Java magazine

JAVA 25TH ANNIVERSARY

The 25 greatest Java apps ever written

From space exploration to genomics, from reverse compilers to robotic controllers, Java is at the heart of today’s world. Here are a few of the countless Java apps that stand out from the crowd.

by Alexa Morales

June 5, 2020

Download a PDF of this article

The story of Java began in 1991, at a time when Sun Microsystems sought to extend their lead in the computer workstation market into the burgeoning personal electronics market. Little did anyone know that the programming language Sun was about to create would democratize computing, inspire a worldwide community, and become the platform for an enduring software development ecosystem of languages, runtime platforms, SDKs, open source projects, and lots and lots of tools. After a few years of secret development led by James Gosling, Sun released the landmark “write once, run anywhere” Java platform in 1995, refocusing it beyond its original design for interactive television to applications for the burgeoning World Wide Web. By the turn of the century, Java was animating everything from smartcards to space vehicles.

Today, millions of developers program in Java. Although Java continues to evolve at an ever-faster pace, on the occasion of the platform's 25th anniversary, *Java Magazine* decided to take a look back at how Java molded our planet.

What follows is a list of the 25 most ingenious and influential Java apps ever written, from Wikipedia Search to the US National Security Agency's Ghidra. The scope of these applications runs the gamut: space exploration, video games, machine learning, genomics, automotive, cybersecurity, and more.

This list, in no particular order, is far from exhaustive. If you think I've left off anything obvious, set the record straight! On Twitter, tweet to @OracleJavaMag with the hashtags #MovedbyJava and #Top25JavaApps. Or send an email to javamag_us@oracle.com.

On a personal note, I started my tenure as editor in chief of *Software Development* magazine in 2000, when the San Francisco Bay Area's savvy new Java developers were making headlines. I remember devouring the first edition of *Head First Java*, by Kathy Sierra and Bert Bates. Their very visual explanations made clear not only the language syntax but also the object-oriented programming concepts that led to Java's success. Little did I know then that, fully 20 years later, Java would remain a powerful force in my own career, now at Oracle.

To those on the Oracle Java development team (many of whom worked at Sun before Oracle acquired the company in 2010) who helped develop this list: Thank you. My thanks also to Andrew Binstock, the former editor in chief of *Java Magazine*; my colleague Margaret Harrist, who in 2006 was the content marketing strategist for Sun's software division; and Java community members such as you, many of whom I talked with while doing this research, including Jeanne Boyarsky, Sharat Chander, Aurelio García-Ribeyro, Manish Gupta, Manish Kapur, Stuart Marks, Mani Sarkar, Venkat Subramaniam, and Dalibor Topic.

And now for the top 25 greatest Java apps ever written…

Final frontier

1. Maestro Mars Rover controller. In 2004, Java became the first programming language to expand humanity's planetary reach. For three months that year, NASA scientists at the Jet Propulsion Laboratory (JPL) in Pasadena, California, used the Java-based Maestro Science Activity Planner built by JPL's robot interface lab to control the Spirit Mars Exploration Rover as it explored the red planet. Experimentation with Java had begun many years earlier at JPL via the creation of a command and control system for the 1995 Mars Sojourner. Java founder James Gosling spent so much time at JPL that he became an advisory board member.

2. JavaFX Deep Space Trajectory Explorer. Planning a space flight? You may need tools from a.i.solutions, a US aerospace contractor whose products and engineering services have been used by defense companies and civilian space agencies for more than 20 years.

The company's JavaFX Deep Space Trajectory Explorer lets trajectory designers calculate deep space three-body system paths and orbits. The application can generate multidimensional views and models for any planet-moon system or asteroid and filter millions of points in a dense visual search.

3. NASA WorldWind. The work of rocket scientists became free for all to use with NASA's release of the open source WorldWind, an SDK for a virtual globe that lets programmers add the US space agency's geographic rendering engine to their own Java, web, or Android apps. Going far beyond Google Earth, WorldWind's geospatial data is generated by NASA engineers who visualize terrain from elevation models and other data sources. According to the website: "Organizations around the world use WorldWind to monitor weather patterns, visualize cities and terrain, track vehicle movement, analyze geospatial data, and educate humanity about the Earth."

