Glossary of Terms

For simplicity, "the Java™ platform" is called Java in the following definitions.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

100% Pure Java(TM)
A Sun Microsystems initiative to guide developers in writing, certifying, and marketing applications written entirely in the Java(TM) programming language.

A

Abstract Window Toolkit (AWT)
A collection of graphical user interface (GUI) components that were implemented using native-platform versions of the components. These components provide that subset of functionality which is common to all native platforms. Largely supplanted by the Project Swing component set. See also *Swing Set*.

abstract
A Java(TM) programming language keyword used in a class definition to specify that a class is not to be instantiated, but rather inherited by other classes. An abstract class can have abstract methods that are not implemented in the abstract class, but in subclasses.

abstract class
A class that contains one or more *abstract methods*, and therefore can never be instantiated. Abstract classes are defined so that other classes can extend them and make them concrete by implementing the abstract methods.

abstract method
A method that has no implementation.

actual parameter list
The arguments specified in a particular method call. See also *formal parameter list*.

alpha value
A value that indicates the opacity of a pixel.

API
Application Programming Interface. The specification of how a programmer writing an application accesses the behavior and state of classes and objects.

appliances
Networked devices such as printers, Java(TM) technology-enabled terminals, and clients, that are managed using applications built using the Java Management API (JMAPI).

applet
A program written in the Java(TM) programming language to run within a web browser compatible with the Java platform, such as HotJava(TM) or Netscape Navigator(TM).

argument
A data item specified in a method call. An argument can be a literal value, a variable, or an expression.

array
A collection of data items, all of the same type, in which each item's position is uniquely designated by an integer.

ASCII

American Standard Code for Information Interchange. A standard assignment of 7-bit numeric codes to characters. See also Unicode.

atomic

Refers to an operation that is never interrupted or left in an incomplete state under any circumstance.

B

Bean

A reusable software component. Beans can be combined to create an application.

binary operator

An operator that has two arguments.

bit

The smallest unit of information in a computer, with a value of either 0 or 1.

bitwise operator

An operator that manipulates two values comparing each bit of one value to the corresponding bit of the other value.

block

In the Java(TM) programming language, any code between matching braces. Example: `{ x = 1; }`.

boolean

Refers to an expression or variable that can have only a true or false value. The Java(TM) programming language provides the boolean type and the literal values true and false.

bounding box

For a Raster object, the smallest rectangle that completely encloses all the pixels that are not fully transparent.

break

A Java(TM) programming language keyword used to resume program execution at the statement immediately following the current statement. If followed by a label, the program resumes execution at the labeled statement.

byte

A sequence of eight bits. The Java(TM) programming language provides a corresponding `byte` type.

bytecode

Machine-independent code generated by the Java(TM) compiler and executed by the Java interpreter.

C

case

A Java(TM) programming language keyword that defines a group of statements to begin executing if a value specified matches the value defined by a preceding "switch" keyword.

casting

Explicit conversion from one data type to another.

catch

A Java(TM) programming language keyword used to declare a block of statements to be executed in the event that a Java exception, or run time error, occurs in a preceding "try" block.

char

A Java(TM) programming language keyword used to declare a variable of type character.

class

In the Java(TM) programming language, a type that defines the implementation of a particular kind of object. A class definition defines instance and class variables and methods, as well as specifying the interfaces the class implements and the immediate superclass of the class. If the superclass is not explicitly specified, the superclass will implicitly be `Object`.

class method

A method that is invoked without reference to a particular object. Class methods affect the class as a whole, not a particular instance of the class. Also called a *static method*. See also *instance method*.

classpath

A classpath is an environmental variable which tells the Java(TM) virtual machine* and Java technology-based applications (for example, the tools located in the JDK(TM) 1.1.X\bin directory) where to find the class libraries, including user-defined class libraries.

class variable

A data item associated with a particular class as a whole--not with particular instances of the class. Class variables are defined in class definitions. Also called a *static field*. See also *instance variable*.

client

In the client/server model of communcations, the client is a process that remotely accesses resources of a compute server, such as compute power and large memory capacity.

