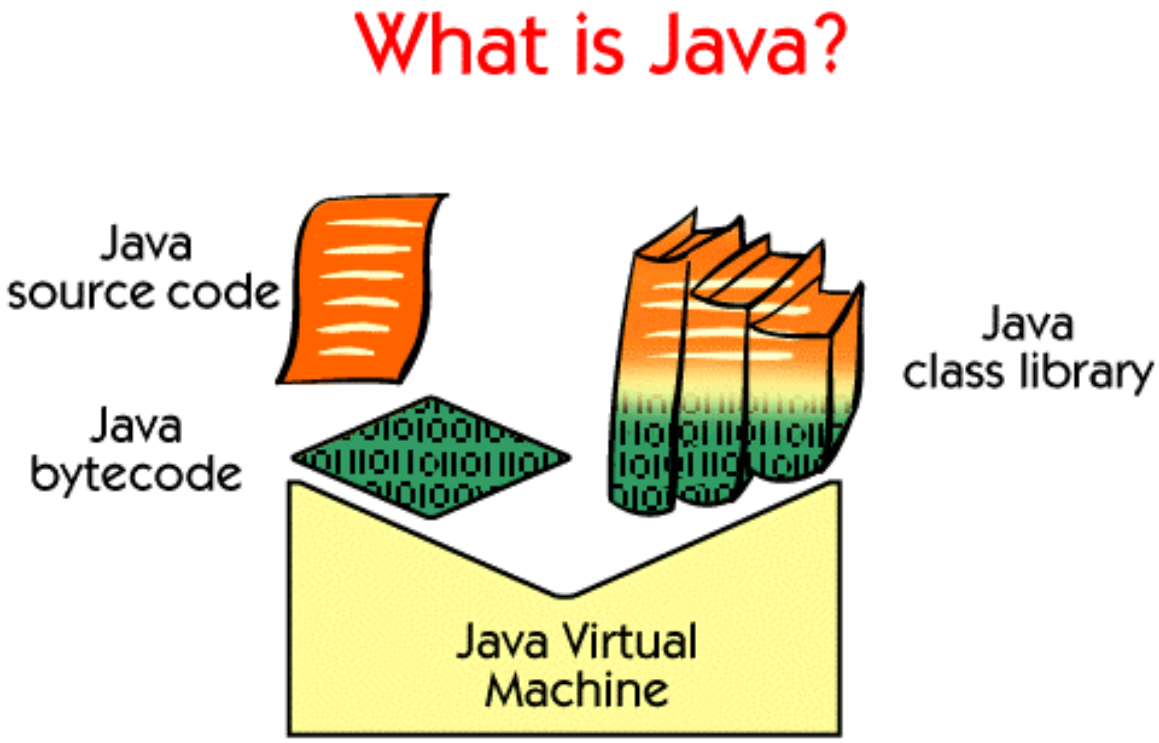
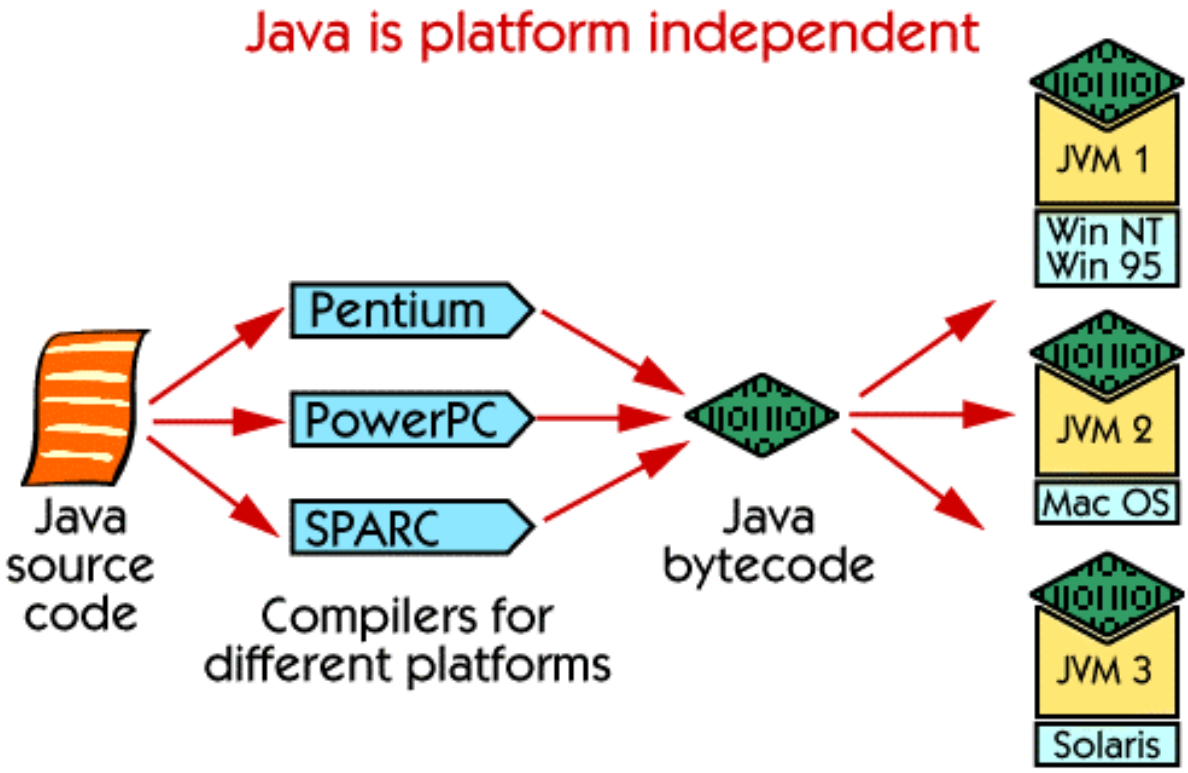
Ch.12 - Java

