About Java

Java is a strongly typed object-oriented programming language. Created in 1995, it has received many improvements since then. Steadily occupies a leading position in rankings of programming languages as well as in rankings of the salaries of software developers.

And most importantly, Java's ranking doesn't jump around from year to year — it remains consistently high. Let's try to understand what makes Java popular.

1. **Cross-platform** — The written code gets converted into bytecode, which then gets executed by the JVM. There are JVM implementations for a variety of platforms. That means that code, once written, will run on Windows, Linux, and macOS, and even various exotic platforms, such as Arduino, smart refrigerators, and vacuum cleaners. In other words, the code will work on different platforms and operating systems, without any need to be adapted for each of them.

2. **Automatic memory management** — The developer doesn't need to think about where the variables are stored in RAM, manually read/write them, or worry about data integrity. When solving a customer's business problem, you ought to think about the problem, not about how and where to write bytes.

3. **Speed (JIT compiler)** — In addition to static compilation, which happens "in advance", Java supports just-in-time (JIT) compilation. This is highly relevant for server code, which can run for months or even years at a time. Frequently-executed code is compiled in different ways and its execution time is measured. The result is that the longer an application runs, the faster it becomes. And that's true for an actively running server. Cool, right?

4. **Backward compatibility** — Code written in older versions of Java will work on newer versions as well. This is convenient: after the Java spec gets updated, you don't have to rewrite half of your project "due to the update", but you can still get the latest security patches.

5. **Object orientation** — Humans think in terms of objects: a table, a trolleybus, a smartphone. Developers aren't forced to think in unusual paradigms as they work, and that helps keep unnecessary errors out of our code. Instead, we can concentrate on only those details that are important in the task. For example, from an interior designer's perspective, a table's size and location in a room are important. Its manufacture date, the name of the worker who cut the wood for the table, and the phone number of the FedEx driver who delivered it are not important. In addition, data and methods for working with that data are stored together in the code.

6. **Static typing (fail fast)** — The compatibility of variable types is checked during the compilation stage. Every developer compiles code, so compilation errors are caught almost instantly. The later the stage at which an error is discovered, the more expensive it is to fix it.

7. **Code as documentation** — Java reads like sentences in English. Accordingly, in most cases, there's no need to spend efforts to maintain documentation, since any developer, having looked at the code, will understand what a method does or what behavior an interface is responsible for. What's more, there are intelligent conventions regarding "proper" naming of all entities in code. A method's name often makes clear what it does.

For example, the getContext() method returns the context, and the age field is responsible for storing the age. In Java, the length of names doesn't affect the amount of system resources required to work with entities. C doesn't handle this as well: when a developer joins a new project, instead of grasping the logic of the code, he or she must decipher it.

8. **Lots of open source libraries and frameworks** — 99% of the tasks that a developer faces in daily practice have already been solved by someone. Over time, the good solutions grow into libraries and even frameworks. Which is better — googling for 5 minutes or reinventing your own bicycle with square wheels?

9. **Big community** — This popular language has a huge developer base that asks a lot of questions on the Internet, gives a lot of answers, writes a lot of code, and faces and solves many problems. And the more developers there are, the more popular the language, and the faster it grows. It's a virtuous cycle.

Many of Java's positive "qualities" have already been mentioned, but I would like to add a few more:

- The JVM (Java Virtual Machine) manages memory for you, which makes it safe and the #1 language for financial tools.
- It's very convenient to write the backend server (server logic) in Java.
- Until recently, Java was the top language for Android applications.

  Kotlin, a JVM language that differs from Java by "syntactic sugar" and a couple of features, now takes its place. Switching from Java to Kotlin and vice versa will take several days. And given that the Java update cycle is now six months, the next Java release may have all the hyped stuff that Kotlin was written for.
- Many famous companies use Java: Google, Facebook, Twitter, Amazon, LinkedIn, eBay, CodeGym and many, many more.