RHI Consulting Quick Reference Guide

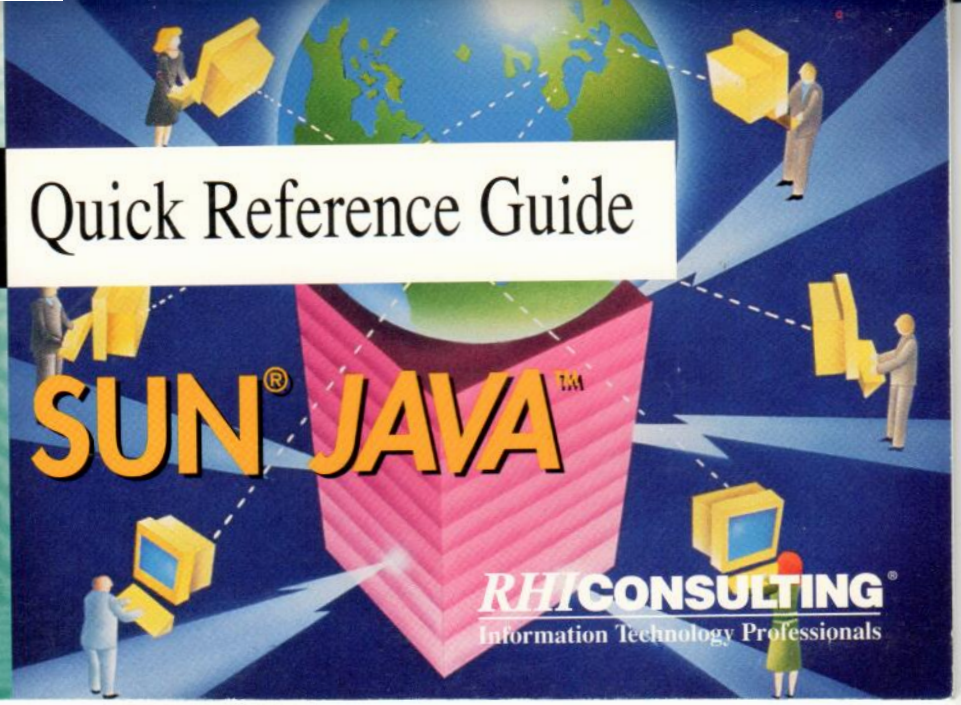


TABLE OF CONTENTS

This quick reference guide has been developed to provide programmers and developers with "at-your-fingertips" information on creating applets and applications with Java from Sun Microsystems, Inc.

Java was developed in response to the growing demand for a more flexible and content-rich environment than could be provided by today's HTML document format of the World Wide Web. It is an object-oriented programming language, very similar to C+, although simpler and with better security. It was developed as a cross-platform system, capable of working with any hardware or software for which products have been created that conform to the Java language specifications published by Sun Microsystems.

Java is currently available in the form of the *Java Development Kit* (see page 2) for Windows NT & Windows 95 (Intel x86), SPARC Solaris (2.3 or later), Intel x86 Solaris, and Macintosh System 7.5. In addition, Java implementations are available or are in the process of being developed for a variety of other platforms.

Java Development Kit

The development tools necessary for creating Java applications and applets can be found in the **Java Development Kit**, available from Sun Microsystem's Web site at **http://www.javasoft.com/**. Other excellent Java programming tools include **Symantec Café** and **Borland Latté** for PCs, and **Symantec Caffeine** for Macintosh. The Java Development Kit includes the Java compiler, interpreter, and several other utilities that are described below.