4. JMARS and JMoon. Publicly available since 2003 and still commonly used by NASA scientists, Java Mission-planning and Analysis for Remote Sensing (JMARS) is a geospatial information system written by the people at Arizona State University's Mars Space Flight Facility. JMARS for the Moon (called JMoon by lunar scientists) analyzes wide-angle camera images from the Lunar Reconnaissance Orbiter, a robotic spacecraft that, since its launch in 2009, has been orbiting the moon at an altitude of 50 to 200 kilometers and sending its observations to NASA's Planetary Data System.

5. Small Body Mapping Tool (SBMT). Popular among space scientists and developed at the Johns Hopkins University Applied Physics Laboratory, SBMT uses data from spacecraft missions to visualize irregular bodies such as asteroids, comets, and small moons in 3D. SBMT is written in Java and uses the open source Visualization Toolkit (VTK) for 3D graphics in Java. Flight mission teams for Dawn, Rosetta, OSIRIS-REx, and Hayabusa2 have all used SBMT as they explored comets, asteroids, and a dwarf planet.

Data intensity

6. Wikipedia Search. It's fitting that an encyclopedia for the people, by the people should run on open source software—and feature a search engine powered by Java. Lucene, written by Doug Cutting in 1999 and named after his wife's middle name, was actually the fifth search engine Cutting developed. He created the others as an engineer for Xerox PARC, Apple, and Excite. In 2014, Wikipedia replaced the Lucene engine with Elasticsearch, a distributed, REST-enabled search engine also written in Java.

7. Hadoop. Lucene isn't the only Cutting creation to make our list. Inspired by a Google research paper describing the MapReduce algorithm for processing data on large clusters of commodity computers, in 2003 Cutting wrote an open source framework for MapReduce operations in Java and named it Hadoop, after his son's toy elephant. Hadoop 1.0 was released in 2006, spawning the big data trend and inspiring many companies to collect "data lakes," strategize on mining their "data exhaust," and describe data as "the new oil." By 2008, Yahoo (where Cutting worked at the time) claimed their Search Webmap, running on a Linux cluster of 10,000 cores, was the largest production Hadoop application in existence. By 2012, Facebook claimed to have more than 100 petabytes of data on the world's largest Hadoop cluster.

8. Parallel Graph AnalytiX (PGX). Graph analysis is about understanding relationships and connections in data. PGX is one of the world's fastest graph analytics engines, according to benchmarks. Written in Java and first publicized in 2014 by a team led by Oracle Labs researcher Sungpack Hong, PGX lets users load up graph data and run analytics algorithms such as community detection, clustering, path finding, page ranking, influencer analysis, anomaly detection, path analysis, and pattern matching on them. Use cases abound in health, security, retail, and finance.

9. H2O.ai. Machine learning (ML) has a steep curve—and that can keep domain experts from implementing great ML ideas. Automated ML (AutoML) helps by inferring some of the steps in the ML process, such as feature engineering, model training and tuning, and interpretation. The open source, Java-based H2O.ai platform created by Java Champion Cliff Click aims to democratize AI and act as a virtual data scientist for those just getting started, as well as to help ML experts become more efficient.

Worlds of fun

10. Minecraft. The peaceful environment of this game—comprising biomes, people, and abodes that you build yourself out of blocks—holds an enduring fascination for children and adults everywhere, making it the most popular video game in history. Developed in Java by Markus "Notch" Persson and released in alpha in 2009, Minecraft and its 3D universe are a never-ending source of creativity, because no two spawned worlds are alike. The video game's use of Java also lets programmers at home and school create their own mods.

11. Jitter robot and leJOS. Before the autonomous Roomba vacuum cleaner, there was the Jitter. A prototype robot built to suck up particles floating in the International Space Station (ISS), Jitter was able to navigate in zero gravity, bouncing off walls and using a gyroscope to orient itself. According to one report, Russian cosmonauts found the robot's *x*, *y*, *z* rotation handling impressively reminiscent of how the ISS itself controlled its orientation. Jitter is the most out-of-this-world prototype of leJOS, the Java Virtual Machine for Lego Mindstorms, Lego's hardware-software environment for developing programmable robots from toy building blocks. The toy OS dates back to the TinyVM project of 1999 started by José Solorzano, which morphed into leJOS and was led by Brian Bagnall, Jürgen Stuber, and Paul Andrews. This full-featured environment has many robotics programming-specific classes simplified by the object-oriented nature of Java, letting anyone take advantage of advanced controllers and behavioral algorithms.

