Java Features/Buzz words

1601-15-733-072

Answer:

The Java features or buzz words are:

Simple:- Java was designed to be easy to learn and use effectively for programmers. Java is not difficult
to master for people with programming experience, especially if they already understand the basic concepts of object-oriented programming.

Secure:- Java has achieved security by confining applets to the Java execution environment and not
allowing it to access other parts of the computer. This is done using the Java Virtual Machine (JVM).
The JVM can contain programs and prevent them from generating side effects outside of the system.

Portable:- Portability is a major aspect of the Internet because there are many different types of computers and operating systems connected to it. The JVM allows the same code to work on all computers.
This is possible through Bytecode. Bytecode is a highly optimized set of instructions designed to be
executed by the JVM. The execution of bytecode by the JVM allows for the creation of portable programs.

Object-Oriented:- Although Java was influenced by its predecessors, it was not designed to be source-
code compatible with any other language. This allowed the Java team to design with a blank slate.

The outcome of this was a clean, usable, pragmatic approach to objects. The object model in Java is
easy to extend while the primitive data types remain as high-performance non-objects.

Robust:- The multiplatformed environment of the Web places extraordinary demands on a program, becuase
the program must execute reliably in a variety of systems. To gain reliability, Java places key restrictions
which forces programmers to find their errors early in the program development. It also frees you

from
worrying about the common causes of errors. Java is a strictly-typed language so it checks the code at
compile time and at run-time. This style of programming also ensures predictability in a program. In
a well written Java program all run-time errors should be managed by your program.

Multithreaded:- Java supports multithreaded programming, which allows you to write programs which can
perform multiple tasks simultaneously. The Java run-time system has an elegant yet sophisticated solution
for multiprocess synchronization that enables programmers to construct smoothly running interactive
systems. This easy approach to multithreading allows the programmer to think about the behavior of the
program instead of the actual multitasking mechanism.

Architecture-Neutral:- An issue for Java designers was code longevity and portability. There was no
guarantee that a program written today will run tomorrow, even on the same machine. Operating system upgrades,
processor upgrades, and changes in core system resources can all combine to make a program malfunction.
Their goal in writing Java was "write once; run anywhere, any time, forever."

Interpreted and High Performance:- Java bytecode was carefully designed so that it would be easy to
translate directly into native machine code for very high performance by using a just-in-time compiler.
Java run-time systems that provide this feature lose none of the benefits of the platform-independent
code.

Distributed:- Java is designed for the distributed environment of the Internet because it handles TCP/IP
protocols. Java supports Remote Method Invocation (RMI). This feature enables a program to invoke methods
across a network.

Dynamic:- Java programs carry substantial amounts of run-time information that is used to verify and

resolve accesses to objects at run-time. This makes it possible to dynamically link code in a safe and expedient manner. This is crucial for the robustness of the Java environment.