FILENAME	DESCRIPTION	COMMAND LINE SYNTAX
JAVA.EXE	Interpreter for running standalone Java applications.	`JAVA [options] Java source file` type `JAVA -help` at command line for options
JAVAC.EXE	Compiles Java applets and applications into bytecodes.	`JAVAC [options] filename.java` type `JAVAC -help` at command line for options
JAVAP.EXE	Disassembler program that prints public methods and variables from class files.	`JAVAP [options] class` type `JAVAP -help` at command line for options
JDB.EXE	Debugger used for locating bugs in Java applets and applications.	`JDB` type `JDB -help` at command line for options
JAVAH.EXE	Creates C header files and C stub files which are necessary in order for Java and C code to interact.	`JAVAH [-v][-version][-l filename] classes...` options: `-v` verbose, `-l` variables
JAVADOC.EXE	Java documentation generator. *(See **Comments**, page 5, for more information.)*	`JAVADOC [options] Java source file` type `JAVADOC -help` at command line for options
APPLETVIEWER.EXE	Runs Java applets.	`APPLETVIEWER [-debug] HTML file` options: `-debug` starts the applet viewer in the Java Debugger (JDB)

HTML <APPLET> Tag Attributes

Sample HTML File with an Applet Tag

```
<HTML>
<TITLE> Hello World </TITLE>
</HEAD>
<BODY>
<APPLET CODE="HelloWorld.class"ALT="Hello
World" WIDTH=200 HEIGHT=200 ALIGN=top>
<PARAM NAME=info VALUE="Hello World.">
</APPLET>
<BODY>
```

<APPLET> Tag Elements

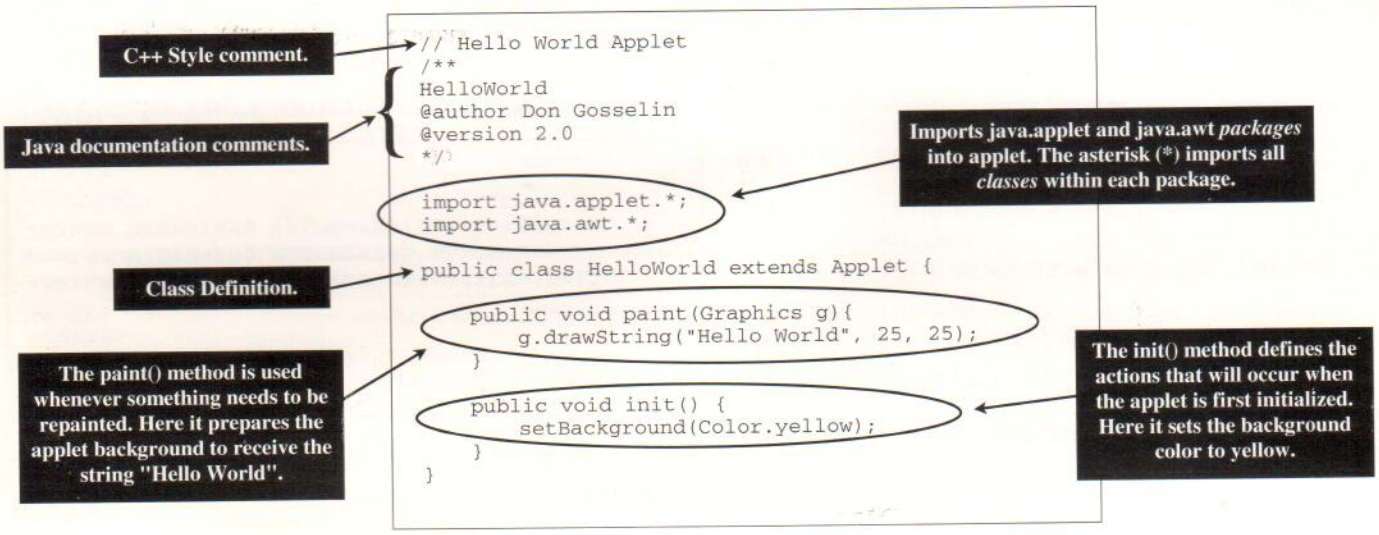
HTML Tag	Name/Description
<APPLET CODE= . . . >	**Begin Applet Tag –** identifies name and attributes of embedded applet.
<PARAM NAME= . . . >	**Parameters Tag –** sends optional information to the applet.
</APPLET >	**Close Applet Tag –** closes the tag.

<APPLET> Tag Attributes

Attribute	Description
ALT=" "	Alternate text to appear in browsers that do not support Java applets.
ALIGN=	Aligns the applet **Left**, **Right**, **Top**, or **Bottom** in relation to the text.
HEIGHT=	Controls the vertical height of the applet.
WIDTH=	Controls the horizontal width of the applet.
BORDER=	Applies a border around the applet.
HSPACE=	Adds extra space to the left or right of an applet in relation to text.
VSPACE=	Adds extra space to the top or bottom of an applet in relation to text.