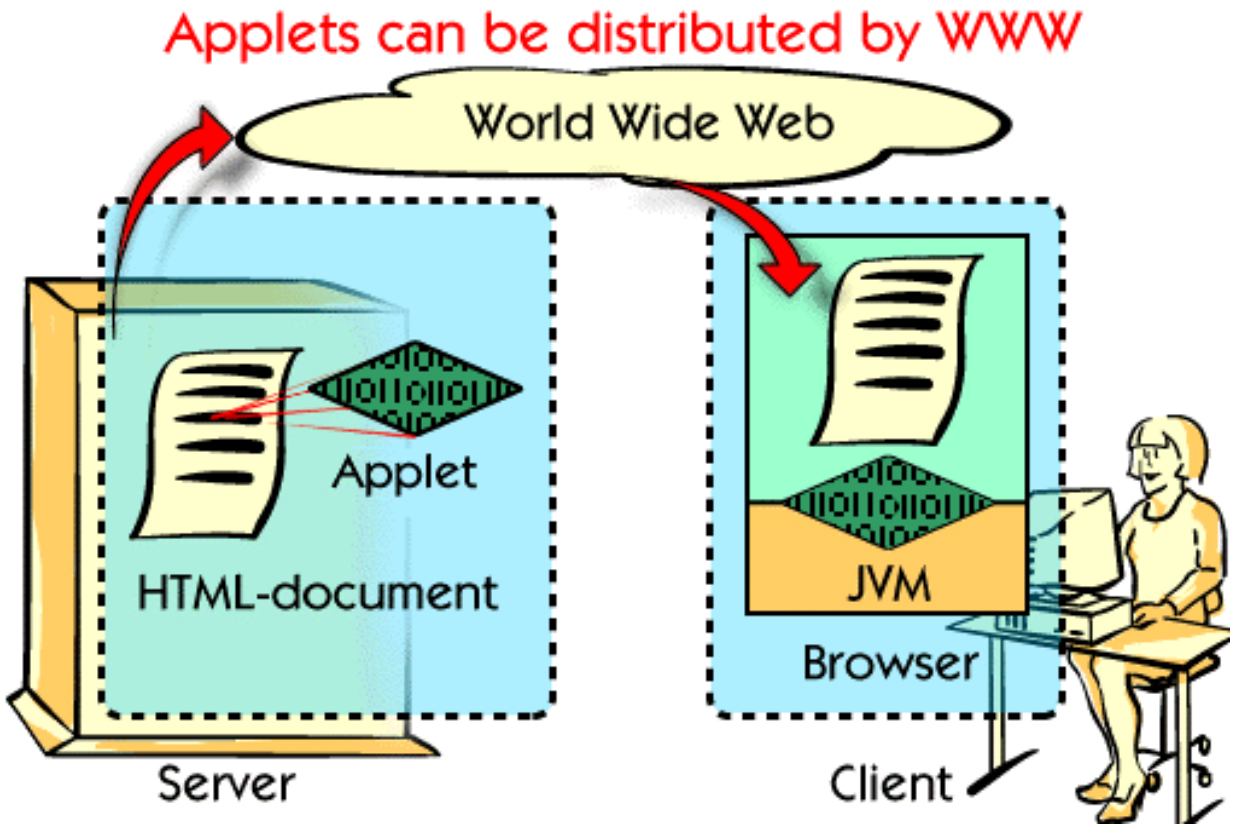


What's Java, really? Well, first one might say that it's a good object oriented programming language, where the best of other languages is extracted and put together with some own supplements, for example the security model which is important to Internet applications. Java has an extensive standard library of classes, making the programming a lot more effective. There is support for network communication, windows management, file management, audio and video, security, data access etc.
When the source code in the Java language is compiled, the Java byte code is created. The byte code is a low-level language, consisting of machine instructions for the Java Virtual Machine. The byte code makes Java independent of any platform.
The Java Virtual Machine is an abstract computer, on which the byte code can execute. The Java Virtual Machine becomes a middle-level between the byte code and any operating system, so, seen from the byte code's perspective, it's always running on the same computer. The Java Virtual Machine has up to now been implemented as software for the operating systems Windows NT, Windows 95, Solaris, and MacOS. For faster performance, Java Virtual Machine will be made in silicone.



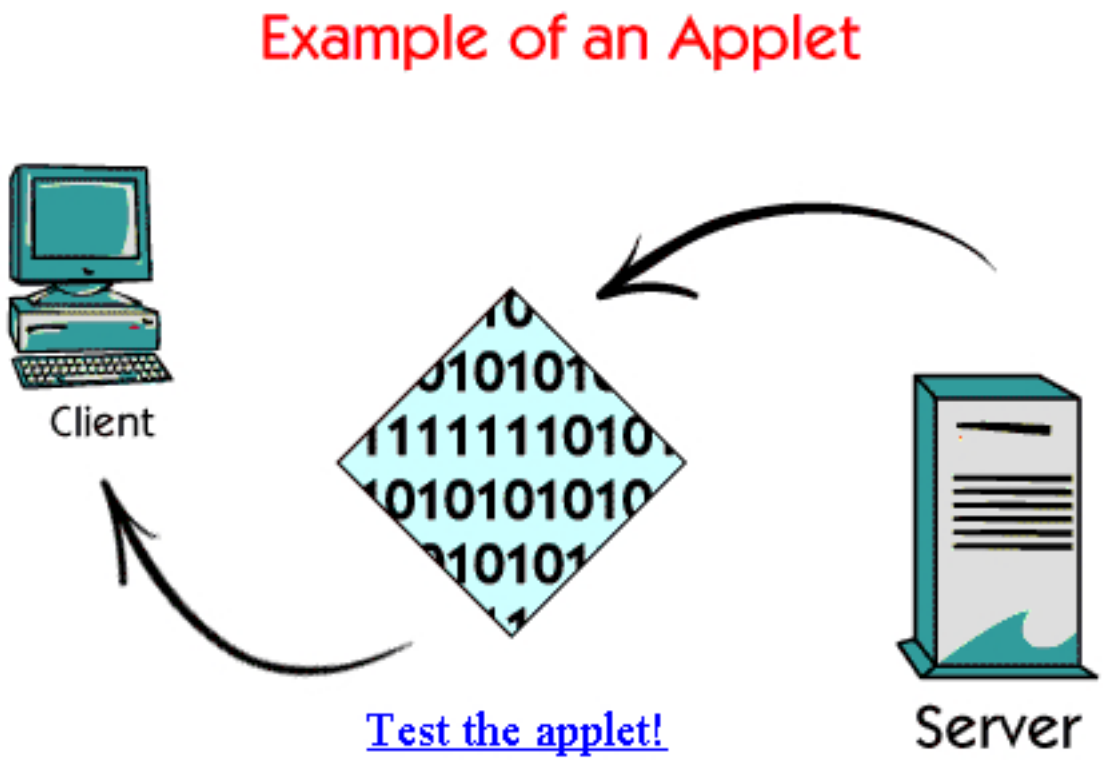
This picture shows that the source code written by the programmer and the byte code are the same for all platforms. The source code may be compiled on different platforms such as Pentium, PowerPC, SPARC and so on, but the result is always the same Java byte code.