12. Java applets. The word *applet* first appeared in 1990 in *PC Magazine*, according to the *Oxford English Dictionary*. But applets didn't truly take off until Java came along in 1995. Java applets were small programs that could launch in a web page (in a frame, a new window, Sun's AppletViewer, or a testing tool) and run on a JVM separate from the browser. Some credit Minecraft's early success to the fact that players could play in a web browser via Java applets rather than having to download and install the game. Although Java applets have been deprecated since Java 9 and they were removed in 2018 from Java SE 11, for a time they were the fastest game in town. Fun fact: Java applets could also access 3D hardware acceleration, making them popular for scientific visualizations.

Code of honor

13. NetBeans and the Eclipse IDE. The integrated development environment that first entranced the Java world was NetBeans, started at Charles University in Prague in 1996 (under the name Xelfi) and commercialized in 1997 by a company of the same name founded by entrepreneur Roman Staněk. Sun bought the modular IDE, which supports all Java application types, in 1999 and open sourced it the following year. In 2016, Oracle donated the entire NetBeans project to the Apache Software Foundation.

Another popular Java-based integrated development environment is the open source Eclipse IDE, which can be used not only for Java coding but also for other programmling languages ranging from Ada to Scala. Launched in 2001 by IBM and based on IBM VisualAge, the Eclipse SDK is meant for Java developers but can be extended via plugins. The Eclipse IDE was spun out of IBM into the Eclipse Foundation in 2004, and it remains one of the top IDEs available.

14. IntelliJ IDEA. There are many IDEs, but IntelliJ IDEA became a favorite after its 2001 introduction. Today, IntelliJ IDEA is the framework for many IDEs for languages as diverse as Python, Ruby, and Go. Written in Java, IntelliJ IDEA and the suite of related JetBrains IDEs add productivity and navigation features that many developers have come to rely upon. These include code indexing, refactoring, code completion (which predates the text autocomplete on smartphones), and dynamic analysis that spots errors (similar to a spellchecker). “IntelliJ IDEA has helped overcome the challenges of managing and debugging Java and JVM-based complex applications under one framework,” says Java Champion Mani Sarkar, a freelance software and data engineer based in the UK. “They make developers feel effective, productive, and above all happy when using their tools.”

15. Byte Buddy. Oslo-based software engineer Rafael Winterhalter, creator of the open source Java library Byte Buddy, admits to dedicating his life—sometimes with maddening focus—to what amounts to a small niche. Nonetheless, the fruit of his dedication is wildly popular: The Byte Buddy runtime code generation and manipulation library, used in Java tools such as Hibernate and Mockito, is downloaded 20 million times a month, says Winterhalter.

16. Jenkins. Created in 2004 by Sun Microsystems engineer Kohsuke Kawaguchi, Jenkins is a powerful open source continuous integration server. Written in Java, Jenkins helps build, test, and deploy applications quickly and automatically. It’s often recognized as one of the early DevOps tools that made “infrastructure as code” possible. Jenkins and its more than 1500 community-contributed plugins tackle myriad deployment and testing tasks, from working with GitHub, to supporting color-blind developers, to providing a MySQL Connector JAR file.

17. GraalVM. A team of researchers in Zurich, led by Oracle Labs’ Thomas Wuerthinger, has spent years honing three ideas: What if you wrote a compiler in Java (the original JVM is written in C)? What if it could run any program written in any language? And what if it were blazingly efficient? Sixty published research papers later, GraalVM beat the odds and became a commercial product. Among the enthusiastic fans of this technology is

Twitter, which uses GraalVM to improve their service's speed and compute efficiency.

18. Micronaut. Developers who code for the cloud need to think carefully about how much memory their application uses—and how the app uses that memory. "You have to make the app very responsive to restarts, and failover, and shutting down and coming back up again, and making it optimized in terms of startup time and memory consumption," says Graeme Rocher, creator of Micronaut, a Java framework for microservices that uses annotation metadata so the JVM can compile the app's bytecode efficiently.

19. WebLogic Tengah. In 1997, WebLogic Tengah became the first substantial implementation of an enterprise Java server. "It predated Java 2 Enterprise Edition and became BEA's principal offering, which eventually led to Oracle's acquisition of BEA Systems," says Andrew Binstock, former editor in chief of *Java Magazine* and of *Dr. Dobb's Journal*. Around the same time, IBM's success with the San Francisco Project, a business object framework, "was what really moved Java from a new, interesting thing that the cool kids were playing with and made it a serious business tool," Binstock says. Today, Oracle WebLogic Server remains the leading Java application server. An alternative still thrives, however: The open source application server GlassFish, started at Sun in 2005, was donated in 2018 to the Eclipse Foundation.