Sample Java Applet

"Hello World"




```
// Hello World Applet
/**
HelloWorld
@author Don Gosselin
@version 2.0
*/

import java.applet.*;
import java.awt.*;

public class HelloWorld extends Applet {

    public void paint(Graphics g){
        g.drawString("Hello World", 25, 25);
    }

    public void init() {
        setBackground(Color.yellow);
    }
}
```


Comments

Comments are used for leaving remarks in your code. Java supports three types of comment styles: C style, C++ style, and Java Documentation style comments. Java documentation style comments are used by the JAVADOC program that is included in the Java Developer's Kit to automatically generate code documentation. (See **Java Developer's Kit**, page 2, for information on how to compile Java style documentation with JAVADOC.)

Comment Styles

C Style Comments

```
/*
Type comments here
*/
```

C++ Style Comments

```
// Type comments here
```

Java Documentation Style

```
/**
Type comments here
*/
```

JAVADOC Method Documentation Tags

`@param parameter-name description...`
Adds a "parameters section" showing which parameters the method accepts.

`@return description...`
Adds a "returns section" containing the description of the return value.

`@exception fully-qualified-classname description...`
Adds a "throw" section containing the names of the exceptions this type of method can throw.

JAVADOC Class Documentation Tags

`@see classname`
Adds a "See Also" hyperlink entry to the specified class.

`@see fully-qualified-classname`
Adds a "See Also" hyperlink entry to the specified class.

`@see fully-qualified-classname#method-name`
Adds a "See Also" hyperlink entry to the method in the specified class.

`@version version text`
Adds an entry with the specified "version".

`@author your-name`
Adds an "author name" entry (there can be multiple author tags).

Data Types

DATA TYPE	DESCRIPTION	MINIMUM	MAXIMUM
byte	An 8-bit whole number.	-256	255
short	A 16-bit whole number.	-32768	32767
int	A 32-bit whole number.	-2147483648	2147483647
long	A 64-bit whole number.	-9223372036854775808	9223372036854775807
float	A 32-bit floating point number.	1.40129846432481707e-45	3.40282346638528860e+38
double	A 64-bit floating point number.	4.94065645841246544e-324	1.79769313486231570e+308

Special Characters

DESCRIPTION	STANDARD DESIGNATION	ESCAPE SEQUENCE	ASCII NUMBER
Backslash	\	\\	92
Backspace	BS	\b	8
Carriage Return	CR	\r	13
Double Quote	"	\"	34
Form Feed	FF	\f	12
Horizontal Tab	HT	\t	9
New Line	NL or LF	\n	10
Single Quote	'	\'	39

Math & Logical Operators

Assignment

=	Simple assignment
+=	Addition and assignment
-=	Subtraction and assignment
*=	Multiplication and assignment
/=	Division and assignment
%=	Modulus and assignment
&=	Bitwise AND with assignment
|=	Bitwise OR with assignment
^=	Bitwise exclusive OR with assignment
<<=	Shift left with assignment
>>=	Shift right with assignment
>>>=	Shift right with assignment with zero-fill and assignment

Binary

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
<<	Shift left
>>	Shift right
>>>	Shift right with zero-fill

Boolean

!	Negation
&	Logical AND
|	Logical OR
^	Logical XOR
&&	Evaluation AND
||	Evaluation OR
==	Equal to
!=	Not Equal To
&=	And assignment
|=	OR assignment
^=	XOR assignment
?:	Ternary - conditional

Unary

++	Increment
--	Decrement
!	Boolean negation (NOT)
-	Numeric negation
~	Bitwise compliment

Loops and Conditional Statements

`if` Statement

```
if (expression) {statements}
```

`if...else` Statement

```
if (expression) {statements}
else {statements}
```

`while` Statement

```
while (expression) {statements}
```

`do...while` Statement

```
do {
      statements
} while (expression);
```

`for` Statement

```
for (initialization; termination; increment)
      {statements}
```

`switch` Statement

```
switch (expression) {
      case expression1: statements; break;
      case expression2: statements; break;
      case expression3: statements; break;
      ...
      default: statements; break
}
```

`label` Statement

```
label: {statements}
      break label;
      continue label;
```


Event Handling

Event handling occurs when a Java applet or application responds to an *event* such as a key press or mouse movement. Events in Java programming can be classified into three main types: *system*, *mouse*, and *keyboard*. Most common events are handled by the **java.awt** package (see page 19 for more information).