The byte code may be executed on different Java Virtual Machines. The difference between Java and other programming languages is that the source code doesn't have to be re-written for different operating systems, nor for new processors.

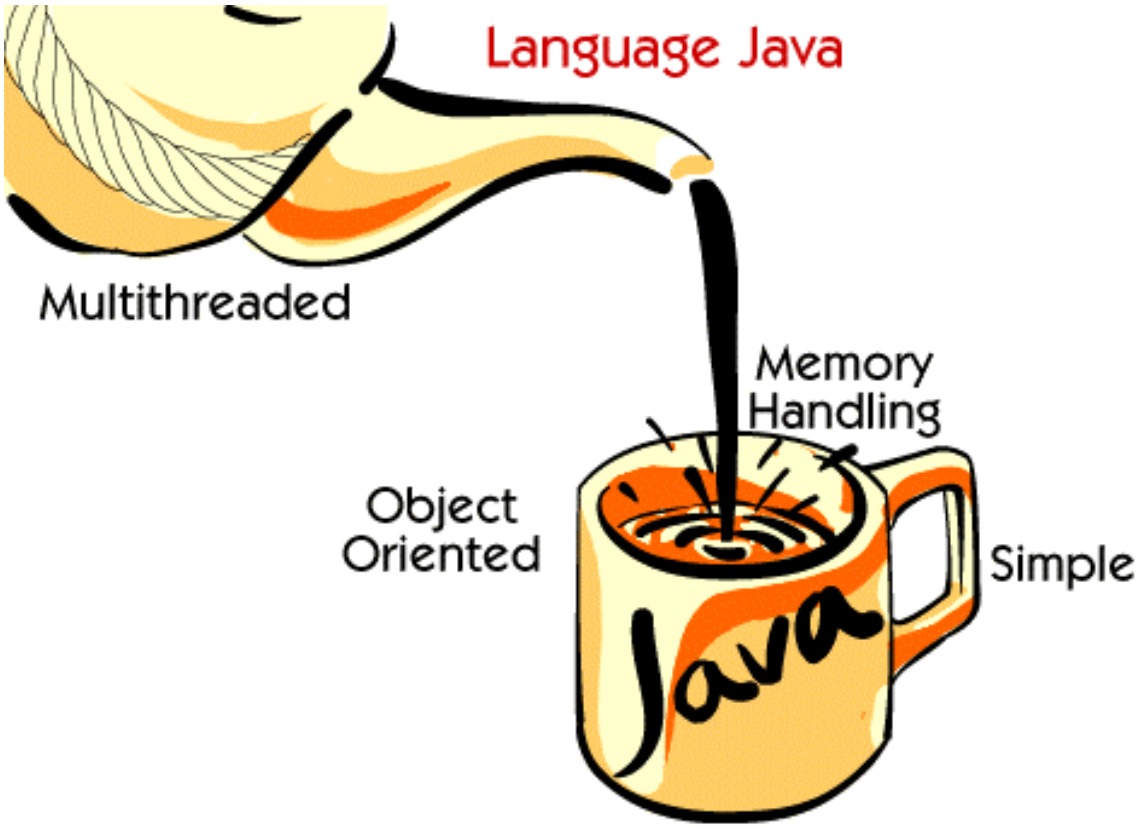


An applet is a Java byte code that can be distributed through the World Wide Web and used in a Web page.
What we see here is a server, the World Wide Web, and a client. On the server machine there is a Web-server, capable of sending HTML-pages and adjoining files. The client machine has a Web-browser.
When the user clicks on a link to a page containing a reference to an applet, the browser will download both the page and the applet. Then it will present the page to the user and start the applet. In order to run, the applet needs a Java Virtual Machine. Most modern Web-browsers have a Java Virtual Machine built in, and since almost everyone surfing the Net has a browser, they also have a Java Virtual Machine.
Since the user has downloaded an applet in the Web-browser on his own computer, he has new possibilities for interactivity, directly on the client. Compare this to using the old CGI-technique, where interaction would have demanded intensive network traffic between the client and server.
There is also a possibility of having more than one Java applet for a Web page. These Java applets can communicate with each other.
The applets may also read parameters from the HTML page itself; in that case there is a possibility to affect certain settings of an applet by using different HTML pages. An example is an applet giving different layout to text on the HTML page, the text and the layout settings could both be parameters in the HTML page red by the applet

