DEFAULT_RELEASE_DELAY);
  }
  ......
}
```

Answer: Whenever you encounter the :

```
this(blah, blah)
```

syntax, it means that another constructor should be called first:

```
public class MyClass{
  MyClass(){
    this(2,2); / / here you make a call to the other constructor
  }
```

```
  MyClass(int a, int b){
  }
}
```

The point here can be, that even if the user doesn't know which parameters to pass to MyClass(int, int) , she gets a default constructor which indirectly gives default parameters.

Important notes:
-calling this(...) is very similar to calling super(...) .
-indeed, this(..) may only be used inside a constructor, and may only be placed as the first instruction in a constructor (all like super(...)).

Note that super(...) will call some constructor from the parent class.
--
Arnaud.

è Question: Simple Question: why constructor doesn't work in following example?

```
class Start {
  public void Start() {
    System.out.println("constructor START");
  }
}

public class Test {
  public static void main(String[] args) {
    Start s = new Start();
  }
}
```

Answer: Because you have included the return-type 'void' in the method declaration, it becomes a normal method, which just happens to have the same name as the class - so it won't get used as a constructor. Remove the 'void' and it should work.
--
Vince Bowdren

P.S. by John: If you do not specifically define any constructors, the compiler inserts an invisible zero parameter constructor "behind the scenes". Often this is of only theoretical importance, but the important qualification is that you only get a default zero parameter constructors if you do not create any of your own.

Your program used this zero parameter constructor and you saw nothing...

è Question: Why we can not declare constructor as final?

Answer: The keyword final when dealing with methods means the method cannot be overridden.

Because constructors are never inherited and so will never have the opportunity to be overridden, final would have no meaning to a constructor.

è Question: In Java, does exist a function like sprintf in C?

Answer: http://www.efd.lth.se/~d93hb/java/printf/index.html a free Java version of fprintf(), printf() and sprintf() - hb.format package

è Question: If I declare an array of an objects, say Dogs, is that memory taken when I create the array or when I create the objects in the array when I declare this array:

```
Dog[] dog = new Dog[100];
```

or does it take the memory when I actually create the Dogs in the array e.g.:

```
  for(int i = 0;i<dog.length;i++)dog[i] = new Dog();
```

Answer: The statement above is actually two-fold. It is the declaration and initialization of the array. Dog[] dog is the declaration, and all this does is declare a variable of type Dog[], currently pointing to null.

You then initialize the array with new Dog[100], which will create 100 elements in the array, all of them referencing null.

It is important to realize that the elements of an array are not actually objects, they only reference objects which exist elsewhere in memory. When you actually create the Dog objects with new Dog(), these objects are created somewhere in memory and the elements in the array now point to these objects.

Pedant point:

Nothing ever points to null. It is a constant that represents the value of a reference variable that is not a pointer to some object new Dog[100] creates an array of 100 null Dog references.

è Question: How do I return more than one value using the return command?

Answer: You could make a new object/class that contains these two values and return it. For example:

Define an object like this:

```
class MyObj {
  public int myInt;
  public double myDouble;
}
```

Then, in your method create one of these, set the corresponding values, and return it.

```
MyObj yourMethod() {
  MyObj obj = new MyObj()
  obj.myInt = 20;
  obj.myDouble = 1.0003
  return obj;
}
```

è Question: How do I use object serialization for an object that has other objects as data member? Do both the class need to implement serialize?
How about static data?

```
class A{
}

class B{
   public A a;
}
```

Answer: Both the object and all the object references it contains need to belong to classes that implement Serializable.

Static and transient fields are not serialized. For more, see,
http://java.sun.com/docs/books/tutorial/essential/io/serialization.html

è Question: I recently learned a bit about "inner classes" but this seems to be different...

I'm a bit new to Java programming so bear with me. My employer bought a package of java graphics library programs to support some chart applets we want to create. We have the source code. I'm trying to create a jar file with all the files I need to run the applet. When I currently run the applet, the browser java tool says that it can't find "TextComponent$1.class".

I recently learned a bit about "inner classes" but this seems to be different. The "TextComponent.java" file does contain some inner classes, but not a class called "1". I'm confused.

Is this an inner class? Or is it something else. Any help would be appreciated. Thanks...

Answer: The TextComponent$1.class is the first anonymous class defined in TextComponent.java. Since nested (inner) classes are compiled to their own .class file, they needed unique names. The javac compiler is just creating a unique file name for an anonymous nested class.

è Question: Hi there, does anybody know a good source of design patterns written in JAVA?

Answer: A pretty good (free to download) book.
http://www.patterndepot.com/put/8/JavaPatterns.htm

è Question: What's the difference between the two: System.err. and System.out? When should we use System.err?

Answer 1: System.out leads the output to the standard output stream (normally mapped to your console screen), System.err leads the output to the standard error stream (by default the console, too). the standard output should be used for regular program output, the standard error for error messages. If you start your console program regularly both message types will appear on your screen.

But you may redirect both streams to different destinations (e.g. files), e.g. if you want to create an error log file where you don't want to be the regular output in.

On an UNIX you may redirect the output as follows:

```
java yourprog.class >output.log 2>error.log
```

this causes your regular output (using System.out) to be stored in output.log and your error messages (using System.err) to be stored in error.log

è Question: What's the difference between the two: System.err. and System.out? When should we use System.err?

Answer 2: System.err is a "special" pipe that usually is directed to the standard console. You can redirect the System.out with the normal pipe control (| or >), but System.err no. If you want to put both the "normal" output and the "error" output to a file you must use the special redirect 2>.

This allow you to send normal messages into a file or in the /null black hole, but still receive the error messages on the console.

è Question: What is the essential difference between an abstract class and an interface? What dictates the choice of one over the other?

Answer: You can only extend one class (abstract or not) whereas you can always implement one or more interfaces. Interfaces are Java's way to support multiple inheritances.

Does anyone know how could I get the size of an Enumeration object? The API for Enumeration only contains getNext () and next ().

Answer 1: You can't. Theoretically, some classes that implement Enumeration may also provide some way to get a size, but you'd have to know about the more specific run-time type and cast to it... and none of the standard java.util Collections classes nor Vector or such provides these methods in their Enumeration implementations.

è Question: What is the essential difference between an abstract class and an interface? What dictates the choice of one over the other?

Answer 2: You can make your own class like this:

```
import java.util.*;

public class MyEnumeration{
  int size;
  int index = 0;
  Enumeration e;
  public MyEnumeration(Vector v){
    size = v.size();
    e = v.elements();
    index = 0;
  }
  public boolean hasMoreElements(){
```

```
        return e.hasMoreElements();
    }

    public Object nextElement(){
        index++;
        return e.nextElement();
    }

    public int size(){
        return size;
    }

    public int getIndex(){
        return index;
    }
}
```

--
by Nicolas Delbing and Victor Vishnyakov

è Question: Is there a way to provide values for a Vector in the source code, analogous to array initializers?

Answer: The Vector class constuctors take no arguments other than Collection (since JDK 1.2), which is abstract, and since a Vector is a structure whose size can change dynamically, it's contents can only be initialized through member methods.
--
Mike Lundy

è Question: How would I add a help file to a java application?
Would it have to be platform specific, or is there a Java API for making help files?
If so, what is it?

Answer: See JavaHelp at http://www.javasoft.com/products/javahelp/
you create HTML pages for the main text, and add some XML files for a hierarchical table of contents and a map from TOC tags to relative URLs giving document locations.

è Question: What is a Just-In-Time (JIT) compiler?

Answer: It is a JVM that compiles Java instructions (called bytecode) into native machine instructions at run time and then uses this compiled native code when the corresponding Java code is needed. This eliminates the constant overhead of interpretation which tradition first generation JVM's used.
--
Dave Lee

è Question: Is there a collection object like the hashmap or hashtable that stores values in an ordered path? Vector does this but I need the key/value functionality. Hashmaps do not guarantee the order of the objects.

Answer: Take a look at java.util.TreeMap.
Red-Black tree based implementation of the SortedMap interface. This class

guarantees that the map will be in ascending key order, sorted according to the natural order for the key's class (see Comparable), or by the comparator provided at creation time, depending on which constructor is used.

Note that this implementation is not synchronized. If multiple threads access a map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally.

è Question: Most people asked why there is an error, but my question is why this is NOT an error...

Please take a look:

r is a number and s is a character, why can I put them together to make a comparison without compilation error? Could somebody tell me... thank you