VARIABLE	DESCRIPTION
ACTION_EVENT	Indicates that the user wants some action to occur.
GOT_FOCUS	A component gains focus.
KEY_ACTION	An "action" key is pressed.
KEY_ACTION_RELEASE	An "action" key is released.
KEY_PRESS	A normal key is pressed.
KEY_RELEASE	A normal key is released
LIST_DESELECT	A list item is deselected
LIST_SELECT	A list item is selected
LOAD_FILE	A file is loaded
LOST_FOCUS	A component loses focus.
MOUSE_DOWN	The mouse button pressed
MOUSE_DRAG	The mouse is moved with the mouse button pressed.
MOUSE_ENTER	The mouse enters a component.
MOUSE_EXIT	The mouse leaves component.
MOUSE_MOVE	The mouse moves.
MOUSE_UP	The mouse button is released.
SAVE_FILE	A file saving event.

Event Handling (continued)

VARIABLE	DESCRIPTION
SCROLL_ABSOLUTE	The bubble is moved in a scroll bar.
SCROLL_LINE_DOWN	The scroll bar moves down a line.
SCROLL_LINE_UP	The scroll bar moves up a line.
SCROLL_PAGE_DOWN	The scroll bar moves down a page.
SCROLL_PAGE_UP	The scroll bar moves up a page.
WINDOW_DEICONIFY	The window is restored from a minimized state.
WINDOW_DESTROY	The window is closed.
WINDOW_EXPOSE	A windows becomes exposed.
WINDOW_ICONIFY	The window is minimized.
WINDOW_MOVED	The window moves.
DOWN	The Down key is pressed.
END	The End key is pressed.
F1 … F12	A function key (F1 through F12) is pressed.
HOME	The home key is pressed.
LEFT	The left arrow key is pressed.
PGDN	The page down key is pressed.
PGUP	The page up key is pressed.
RIGHT	The right arrow key is pressed.

Event Handling (continued)

VARIABLE	DESCRIPTION
UP	The up arrow key is pressed.
ALT_MASK	The "alt" key was down when the event occurred or the middle mouse button was pressed or released.
CTRL_MASK	The control key was down when the event occurred.
SHIFT_MASK	The shift key was down when the event occurred.
META_MASK	The meta key was down when the event occurred or the right button was pressed or released.
Arg	Arbitrary argument - depends on the type of event.
clickCount	Indicates number of consecutive clicks for MOUSE_DOWN events.
Evt	Next event.
id	Type of event.
key	Key that was pressed in a keyboard event.
modifiers	State of the modifier keys.
target	Target component.
when	Time stamp of the event.
x	X coordinate of the event.
y	Y coordinate of the event.

Packages

Packages contain groups of Java classes and interfaces. Below are the packages supplied with the **Java Developer's Kit** (see page 2) along with a description of their contents:

java.lang	Contains essential Java classes, including numerics, strings, objects, compiler, runtime, security, and threads. This package is automatically imported into every program and applet.
java.io	Provides classes to manage input and output streams to read data from and write data to files, strings, and others.
java.util	Contains various utility classes, including string manipulation, random number generation, system properties, generic data structures, bit sets, time, date, notification, and enumeration of data structures.
java.net	Provides classes for network support, including URLs, TCP sockets, UDP sockets, IP addresses, and a binary-to-text converter.
java.awt	Abstract Window Toolkit. Contains components to manage user interface such as windows, dialog boxes, buttons, checkboxes, lists, menus, scrollbars, and text fields.
java.awt.image	Contains classes for managing image data, including color models, cropping, color filtering, setting pixel values, and grabbing snapshots.
java.awt.peer	Connects AWT components to their platform-specific implementations (such as Microsoft Windows controls).
java.applet	Enables the creation of applets through the Applet class and includes several interfaces that connect an applet to its document and to resources for playing audio.

