

International Journal of Advance Research in Computer Science and Management Studies

Research Paper

Available online at: www.ijarcsms.com

A Study on Buzzword in Java

Abhilasha M. Raval

M.C.A. Department

Saurashtra University

IGNOU Rajkot Regional

India

Abstract: There are many reasons why Java is so popular and some of the reasons are javas' strongly supports features.

These features have made Java the first application language of the World Wide Web. Java will also become the premier language for general purpose stand-alone applications.

Keywords: Platform Independent, Object Oriented, Easy to Learn, Robust, Secure, Multi-Threading, Dynamic, High Performance.

I. Introduction

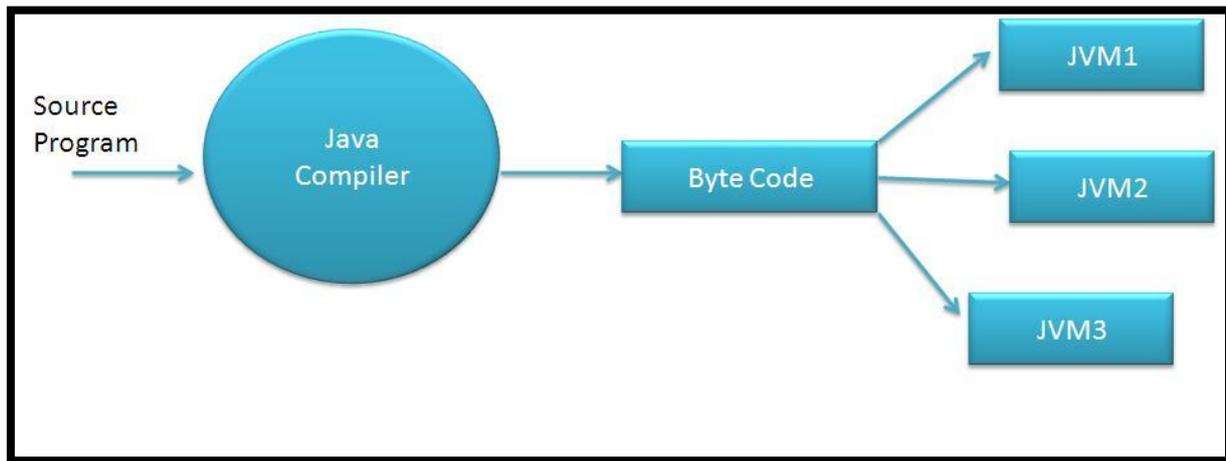
The features of Java this torrent computer-speak jargon has often been labelled the "Buzzword description" and was doubtless intended with tongue in cheek, it nevertheless accurately identifies many of the features of Java that they make it so well-suited for programming internet applications.

The following Java Buzzwords:

- 1) Platform Independent
- 2) Secure
- 3) Portable
- 4) Object-Oriented
- 5) Robust
- 6) Architecture-neutral
- 7) Multithread
- 8) Interpreted
- 9) High performance
- 10) Distributed
- 11) Dynamic

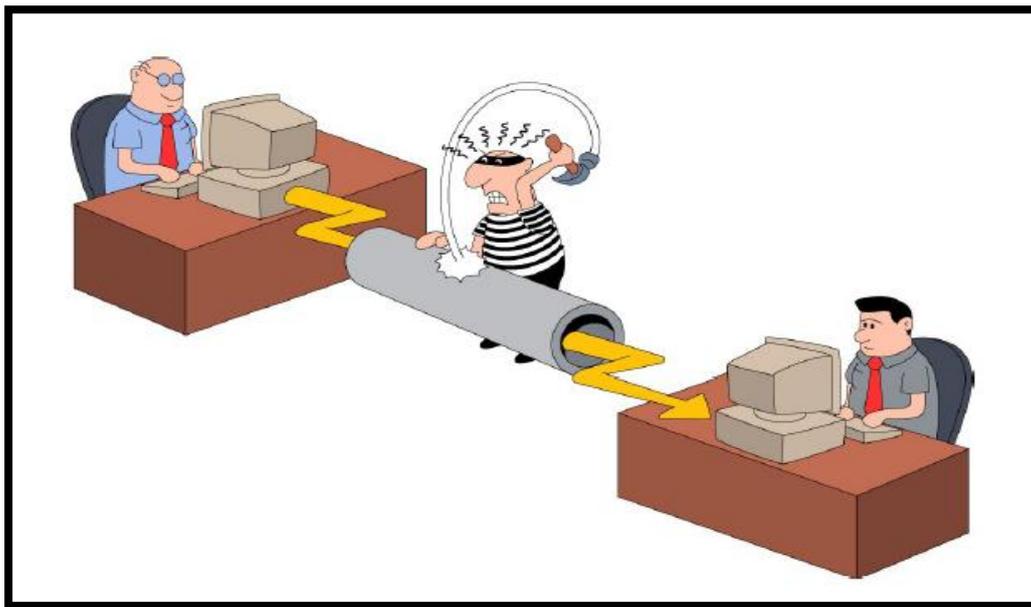
II. Platform Independent

Java is Platform independent. The meaning of the platform here, in the computer industry it typically means some combination of hardware and system software. Java is compiled to an intermediate form called **Java byte-code** or simply byte code Java program never really executes immediately after compilation on the host machine. Rather, this special program called the Java interpreter or Java Virtual Machine reads the byte code, translates it into the corresponding host machine instructions and then executes the machine instruction. A Java program can run on any computer system for which a JVM (**Java Virtual Machine**) so Java is platform independent.