codebase

Works together with the `code` attribute in the `<APPLET>` tag to give a complete specification of where to find the main applet class file: code specifies the name of the file, and codebase specifies the URL of the directory containing the file.

comment

In a program, explanatory text that is ignored by the compiler. In programs written in the Java(TM) programming language, comments are delimited using `//` or `/*`...`*/`.

compilation unit

The smallest unit of source code that can be compiled. In the current implementation of the Java(TM) platform, the compilation unit is a file.

compiler

A program to translate source code into code to be executed by a computer. The Java(TM) compiler translates source code written in the Java programming language into bytecode for the Java virtual machine*. See also *interpreter*.

compositing

The process of superimposing one image on another to create a single image.

constructor

A pseudo-method that creates an object. In the Java(TM) programming language, constructors are instance methods with the same name as their class. Constructors are invoked using the `new` keyword.

const

This is a reserved Java(TM) programming language keyword. However, it is not used by current versions of the Java programming language.

continue

A Java(TM) programming language keyword used to resume program execution at the end of the current loop. If followed by a label, "continue" resumes execution where the label occurs.

core class

A public class (or interface) that is a standard member of the Java(TM) Platform. The intent is that the core classes for the Java platform, at minimum, are available on all operating systems where the Java platform runs. A program written entirely in the Java programming language relies only on core classes, meaning it can run anywhere. See also, *100% Pure Java(TM)*.

Core Packages
The required set of APIs in a Java platform edition which must be supported in any and all compatible implementations.

critical section
A segment of code in which a thread uses resources (such as certain instance variables) that can be used by other threads, but that must not be used by them at the same time.

D

declaration
A statement that establishes an identifier and associates attributes with it, without necessarily reserving its storage (for data) or providing the implementation (for methods). See also *definition*.

default
A Java(TM) programming language keyword optionally used after all "case" conditions in a "switch" statement. If all "case" conditions are not matched by the value of the "switch" variable, the "default" keyword will be executed.

definition
A declaration that reserves storage (for data) or provides implementation (for methods). See also *declaration*.

deprecation
Refers to a class, interface, constructor, method or field that is no longer recommended, and may cease to exist in a future version.

derived from
Class X is "derived from" class Y if class X *extends* class Y. See also subclass, superclass.

distributed
Running in more than one address space.

do
A Java(TM) programming language keyword used to declare a loop that will iterate a block of statements. The loop`s exit condition can be specified with the "while" keyword.

double
A Java(TM) programming language keyword used to define a variable of type double.

double precision
In the Java(TM) programming language specification, describes a floating point number that holds 64 bits of data. See also single precision.

E

else
A Java(TM) programming language keyword used to execute a block of statements in the case that the test condition with the "if" keyword evaluates to false.

EmbeddedJava(TM) Technology
The availability of Sun's Java 2 Platform, Micro Edition technology under a restrictive license agreement that allows a licensee to leverage certain Java technologies to create and deploy a closed-box application that exposes no APIs.

encapsulation

The localization of knowledge within a module. Because objects encapsulate data and implementation, the user of an object can view the object as a black box that provides services. Instance variables and methods can be added, deleted, or changed, but as long as the services provided by the object remain the same, code that uses the object can continue to use it without being rewritten. See also *instance variable*, *instance method*.

exception

An event during program execution that prevents the program from continuing normally; generally, an error. The Java(TM) programming language supports exceptions with the try, catch, and throw keywords. See also *exception handler*.

exception handler

A block of code that reacts to a specific type of *exception*. If the exception is for an error that the program can recover from, the program can resume executing after the exception handler has executed.

executable content

An application that runs from within an HTML file. See also *applet*.

extends

Class X extends class Y to add functionality, either by adding fields or methods to class Y, or by overriding methods of class Y. An interface extends another interface by adding methods. Class X is said to be a subclass of class Y. See also *derived from*.