Classes

Classes are components contained in Java *packages* (see previous page) that are also found in C++. They are structures that are used to describe program code known as *objects*, along with associated methods and data. Classes can be used as they are or extended and modified into user-defined classes.

Package java.awt Classes

NAME	DESCRIPTION
TextComponent	The superclass of any component that allows the editing of some text.
TextField	A component that presents the user with a single editable line of text.
Toolkit	The abstract superclass of all actual implementations of the Abstract Window Toolkit.
Window	A Window is a top-level window that has no borders or menubar.

Package java.applet Classes

NAME	DESCRIPTION
Applet	The superclass of any applet that is to be embedded in a Web page or viewed by the Java Applet Viewer.

Package java.lang Classes

NAME	DESCRIPTION
Boolean	Wraps a value of the primitive type boolean in an object and provides a number of methods for converting a boolean to a String and a String to a boolean as well as other constants and methods useful when dealing with a boolean.
Character	Wraps a value of the primitive type char in an object and provides a number of methods for determining the type of a character and converting characters from uppercase to lowercase and vice versa.
Class	Represent classes and interfaces in a running Java application.
ClassLoader	Subclasses are implemented in order to extend the manner in which the Java Virtual Machine dynamically loads classes.
Compiler	Provided in support of Java-to-native-code compilers and related services.
Double	Wraps a value of the primitive type double in an object and provides a number of methods for converting a double to a String and a String to a double as well as other constants and methods.
Float	Wraps a value of primitive type float in an object and provides a number of methods for converting a float to a String and a String to a float, as well as other constants and methods useful when dealing with a float.
Integer	Wraps a value of the primitive type int in an object and provides a number of methods for converting an int to a String and a String to an int, as well as other constants and methods useful when dealing with an int.
Long	Wraps a value of the primitive type long in an object and provides a number of methods for converting a long to a String and a String to a long, as well as other constants and methods useful when dealing with a long.

Package java.lang Classes (continued)

NAME	DESCRIPTION
Math	Contains methods for basic numerical operations such as the elementary exponential, logarithm, square root, and trigononetric functions.
Number	The superclass of classes Float, Double, Integer, and Long.
Object	The root of the class hierarchy. Every class has Object as a superclass parent. All objects implement the methods of this class.
Process	Return an instance of a subclass of Process used to control the process and obtain information about it.
Runtime	Allows the application to interface with the environment in which the application is running.
SecurityManager	Allows applications to implement a security policy.
String	Represents character strings.
StringBuffer	Implements a mutable sequence of characters.
System	Contains various class fields and methods including standard input, output, and error output streams, access to externally defined "properties", a means of loading files and libraries, and a utility method for quickly copying a portion of an array.
Thread	The Java Virtual Machine allows an application to have multiple threads of executing running concurrently.
ThreadGroup	Constructs a new thread group, whose parent is the thread group of the currently running thread.
Throwable	Superclass of all errors and exceptions in the Java language.

Package java.io Classes

NAME	DESCRIPTION
BufferedInputStream	Implements a buffered input stream.
BufferedOutputStream	Implements a buffered output stream.
ByteArrayInputStream	Allows an application to create an input stream in which the bytes read are supplied by the contents of a byte array.
ByteArrayOutputStream	Implements an output stream in which the data is written into a byte array.
DataInputStream	Allows an application to read primitive Java data types from an underlying input stream in a machine-independent way.
DataOutputStream	Allows an application to write primitive Java data types from an output stream in a portable way.
File	Represent the name of a file or directory on the host file system.
FileDescriptor	Serves as an opaque handle to the underlying machine-specific structure representing an open file or an open socket.
FileInputStream	Used for reading data from a File or FileDescriptor.
FileOutputStream	Used for writing data to a File or FileDescriptor.
FilterInputStream	The superclass of all classes that filter input streams.
FilterOutputStream	The superclass of all classes that filter output streams.
InputStream	The superclass of all classes representing an input stream of bytes.