```
double r = 34.5;

char s = 'c';
if (r > s) {
   System.out.println("r > s");
} else {
   System.out.println("r < s");
}
```

Answer: Yes, char is indeed a 16-bit value. However, the actual answer is in the Java Language Specification, section 5.6.2, which is at the following URL: http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#170983

In summary, the char is automatically promoted to a double. No explicit cast is necessary since the language rules say that it gets "promoted" to a double
--
by John O'Conner

è Question: == and equals ()... These two still make me confuse a lot of time. Can somebody give me some thumb rule or explain it to me?

Answer: When you use == with a primitive -int, double, char... you are checking that the values are identical. But if you use == with an object, you are checking that the 2 objects are stored at the same address. In other words the references pointing to the same object...

Method equals () is different.
It is the same as ==, if it isn't overridden by the object class.

Many classes override the method equals (). In this case this method will check that content of the object is the same or not, not addresses.

è Question: Why do I get message like “wrong magic number” when I am trying to run applet? What is magic number?

Answer: The first thing a JVM does when it loads a class is check that the first four

bytes are (in hex) CA FE BA BE. This is the "magic number" and that's why you are getting that error, you are trying to load a file that isn't a class and so the class loader in the JVM is throwing out that exception.

Make sure you transfer the class files to site in binary mode, rather than text or ASCII mode.

An error from the browser saying "cannot start applet ... bad magic number" usually means that one of the class files on the server is corrupted. '

Replace your class binary files on the web server; clean up the cache of your browser, and reload your applet.

è Question: In java, I found a lot of methods, which let you enter a line (read (), readLine () etc). They all wait until return is pressed, and then start providing you the information.

Does anyone know if there is a read method available whit the desired behavior, i.e. which doesn't wait for return being pressed?

Answer: Java does not provide it, the terminal itself waits until return is pressed before sending the entered line to Java.

You need to use some platform specific mechanism to change the terminal settings.

è Question: Can I pass a function as a parameter in Java? If so, how?

Answer: No, you cannot. But you can pass an object with method and then just use this method like this:

```
myFunction (object); // object has method useMyMethod(){ do smth here...}
..
..
somewhere in another place where use this function..
object.useMyMethod();
```

è Question: What is difference capacity() and size() methods for vector?

What is difference between

public final int capacity()
Returns the current capacity of this vector.

and

public final int size()
Returns the number of components in this vector.

Answer: please read this method: ensureCapacity()

public final synchronized void ensureCapacity(int minCapacity)

Increases the capacity of this vector, if necessary, to ensure that it can hold at least

the number of components specified by the minimum capacity argument.

Parameters:
minCapacity - the desired minimum capacity.

è Question: Can I use try/catch block to control program flow? Will it affect program performance?

Answer: Try/catch block itself does not add much overhead to your program. Handling of thrown exception can be expensive.
Use try/catch block only for error handling, not to control program flow. Especially try to avoid placing within loops.

è Question: I want to create a logging system that handles the user, database, application server logging, etc.

What are options?

Answer: Sun has introduced a new API for logging in JDK 1.4. It is in the package java.util.logging.

If you use a JDK prior to 1.4, use alternative Log4j from the Apache project (jakarta.apache.org/log4j).

It is an open source logging API. Very easy to use!

If you want to use something else on JDK 1.2 and 1.3 but similar to the final java.util.logging API in 1.4, use:

http://javalogging.sourceforge.net

è Question: I have a method that names something like this - public String doSmth()...

I do not need any Object for this method. Can I make it static? Is it good idea to make all methods that do not use Objects static?

Answer: yes, I think so. It is bad idea to create an instance of the class just to use one method without needing the instance itself.

Wasting of memory and CPU performance for further garbage collecting of the Object that has not been used :-)

è Question: What're the differences between classpath and import in the java application?

Do I need to have the package in the classpath first before importing in a java application or need not?

Answer: Classpath is an environment variable of your OS, you have to set it (or better: Java sets it) to tell Java where to search for classes.

You use import in a program to let Java search for the specified classes within the classpath. This implies that the package must be in the classpath.
--
Stephan Effelsberg

è Question: What is difference between Iterator and Enumeration?

First of all Java FAQ Team wish you HAPPY NEW YEAR!!! and then

Answer: from
http://java.sun.com/docs/books/tutorial/collections/interfaces/collection.html

The object returned by the iterator method deserves special mention. It is an Iterator, which is very similar to an Enumeration, but differs in two respects:

Iterator allows the caller to remove elements from the underlying collection during the iteration with well-defined semantics.

Method names have been improved.

The first point is important: There was no safe way to remove elements from a collection while traversing it with an Enumeration. The semantics of this operation were ill-defined, and differed from implementation to implementation.

The Iterator interface is shown below:

```
public interface Iterator {
  boolean hasNext();
  Object next();
  void remove(); // Optional
}
```

The hasNext method is identical in function to Enumeration.hasMoreElements, and the next method is identical in function to Enumeration.nextElement. The remove method removes from the underlying Collection the last element that was returned by next. The remove method may be called only once per call to next, and throws an exception if this condition is violated. Note that Iterator.remove is the only safe way to modify a collection during iteration; the behavior is unspecified if the underlying collection is modified in any other way while the iteration is in progress.

The following snippet shows you how to use an Iterator to filter a Collection, that is, to traverse the collection, removing every element that does not satisfy some condition:

```
static void filter(Collection c) {
  for (Iterator i = c.iterator(); i.hasNext(); )
     if (!cond(i.next()))
     i.remove();
}
```

Two things should be kept in mind when looking at this simple piece of code:

The code is polymorphic: it works for any Collection that supports element removal, regardless of implementation. That's how easy it is to write a polymorphic algorithm under the collections framework!

It would have been impossible to write this using Enumeration instead of Iterator, because there's no safe way to remove an element from a collection while traversing it with an Enumeration.

è Question: How can I find the first dimension length of the 2-dimenstions array? I use the array [].length but it does not work, how can I solve this problem?

Answer: Java doesn't really have "multidimensional arrays", only arrays of arrays. So try: array[0].length and you will get this dimension.

è Question: I guess what I'm asking is "Is java.util.Hashtable thread safe?"

It's been a while since I've used hashtables for anything significant, but I seem to recall the get() and put() methods being synchronized.

The JavaDocs don't reflect this. They simply say that the class Hashtable is synchronized. What can I assume? If several threads access the hashtable at the same time (assuming they are not modifying the same entry), the operations will succeed, right? I guess what I'm asking is "Is java.util.Hashtable thread safe?"

Answer: That is right! It is recommendable, if you have questions like these, always look at source for the API, it's freely available.

è Question: I try to copy an object of my own using the clone() method from java.lang.Object, but this is a protected method so I can't use it. Is there some other way to get my objective of duplicating an arbitrary object?

Answer: If you want to clone your object, you need to make it cloneable. To achieve this, you need to do two things:

1. implement the interface Cloneable
2. override the method clone(), so that it
   a. becomes public
   b. calls super.clone()
   c. if necessary, clones any members, or
   d. if a member can't be cloned, creates a new instance.

Simple example:

```
public MyClass implements Cloneable {
   int someNumber;
   String someString;

   public Object clone() {
   // primitives and Strings are no
   // problem
      return super.clone();
  }
}
```

In this case the method clone() of the class MyClass returns a new instance of MyClass, where all members have exactly the same value. That means the object reference 'someString' points to the same object. This is called a shallow copy. In many cases this is no problem. Strings are immutable and you do not need a new copy. But if you need new copies of members, you have to do it in the clone() method. Here is another simple example:

```
public class SomeMember implements Cloneable {
  long someLong;
  public Object clone() {
    return super.clone();
  }
}

public AnotherClass extends MyClass {
   SomeMember someMember;
   public Object clone() {
      AnotherClass ac = (AnotherClass)(super.clone());
      if (someMember != null) {
         ac.someMember = (SomeMember)(someMember.clone());
      }
      return ac;
   }
}
```

Note that the class AnotherClass, that extends MyClass, automatically becomes Cloneable, because MyClass is Cloneable.

Also note, that super.clone() always returns an Object of the type of the actual object, although the superclass doesn't know anything about that sub class. The reason is, that Object.clone() is a native method, which just allocates new memory for the new object and copies the bytes to that memory. Native code has it's own ways of finding out which type to return ;-)
--
Karl Schmidt

è Question: I was just wondering about the usefulness of Interfaces...
I was just wondering about the usefulness of Interfaces. I was under the impression that interfaces could be used to perform multiple inheritance. But an interface only declares a method - in a very abstract way.

A class that implements an interface needs to define its own implementation of a certain method. What is the use of having an interface when nothing is being gained...?

Answer: If two classes implement the same interface, you can get a reference to the interface instead of the effective class without bother what class are you managing.

This is very useful in RMI (for example) or in any condition when you have to take an object without knowing exactly his class, but only the interface that it implement.

For example:

```
public void recurseList( List l )
```

the generic List ensure that you can use every List for this method (ArrayList, AbstractList, Vector...), so your calling method can be:

```
ArrayList l = new ArrayList(); or
Vector l = new Vector();
recurseList( l );
```

Without any problem.
--
by Davide Bianchi

è Question: I got a problem with an array/vector... I got a problem with an array/vector.

my class contains a member:

```
static Vector quad[][];
....
```

in my constructor I got:

```
Vector quad[][] = new Vector[row][col];
for (int i = 0; i < row; i++){
   for (int j = 0; j < col; j++){
      quad[i][j] = new Vector (0,1);
   }
}
```

// row and col are int between (10..50) -- it's a big I know, but that might not be the problem.

My PROBLEM that (and I don't know what to do, really) I can't access quad[x][y] outside of the constructor!!!! Within the constructor I've got full access on quad[x][x]. Java (1.2) returns a NullPointerException on any method except within the constructor!!!

What's my fault!???

è Question: What is Java Space technology?

Answer: The dominant model of computation in distributed computing is the Client\Server model. This model is based on the assumption that local procedure calls are the same as remote procedure calls.

Javaspace technology is based on a new model of computation called the tuplespace model and aims to overcome some of the key difficulties related to Client\Server communication.

Javaspaces overcome the problems of synchronization, latency and partial failure, inherent in distributed systems, by providing loosely coupled interactions between the components of distributed systems.

Javaspace processes communicate through a space, not than directly. Communication between processes on different physical machines is asynchronous and free from the main limitation of the traditional client/server model, when client/server communication requires simultaneous presence on network both parts - client and server.

JavaSpace technology. Sender and receiver in Java Space don't need to be synchronized and can interact when network is available...

In a distributed application, JavaSpaces technology acts as a virtual space between providers and requestors of network resources or objects. This allows participants in a distributed solution to exchange tasks, requests, and information in the form of Java technology-based objects.

n Question: How is JavaSpaces technology different from a database?

Answer: Databases require a tremendous amount of overhead in structured programming, screen design, type matching, and indexing to achieve their results. Clients post specific questions to the server, and receive a specific answer. JavaSpaces technology provides a more loosely coupled repository of information. The identity of a client or a server is no longer relevant; data packets are treated just like any other object posted to the space as an anonymous service.

More importantly, JavaSpaces technology can host more than just data. A space can find, match and reference objects by both type and value, meaning they can store objects as information or behavior. This means any object-based program, device, or user profile can join a JavaSpace system. This capacity greatly enhances the ability to coordinate different functions and processes in a network environment.

Java Space technology is ready for usage product although it is still under development. One example of Java Space based product here:

http://j-spaces.com/

This newsletter is based on:

"JAVASPACES: A CRITIICAL ANALYSIS OFF CONCEPTS & TECHNIQUES"

http://www.dcs.shef.ac.uk/teaching/eproj/ug2001/pdf/u8sm.pdf

"JavaSpacesTM FAQ"

http://java.sun.com/products/javaspaces/faqs/jsfaq.htm

n Question: What is meant by persistence? What is relation of persistence to high availability?

Answer: Persistence indicates that a collection of data remains intact even if its source is no longer attached to the network. The objects in JavaSpaces for example, remain available to other users even if the source has temporarily disconnected from the network. This feature is critical to keep distributed systems highly available.

Question: I heard many times from Java lectors: "If you are interested how new feature was implemented in JDK please check the source code...". I tried to find the source code but have not succeeded.

Of course, I can decompile class files but could be good to have "original" source code from SUN, not from hacking tools :-)

Answer: I found the SUN's page and what was there:

"Sun is making the source code to the Java 2 platform available to the developer community. This offering is intended to further propel innovation of the Java platform and provide developers with easier access to Java platform sources while still maintaining compatibility.

Do You Need The Source Code?

In most cases, you do not need the source. In fact, most developers will find all the basic resources and tools they need in the Java 2 Software Development Kit (formerly known as the JDK 1.2).

The source code to the Java 2 platform can be invaluable if you are doing any of the following:

conducting research work
debugging possible platform bugs
adding enhancements which require access to platform internals

But building, porting, and testing the Java 2 platform source code requires a high level of technical expertise. "

P.S. About last sentence: I am pretty sure that all readers of my newsletter have it.

n Question: I am getting "OutOfMemoryError". How do I prevent such memory leaks?

Answer: It is not always a memory leak that causes "OutOfMemoryError".

A memory leak is a situation when some application, for example JVM allocates some amount of memory and then loses

pointers to that area. So, the allocated area becomes a "black box" which you can not access and you need a key to open it.

It is very rare situation when some Java code causes memory leaks. They are reported and can be found at "Bug parade" on SUN's web site.

"OutOfMemoryError" often caused by bad design. It can be for example:

1. Your program trying to process too much data. Increase the amount of memory or decrease the amount of data to be processed. Actually you can do it in a wise way: control how much memory left and if it less than, let say 20 %, decrease the processing speed. It was described in one of our tips while ago...

2. Your program holding too many objects and then program is running out of memory.

You need to use close methods if you do need them. You can even explicitly set a reference to null to speed up garbage collection.

3. You use too many objects with finalizers... They consume your memory before you get any chance to run them...

- Question: I have an application which can be run using 64 bit JNI libraries or 32 bit libraries depending on the command line parameters. I want to know if there is any way for me to find out if the JVM is running in 64 bit mode (the -d64 parameter was

  sent to it) or 32 bit mode.

  Answer:

  Run this small program and you will get a long list of System properties...

```
public class ShowSystemProperties {
  public static void main(String[] args) {
    System.getProperties().list(System.out);
  }
}
```

  One of lines will be

  sun.arch.data.model=32

  or

  sun.arch.data.model=64

  Just search for it and you will know which version you are running on.

- Question: What is the maximum number of dimensions that a Java array can take?

  Answer: It is not limited by Java Language Specification, but limited by implementation...

  For example, for java.lang.reflect.Array newInstance method I found this:

  "The number of dimensions of the new array must not exceed the number of array dimensions supported by the implementation (typically 255)."

- Question: Is there any way to keep an object from being collected by Garbage Collector?

  Answer: 8-)

Yes, of course! Just keep a reference to an object and it will not be garbage collected (according to definition of GC)!

n Question: Reading Java's API version 1.41 I noticed that many classes contain now valuable information about thread safety. Are all classes having such useful info?

Answer: That is almost right. SUN added descriptions to many classes. For example see bug No 4264153

(http://developer.java.sun.com/developer/bugParade/bugs/4264153.html)

I am not quite sure that all classes have a description. At least I have not found any information that SUN updated ALL descriptions...

My recommendation is to read API... If there is no information on thread safety for particular class to assume that class is not safe. It will help to avoid strange runtime errors which are difficult to debug.

n Question: Do you know about "Full Speed Debugging" and "HotSwap" Class File Replacement in Java Debugger since v.1.4.1? Part 1.

Answer:

Full Speed Debugging

The Java HotSpot[TM] virtual machine now uses "full-speed debugging". In the previous version of HotSpot, when debugging was enabled, the program executed using only the interpreter. Now, the full performance advantage of HotSpot Technology is available to programs running with debugging enabled. The improved performance allows long running programs to be more easily debugged. It also allows testing to proceed at full speed and the launch of a debugger to occur on an exception:

```
-
Xrunjdwp:transport=dt_socket,server=y,onuncaught=y,launch=myDebuggerLaunch
Script
```

For details on launch "onthrow" and launch "onuncaught", see Connection and Invocation Details . Setting a breakpoint only inhibits compilation (full speed execution) for the method containing the breakpoint. Execution reverts to the interpreter during some debug operations such as single-stepping and when method entry/exit or watch points are set.

As of version 1.4.1, full-speed debugging is available with both the Java HotSpot Client Vm (default) and Java HotSpot Server VM (invoked with the -server command-line flag). In the Java 2 SDK v1.4.0, full-speed debugging was available only with the Java HotSpot Client VM.

This tip is based on release info for the version 1.4.1 family of the Java 2 SDK.

n Question: Do you know about "Full Speed Debugging" and ”HotSwap" Class File Replacement in Java Debugger since v.1.4.1? Part 2.

"HotSwap" Class File Replacement.

This new feature encapsulates the ability to substitute modified code in a running application through the debugger APIs. For example, one can recompile a single class and replace the old instance with the new instance.

This change was made to address these issues:

Tool (IDE) vendors want the ability to do fix-and-continue debugging. That is, while debugging, identify a problem, fix it, and continue debugging with fixed code.

Organizations deploying long running servers wish to be able to fix bugs without taking down the server.

HotSwap adds functionality to the Java TM Platform Debugger Architecture (JPDA) to allow a class to be updated while under the control of a debugger. The two central components of this functionality are RedefineClasses which replaces the class definitions and PopFrame which pops frames off the stack allowing a method which has been redefined to be re-executed.

In the reference implementation, this functionality is implemented at the Java Virtual Machine Debug Interface (JVMDI) layer and made available through the higher layers of JPDA - the Java Debug Wire Protocol (JDWP) and the Java Debug Interface (JDI).

This tip is based on release info for the version 1.4.1 family of the Java 2 SDK.

n Question: I see that in Java a lot of Class loaders: bootstrap, extension, system, context and even user defined! It is difficult to understand Class loader hierarchy in Java..

Answer: It was much better in Java 1.1 time :-)

There were no relations between system class loader (responsible for loading in the Java runtime, the application, and classes and resources in the application's CLASSPATH) and applet class loader (loaded classes over networks).

Since Java 1.2 we have three types of class loaders:

1. Class loaders created automatically by the JVM
2. Program defined class loaders
3. Context class loaders.

There are three Class loaders in first group:

bootstrap class loader - loads classes from ../jre/lib/rt.jar
It is the "root" in the class loader hierarchy.
extensions class loader - loads classes from ../jre/lib/ext/*.jar
system class loader - it is responsible for loading in the application, as well as for loading classes and resources in the application's CLASSPATH.

Second group includes system class loader - parent class loader by default additional parent class loader can be specified explicitly

Third group is context class loader. A thread's context class loader is, by default, set to the context class loader of the thread's parent. The hierarchy of threads is rooted at the primordial thread (the one that runs the program). The context class loader of the primordial thread is set to the class loader that loaded the application. So unless you explicitly change the thread's context class loader, its context class loader will be the application's class loader. That is, the context class loader can load the classes that the application can load.

To change a thread's context class loader, you use Thread.setContextClassLoader().

n Question: I am getting "StackOverflowError" error...

Where should I look at?

Answer: "StackOverflowError" usually caused by an infinite recursion.

è Question: I got a problem with an array/vector...

my class contains a member:

```
static Vector quad[][];
....
```

in my constructor I got:

```
Vector quad[][] = new Vector[row][col];
for (int i = 0; i < row; i++){
  for (int j = 0; j < col; j++){
    quad[i][j] = new Vector (0,1);
  }
}
```

// row and col are int between (10..50) -- it's a big I know, but that might not be the problem

My PROBLEM (and I don't know what to do, really), I can't access quad[x][y] outside of the constructor!!!! Within the constructor I've got full access on quad[x][x]. Java (1.2) returns a NullPointerException on any method except within the constructor!!!

What's my fault!???

Answer: I guess you shouldn't write Vector here:

```
Vector quad[][] = new Vector[row][col];
```

so, the correct variant may be:

```
quad[][] = new Vector[row][col];
```

I guess You just overridden your static variable with one defined in your constructor: Vector quad[][].

Thus, you're initializing NOT your class-scope static variable but your constructor-scope quad. It's not reachable outside the constructor. And as for static quad, it has never been initialized! And a first reference to it causes NullPointerException. I guess. I hope I'm right :)
--
Xao Rom

è Question: I know that Java file should have the same name as public class in the file.

But what are rules for files that have no public classes?

Answer: No rules are defined for such case! It is up to you how to name it. Please check my example:

```
// ****** ABCDEFG.java file *******************
class G {
  public static void main(String[] args) {
    System.out.println("This is class G!");
  }
}

class Z {
  public static void main(String[] args) {
    System.out.println("This is another class Z!");
  }
}
class AB {}
class CD {}
class EF {}
```

After compilation you will get AB, CD, EF, G and Z classes. You can run G and Z (they have main method), but not AB, CD, EF.

Try it:

```
java G
```

or

```
java Z
```

è Question: I mainly use C++ but I have a program to write and java seems to be the right tool for this job. That said my java skills are a bit rusty. I downloaded a free IDE and am ready to go but I'm not a fan of putting my method implementations in the class definition.
Is there a way to separate the implementation from the class definition? Like in c++ there are .h files and .cpp files.

Answer: You can use an interface and an implementation class, if you like.

The interface has just the method signatures; the class has their implementations. You can then declare variables having the type of the interface, and assign objects of the implementation class created with "new", or by a factory method.

Java is very different from C++ in that the interface of a class is defined by a compiled view of that class, not by textual inclusion of source for a separate interface description. Therefore the concept of splitting interface from implementation does have the same meaning in Java as in C++.

In Java, you can separate the description of pure interfaces (or abstract classes) from concrete classes that implement (or complete) them. This is a useful technique - compare java.util.Collection and java.util.List to java.util.ArrayList and java.util.LinkedList.

Also, it is traditional to define the return type from factory methods and other such producers to be pure interfaces to allow the factory or other class to decide which implementation to give you - this is used extensively in java.net and java.sql packages.
--
Phil Hanna (Author of JSP: The Complete Reference
http://www.philhanna.com), Chuck

è Question: How do we exchange data between Java and JavaScript and vice-versa?

Answer: Public variable and methods of Java Applet are visible to a HTML document. So using JavaScript you can access the public variables and public functions.

The syntax is:
var some_var = document.appletname.variable_name

With this you will get the value of the variable variable_name in your JavaScript variable some_var.

è Question: Constructors and methods: are they the same?

I need a little help here...I have been teaching that constructors are not methods. This is for several reasons, but mainly because JLS says "constructors are not members" and members are "classes, interfaces, fields, and methods."

So, now the rest of the staff is ganging up on me and making life a little nasty. They quote Deitel and Deitel, and Core Java (which references "constructor methods") and who knows how many other books.

The one we are teaching in is loaded with so many errors that even though it calls constructors methods NOBODY will quote it as an authority.
How can so many people call constructors methods if they aren't.

Okay. Are they or aren't they? I holding to the definition that they are not unless it is so common to call them that, that I will have to change. Comments?

Answer: If you go by the JLS (Java Language Specification) and the Java API (and you should), then no, constructors are not methods. Consider that Class.getMethods() returns an array of Method instances and Class.getConstructors() returns an array of Constructor instances, and Constructor and Method or not interchangeable (one is not derived from the other), but both implement the Member interface. Seems to me that Java is going out of its way to differentiate them.

Besides, the mechanics of constructors are so different from the mechanics of methods, there seems to be no value to considering one a member of the set of the other.

Now, as far as teaching the language goes:
Methods:
+ return types
+ called by name
+ executed multiple times

Constructors:
+ super(...) or this(...) as first instructor (often implicit)
- everything else

I very much do not like trying to lump the two concepts together, especially in introductory courses. Conceptually they are very, very different things.

A constructor is a unique element (even if there are several). It has the name of the class, its declaration is different, and it doesn't have the same syntax as a method. It plays a unique role. You can call a method from a constructor, but you cannot call a constructor from a method.

I say constructors are distinct from methods, and for students, blurring the distinction will lead to problems.
--
by Chuck McCorvey, Chris Wolfe,
-- Paul Lutus, www.arachnoid.com

è Question: I am wondering if a constructor has return type...

Answer: It has no return type, not even void...
But speaking more generally we can assume that implicit return type is a class type itself.

è Question: Constructors are similar to methods... I know that they do not return any value. Could I say that they have void type (no return type)?

Answer: Not. Constructors are not methods and they are different because they:
* always have the same names as the class name
* have no return type, even void!
* are not inherited and that's why (by the way) you can declare them final.

è Question: Ok, I know that one cannot put primitive types into a hashmap (only objects or references to them) and I know how to deal with that (write some kind of wrapper class).

What I'm interested in is: 'Why is that?' Why can I not put a primitive type into a hashmap? Something to do with this 'heap' thing… Right?

Answer: HashMap requires a key or a value to be assignment- compatible with java.lang.Object (i.e. to be an object or an array). Primitive types aren't ones.

You can use wrappers like java.lang.Integer for this purpose. All container classes require some basic operations to be defined for all of its contained objects in order for it to organize the data.

For example, a hashmap requires a hashing method hashCode (). A primitive type doesn't have any methods associated with it.

The distinction between primitive types (int) and their wrapper classes (Integer) is made purely for optimization purposes. For example, we wouldn't want to have to instantiate actual objects in an array instead of just allocating a block of memory for an array of integers.
--
Gary

è Question: I propose that Java should allow multiple inheritances if...

Everyone knows the potential problem with multiple inheritances is when you run into the problem of having two instances of a grand parent super class.

For example:

```
class A extends D {int i; }
class B extends D {int i; }
class C extends A,B {}
```

Potentially, you could have two copies of D for each instance of C.
However, I propose that Java should allow multiple inheritance if there are no instance variables associated with the abstracts that the base class is extending.

```
abstract class A { public setX(); public setY(); public setAll() {setX (); setY(); }
abstract class B { public setC(); public setD(); public setBoth()
{setC(); setD(); }

class C extends A,B {}
```

You won't have two instances of some grandfather class, since A and B doesn't have instances variables.

I hope the next versions of Java explore this issue.

Answer: It does. They're called interfaces:

```
interface A { public void setX(); public void setY(); public void setAll(); }
interface B { public void setC(); public void setD(); public void setBoth(); }

interface C extends A,B {};
public abstract class D implements C {
}
```

--
jim

è Question: Can I access a private variable of one Object from another Object?

Answer: Yes, if this object of the same class. You can access private field from static

method of the class through an instance of the same class.

Check one example below:

```
public class Example {
  private int privateVar;
  Example(int a) {
    privateVar = a;
    System.out.println("private privateVar = " + privateVar);
  }

  public static void main(String[] args) {
    Example ex = new Example(5);
    System.out.println(ex.privateVar);
  }
}
```

--
AP (JA)

è **Question:** Is there a way to know from class X which class called the method foo()?

If class A and class B are calling a method foo() on class X, is there a way to know from class X which class called the method foo() (they can be either A or B). I know that this can be done by capturing the stack trace and examining it, but that solution looks expensive as I have to create a new Throwable object every time and capture stack trace (And I do this quite frequently).

Is there any other elegant solution to do this, any help and directions appreciated.

Answer: Pass a reference to the class to the foo() method.

```
foo(Object x){
    System.out.println(x.getClass());
}
```

should do it.

è **Question:** Why does this simple application never exit?

```
public class UIQuitTest {
   public static void main (String[] args) {
      java.awt.Frame f = new java.awt.Frame();
      f.dispose();
      f = null;
  } // end of main ()
}
```

The application above never quits, is it a bug or a (mis)feature? Win98, JRE 1.3.0

Answer: By creating an AWT object, you now have started the AWT thread. In order to end the application now, you have to do a System.exit (0) that will kill all non-daemon threads, including the AWT thread.

è Question: Is it possible to stop an object from being created during construction?

For example if an error occurs inside the constructor (e.g. the parameters pass in were invalid) and I wanted to stop an object being created would it be possible to return null rather than a reference to a new object. (I know the term return is technically correct in this case but you know what I mean).

Basically, is it possible to cancel object creation?

Answer: Yes, have the constructor throw an exception. Formally, an object _will_ be created (since the constructor is a method invoked after the actual method creation), but nothing useful will be returned to the program, and the dead object will be later reclaimed by Garbage Collector.

But the clean way is as another reply suggests, that you leave calls to the constructor to a static factory method which can check the parameters and return null when needed.

Note that a constructor - or any method in general - throwing an exception will not "return null", but will leave the "assign target" as it was.
--
Tor Iver Wilhelmsen

è Question: What does mean "volatile"?
For the past couple of hours, I've seen quite a few set of codes that has the _volatile_ keyword.

E.g.
private volatile somethingsomething....

What does it mean?

Answer: See JLS 2nd Edition, which just came out last year, still mentions it.

http://java.sun.com/docs/books/jls/second_edition/html/classes.doc.html#36930

"A field may be declared volatile, in which case a thread must reconcile its working copy of the field with the master copy every time it accesses the variable. Moreover, operations on the master copies of one or more volatile variables on behalf of a thread are performed by the main memory in exactly the order that the thread requested."

Seems just like the idea in C++ and appears still to me present in the language.

Synchronization certainly has it place in many applications, that doesn't mean that volatile is not longer used or part of the language.
--
Paul Hill

è Question: suppose I put a file a.txt in package com.xyz and the try access it like following. Will it work?