F

FCS

First Customer Ship. The day in which a product is released/shipped to the customer.

field

A data member of a class. Unless specified otherwise, a field is not static.

final

A Java(TM) programming language keyword. You define an entity once and cannot change it or derive from it later. More specifically: a final class cannot be subclassed, a final method cannot be overridden and a final variable cannot change from its initialized value.

finally

A Java(TM) programming language keyword that executes a block of statements regardless of whether a Java Exception, or run time error, occurred in a block defined previously by the "try" keyword.

float

A Java(TM) programming language keyword used to define a floating point number variable.

for

A Java(TM) programming language keyword used to declare a loop that reiterates statements. The programmer can specify the statements to be executed, exit conditions, and initialization variables for the loop.

FTP

The basic Internet File Transfer Protocol. FTP, which is based on TCP/IP, enables the fetching and storing of files between hosts on the Internet. See also *TCP/IP*.

formal parameter list

The parameters specified in the definition of a particular method. See also *actual parameter list*.

G

garbage collection

The automatic detection and freeing of memory that is no longer in use. The Java(TM) runtime system performs garbage collection so that programmers never explicitly free objects.

goto

This is a reserved Java(TM) programming language keyword. However, it is not used by current versions of the Java programming language.

GUI

Graphical User Interface. Refers to the techniques involved in using graphics, along with a keyboard and a mouse, to provide an easy-to-use interface to some program.

H

hexadecimal

The numbering system that uses 16 as its base. The marks 0-9 and a-f (or equivalently A-F) represent the digits 0 through 15. In programs written in the Java(TM) programming language, hexadecimal numbers must be preceded with 0x. See also *octal*.

hierarchy

A classification of relationships in which each item except the top one (known as the root) is a specialized form of the item above it. Each item can have one or more items below it in the hierarchy. In the Java(TM) class hierarchy, the root is the `Object` class.

HotJava(TM) Browser

An easily customizable Web browser developed by Sun Microsystems, which is written in the Java(TM) programming language.

HTML

HyperText Markup Language. This is a file format, based on SGML, for hypertext documents on the Internet. It is very simple and allows for the embedding of images, sounds, video streams, form fields and simple text formatting. References to other objects are embedded using URLs.

HTTP

HyperText Transfer Protocol. The Internet protocol, based on TCP/IP, used to fetch hypertext objects from remote hosts. See also *TCP/IP*.

I

IDL

Interface Definition Language. APIs written in the Java(TM) programming language that provide standards-based interoperability and connectivity with CORBA (Common Object Request Broker Architecture).

identifier

The name of an item in a program written in the Java(TM) programming language.

if

A Java(TM) programming language keyword used to conduct a conditional test and execute a block of statements if the test evaluates to true.

implements

A Java(TM) programming language keyword optionally included in the class declaration to specify any interfaces that are implemented by the current class.

import

A Java(TM) programming language keyword used at the beginning of a source file that can specify classes or entire packages to be referred to later without including their package names in the reference.

inheritance

The concept of classes automatically containing the variables and methods defined in their *supertypes*. See also *superclass*, *subclass*.

instance

An object of a particular class. In programs written in the Java(TM) programming language, an instance of a class is created using the `new` operator followed by the class name.

instance method

Any method that is invoked with respect to an instance of a class. Also called simply a *method*. See also *class method*.

instance variable

Any item of data that is associated with a particular object. Each instance of a class has its own copy of the instance variables defined in the class. Also called a *field*. See also *class variable*.

instanceof

A two-argument Java(TM) programming language keyword that tests whether the run-time type of its first argument is assignment compatible with its second argument.

int

A Java(TM) programming language keyword used to define a variable of type integer.

interface

A Java(TM) programming language keyword used to define a collection of method definitions and constant values. It can later be implemented by classes that define this interface with the "implements" keyword.

Internet

An enormous network consisting of literally millions of hosts from many organizations and countries around the world. It is physically put together from many smaller networks and data travels by a common set of protocols.