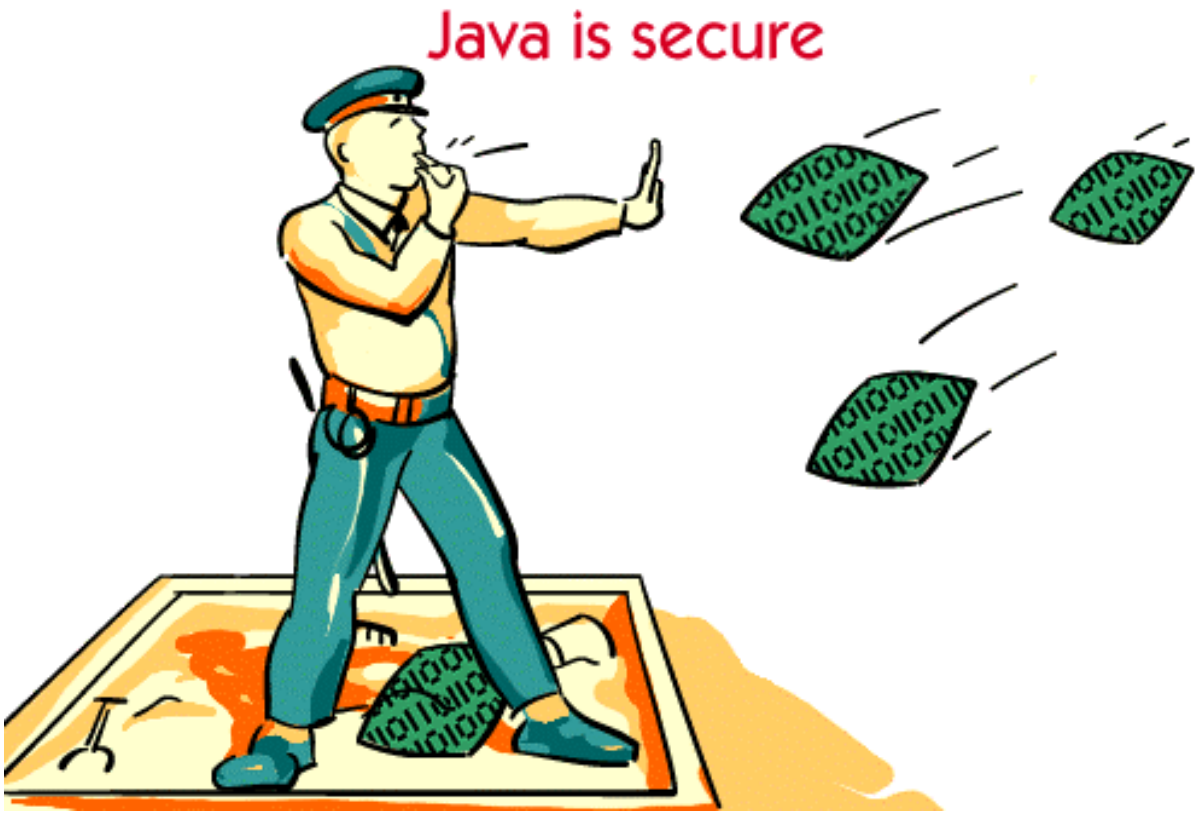
To keep a high level of security, in order to stop the spreading of viruses, Netscape and Microsoft both chose to strongly limit what an applet is allowed to do. It may not, for example, communicate with a different computer than the one from which it was downloaded. Also, it may not read nor write on any file in the client machine.



Here is link to an example of an applet. This is a free applet that runs on a client machine, that is, it has no communication with the server after getting downloaded.
Click on the link "Go to Snake Game"
http://www.halcyon.com/mach/java/wormgame/wormpage.htm
and test it. Don't forget to return when you have finished the testing.



Java is an elegant object oriented language, with a simple construction.
James Gosling, the creator of Java at SUN, chose and combined the best parts of several different languages. The syntax of Java is almost the same as in C++, but Java is simpler and more object oriented.
There is support for multithreaded applications in Java, so several tasks may run parallel. One thread could manage the user interface, at the same time as another thread is scanning the net.
Java has an automatic memory management, which lessens the complexity of the programming. The memory is not directly accessible, which leads to fewer viruses and other bugs in the programs.



The security model of Java is usually called the "sand-box-model" since the Java programs only get a limited part of the computer to execute within.
With the help of a Security Manager, and a self-described security policy, the things Java is allowed to do at an application level, are limited. With Security Manager, you can determine whether Java is allowed to read or write to a file, the computers it is allowed to contact or if it is allowed to call a C-program.
No direct access to the computer's memory is possible from a pure Java program, which means that the risk of applets deleting or changing files from your hard disk is significantly reduced.
Furthermore the code is not only controlled when it's compiled, but also when it's loaded for execution. No illegal Java code is allowed to run in the Java Virtual Machine.

Java in four versions

Embedded Java

Personal Java

Standard Java

Enterprise Java

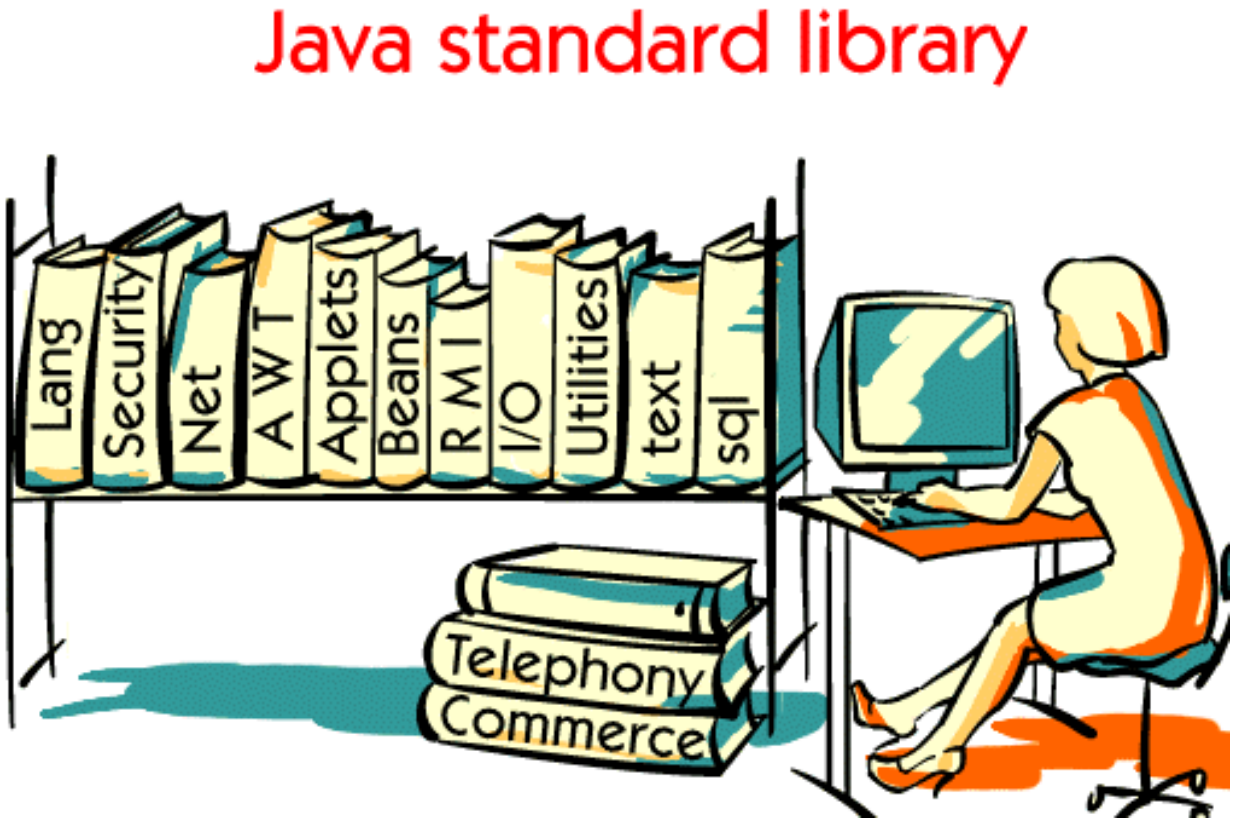
The Java Company of Sun, Javasoft, is currently developing four different versions of Java. The language is always the same, but the difference is the size of the class library that's included.