```
import com.xyz.*;
public class Hello{
```

```
    File f = new File("a.txt");
    ...
}
```

it is not working for me. Is there any workaround?

Answer: If the source and the text file are in the jar file, then you access the file by:

URL fileURL = getClass().getResource("file.txt");

You can then read the file by using a reader (or whatever you choose), e.g.:

_istream = new BufferedReader(new InputStreamReader(fileURL.openStream()));
--
johneweber

Or, simpler
getClass().getResourcesAsStream("file.txt"), but you must be sure that file is in the same directory (package) as your class, otherwise you need play with getClassLoader().getResourceAsStream(<path>"/file.txt");
--
Oleg

è Question: Difference between loading and instantiating a class?
Well, the subject says it all. What is the difference between loading and instantiating a class in a JVM?

Second Question: What would happen if at runtime I update a class file? Will the JVM know to use that instead?

Answer: The difference is that when a class is loaded by a ClassLoader it is read in as a stream of bytes, presumably from a file, but it could just as easily be from over the network, and then processed or "cooked" into a representation that the VM can use to make instances of Objects of that classes type. This last part is the instantiation. You can load a class at runtime with:

Class.forName("MyClass");

and instantiate one with:

MyClass mc = Class.forName("MyClass").newInstance();

Cool, ehh. You don't have to know the name of a class at compile time.

Answer to second Question: What would happen if at runtime I update a class file?

Loaded classes are cached because it's quite costly to do the "cooking" I mentioned above. So it will not be loaded. You may create a separate ClassLoader with new SecureClassLoader but that will cause all classes _it_ loads to be loaded from this new ClassLoader but that's not what you want.

I don't know if you can specify that a class should be loaded from disk again using the normal ClassLoader. You could very easily make your own ClassLoader in which

case you would have explicit control over such things. Look at java.lang.ClassLoader and java.lang.Class.
--
Michael B. Allen

è Question: Why developers should not write programs that call 'sun' packages?

Answer: Java Software supports into the future only classes in java.* packages, not sun.* packages. In general, API in sun.* is subject to change at any time without notice.

A Java program that directly calls into sun.* packages is not guaranteed to work on all Java-compatible platforms. In fact, such a program is not guaranteed to work even in future versions on the same platform.

For these reasons, there is no documentation available for the sun.* classes. Platform-independence is one of the great advantages of developing in the Java programming language. Furthermore, Sun and our licensees of Java technology are committed to maintaining backward compatibility of the APIs for future versions of the Java platform. (Except for code that relies on serious bugs that we later fix.) This means that once your program is written, the class files will work in future releases.

For more details, see the article Why Developers Should Not Write Programs That Call 'sun' Packages.
http://java.sun.com/products/jdk/faq/faq-sun-packages.html

è Question: Can garbage collector remove my singleton? Part 1
A usually singleton

```
public class Single{
   private static Single single;
   private Single {}
   public static Single getInstance(){
      if(single==null){
         single = new Single();
      }
      return single;
  }
}
```

Well, seems good?

But classes are objects too...so do Java 2 v1.3 class garbage collecting? Meaning my singleton could disappear if I don’t keep a reference to it (or the class itself) somewhere?

If classes are not garbage collected, that's pretty stupid, I don’t want classes taking up memory when I perhaps never will use it again...

Answer: No. Classes can define objects. That is, only the dynamic part of the class defines objects. The static part exists only in one place in memory and can not be duplicated. You can call the getInstance() method from anywhere in your program.

Java requires however that you tell where to find the method, in this case in the

Single class.
Therefore, you should use

Single.getInstance()

to get the instance. This is (though it looks much like it) not an execution of a method on an object, but just a method call without object. Single is only used to find out which getInstance() method should be used, and where it is.

You could add a delete() method if you don't need the instance anymore:

```
public class Single{
   private static Single single;
   private Single {}
   public static Single getInstance(){
      if(single==null)
          single = new Single();
      return single;
    }

   public static delete(){
      single = null;
   }
}
```

The garbage collector can now remove the single object after delete() is called if memory is needed.
--
Rijk-Jan van Haaften
Dept of Computer Science, Utrecht University, the Netherlands

è Question: Can garbage collector remove my singleton? Part 2
P.S by John:

Doing deeper investigation of this question I found one very good article about this topic. Everybody who are interested can read full article here:

http://developer.java.sun.com/developer/technicalArticles/Programming/singletons/

For the rest of our audience shortly:
A Singleton class can be garbage collected and when
".. a Singleton class is garbage-collected and then reloaded, a new Singleton instance is created. Any class can be garbage-collected when no other object holds reference to the class or its instances. If no object holds a reference to the ' Singleton object, then the Singleton class may disappear, later to be reloaded when the Singleton is again needed. In that case, a new Singleton object will be created.

Any static or instance fields saved for the object will be lost and reinitialized.

This problem exists in older Java TM Virtual Machines1. JDK 1.2 VMs, in particular, conform to a newer class garbage collection model that forbids any class in a given classloader to be collected until all are unreferenced".

And you "... can also set your VM to have no class garbage collection (-Xnoclassgc on

the JRE 1.3, or -noclassgc on the IBM JVM). Keep in mind that if you have a long-running program that frequently reloads classes (perhaps through special class loaders such as the remote class loaders), you have to consider whether that could cause a problematic buildup of garbage classes in the VM."

Also some people asked what is a Singleton and what is relation has it to Java. Shortly a Singleton is one of classical design patterns that are used in software development. More please read in free book here:

http://www.patterndepot.com/put/8/JavaPatterns.htm

è Question: I study patterns now and would be lost and forget everything very soon!

I am very new to design patterns and just bought the GoF book to learn about it. But as I complete the first couple of chapters right away, I see that they have 23 different patterns and I would be lost and forget everything very soon if I sequentially (page by page) read the entire book!

Do any of you have recommendations on how to proceed with this book so that I 'll remember at least some of them by the time I am done with the book? I can see that many of the classes in java API use composition, facade etc...

But other than that I don't think I will be able to gather anything unless I am able to remember where & when to use particular pattern A, B or C...

Answer: Glad to hear you got the GoF book, it's a great reference manual for patterns. As you've found, however, it's a bit heavy to just "read." What I recommend to people is that they pick a few of the easier, more commonly used patterns: Singleton is a no-brainer that pops up a lot. Adapter tends to get used here and there and isn't that difficult to understand. If you're doing Swing, then definitely go learn the Observer pattern. It'll help to keep you from mixing data and interface code.
Once you've learned three or four and have used them a few times, then as you start new projects; look back to the text to see if there are opportunities in your project where other patterns can be used.

You'll find that over time you'll use more and more of the patterns (some a lot more than others, obviously). I've often found cases where I missed a pattern during design and had "the light go on" after I'd written a bunch of code and realized I was either using a known pattern by accident, or could have used a pattern to my advantage. When possible, I then go back and adjust the design/code to match the pattern.

Keep in mind that the patterns generally don't appear as "absolute." It's expected that you may have variations to satisfy your application's needs.

It's really helpful to others, however, if you make a quick note in your design doc/code about what pattern you were using (which helps them learn patterns too, and helps them understand what you were up to if they know the pattern already).
--
Rico Trooper

è Question: When I used java.util.Hashtable, I also used interface Enumeration. But I found that this interface contains defined methods.

How can an interface contain defined methods? Isn't an interface supposed to hold abstract methods only?

Answer:
> How can an interface contain defined methods?
No.

> Isn't an interface supposed to hold abstract methods only?
Yes. (and fields). I'm not sure why you think Enumeration contains defined methods, since it does not (I checked the source to be sure). Keep in mind that an implementing class is also of type Enumeration. So Hashtable's keys method which returns an Enumeration will actually be returning an inner class that implements Enumeration (and therefore defining hasMoreElements and nextElement in a manner that is specific to traversing a Hashtable).

If this doesn't help, maybe you could clear up what you mean by Enumeration having defined methods.
--
Eric

è Question: can anyone provide an example on how to use clone () and Cloneable interface?

Answer:

```
//Class Point uses default clone implementation that returns
//shallow copy of Point (which is sufficient in this case).