IP

Internet Protocol. The basic protocol of the Internet. It enables the unreliable delivery of individual packets from one host to another. It makes no guarantees about whether or not the packet will be delivered, how long it will take, or if multiple packets will arrive in the order they were sent. Protocols built on top of this add the notions of connection and reliability. See also *TCP/IP*.

interpreter

A module that alternately decodes and executes every statement in some body of code. The Java(TM) interpreter decodes and executes bytecode for the Java virtual machine*. See also *compiler*, *runtime system*.

J

JAE

Java(TM) Application Environment. The source code release of the Java Development Kit (JDK(TM)) software.

JAR Files (.jar)

Java ARchive. A file format used for aggregating many files into one.

JAR file format

JAR (Java Archive) is a platform-independent file format that aggregates many files into one. Multiple applets written in the Java(TM) programming language, and their requisite components (.class files, images, sounds and other resource files) can be bundled in a JAR file and subsequently downloaded to a browser in a single HTTP transaction. It also supports file compression and digital signatures.

Java(TM)
Sun's trademark for a set of technologies for creating and safely running software programs in both stand-alone and networked environments.

Java Application Environment (JAE)
The source code release of the Java Development Kit (JDK(TM)) software.

JavaBeans(TM)
A portable, platform-independent reusable component model.

Java Blend(TM)
A product that enables developers to simplify database application development by mapping database records to objects in the Java(TM) programming language (Java objects) and Java objects to databases.

Java Card(TM) API
An ISO 7816-4 compliant application environment focused on smart cards.

JavaCheck(TM)
A tool for checking compliance of applications and applets to a specification.

JavaChip(TM)
Sun's processor, which executes bytecode for the Java(TM) virtual machine* natively. With a JavaChip processor, bytecode bypasses the virtual machine or just-in-time compiler stage to go directly to the processor.

Java(TM) Compatibility Kit (JCK)
A test suite, a set of tools, and other requirements used to certify a Java platform implementation conformant both to the applicable Java platform specifications and to Java Software reference implementations.

Java Database Connectivity (JDBC(TM))
An industry standard for database-independent connectivity between the Java(TM) platform and a wide range of databases. The JDBC(TM) provides a call-level API for SQL-based database access.

Java Developer Connection(SM)
A service designed for individual developers, providing online training, product discounts, feature articles, bug information, and early access capabilities.

Java Development Kit (JDK(TM))
A software development environment for writing applets and applications in the Java programming language.

Java(TM) Electronic Commerce Framework
A structured architecture for the development of electronic commerce applications in the Java(TM) programming language.

Java(TM) Enterprise API
This API makes it easy to create large-scale commercial and database applications that can share multimedia data with other applications within an organization or across the Internet. Four APIs have been designed within the Java(TM) Enterprise API family.

Java(TM) Foundation Classes (JFC)
An extension that adds graphical user interface class libraries to the Abstract Windowing Toolkit (AWT).

Java(TM) Interface Definition Language (IDL)
APIs written in the Java programming language that provide standards-based interoperability and connectivity with CORBA (Common Object Request Broker Architecture).

Java(TM) Media APIs

A set of APIs that support the integration of audio and video clips, 2D fonts, graphics, and images as well as 3D models and telephony.

Java(TM) Media Framework

The core framework supports clocks for synchronizing between different media (e.g., audio and video output). The standard extension framework allows users to do full audio and video streaming.

Java Naming and Directory Interface(TM) (JNDI)

A set of APIs that assists with the interfacing to multiple naming and directory services.

JavaOS(TM)

An Java(TM) technology-based operating system that is optimized to run on a variety of computing and consumer platforms. The JavaOS(TM) operating environment provides a runtime specifically tuned to run applications written in the Java programming language directly on hardware platforms without requiring a host operating system.

JavaPlan(TM)

An object-oriented design and diagramming tool written in the Java(TM) programming language.

Java(TM) Platform

Consists of a language for writing programs ("the Java(TM) programming language"); a set of APIs, class libraries, and other programs used in developing, compiling, and error-checking programs; and a virtual machine which loads and executes the class files.

In addition, the Java platform is subject to a set of compatibility requirements to ensure consistent and compatible implementations. Implementations that meet the compatibility requirements may qualify for Sun's targeted compatibility brands.