Embedded Java is a minimal part of Java meant for use in small platforms, e.g. smart cards, or software controlled electrical components in which you want to download the software from a network and dynamically change it.

Personal Java is intended for products with some sort of display, although lacking keyboard. Those products could be faxes, copy machines, TV sets, videos or other home-electronics on which one might want to change the software dynamically.

Standard Java is the original and most spread Java environment, which has the potential to become one of the big developing environments in the next decade.

Enterprise Java is meant to be used in business critical applications, with a more developed security model and more attachments to commercial systems than standard Java.



There is an extensive standard class library for Java. Developers may count on it being on every standard Java platform. Besides from this library, there are many other classes from Sun and other tool suppliers.

The standard library includes a package to manage user interface, AWT or Abstract Window Toolkit. This is largest, and most important of the standard Java packages. It includes classes that let you manipulate Windows, Dialogs, Graphics, Buttons, and so on. A program written with AWT will look somewhat different on different platforms, due to the fact that Java Virtual Machine uses the components of the respective operating systems to display the AWT-components on-screen.

Another frequently used package is the Applet-package which is used for building applets.

The Net-package is used for network communication management, also supporting TCP/IP communication.

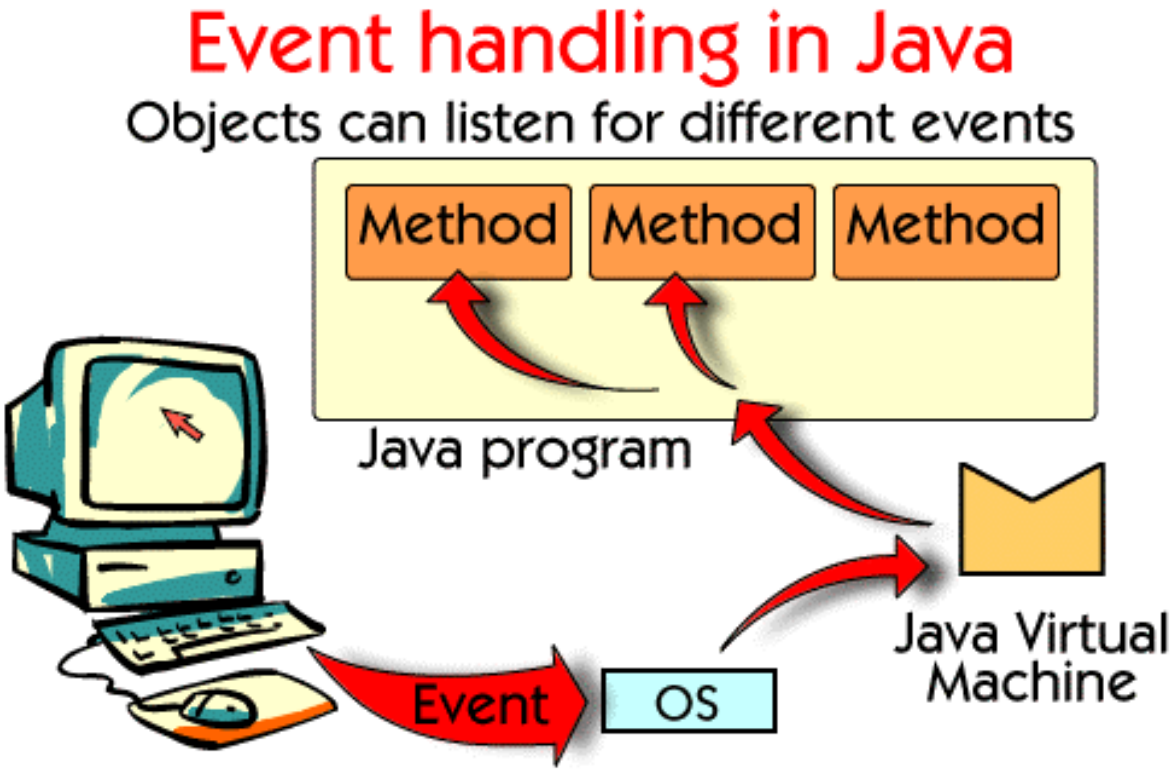
The I/O-package is used to read and write files, including files accessed over the Internet.

In the SQL-package there is support for accessing an ODBC-database with help of the SQL standard language.