Package java.io Classes (continued)

NAME	DESCRIPTION
LineNumberInputStream	Provides the added functionality of keeping track of the current line number.
OutputStream	The superclass of all classes representing an output stream of bytes.
PipedInputStream	The receiving end a communications pipe.
PipedOutputStream	The sending end a communications pipe.
PrintStream	Provides convenient methods for printing types other than bytes and arrays of bytes.
PushbackInputStream	Provides a one-byte push back buffer.
RandomAccessFile	Supports both reading and writing to a random access file.
SequenceInputStream	Combine several input streams serially and make them appear as if they were a single input stream.
StreamTokenizer	Allows "tokens" to be read one at a time from an input stream.
StringBufferInputStream	Creates an input stream in which the bytes read are supplied by the contents of a string.

Package java.net Classes

NAME	DESCRIPTION
ContentHandler	The superclass of all classes that read an Object from a URLConnection.
DatagramPacket	Used to implement a connectionless packet delivery service. Represents a datagram packet.
DatagramSocket	Represents a socket for sending and receiving datagram packets.
InetAddress	Represents an Internet Protocol (IP) address.
ServerSocket	Implements server sockets.
Socket	Implements client sockets.
SocketImpl	Used to create both client and server sockets.
URL	Represents a Uniform Resource Locator-a pointer to a "resource" on the World Wide Web.
URLConnection	Represents a communications link between the application and a URL.
URLEncoder	Converting a String into a MIME format called "x- www-form-urlencoded" format.
URLStreamHandler	The common superclass for all stream protocol handlers.

Package java.awt Classes

NAME	DESCRIPTION
BorderLayout	Lays out a container using members named "North", "South", "East", "West", and "Center".
Button	Creates a labeled button.
Canvas	Represents a blank rectangular area of the screen onto which the application can draw or from which the application can trap input events from the user.
CardLayout	A layout manager for a container that contains several "cards".
Checkbox	A graphical component that has true and false states.
CheckboxGroup	Groups together a set of Checkbox buttons.
CheckboxMenuItem	Represents a check box that can be included in a menu.
Choice	Presents a pop-up menu of choices.
Color	Uses the RGB format to encapsulate colors.
Component	The abstract superclass of many of the Abstract Window Toolkit classes.
Container	Represents all components that can hold other components.
Dialog	Represents a window that takes input from the user (a dialog window).

Package java.awt Classes (continued)

NAME	DESCRIPTION
Dimension	Encapsulates the width and height of a component in a single object.
Event	Encapsulates user events from the local Graphical User Interface (GUI) platform.
FileDialog	Allows the user can select a file from a dialog window.
FlowLayout	Arranges components in a left-to-right flow.
Font	Represents a font.
FontMetrics	Represents a font metrics object.
Frame	A top-level window with a title and a border - can also have a menu bar.
Graphics	Base class for all graphics contexts which allow an application to draw onto components or onto off-screen images.
GridBagConstraints	Specifies constraints for components that are laid out using the GridBagLayout class.
GridBagLayout	Aligns components horizontally and vertically, without requiring that the components be the same size.
GridLayout	Arranges components in a rectangular grid.
Image	Represents all classes that display graphical images.
Insets	Represents the borders of a container.

Package java.awt Classes (continued)

NAME	DESCRIPTION
Label	Used for placing text in a container.
List	Creates a scrolling list of text items.
MediaTracker	Traces the status of a number of media objects.
Menu	Item of a menu bar.
MenuBar	Contains menu items.
MenuComponent	The superclass of all menu related components.
MenuItem	Items in a menu belong to either MenuItem class or a subclass.
Panel	A simple container class.
Point	Represents an (x, y) coordinate.
Polygon	Defines the sides of the polygon with successive pairs of (x,y) coordinates.
Rectangle	Defines a rectangle.
Scrollbar	Defines a scroll bar.
TextArea	Displays multiline text areas.