class Point implements Cloneable {
    int x, y;
}

class ImmutableRectangle {
  private Point lowerLeft = new Point();
  private Point upperRight = new Point();
  public Point getLowerLeftPoint() {
    try {
      return lowerLeft.clone();
    }
    catch(CloneNotSupportedException e) {
        //shouldn't occur since Point is cloneable
        throw new InternalError();
    }
  }
}
```

we don't want Rectangle points to be changed outside its class. the best way would be to create a copy of Point object and pass it.

i5mast
MVP for Java 2

http://www.brainbench.com

è Question: I need to programmatically replace an entry in a zip file.
I could not quite get it using the ZipOutputStream because it simply creates a new file and write only that entry for me. The rest of the original entries are gone.
Does anyone have a solution for this?

Answer:
1) Read the file (myfile.properties) out of the original Zip Archive (original.zip)
2) Make your changes, write the file to the file system
3) Create a New Archive (originalNew.zip)
4) Write your edited file (myfile.properties) to originalNew.zip
5) loop through all the entries in the original Zip archive (original.zip), adding them to the new archive (originalNew.zip) EXCEPT for the file you are replacing (myfile.properties)
6) When you're done, erase the original and rename your new one to original.zip

I believe that this may be the only way to do this, since there doesn't seem to be any random access in the ZIP file.
--
Kevin T. Smith

è Question: What is better to use: array or vector?
Just wondering as I am using Vectors to store large amounts of objects from 50 to 4000 and each one has to be "looked at" every time paint is called...

Just wondering if it would be better to use an array, list etc?

Answer 1: Since the Vector method uses an array for storage but has extra steps involved in getting an element, use an array for fastest access.
--
WBB Java Cert mock exams http://www.lanw.com/java/javacert/

Answer 2: arrays are faster, vectors are more dynamic.

This should be evident just looking at the amount of code you need to traverse one versus the other. It might also be beneficial to write a linkedlist class and use that. That way you have a dynamic container which has potential to be faster than a vector (though still not as fast as an array). The problem with arrays is that if you need more space than the current size, you have to hardcode their copying into a bigger array.

Conversely, if you never (or rarely) use the entire array, its a waste of space and memory.

The following are benchmark test results of vector vs. array (ran on a 200-Mhz Pentium w/ 96 Mbytes of memory and Windows95):

Allocating vector elements: 17910 milliseconds
Allocating array elements: 4220 milliseconds
Accessing Vector elements: 18130 milliseconds
Accessing array elements: 10110 milliseconds

One other reason for vectors being slower that I did not mention above is that vector

methods are synchronized, which creates a performance bottleneck.
--
MSW

è Question: Would anyone know the performance issues regarding Vector's?
I am actually talking about resource pooling. I have objects that wait in a queue. It is a vector that keeps growing, as the queue gets bigger.

Do Vectors have much performance hit? Is there a better way to implement vectors to get the best out of them? Or am I better of creating a fixed size array?

Answer: If you just want a LIFO or LILO queue, you may be better off with LinkedList than with Vector, as then you'll never have to wait for the contents to be copied.

Vectors perform pretty well, but if you know (even roughly) how big you're going to need it to be, specifying that in the constructor call can help.

How sure are you that this will be a performance bottleneck? Premature optimization is the root of all evil...

The Vector class is thread-safe. By that I mean that there is no way to corrupt the internal representation of the data by accessing the vector from more than one thread. However, it is still possible, very easy in fact, to use a vector in a way that is not thread safe.

Consider this code:

```
for (int i = 0; i < vector.size(); i++) {
  System.out.println(vector.elementAt(i));
}
```

It looks safe, but there's a subtle flaw...

è Question: Isn't the Vector class included in Java 1?
I'm writing an applet where I need to use something like ArrayList or Vector. Is there anything in Java1 I can use? When I try using the Vector Explorer gives me the message:

```
java.lang.NoSuchMEthodError: java/util/Vector:
method add (Ljava/lang/Object;)Z not found
```

Answer: Before Java2 there was only the method addElement(Object). The method add(Object) was included with the List interface.

I suggest you downloaded the JDK1.1.8 documentation and referred to that (to decide which methods to use).

To avoid such problems in a future please check the 1.1 Vector doc online at:

http://java.sun.com/products/jdk/1.1/docs/api/java.util.Vector.html

or, better still, install the 1.1 docs (download from

http://java.sun.com/products/jdk/1.1/docs.html

è Question: How do I copy one array to another? Given that I have an byte array defined like this:

```
byte byteSmall = new byte[23];
```

and another larger byte array defined like this:

```
byte byteBig = new byte[30];
```

How do I copy byteSmall into byteBig starting at index 7 without a for loop like this:

```
for(int i = 0; i < 23; i++){
  byteBig[i + 7] = byteSmall;
}
```

Answer: See System.arraycopy:

"Copies an array from the specified source array, beginning at the specified position, to the specified position of the destination array. A subsequence of array components are copied from the source array referenced by src to the destination array referenced by dst. The number of components copied is equal to the length argument. The components at positions srcOffset through srcOffset+length-1 in the source array are copied into positions dstOffset through dstOffset+length-1, respectively, of the destination array.

If the src and dst arguments refer to the same array object, then the copying is performed as if the components at positions srcOffset through srcOffset+length-1 were first copied to a temporary array with length components and then the contents of the temporary array were copied into positions dstOffset through dstOffset+length-1 of the argument array."

è Question: More about Robot! I met with a problem in using class Robot.mousePress...

The compiling process is successful. But when I run it, I receive "IllegalArgumentException:

Invalid combination of button flags". I don't quit understand this information. Part of my code is as following:

```
Robot rMouse=new Robot();
int button=1;
rMouse.mousePress(button);
rMouse.mouseRelease(button);
```

I am really confused. Will you please give me some advice? Thank you in advance!

Answer: You are not using a valid value for the argument to the mousePress() and mouseRelease() methods. If you check the API documentation, you'll find the valid values are a combination of one or more of the following constants:

```
InputEvent.BUTTON1_MASK
InputEvent.BUTTON2_MASK
InputEvent.BUTTON3_MASK
```

plus others which represent the Ctrl, Alt, and Shift keys. To press the left mouse button, you want to use:

```
rMouse.mousePress(InputEvent.BUTTON1_MASK);
```

--
Lee Weiner

è Question: In what situation an exception has to be caught otherwise the compiler will complain?

E.g. IOException does NOT have to be explicitly caught, however, SQLException has to be caught otherwise VisUalAge will not compile the program.

Answer: The only unchecked exceptions in Java are RuntimeException and its subclasses. This includes such familiar classes as NullPointerException, ClassCastException, and IndexOutOfBoundsException.

IOException is not one of these, and *does* have to be explicitly caught or thrown
--
jeff_robertson

è Question: Is it possible to use switch with range of values?
Ex:

```
switch (iBegins){
  case 2 to 10:
}
```

Answer:
not exactly like this but:

```
switch (iBegins){
  case 2:
  case 3:
  case 4:
  case 5:
  case 6:
  case 7:
  case 8:
  case 9:
  case 10:
     /* do stuff */
     break;
  case 11:
     /* other stuff */

}
```

Jan Schaumann http://www.netmeister.org

è Question: Is there a general reason why null pointer exceptions occur?

Answer: Of course there is. A NullPointerException happens when you have a reference, its set to null (or is a class or instance variable and has never been initialized), and you try to perform some operation on the object it points to. Since it really doesn't point to an object, you can't do that.
--
Chris Smith

è Question: I am aware that the exceptions in the catch statements must be of the same type or a super type of the exception thrown in the try block.

My question is this:
Is there any significance in the ordering of the catch statements after a try block?

Answer: It is required to design you catch statement sequences with most derived exception class type first and progressing to the highest superclass. Else, the code will not compile.

For a obvious reason, if the a catch block for a given class precedes a catch block for a type that is derived from the first, the second catch block will never be executed.
--
D. Michael Nelson

è Question: I wrote a program that uses a few RS232 ports. The operators are unskilled and often start multiple instances of the program. Will someone please be so kind and tell me how I can prevent them doing it?

Answer 1: The first instance might write a file. Subsequent instances could check for the existence of that file, or else check its contents.

Another method could involve creating a server socket on a specific port. Subsequent efforts to create a socket on that same port would throw an exception.

Answer 2: Actually a better way is to (on launch):

1) Check if the file exists. If not, create it, open it and run. Leave it open until you quit, upon which time you close it.

2) If the file _does_ exist, try to delete it. If the delete fails, then someone else has it open, which means another copy of your app is running. Inform the user and quit.

3) If you succeeded in deleting it, then you are the first. Now create, open and run.

Doing the above prevents the problem of having the semaphore file left around when the system crashes. I implemented it recently in one of our apps, and it works like a charm.
--
Burt Johnson
MindStorm Productions, Inc.
http://www.mindstorm-inc.com

è Question: Can you call a class static abstract method from an abstract class or does it need to be extended and then called from its concrete class?

I've been told that abstract classes do not actually have any code in them cause they are like a placeholder, so I guess you wouldn't bother calling a static method in an abstract class cause it wouldn't have any code to begin with....?

Answer: You have been told wrong. Abstract classes can and do have code in them.

See, for example, java.awt.Component, an abstract class with a lot of code and no abstract methods at all. If a class has any abstract method member, directly declared or inherited, it is required to be declared abstract. If not, it is the programmer's decision and should be based on whether it would make sense to have an instance of that class.

Perhaps whoever told you was confusing abstract classes with interfaces, which don't contain implementation, only abstract method and constant declarations.

You cannot declare a method to be both static and abstract. Abstract requires the method to be overridden before you can have a concrete class, static prevents overriding. You can have a static method in an abstract class - such a method could be called without creating an instance of the class, the only thing that is prohibited for abstract classes.

And when a subclass of an abstract method has been instantiated, all the methods from the original class will keep the same code in the instance. Most of the time an abstract class will have abstract methods.

However, there are several examples of abstract classes that don't have any abstract methods at all. Some examples are Component and FontMetrics from the AWT. It doesn't make sense to have just a Component that's not a specific type of component. It doesn't make sense to have a FontMetrics that doesn't measure any specific kind of Font.

Also being abstract never prevents overriding, it just requires overriding in order to derive a non-abstract subclass. And if a class is a subclass of an abstract class, it only MUST override those methods declared abstract. The other methods do not require overriding.

è Question: Why can not I compile this simple example?