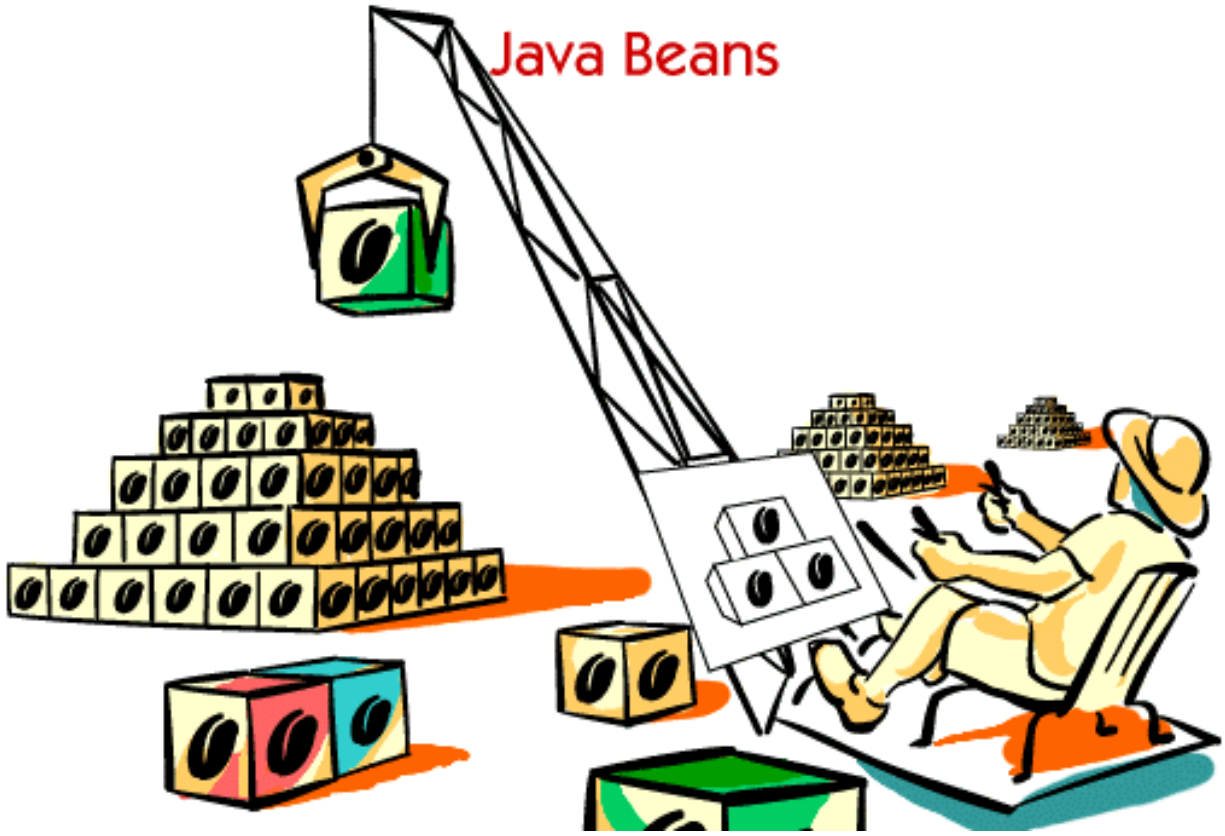
The Security package supports data-encryption. There is support for digital signatures, and public-key encryption.

The RMI package, which stands for Remote Method Invocation, supports building applications distributed on a network. With RMI a program is able to communicate with programs that run on other computers. In this way you can divide a job between several computers.

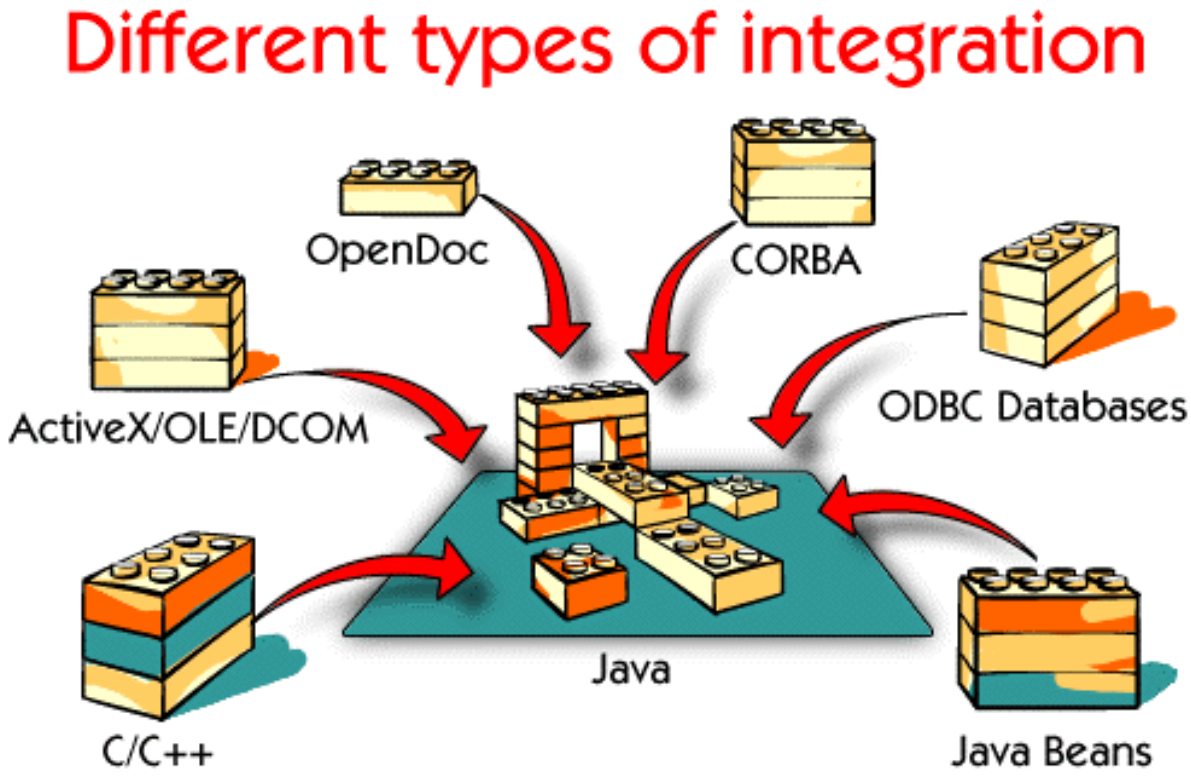
The class library makes the program development quicker, since the programmer is able to use what's already written and well functioning, without having to do it all by himself. Another advantage is that only small programs need to be transferred over the net, since the class library already exists on the Java platforms.



The user interface in Java is event controlled. When a user pushes a key or moves the mouse, the operating system generates events. These events are caught by the Java environment, and transformed into Java events. There are events for different classes, and the different parts of the Java program may choose which ones to listen for.
Events don't always need to come from the operating system. They could be directly generated by the Java program. In this way, the programmer can gain increased control and events may be simulated.