Package java.awt.image Classes

NAME	DESCRIPTION
ColorModel	Encapsulate methods for translating from pixel values to their alpha (transparency), red, green, and blue components.
CropImageFilter	An image filter for cropping images.
DirectColorModel	Specifies a translation from pixel values to alpha, red, green, and blue components using the actual bits of the pixel value.
FilteredImageSource	Implements the image producer interface which takes an existing image and an image filter to produce a new filtered version of the original image.
ImageFilter	Filter the data delivered from an ImageProducer to an ImageConsumer.
IndexColorModel	Specifies a color model in a which a pixel value is converted into alpha, red, green, and blue component by using the pixel value as an index into a color map.
MemoryImageSource	An implementation of the image producer interface.
PixelGrabber	An image consumer which can be attached to an image or image producer object to retrieve a subset of the pixels in that image.
RGBImageFilter	Create an image filter which modifies the pixels of the original image by converting them one at a time in the default RGB color model.

Package java.util Classes

NAME	DESCRIPTION
BitSet	Implements a vector of bits that grows as needed.
Date	Provides anabstraction of dates and times.
Dictionary	The abstract parent of any class, such as Hashtable, which maps keys to values.
Hashtable	Implements a hash table, which maps keys to values.
Observable	Represents an observable object, or data in the model-view paradigm.
Properties	Represents a persistent set of properties.
Random	Used to generate a stream of pseudo-random numbers.
Stack	Represents a last-in-first-out (LIFO) stack of objects.
StringTokenizer	Allows an application to break a string into tokens.
Vector	Implements a growable array of objects.

Interfaces

An interface is a form of a template that declares a collection of methods and variables that will be implemented at a later time. These protocols are then used at various stages in the application development to force you to conform to the interface structure.

Package java.lang Interfaces

NAME	DESCRIPTION
Cloneable	A class implements the Cloneable interface to indicate to the clone method in class Object that it is legal for that method to make a field-for-field copy of instances of that class.
Runnable	Should be implemented by any class whose instances are intended to be executed by a thread.

Package java.io Interfaces

NAME	DESCRIPTION
DataInput	Implemented by streams that can read primitive Java data types from a stream in a machine-independent manner.
DataOutput	Implemented by streams that can write primitive Java data types to an output stream in a machine-independent manner.
FilenameFilter	Instances of classes that implement this interface are used to filter filenames.

Package java.util Interfaces

NAME	DESCRIPTION
Enumeration	An object that implements the Enumeration interface generates a series of elements, one at a time.
Observer	A class can implement the Observer interface when it wants to be informed if observable objects changes.

Package java.net Interfaces

NAME	DESCRIPTION
ContentHandlerFactory	Defines a factory for content handlers. An implemention of this interface should map a MIME type into an instance of ContentHandler.
SocketImplFactory	Defines a factory for socket implementations. It is used by the classes Socket and ServerSocket to create actual socket implementions.
URLStreamHandlerFactory	Defines a factory for URL stream protocol handlers. It is used by the URL class to create a URLStreamHandler for a specific protocol.

Package java.awt Interfaces

NAME	DESCRIPTION
LayoutManager	Specifies the methods that all layout managers must implement. A layout manager is a class for laying out the components of a Container.
MenuContainer	Specifies the methods that all menu-related containers must implement. Note that menu containers are not required to be full-fleged Container objects.

Package java.awt.image Interfaces

NAME	DESCRIPTION
ImageConsumer	Specifies the methods that all image consumers must implement. An image consumer is an object interested in data produced by the image producers.
ImageObserver	Specifies the methods that all image observers must implement. An image observer is interested in receiving asynchronous notifications about the image as the image is being constructed.
ImageProducer	Specifies the methods that all image producers must implement. Every image contains an image producer which can reconstruct the image whenever it is needed by an image consumer.