20. Eclipse Collections. There's a reason many high-paying developer jobs at investment banks, stock exchanges, and other financial services companies require strong Java skills: The programming language is adept at handling concurrency, the management of multiple execution threads common in high-frequency trading and other large-scale financial transactions. Originally known as Goldman Sachs Collections and later donated to the Eclipse Foundation, Eclipse Collections expands on native Java's performance-intensive features "with optimized data structures and a rich, functional, and fluent API," says Java Champion Mani Sarkar. Sarkar notes that Eclipse Collections contains caching, primitives support, concurrency libraries, common annotations, string processing, input/output, and more.

21. NSA Ghidra. At the 2019 RSA Conference in San Francisco, the U.S. National Security Agency introduced a Java-based open source tool called Ghidra, which security researchers and practitioners now use to understand how malware works and check their own code for weaknesses. This reverse-engineering platform can decompile software from machine language back into source code (languages such as Java). The tool has a storied, if not infamous, history: Its existence was revealed to the public in March 2017 via WikiLeaks.

Mapping the genome

22. Integrated Genome Browser. The race to map the human genome started in 1990 and ended 13 years later, when medical researchers succeeded in sequencing the 3 billion DNA base pairs of biotechnologist Craig Venter, after a decade of work involving 3,000 people at a cost of US$3 billion. Once the sequence was completed, scientists were raring to dive into our species' source code—but how? Enter the Java-based genome browser, a visualization tool developed by a team including bioinformatics professor Ann Loraine, to explore both the foundational datasets and the reference gene annotations. The open source Integrated Genome Browser lets researchers zoom into, pan, and graph genomic data in order to identify and annotate genetic features. In keeping with this worldwide effort, a similar tool comes from the University of California Santa Cruz in the form of the Genome Browser, managed by Jim Kent.

23. BioJava. Launched in 2000 and still going strong, BioJava is an open source library for processing biological data, a field known as bioinformatics. Scientists using this library can work with protein and nucleotide sequences and study data on gene-to-protein translation, genomics, phylogenetic development, and macromolecular structures. The project is supported by the Open Bioinformatics Foundation, and its contributors worldwide are funded by a variety of pharmaceutical, medical, and genomics interests. "BioJava is a popular option for method and software development thanks to the tooling available for Java and its cross-platform portability," writes Aleix Lafita and colleagues in a 2019 paper titled "BioJava 5: A community-driven open source bioinformatics library." The paper goes on to note that BioJava had accepted contributions from 65 different developers since 2009, and over the previous year had accumulated 224 forks and 270 stars on GitHub and had been downloaded more than 19,000 times.

Favorite 'things'

24. VisibleTesla. This Java-based app was created by Tesla automobile enthusiast Joe Pasqua in 2013 as a free program to monitor and control his Tesla Model S. VisibleTesla, inspired by the Tesla Motors Club community, offers features similar to those found on the electric car maker's official mobile app. Users can set up geofencing and notifications for things such as an unlocked door or the state of charge, as well as collect and manipulate trip data. The project's open source code is on GitHub.

25. SmartThings. The Internet of Things (IoT) application developed by SmartThings, cofounded in 2012 by Alex Hawkinson and later funded with US$1.2 million raised on

Kickstarter, lets you control and automate everything from household lights, locks, coffee makers, thermostats, and radios to home security systems—all through a smartphone or tablet. The application uses the Java-based Micronaut framework (see #18) so its cloud-based services can operate at subsecond speeds. The company was acquired by Samsung Electronics in 2014 for a cool US$200 million.

Alexa Morales

Alexa Weber Morales is director of developer content at Oracle. Morales is the former editor in chief of *Software Development* magazine and has more than 15 years of experience as a technology content strategist and journalist.

Share this Page

Contact

US Sales: +1.800.633.0738
Global Contacts
Support Directory
Subscribe to Emails

About Us

Careers
Communities
Company Information
Social Responsibility Emails

Downloads and Trials

Java for Developers
Java Runtime Download
Software Downloads
Try Oracle Cloud

News and Events

Acquisitions
Blogs
Events
Newsroom

ORACLE | Integrated Cloud
Applications & Platform Services

© Oracle | Site Map | Terms of Use & Privacy | Cookie Preferences | Ad Choices