

About Java

Liv

2021-02-14

1 Definitions

1.1 Simple

The Java programming language can be described in the sense that it is simple to make a functional program in it.

1.2 Object oriented

Java can be categorized as "Object oriented" because it is an object oriented programming language. This means it is organized around the concept of "objects", which contain data structures, as opposed to software structured around logic or functions.

1.3 Distributed

Java is a distributed language because you have the ability to split a program into many parts across different machines. Examples of technologies that allow you to do this include Java RMI-IIOP and Java IDL API.

1.4 Multithreaded

Java can be categorized as a multithreaded language because it supports multithreading. Multithreading is the task of distributing operations in a single program across multiple threads.

1.5 Dynamic

Java is a dynamic language because the same code written somewhere can be compiled to bytecode and executed on any platform with the JVM. It can also load classes dynamically at runtime.

1.6 Architecture neutral

Java can be categorized as an architecture neutral language because the JVM supports many different architectures, such as ARM and x86.

1.7 Portable

Java is a portable language because of all the different platforms it runs on, with only needing to be compiled to bytecode on one. The JVM can run on Linux, Solaris, Windows, Android, and different macOS platforms (including the M1).

1.8 High performance

Java can be categorized as a high performance language because the Java bytecode is much closer to native code than Python and Ruby, which are interpreted instead of compiled. It is still slightly slower than its C and C++ counterparts, but higher level and easier to apply in most situations.

1.9 Robust

Java is a robust language because it lacks pointers which could cause language security problems. It also has its own garbage collection system for memory to get rid of unused objects, though this comes at the cost of runtime stutters for high memory usage applications.

1.10 Secure

Java is a secure language because Java programs run inside the JVM, which is essentially a sandbox. The Java bytecode verifier also checks code fragments for "illegal" code which could break out of the permission checks which Java has, though this can be disabled for experimental reasons, such as bytecode obfuscation. Java also does not support pointers which also prevents security issues.

2 Advantages over other languages

- Java is high performance.
- Java makes it easy to apply design concepts.
- Java is secure.
- Java is cross platform.
- Java is simple and easy to use.

3 Longevity over other languages

Java is slowly being replaced by other JVM languages in more niche applications, but for the most part is still used to maintain existing applications which are written in Java. Java is still the choice for some because it is maintainable, safe and has many libraries and bindings for cross-operation across different programming languages, which makes it easy to integrate into existing systems.

Developers preferred Java alternatives. StackOverflow's 2020 developer survey reported that 7.8% of developers have used Kotlin, while 40.2% have used Java. This is a stark contrast compared to how many developers have said they enjoy writing Kotlin, which is 62.9%, much higher than Java's 44.1%.