```
public class Test{
   int a;
   final static int fin;
   final int finint;
}
```

Answer: You can not compile it because you have to initialize final variables. It must be done explicitly.

è Question: I write java about 2 years, but I still confuse one thing that is why should we use interface???

If I need to implement an interface and just use its every methods name. Why

shouldn't just write every methods statements in a class, not in interface?

I only can think about that if I extend a class, than can implement another or the others interface.

As you can saw, I really confuse about this. And I do see many books for the reasons, but I can't get the answer, please tell me!

Answer: "Interface" is the Java way to do multiple inheritances, or a better way to think of it is as a way to design plug-ins.

For example, let's say we have an application that monitors a network of computers. Our monitors might check for web pages, or they may check for other ports, or they may have hooks for hardware checks.

The interface to our main control panel is always the same: We need some means to poll the monitor object for an answer. This is the "NetworkMonitor" interface and all network monitors will share this interface, but they may have a class hierarchy that is very different, for example, port-monitors may all fork a thread that periodically checks whereas our control panel interface just asks for the most recent answer; hardware monitors may ask for their data in real-time or over RPC and thus have no need of inheriting from Thread.

Because they share the same Interface definition, the control panel application does not need to know if they are polling monitors or real-time monitors because, from the control panel's point of view, it does not matter
--
P.S. by John
Also interfaces make our life (I mean programmers) much easier.
Imagine a big project (a lot of programmers, many packages, thousands of files): it is impossible all the time to be aware about particular implementation of some method in specific class in another package!

Much easier to define interfaces between packages as some kind of bridges or gates into another package and their functionality, with hidden (for us) implementation. We just know interface and method names. It is enough to use those methods.

How it is implemented there does no matter... It is working!

è Question: I find such function, but in Class class. Is it possible to use it (or something like this) in another class? Exactly I would like call:

```
super.super.method1();
```

Answer: No, you can only access methods of your direct ancestor. The only way to do this is by using reflection:

```
import java.lang.reflect.*;

Class superduper = this.getClass().
getSuperClass().
getSuperClass();
Method ssMethod = superduper.getMethod("method1",null);
ssMethod.invoke(this,null);
```

(I didn't check the exact syntax and parameters of the above code.)

However the necessity to skip a hierarchy shows that there is something wrong with the design of your application. You should fix this instead of using dirty tricks like the above.

Carl Rosenberger
db4o - database for objects - http://www.db4o.com

è Question: I am facing a problem that might sound common to you all. I have to write a switch statement with 40 case statements. I am wondering if there can be some way to reduce this. If anybody can provide some tips on how to design this smartly. ;)

Answer: The alternatives are:

- an array you index into to get values or delegates.
- an array you binary search.

--
Roedy Green
http://www.mindprod.com or http://209.153.246.39

è Question: Does anyone know what the character limit for a class name would be? I have my class name in a variable ft and I am doing a Class.forName(ft);

My class name with qualifiers is 44 characters long, I can only imagine that there must be a limit ...

Answer: From the JVM Spec 2 - 4.2

Class and interface names...fully qualified name...as
CONSTANT_Utf8_info.

That structure, CONSTANT_Utf8_info, places a limit of 64k on the length. But for a class this is the fully qualified name, so the maximum length of the name could be less.

--
jschell

è Question: I read it on:

http://java.sun.com/docs/books/tutorial/java/javaOO/override.html:

"Also, a subclass cannot override methods that are declared static in the superclass. In other words, a subclass cannot override a class method. A subclass can HIDE a static method in the superclass by declaring a static method in the subclass with the same signature as the static method in the superclass. "

My Question: It looks like play with words: override, hide.. Why is it written HIDE?

If I can write a static method in subclass with the same signature it means that I

override that method in superclass, I really changed that method here in subclass. Am I wrong?

Answer: Let's start from the very beginning - definition for overriding. It includes three main points that overridden method must:

```
* have the same name
* have the same signature
* have the same data type.
```

In your case the static method in subclass can have another data type.
Example:

```
************ Alex.java *************
package alex;
public class Alex {
   static String al(String name){
      return name;
   }
}
******************** EOF ***********

************ John.java *************
package alex;
public class John extends Alex {
   static int al(int name){
      return name;
   }
}
******************** EOF ***********
```

It compiles! You see that in Alex class the method "al" returns String and in John class - int.

It is hiding, not overriding.

Advice: try to avoid such situation when you hide methods in super class. Use different names instead and you will save a lot of time.

è Question: Why C++ is not platform independent?

Answer: C++ compiles to binary code (.obj, .exe, .dll, a.out etc.). Binary code (Machine code, 0's and 1's) are machine dependent.

Java compiles to byte code which is independent of any machine. It need to be interpreted to binary code by JVM, and executed by the machine

è Question: I know that void method does not return any value. But I still write the code like this:

```
void nothing() {};
void nothing2() {return;};
```

Why can we still use "return" in the body of method?

Answer: To be able to exit method when it is necessary.
You can exit the method when you reach the end of method and do not need to use "return". If due to some condition program must exit the method then you use "return".

è Question: I can't find the API documentation on any classes in the sun.* packages. Where is it?

Answer: The short answer is that SUN provides documentation only for the public classes in java.*. SUN does not provide documentation for sun.* because those are the Sun-specific implementation, and specifically not part of the Java technology API standard, and are therefore subject to change without notice.

In general, SUN doesn’t provide javadoc documentation for sun.* classes in order to discourage developers from writing programs that use them. For further explanation, see the next question.

However, if you must have it, the documentation for sun.* is available separately, here: http://java.sun.com/communitysource/index.html
For example, the doc comments for sun.net are in the source files located at:

/src/share/sun/sun/net/*.java

This source code release does not include javadoc-generated documentation. You would have to generate those docs yourself using javadoc.

source: http://java.sun.com/products/jdk/faq.html#A12

è Question: Where can I get source code for Java 2 platform?

Answer: In most cases you do need the code itself. Often it is enough all those tools and resources that are included into standard Java SDK.

Anyway you can always find it here:
http://www.sun.com/software/communitysource/java2

è Question: Does the java class that call the JNI need to be in the same directory of the DLL?

I have the java class in a package and in a jar file. The DLL it uses is outside of the jar file. The JNI call does not work. I am confused with the path.

Answer: As long as the library is in your "path" you should be able to load it with java.lang.System.loadLibrary.

On Linux it should be in your LD_LIBRARY_PATH and on Windows in your PATH.
With java.lang.System.load you can load the library from a predefined location.
--
Flaps

è Question: ActionListener is an interface and we cannot instantiate an interface. Then how is it possible to do something like this (see below)...?

```
textField[0].addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent e){
      handleSearch();
   }
});
```

Answer: That notation is essentially shorthand.
The compiler fills in the equivalent of

```
XXXX extends Object implements ActionListener {
}
```

--
William Brodgen

è Question: By default an application has no security manager and Java runtime environment does not create automatically a security manager for my program. How then applet where I am not creating any security manager already prevented from many operations?

Answer: It is true - Java by default let you do whatever you want and then it is your responsibility to restrict something. In case with applets a little bit different story - applet viewers and browser have THEIR OWN security manager, not from JRE.

That's why even if you did not define any security manager in the code of your applet, during the start up time browser/viewers will use their own security manager. This manager is built in into their application (browser, viewer), not your Java applet.
--
AP (J.A.)

è Question: Does java support ASM?

Answer: You can't directly embed ASM code in your java source. However, you could just use the Java Native Interface to call a C (or whatever else you like) library that embeds the ASM code. This will make you platform-dependent!
--
Sebastian

è Question: I am a Java beginner. I just want to ask what's the difference between

```
int [] number;
```

and

```
int number[];
```

Answer: The "postfix []" syntax is to make the language more comfortable with C programmers. In Java, the "postfix []" binds to the variable name, while the "prefix []" binds to the type name.

In C, there is only the "postfix []". But C also includes pointers, which Java doesn't.

The pointers are identified by having a * between the type name and the variable name.

While there is no danger of ambiguity in this declaration:

```
int a[], b;
```

there IS danger of ambiguity in this declaration:

```
int *a, b;
```

In the latter case, the C rules are that a is declared a pointer, while b is not.

Java offers the following different ways of declaring arrays:

```
int a[], b; <--> int[] a, b;
```

These are not equivalent, because b is an array only on the right side. Whereas with C-style pointers these would become:

```
int *a, b; <--> int* a, b;
```

which breaks the concept of "white space doesn't matter", so it will not be as easy to scan and parse.

So the designers of C decided to have only 1 form of defining variables with pointer types or array types, and this was that the * or the [] binds to the variable, not to the type.

Java carried over the variable binding, but also introduced the type binding, because there were no more pointers, so the ambiguity was removed.
--
Joona Palaste

And think also about marketability!
Making Java's syntax highly similar to C and C++'s during its early years:

* Made the language seem more familiar to C/C++ programmers.
* Decreased the learning curve for C/C++ programmers.
* Gained credibility through the frequent inference that it was a next-generation descendant of C++.
* Increased its pool of programmers by attracting C/C++ programmers through the effects above.

This isn't the only instance of anachronistic syntax; consider the optional semicolon at the end of a class declaration.
--
Andrew R.

è Question: Besides speed, what does the IBM JDK have that sun’s does not? Are there any compatibility problems?

Answer: If there are problems, then they represent bugs in one JDK or the other. To the developer, they should present identical APIs and behavior.

--
Jim Sculley

è Question: I've seen a few examples were for loops are used in different way

```
for(;;) {
}
```

does this do the same as a

```
while(true)
```

is this a speed issue?

Answer: The two are identical. It's simply a matter of style and personal taste.
--
Kevin Riff

è Question: When do I need to use overloading in a program? And why?

Answer: Overloading is best used when you want to have a function that has the same name but can accept different arguments. The only restriction is that it must return the same type.

e.g.