Java Beans is a standard for software components in Java. The programmer can build applications faster since he can assemble already made pieces into bigger pieces. The purpose of Java Beans is to make Bean-components available for purchase from different suppliers so that a programmer doesn't need to program everything from scratch.
A Java bean is a software component which can be graphically handled by different software tools. The idea is that a programmer can combine Java beans graphically, without having to write code. The tools generate the code automatically.
Java Beans could be compared to ActiveX, the industrial standard from Microsoft for software components. ActiveX is in this context an old and reliable technique that is based on OLE and DCOM. This means that there already exists quite a lot ActiveX-components. Unlike Active X, Java Beans components are written totally in Java, and will therefore become platform independent.



Java is an open system, which means that Java can co-operate with other types of systems.

There are some attachments to other object models and languages, also over networks.

In order for a Java program to be integrated with another Java program in a network, RMI (Remote Method Invocation) is preferably used, since it contains Java specific optimization. The Java Beans architecture may also be used at a higher level.

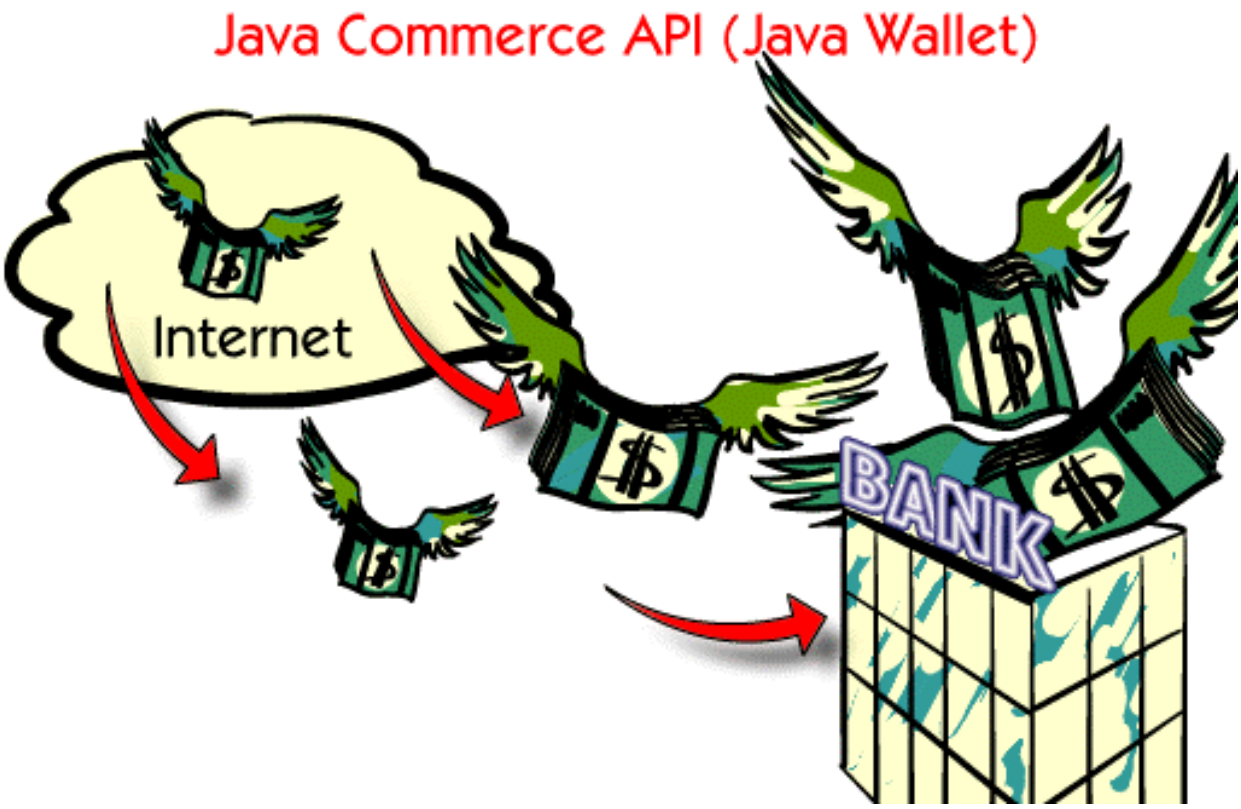
If instead the program is written in, say, C++ or Smalltalk, CORBA may be used, supported by Java to integrate the system.

There is a built-in support in Java for the communication with ODBC databases.

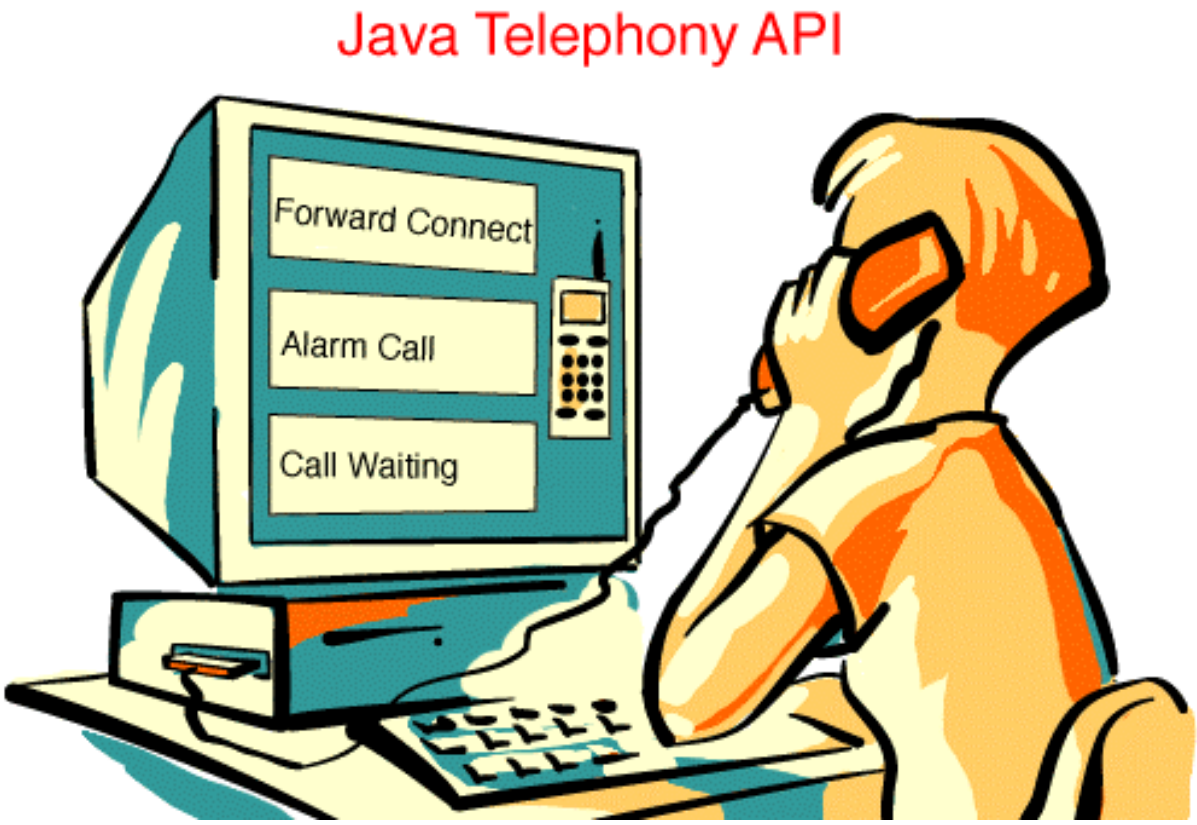
There are also connections to Apple's and IBM's OpenDoc.