III. Secure

Java is intended to work in networked and distributed environments by providing security. All the references to memory are **symbolic references**, meaning that the user is not aware where in the memory program is present; it totally depends on the JVM and the machine on which the program is running. Each applet is loaded on its own memory space, which avoids the information interchange between applets. Introducing **viruses**, deleting and modifying file in the host computer. The Java enabled web browser checks the byte code of applets to ensure that it should not do anything wrong before it will run the applet. The major security issue in today's software world is BUGS.



Unintended bugs are responsible for more data loss than data loss because of viruses. In Java it is easier to write bug-free code than in other languages.

IV. Portable

Early form portability involved carrying media, for example, floppy disks, from one system to another. Portability becomes a much larger problem once different sorts of computers were interconnected to form the internet.

Java programs contain no implementation-dependent aspects, so the result of executing a series of Java byte codes should always be the same, no matter on what system they are executed. Moreover, the Java run-time system itself, though it is written in c is written in a way that simplifies porting the Java run-time for a new computer system.

V. Object-Oriented

In objects represent object-oriented languages' data. Objects have two sections. The first is **Data** (instance variables) and the second is **methods**. Data represents what an object is. A method represents what an object does. The Data and methods are closely related to the real world structure and behavior of objects. Java is a true object-oriented language, which provides a platform to develop an effective and efficient application and program real life complexities. Java does not allow methods without class, thus an application consists of only the object which makes it true OOP. Most of the Object-oriented concepts in Java are inherited from C++ which makes it easy for traditional programmers to understand it.

VI. Robust

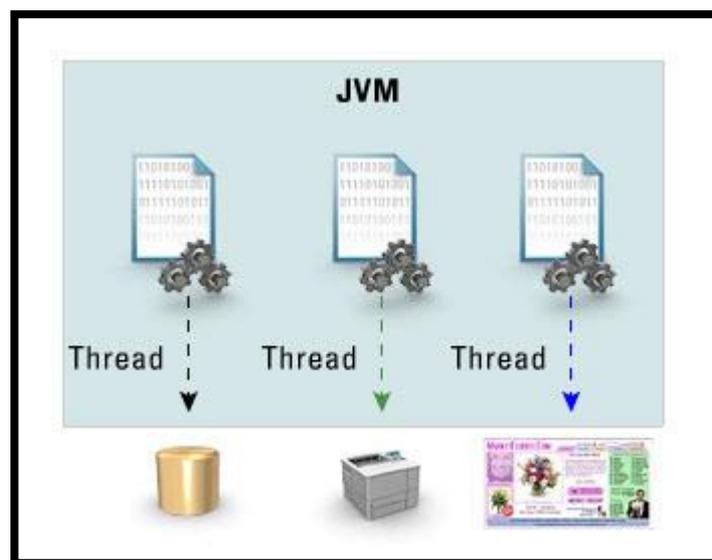
Java provides checking for possible problems at two levels, one at the compile time and the other at the run time, so programs are highly **reliable** and eliminate situations that are error-prone compared to C/C++. The best and worst features of C and C++ are **pointers** that help in direct manipulation of memory addresses. The power of pointers is as a great tool used by expert programmers in developing system software, driver etc. but many times pointers is the main cause of **runtime errors** because of improper use of memory. Java eliminates pointer manipulation completely from the language, and therefore eliminates a large source of runtime errors. Java programmers need not remember to de-allocate memory in programs since there is a **garbage collection mechanism** which handles de-allocation of memory. It provides powerful a robust exception handling mechanism to deal with both expected and unexpected errors at run time.

VII. Architecture-Nature

The word architecture in this phase does not refer to the building in which live. But to the home in which your computer program lives, we can say that the computer systems. Java's byte code are designed to be read and interpreted in exactly the same manner on any computer hardware or operating system that supports a Java run time

VIII. Multithread

A thread is a program's path of execution. In any programming problems, when multiple Events or actions need to occur at the same time. Multithreaded applications deliver their potent power by running many threads **Concurrently** within a single program.



Java is inherently multi-threaded, for example garbage collection subsystem runs as a low-priority thread. A single Java program can have many different threads executing independently and continuously, for example, different Java applets on the same web page can run together with getting equal time from the processor. Because multithreaded applications share data and all threads of an application exists in the same data space therefore for maintaining reliability is sometime difficult. To make easy the use of threads Java offers features for synchronization between threads.

IX. Interpreted

One of the first advances, after the intention of the programs itself, was the of the program interpreter. Like the early short – code and speed code, Java is an interpreted language.