```
public String aName(String str) {
   String retStr;
   retStr = "Hello" + str;
   return retStr;
}
// Now the same function but overloaded
public String aName(String str, String str2){
   String retStr;
   retStr = "Hello " + str + str2;
   return retStr;
}
```

Both functions return the same type (a String) but have different signatures.

You will find this used a lot with constructors. See any recommended text on Java for a much better explanation.
--
Anthony Miller

è Question: Why people claim that java is a secure language...

Answer: The java programming language itself is just as secure or insecure as any other programming language. A Java Applet (which are used on the web) is a different matter. They are quite secure.

An applet does not have access to system resources outside of the directory that it is created in. There are ways that you can give an applet access to this information,

but they are pretty explicit and you will most likely know that the applet is trying to do this.

I am not really familiar with the subject but did a little reading on the sun website. Check it out for more info:

http://developer.java.sun.com/developer/technicalArticles/Security/

You aren't supposed to be able to break the rules easily at run time. The elimination of pointer arithmetic is supposed to improve type safety.

The Code Red worm is a good example of the weakness of the 'old-style' languages. This worm uses a 'buffer overflow' attack to 'trick' the IIS program in to doing it's bidding.

The Java system is designed to make it harder to make the programming mistake that Code Red exploits. It is important that you understand that Code Red would not have been possible without careful attention to a fast buck by Mr. William Gates III and his cohorts.
--
Dr Hackenbush

è Question: This is a basic question... While looking at java API s I noticed that there are classes that start with java and javax (like java.lang.util). What is the basis for this?

Answer: The size of the core part of the Java platform has been growing steadily since the release of version 1.0. The first Java platform had 8 core packages, in version 1.1 there were 22 packages, and in version 1.2 there are over 50! The extensions framework provides a standard means to add functionality to the Java platform for use when needed, without having to increase the size of the core API. Standard extensions have names in the javax.* namespace.

However, just because a package name begins with javax does not guarantee that that package is an extension rather than a part of the core platform. The Swing packages have names beginning with javax indicating that they were non-core extensions prior to version 1.2 of the platform. They are part of the core 1.2 platform even though they are in the javax.* namespace.

For more details please see:
http://java.sun.com/products/jdk/1.2/docs/guide/extensions/ext_faq.html

è Question: Does anyone know if there is a way to prevent System.exit() being called?

Answer: Look into java.lang.SecurityManager. It has a method called checkExit() with which you can prevent System.exit().

This method throws a SecurityException if the calling thread is not allowed to cause the Java Virtual Machine to halt with the specified status code.
--
David Zimmerman

è Question: I want forcibly terminate current running JVM and do not want initiate its shutdown sequence. Just STOP it!

Answer: Use method halt() (Runtime) since: it forcibly terminates the currently running Java virtual machine. This method never returns normally.

This method should be used with extreme caution. Unlike the exit method, this method does not cause shutdown hooks to be started and does not run uninvoked finalizers if finalization-on-exit has been enabled. If the shutdown sequence has already been initiated then this method does not wait for any running shutdown hooks or finalizers to finish their work.

è Question: What is difference between virtual-machine shutdown hook and using the method exit()?

Answer: Sophisticated applications often require the ability to perform specific actions when the virtual machine shuts down, whether due to voluntary exit or involuntary termination. Typical actions include cleanly closing down network connections, saving session state, and deleting temporary files.

Every major operating system provides the means for native applications to do this, though of course the details vary from platform to platform.

Shutdown hook is just one of three steps that executed when you run exit() method.

Terminates the currently running Java virtual machine by initiating its shutdown sequence. This method never returns normally. The argument serves as a status code; by convention, a nonzero status code indicates abnormal termination.

The virtual machine's shutdown sequence consists of three phases:

- In the first phase all registered shutdown hooks, if any, are started in some unspecified order and allowed to run concurrently until they finish.

- In the second phase all uninvoked finalizers are run if finalization-on-exit has been enabled.

- And finally when this is done the virtual machine halts.

è Question: What should I take into consideration when I add my Shutdown hook?

Answer: If you look at the signature of the method:

public void addShutdownHook(Thread hook)

you see that this method uses a thread as an argument. A shutdown hook is simply an initialized but unstarted thread. When the virtual machine begins its shutdown sequence it will start all registered shutdown hooks in some unspecified order and let them run concurrently. When all the hooks have finished it will then run all uninvoked finalizers if finalization-on-exit has been enabled. Finally, the virtual machine will halt. Note that daemon threads will continue to run during the shutdown sequence, as will non-daemon threads if shutdown was initiated by invoking the exit method.

Once the shutdown sequence has begun it can be stopped only by invoking the halt method, which forcibly terminates the virtual machine.

Once the shutdown sequence has begun it is impossible to register a new shutdown hook or de-register a previously-registered hook. Attempting either of these operations will cause an IllegalStateException to be thrown.

Shutdown hooks run at a delicate time in the life cycle of a virtual machine and should therefore be coded defensively. They should, in particular, be written to be thread-safe and to avoid deadlocks insofar as possible. They should also not rely blindly upon services that may have registered their own shutdown hooks and therefore may themselves in the process of shutting down.

Shutdown hooks should also finish their work quickly. When a program invokes exit the expectation is that the virtual machine will promptly shut down and exit. When the virtual machine is terminated due to user logoff or system shutdown the underlying operating system may only allow a fixed amount of time in which to shut down and exit. It is therefore inadvisable to attempt any user interaction or to perform a long-running computation in a shutdown hook.

In rare circumstances the virtual machine may abort, that is, stop running without shutting down cleanly. This occurs when the virtual machine is terminated externally, for example with the SIGKILL signal on UNIX or the TerminateProcess call on Microsoft Windows. The virtual machine may also abort if a native method goes awry by, for example, corrupting internal data structures or attempting to access nonexistent memory. If the virtual machine aborts then no guarantee can be made about whether or not any shutdown hooks will be run.

Based on Java API description

è

8. Java Bugs

9. Java Hardware

This is DEMO version!

10. Java Tools and Software

This is DEMO version!

11. Job, fun

This is DEMO version!

12. Miscellaneous

This is DEMO version!

13. Mobile Java

This is DEMO version!

14. Networking, JSP, Mail... etc.

This is DEMO version!

15. Operation Systems and Java

This is DEMO version!

16. RMI

This is DEMO version!

17. Security

è This is DEMO version!

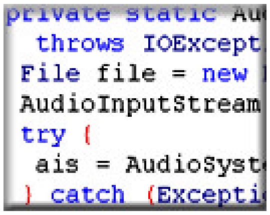
18. Servlets, Servers, Mail etc.

This is DEMO version!

19. Sound, Speech and Multimedia

[This is DEMO version!]

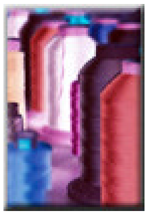
20. String, text, numbers, I/O



```
private static Au
  throws IOExcept
File file = new 
AudioInputStream
try {
  ais = AudioSyst
} catch (Excepti
```

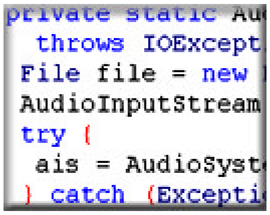
[This is DEMO version!]

21. Threads



[This is DEMO version!]

22. Code Examples



```
private static Au
  throws IOExcept
 File file = new 
 AudioInputStream
 try {
  ais = AudioSyst
 } catch (Excepti
```

This is DEMO version!

23. More Java Tips by Dr. Heinz M. Kabutz

This is DEMO version!

24. XML

This is DEMO version!

This is DEMO version!

è

è

25. About The Author

John Andersson is my pseudonym which I use in my newsletters and on my website JavaFAQ.nu. It will be probably long explanation why I, as well as other people, use pseudonyms. So, I skip it for "clarity" :-)

A few words about me…

I graduated from Moscow State University, physical faculty: computer design. Also I studied nuclear physics and plasma physics extensively during first four years of studying.

After University worked in many telecommunication companies and now I am working at Ericsson AB in Stockholm as programmer in GSM area. During last years I was working mostly with Java and also High Availability UNIX Servers (where we run our SW).

I am married and have two sons. Most time I spend with them and my wife.
Also in my free time I like to “ride” my bike and twice a week I am training in sport hall.

è