If there is a need to use ActiveX, there is a bridge, so a Java Bean can easily be transformed into an ActiveX component.

There is also support to call a C++ method directly from within a Java program.

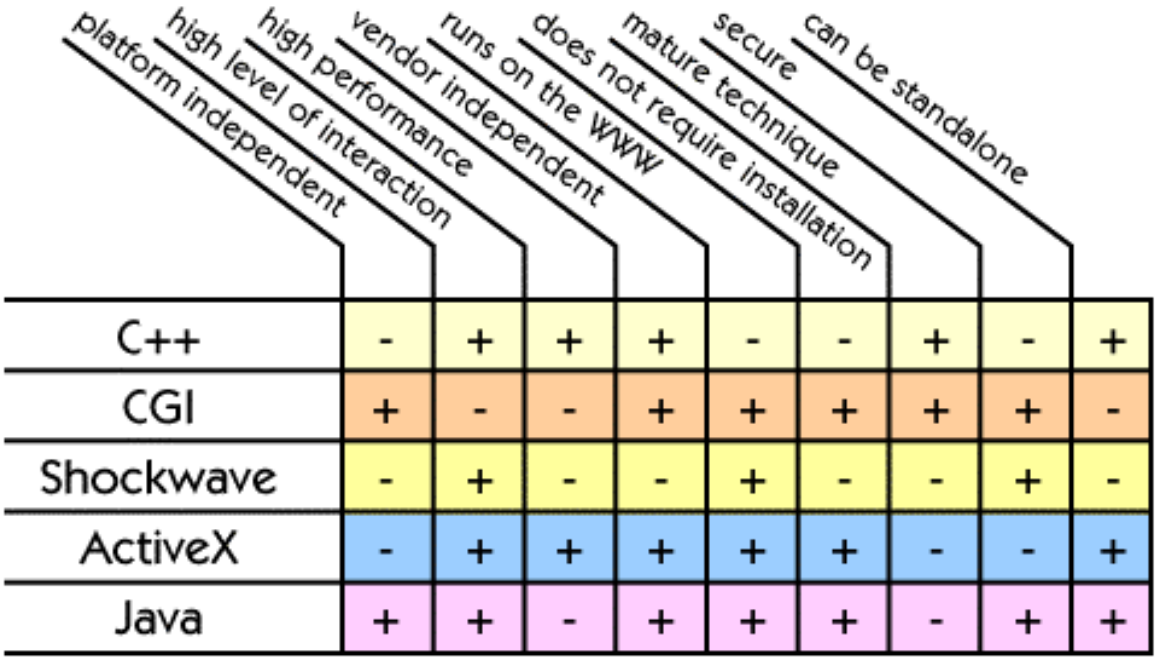


Java Commerce API is an extension of Java's class library. In this extension there will be support for trading on the Net. It will be easier to accept payment for program and data usage, and an open platform will be available for bank and finance applications. In the Java commerce API there will be methods for encryption of messages and authorization control. Included in the Java Commerce API there will be partially prefabricated programs for handling sales, and high-level components for finance applications.



Java Telephony API is a package for integration of computers and telephones. With this API it's possible to control a telephone card in the computer. There are great possibilities to control a telephone, ordinary or cellular, through a Java program.
One example is to get a graphical interface for your telephone including the services you want to order. Alarm Call, Call Waiting and other services could be shown pedagogically so ordering will become easier. At the same time new possibilities will open up for more advanced functions.

Alternative to Java



	platform independent	high level of interaction	high performance	vendor independent	runs on the WWW	does not require installation	mature technique	secure	can be standalone
C++	-	+	+	+	-	-	+	-	+
CGI	+	-	-	+	+	+	+	+	-
Shockwave	-	+	-	-	+	-	-	+	-
ActiveX	-	+	+	+	+	+	-	-	+
Java	+	+	-	+	+	+	-	+	+

How does Java compare to other techniques on Internet?

Java shows big similarities with C++. Java is platform independent which C++ is not, but C++ is more mature, and performs better.

The CGI-technique has been used for long time to create forms and active surfaces on HTML-pages. Although this technique is platform independent on the client part, the performance is so low due to constant network communication, that only a small part of Java's functionality is achieved.

ActiveX shouldn't be directly compared with Java since it's an architecture for software components. Compared to Java Beans, which is the equivalent of ActiveX, the difference is that ActiveX is faster but not platform independent.

Shockwave is a family of plug-ins from Macromedia. It has high level of interactivity but it requires installation on the client side which Java doesn't.

In summary, considering platform independence, security, the possibility of expanding a program with new parts while running it, and the very strong industrial support, one might say that Java is a good alternative for many applications.