This means the Java executable files are composed of so called byte codes that are instructions and data relating to a hypothetical computer called Java virtual machines.

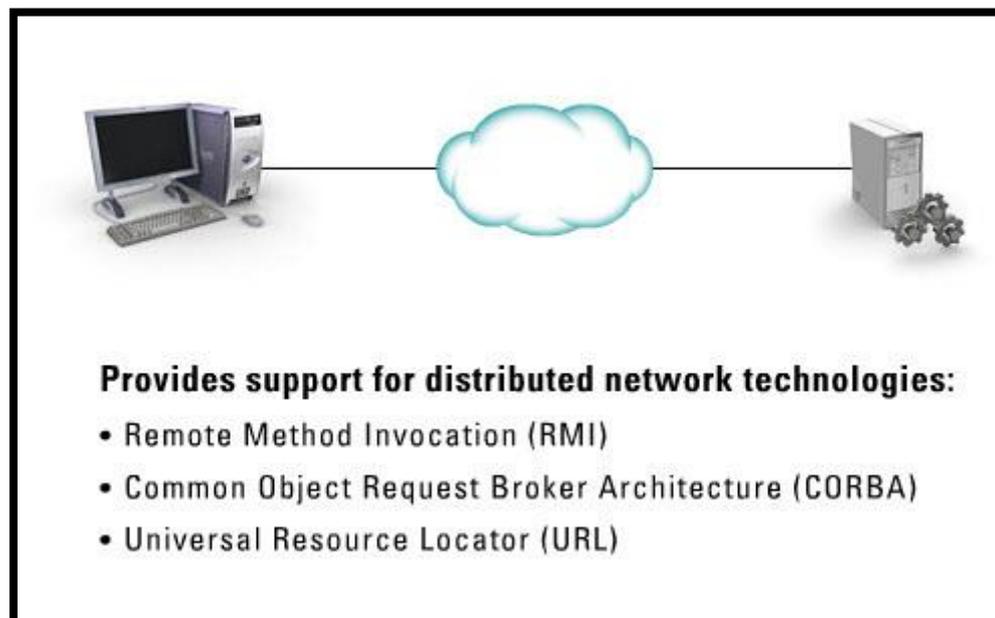
X. High Performance

In Java first compile the program, then execute it using the Java interpreter. In general, interpreters are slow, because an interpreter executes program instruction by instruction while Java is a fast-interpreted language. Java has also been designed so that the run-time system can optimize their performance by compiling byte code to native machine code on the fly (execute immediately after compilation). Many companies are a working on a native - machine-architecture compilers for Java. These will produce an executable code that does not require a separate interpreter, and that is indistinguishable in speed from C++.

Java offers two flavors of programming, Java applets and Java application. Applets are small Java programs (mostly) that can be downloaded over a computer network and run from a web page by using a Java enabled browser like Netscape / Microsoft Internet Explorer.

XI. Distributed

Java is designed as a distributed language for creating applications on networks.



It has ability to share both data and programs.java applications can open and access remote objects on internet as easily they can do in a local system .This enables multiple programs at multiple remote locations to collaborations and work together s on a single project.

XII. Dynamic

Java was designed to adapt to an evolving environment, therefore the Java compiler is smart and dynamic. If you are compiling a file that depends on other non-compiled files, then the compiler will try to find and compile them also. The compiler can handle methods that are used before they're declared. It can also determine whether a source code has been changed since the last time it was compiled. Java classes that were unknown to a program when it was compiled can still be loaded into it at runtime. For example, a web browser can load applets of other classes without recompilation.

XIII. Conclusion

In this paper, I have studied the Buzzword of Java and also why Java is popular language now days.

Java an Object Oriented, general-purpose programming language. After its birth it became popular because of many reasons like security, robustness and multithreadedness but mainly because of its Buzzword of Architecture Neutral and platform independent. The logic and magic behind its platform independence is "BYTECODE".

References

Books:

1. Stephen Gilbert and Bill McCarty. Object –Oriented Programming in Java 1st Edition, New Delhi-2 : Techmedia
2. E Balagurusamy, Programming with JAVA 2nd Edition, New Delhi: Tata McGraw-Hill Publishing.
3. Herbert Schildt, The Complete Reference Java2 5th Edition McGraw-Hill/Osborne.
4. R. Nageswara Rao, Core JAVA : An Integrated Approach 1st Edition, Dreamtech Press.
5. Herb Schildt, Java: A Beginner's Guide 5th Edition, Tata McGraw-Hill Education Pvt. Ltd.
6. Katherine Sierra, Bert Bates - SCJP Sun Certified Programmer for Java 6th Edition Tata McGraw-Hill
7. Bryan Basham, Kathy Sierra, Bert Bates - Head First Servlets And JSP 2nd Edition, O'reilly.
8. Bruce Eckel, Thinking in Java 4th Edition, Pearson.

Web sites:

1. Java Features available at: <http://www.javatpoint.com>