The Java(TM) 2 platform is the current generation of the Java platform.

Java(TM) Platform Editions

A Java platform "edition" is a definitive and agreed-upon version of the Java platform that provides the functionality needed over a broad market segment.

An edition is comprised of two kinds of API sets: (i) "core packages," which are essential to all implementations of a given platform edition, and (ii) "optional packages," which are available for a given platform edition and which may be supported in a compatible implementation.

There are 3 distinct editions of the Java Platform:

* Java 2 Platform, Enterprise Edition:
The edition of the Java platform that is targeted at enterprises to enable development, deployment, and management of multi-tier server-centric applications.

* Java 2 Platform, Standard Edition:
The edition of the Java platform that enables development, deployment, and management of cross-platform, general-purpose applications.

* Java 2 Platform, Micro Edition:
The edition of the Java platform that is targeted at small, standalone or connectable consumer and embedded devices to enable development, deployment, and management of applications that can scale from smart cards through mobile devices and set-top boxes to conventional computing devices.

Java(TM) Remote Method Invocation (RMI)

A distributed object model for Java(TM) program to Java program, in which the methods of remote objects written in the Java programming language can be invoked from other Java virtual machines*, possibly on different hosts.

Java(TM) Runtime Environment (JRE)

A subset of the Java Development Kit (JDK(TM)) for end-users and developers who want to redistribute the runtime environment alone. The Java runtime environment consists of the Java virtual machine*, the Java core classes, and supporting files.

JavaSafe(TM)

A tool for tracking and managing source file changes, written in the Java(TM) programming language.

JavaScript(TM)

A Web scripting language that is used in both browsers and Web servers. Like all scripting languages, it is used primarily to tie other components together or to accept user input.

Java Studio(TM)

The first program that allows you to easily create Java(TM) technology-based applications and applets without having to know the Java programming language.

Java(TM) Technologies

A set of technologies that enable the creation and safe running of software programs in both stand-alone and networked environments.

*Java(TM) virtual machine (JVM)**

Sun's specification for or implementation of a software "execution engine" that safely and compatibly executes the byte codes in Java class files on a microprocessor (whether in a computer or in another electronic device).

* Java HotSpot(TM) performance engine - Sun's ultra-high-performance engine for implementing the Java runtime environment which features an adaptive compiler that dynamically optimizes the performance of running applications.

* KJava(TM) virtual machine - Sun's small-footprint, highly optimized foundation of a runtime environment within the Java 2 Platform, Micro Edition. Derived from the Java virtual machine, it is targeted at small connected devices and can scale from 30KB to approximately 128KB, depending on the target device's functionality.

* Java Card(TM) virtual machine - Sun's ultra-small-footprint, highly-optimized foundation of a runtime environment within the Java 2 Platform, Micro Edition. Derived from the Java virtual machine, it is targeted at smart cards and other severely memory-constrained devices and can run in devices with memory as small as 24K of ROM, 16K of EEPROM, and 512 bytes of RAM.

Java Web Server(TM)

The easy-to-use, extensible, easy-to-administer, secure, platform-independent solution to speed and simplify the deployment and management of your Internet and Intranet Web sites. It provides immediate productivity for robust, full-featured, Java technology-based server applications.

Java Workshop(TM)

A complete set of tools integrated into a single environment for managing programming with Java technology. The Java Workshop software uses a highly modular structure that enables you to easily plug new tools into the overall structure.

Java(TM) wallet

A user interface, built on the Java(TM) Electronic Commerce Framework, which allows for online purchases, value transfers, and administrative functions.

JavaSpaces(TM)

A technology that provides distributed persistence and data exchange mechanisms for code in the Java(TM) programming language.

JavaSoft(TM)

A former business unit of Sun Microsystems, Inc., currently known as Sun Microsystems, Inc., Java Software division.

JDBC(TM)

Java(TM) Database Connectivity. An industry standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based database access.

JDK(TM)

Java(TM) Development Kit software. A software development environment for writing applets and application in the Java programming language.

