Compiler and Bytecode

1. Java is the most popular programming language

Did you know that *Java* is the world's most popular programming language?

According to the TIOBE ranking, the *Java* programming language is used by more than **17%** of all programmers in the world. *C* comes in second with **16%**. 20 years ago, when the Java language just appeared, the undisputed leader was C++, but now its share is less than 7%.

Java appeared in the mid-1990s and quickly gained popularity. Programmers have switched from C ++ to Java by the thousands. Which only confirms once again that **Java is a very cool programming language**.

So what's so cool about it? What features did its creators give it?

You will be very surprised if you compare Java and C ++: Java is very similar to a heavily trimmed C++!

Yes, the Java language is to some extent a heavily stripped down C++. If C++ lets you do something in 20 ways, then Java lets you do it in only one way. So what is the advantage here, you ask?

Well, today's programs are very large, programmers often spend up to 90% of their time working to understand code written by other people. And only 10% is spent on writing new code. So yes, simplicity is an advantage.

---

2. Java compiler

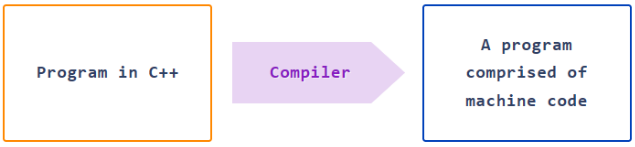
By the way, you will hear more than once that Java's unmatched advantage is its **platform independence**. What is that and what's it for, you ask? Let's start from the beginning.

A computer can execute only the simplest commands.

When training dogs, we use commands like 'Heel', 'Shake', etc. to make a dog do something we want it to do. For computers, numbers play the role of such commands: each command is encoded by a certain number (it is also called a machine code).

But it's really hard to write a program using only numbers, so people invented *programming languages* and *compilers*. A programming language can be understood by both humans and compilers. A *compiler* is a special program that converts a program written in a programming language into a series of machine codes.

A programmer usually writes a program in a programming language and then runs a compiler, which turns the program code files written by the programmer into a single file with machine code – the final (compiled) program.

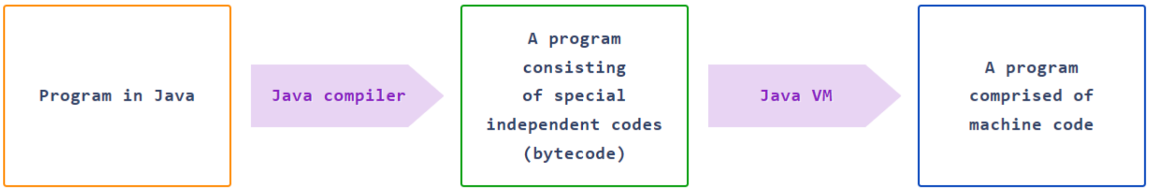


Compilation stages for the C++ language

The resulting program can be executed by the computer immediately. The bad news is that the code of the final program strongly depends on the processor and operating system. This means that a program compiled for Windows will not work on an Android smartphone.

If you write a program for *Android*, then *it will not run* on
the Windows operating system!

However, *Java* uses a much more innovative approach.


Program in Java
Java compiler
A program consisting of special independent codes (bytecode)
Java VM
A program comprised of machine code

Compilation stages for the Java language

A Java compiler doesn't compile all the classes into one machine-code program. Instead, it compiles every class independently and, what's more, not into machine code, but into a special intermediate code (bytecode). The bytecode is compiled into machine code when the program is started.

So, who compiles the program into machine code when it's being executed?

There is a special program for this called the Java virtual machine (JVM). It gets launched first, and then the program comprised of bytecode. Then the JVM will compile the bytecode into machine code before the program is executed.

It's a very powerful approach and one of the reasons for Java's total domination.

---

3. Areas where Java dominates

The advantages described above allow programs written in *Java* to be run on almost any device — computers, smartphones, ATMs, toasters, and credit cards.

This approach has many advantages. It's why *Android* programs are also written in *Java*. Thanks to the quick growth of the mobile phone industry, Java dominates the following areas of programming:

1. *Enterprise*: heavy server-oriented applications for banks, corporations, investment funds, etc.
2. *Mobile*: mobile development (smartphones, tablets), thanks to Android.
3. *Web*: PHP is in the lead, but Java has captured a solid share of the market.
4. *Big Data*: distributed computing in clusters consisting of thousands of servers.
5. *Smart Devices*: programs for smart homes, electronic devices, IoT refrigerators, etc.

Java is not just a language, but a whole ecosystem: millions of ready-made modules that you can use in your program. Thousands of online communities and message boards where you can get help or advice.

The more you write programs in Java, the more answers you will find to the question 'Why Java?'.