Package java.awt.peer Interfaces

NAME	DESCRIPTION
ButtonPeer	Specifies the methods that all implementations of Abstract Window Toolkit buttons must define.
CanvasPeer	Specifies the methods that all implementations of Abstract Window Toolkit canvases must define.
CheckboxMenuItemPeer	Specifies the methods that all implementations of Abstract Window Toolkit check box menus must define.
CheckboxPeer	Specifies the methods that all implementations of Abstract Window Toolkit check boxes must define.
ChoicePeer	Specifies the methods that all implementations of Abstract Window Toolkit choice menus must define.
ComponentPeer	Specifies the methods that all implementations of Abstract Window Toolkit components must define.
ContainerPeer	Specifies the methods that all implementations of Abstract Window Toolkit containers must define.
DialogPeer	Specifies the methods that all implementations of Abstract Window Toolkit dialog windows must define.
FileDialogPeer	Specifies the methods that all implementations of Abstract Window Toolkit file dialog windows must define.
FramePeer	Specifies the methods that all implementations of Abstract Window Toolkit frames must define.
LabelPeer	Specifies the methods that all implementations of Abstract Window Toolkit label must define.

Package java.awt.peer Interfaces (continued)

NAME	DESCRIPTION
ListPeer	Specifies the methods that all implementations of Abstract Window Toolkit scrolling lists must define.
MenuBarPeer	Specifies the methods that all implementations of Abstract Window Toolkit menu bars must define.
MenuComponentPeer	Specifies the methods that all implementations of Abstract Window Toolkit menu components must define.
MenuItemPeer	Specifies the methods that all implementations of Abstract Window Toolkit menu items must define.
MenuPeer	Specifies the methods that all implementations of Abstract Window Toolkit menus must define.
PanelPeer	Specifies the methods that all implementations of Abstract Window Toolkit panels must define.
ScrollbarPeer	Specifies the methods that all implementations of Abstract Window Toolkit scroll bars must define.
TextAreaPeer	Specifies the methods that all implementations of Abstract Window Toolkit text areas must define.
TextComponentPeer	Specifies the methods that all implementations of Abstract Window Toolkit text components must define.
TextFieldPeer	Specifies the methods that all implementations of Abstract Window Toolkit text fields must define.
WindowPeer	Specifies the methods that all implementations of Abstract Window Toolkit windows must define.

Package java.applet Interfaces

NAME	DESCRIPTION
AppletContext	Corresponds to the document containing the applet and the other applets in the same document.
AppletStub	Serves as the interface between the applet and the browser environment or applet viewer environment in which the application is running.
AudioClip	Plays sound clips. Sounds can be mixed together to produce a composite by playing multiple items at the same time.

Online Java Resources

Name	URL
AltaVista Java Searching	http://altavista.digital.com
Black Star's Link	http://www.blackstar.com
Café Del Sol	http://www.xm.com/café
Coffe Pages	http://answer.questions.com/date/960101/date.html
Dimension X	http://www.dimensionx.com/dnx/java.html
Diva	http://www.inch.com/~friskel/rant.htm
Gamelan	http://www.gamelan.com
Java Applet Rating Service	http://www.jars.com
Java Boutique	http://weber.u.washington.edu/~jgurney/java/
Java Newsgroup	comp.lang.java
Java Report	http://www.sigs.com
Java World	http://www.javaworld.com
Javamaker	http://net.info.samsung.co.kr/~hcchoi/javmaker.html
JavaSoft	http://www.javasoft.com
JavaWorld	http://www.xm.com/café/
Kalimantan Compiler	http://www.dstc.edu.au/projects/kalimantan/
Netscape's Java Applet Page	http://home.mcom.com/comprod/products/navigator/version_2.0/java_applets/index.html.
Roaster Home Page	http://www.natural.com/pages/products/roaster/flyer.html
Sun Users Groups	http://www.sug.org/java-sig.html
Symantec's Java Central	http://café.symantec.com

NOTES:

NOTES:

NOTES:

NOTES:

RHI Consulting is...

... The world's leader in information technology staffing. Finding the right information technology support for your organization can be difficult. Experience, personality, technical skills, and cost must be weighed and factored into your decision.

Through ProSCAN℠, our specialized evaluation process, *RHI Consulting* guarantees that our consultants possess the knowledge and skills necessary to get the job done right. And, our INFINITY℠ advanced technical training program assures that our consultants maintain leading-edge skills for the IT industry.

Call today to be connected with your nearest *RHI Consulting* location or contact us online for more information.

RHICONSULTING®

Information Technology Professionals

800-793-5533 U.S. • 800-268-1948 CANADA

www.rhic.com

RHIC-996-3502