JFC

Java(TM) Foundation Class. An extension that adds graphical user interface class libraries to the Abstract Windowing Toolkit (AWT).

Jini(TM) Technology

Sun's Jini technology includes a set of APIs that may be incorporated an optional package for any Java 2 platform edition. This set of APIs enables transparent networking of devices and services and eliminates the need for system or network administration intervention by a user.

The Jini technology is currently an optional package available on all Java 2 platform editions.

JMAPI

Java(TM) Management API. A collection of Java programming language classes and interfaces that allow developers to build system, network, and service management applications.

JNDI

Java Naming and Directory Interface(TM). A set of APIs that assist with the interfacing to multiple naming and directory services.

JPEG

Joint Photographic Experts Group. An image file compression standard established by this group. It achieves tremendous compression at the cost of introducing distortions into the image which are almost always imperceptible.

JRE

Java(TM) runtime environment. A subset of the Java Developer Kit for end-users and developers who want to redistribute the runtime environment. The Java runtime environment consists of the Java virtual machine*, the Java core classes, and supporting files.

Just-in-time (JIT) Compiler

A compiler that converts all of the bytecode into native machine code just as a Java(TM) program is run. This results in run-time speed improvements over code that is interpreted by a Java virtual machine*.

JVM

Java(TM) Virtual Machine*. The part of the Java Runtime Environment responsible for interpreting bytecodes.

K

keyword

The Java(TM) programming language sets aside words as keywords - these words are reserved by the language itself and therefore are not available as names for variables or methods.

L

Pertaining to how the characters in source code are translated into tokens that the compiler can understand.

linker

A module that builds an executable, complete program from component machine code modules. The Java(TM) linker creates a runnable program from compiled classes. See also *compiler*, *interpreter*, *runtime system*.

literal

The basic representation of any integer, floating point, or character value. For example, 3.0 is a double-precision floating point literal, and "a" is a character literal.

local variable

A data item known within a block, but inaccessible to code outside the block. For example, any variable defined within a method is a local variable and can't be used outside the method.

long

A Java(TM) programming language keyword used to define a variable of type long.

M

member

A *field* or *method* of a class. Unless specified otherwise, a member is not static.

method

A function defined in a class. See also *instance method*, *class method*. Unless specified otherwise, a method is not static.

Mosaic

A program that provides a simple GUI that enables easy access to the data stored on the Internet. These data may be simple files or hypertext documents. Mosaic was written by a team at *NCSA*.

multithreaded

Describes a program that is designed to have parts of its code execute concurrently. See also *thread*.

N

native

A Java(TM) programming language keyword that is used in method declarations to specify that the method is not implemented in the same Java source file, but rather in another language.

NCSA

National Center for Supercomputer Applications. See also *Mosaic*.

new

A Java(TM) programming language keyword used to create an instance of a class.

null

A Java(TM) programming language keyword used to specify an undefined value for reference variables.

O

object

The principal building blocks of object-oriented programs. Each object is a programming unit consisting of data (*instance variables*) and functionality (*instance methods*). See also *class*.

object-oriented design

A software design method that models the characteristics of abstract or real objects using classes and objects.

octal

The numbering system using 8 as its base, using the numerals 0-7 as its digits. In programs written in the Java(TM) programming language, octal numbers must be preceded with 0. See also *hexadecimal*.

Optional Packages

The set or sets of APIs in a Java platform edition which are available with and may be supported in a compatible implementation.

Over time, optional packages may become required in an edition as the marketplace requires them.

overloading

Using one identifier to refer to multiple items in the same scope. In the Java(TM) programming language, you can overload methods but not variables or operators.

overriding

Providing a different implementation of a method in a subclass of the class that originally defined the method.

P

package

A group of *types*. Packages are declared with the `package` keyword.

peer

In networking, any functional unit in the same layer as another entity.

PersonalJava(TM)

A Java runtime environment for network-connectable applications on personal consumer devices for home, office, and mobile use.

pixel

The smallest addressable picture element on a display screen or printed page.

POSIX

Portable Operating System for UNIX(TM). A standard that defines the language interface between the UNIX operating system and application programs through a minimal set of supported functions.

private

A Java(TM) programming language keyword used in a method or variable declaration. It signifies that the method or variable can only be accessed by other elements of its class.

process

A virtual address space containing one or more threads.

property

Characteristics of an object that users can set, such as the color of a window.

Profiles

A Profile is a collection of Java APIs that complements one or more Java 2 Platform Editions by adding domain-specific capabilities. Profiles may also include other defined Profiles. A profile implementation requires a Java 2 Platform Edition to create a complete development and deployment environment in a targeted vertical market. Each profile is subject to an associated set of compatibility requirements.

Profiles may be usable on one or more editions.

Some examples of profiles within the Java 2 Platform, Micro Edition are:

* PersonalJava(TM) - for non-PC products that need to display web-compatible Java-based content

* Java Card(TM) - for secure smart cards and other severely memory-constrained devices.

protected

A Java(TM) programming language keyword used in a method or variable declaration. It signifies that the method or variable can only be accessed by elements residing in its class, subclasses, or classes in the same package.

public

A Java(TM) programming language keyword used in a method or variable declaration. It signifies that the method or variable can be accessed by elements residing in other classes.

Q

R

raster

A line of pixels.

reference

A data element whose value is an address.

return

A Java(TM) programming language keyword used to finish the execution of a method. It can be followed by a value required by the method definition.

RFE

Request for Enhancement.

RMI

See *Java Remote Method Invocation*.

root

In a hierarchy of items, the one item from which all other items are descended. The root item has nothing above it in the hierarchy. See also *hierarchy*, *class*, *package*.

RPC

Remote Procedure Call. Executing what looks like a normal procedure call (or method invocation) by sending network packets to some remote host.

runtime system

The software environment in which programs compiled for the Java(TM) virtual machine* can run. The runtime system includes all the code necessary to load programs written in the Java programming language, dynamically link native methods, manage memory, handle exceptions, and an implementation of the Java virtual machine, which may be a Java interpreter.

S

Sandbox

Comprises a number of cooperating system components, ranging from security managers that execute as part of the application, to security measures designed into the Java(TM) virtual machine* and the language itself. The sandbox ensures that an untrusted, and possibly malicious, application cannot gain access to system resources.

scope

A characteristic of an identifier that determines where the identifier can be used. Most identifiers in the Java(TM) programming environment have either class or local scope. Instance and class variables and methods have class scope; they can be used outside the class and its subclasses only by prefixing them with an instance of the class or (for class variables and methods) with the class name. All other variables are declared within methods and have local scope; they can be used only within the enclosing block.

Secure Socket Layer (SSL)

A protocol that allows communication between a Web browser and a server to be encrypted for privacy.

servlet

A server-side program that gives Java(TM) technology-enabled servers additional functionality.

short

A Java(TM) programming language keyword used to define a variable of type short.

single precision

In the Java(TM) language specification, describes a floating point number with 32 bits of data. See also *double precision*.

SGML

Standardized Generalized Markup Language. An ISO/ANSI/ECMA standard that specifies a way to annotate text documents with information about types of sections of a document.

static

A Java(TM) programming language keyword used to define a variable as a class variable. Classes maintain one copy of class variables regardless of how many instances exist of that class. "static" can also be used to define a method as a class method. Class methods are invoked by the class instead of a specific instance, and can only operate on class variables.

static field

Another name for *class variable*.

static method

Another name for *class method*.

subarray

An array that is inside another array.

subclass

A class that is derived from a particular class, perhaps with one or more classes in between. See also *superclass*, *supertype*.

subtype

If type X *extends* or implements type Y, then X is a subtype of Y. See also *supertype*.

superclass

A class from which a particular class is derived, perhaps with one or more classes in between. See also *subclass*, *subtype*.

super

A Java(TM) programming language keyword used to access members of a class inherited by the class in which it appears.

supertype

The supertypes of a type are all the interfaces and classes that are extended or implemented by that type. See also *subtype*, *superclass*.

switch
A Java(TM) programming language keyword used to evaluate a variable that can later be matched with a value specified by the "case" keyword in order to execute a group of statements.

Swing Set
The code name for a collection of graphical user interface (GUI) components that runs uniformly on any native platform which supports the Java(TM) virtual machine*. Because they are written entirely in the Java programming language, these components may provide functionality above and beyond that provided by native-platform equivalents. (Contrast with AWT.)

synchronized
A keyword in the Java programming language that, when applied to a method or code block, guarantees that at most one thread at a time executes that code.

T

TCP/IP
Transmission Control Protocol based on IP. This is an Internet protocol that provides for the reliable delivery of streams of data from one host to another. See also *IP*.

Technology Compatibility Kit (TCK)
A test suite, a set of tools, and other requirements used to certify an implementation of a particular Sun technology conformant both to the applicable specifications and to Sun or Sun-designated reference implementations.

Thin Client
A system that runs a very light operating system with no local system administration and executes applications delivered over the network.

this
A Java(TM) programming language keyword that can be used to represent an instance of the class in which it appears. "this" can be used to access class variables and methods.

thread
The basic unit of program execution. A process can have several threads running concurrently, each performing a different job, such as waiting for events or performing a time-consuming job that the program doesn't need to complete before going on. When a thread has finished its job, the thread is suspended or destroyed. See also *process*.

throw
A Java(TM) programming language keyword that allows the user to throw an exception or any class that implements the "throwable" interface.

throws
A Java(TM) programming language keyword used in method declarations that specify which exceptions are not handled within the method but rather passed to the next higher level of the program.

transient
A keyword in the Java programming language that indicates that a field is not part of the serialized form of an object. When an object is serialized, the values of its transient fields are not included in the serial representation, while the values of its non-transient fields are included.

try
A Java(TM) programming language keyword that defines a block of statements that may throw a Java language exception. If an exception is thrown, an optional "catch" block can handle specific exceptions thrown within the "try" block. Also, an optional "finally" block will be executed regardless of whether an exception is thrown or not.

A class or interface.

U

Unicode

A 16-bit character set defined by ISO 10646. See also *ASCII*. All source code in the Java(TM) programming environment is written in Unicode.

URL

Uniform Resource Locator. A standard for writing a text reference to an arbitrary piece of data in the WWW. A URL looks like "protocol://host/localinfo" where protocol specifies a protocol to use to fetch the object (like HTTP or FTP), host specifies the Internet name of the host on which to find it, and localinfo is a string (often a file name) passed to the protocol handler on the remote host.

V

variable

An item of data named by an identifier. Each variable has a type, such as `int` or `Object`, and a scope. See also *class variable*, *instance variable*, *local variable*.

virtual machine

An abstract specification for a computing device that can be implemented in different ways, in software or hardware. You compile to the instruction set of a virtual machine much like you'd compile to the instruction set of a microprocessor. The Java(TM) virtual machine* consists of a bytecode instruction set, a set of registers, a stack, a garbage-collected heap, and an area for storing methods.

void

A Java(TM) programming language keyword used in method declarations to specify that the method does not return any value. "void" can also be used as a nonfunctional statement.

volatile

A Java(TM) programming language keyword used in variable declarations that specifies that the variable is modified asynchronously by concurrently running threads.

W

wait

A UNIX® command which will wait for all background processes to complete, and report their termination status.

while

A Java(TM) programming language keyword used to declare a loop that iterates a block of statements. The loop`s exit condition is specified as part of the while statement.

world readable files

Files on a file system that can be viewed (read) by any user. For example: files residing on web servers can only be viewed by Internet users if their permissions have been set to world readable.

wrapper

An object that encapsulates and delegates to another object to alter its interface or behavior in some way.

WWW

World Wide Web. The web of systems and the data in them that is the Internet. See also *Internet*.

X

Z

***As used on this web site, the terms "Java virtual machine" or "JVM" mean a virtual machine for the Java platform.**