

Java version history

The Java language has undergone several changes since JDK 1.0 as well as numerous additions of classes and packages to the standard library. Since J2SE 1.4, the evolution of the Java language has been governed by the Java Community Process (JCP), which uses *Java Specification Requests* (JSRs) to propose and specify additions and changes to the Java platform. The language is specified by the *Java Language Specification* (JLS); changes to the JLS are managed under JSR 901 (<http://www.jcp.org/en/jsr/detail?id=901>). In September 2017, Mark Reinhold, chief Architect of the Java Platform, proposed to change the release train to "one feature release every six months" rather than the then-current two-year schedule.^{[1][2]} This proposal took effect for all following versions, and is still the current release schedule.

In addition to the language changes, other changes have been made to the Java Class Library over the years, which has grown from a few hundred classes in JDK 1.0 to over three thousand in J2SE 5. Entire new APIs, such as Swing and Java2D, have been introduced, and many of the original JDK 1.0 classes and methods have been deprecated. Some programs allow conversion of Java programs from one version of the Java platform to an older one (for example Java 5.0 backported to 1.4) (see Java backporting tools).

Regarding Oracle Java SE Support Roadmap,^[3] version 20 is the latest one, and versions 17, 11 and 8 are the currently supported long-term support (LTS) versions, where Oracle Customers will receive Oracle Premier Support. Java 8 LTS the last free software public update for commercial use was released by Oracle in March 2022, while Oracle continues to release no-cost public Java 8 updates for development^[3] and personal use indefinitely.^[4] Java 7 is no longer publicly supported. For Java 11, long-term support will not be provided by Oracle for the public; instead, the broader OpenJDK community, as Eclipse Adoptium or others, is expected to perform the work.^[5]

Java 17 the latest (3rd) LTS was released on September 14, 2021.^[6]

Release table

Version	class file format version ^[7]	Release date	End of Free Public Updates ^{[8][9][10][11][12][13][14]}	Extended Support Until
JDK 1.0	45	23rd January 1996	May 1996	—
JDK 1.1	45.3	2nd February 1997	October 2002	?
J2SE 1.2	46	4th December 1998	September 2003	?
J2SE 1.3	47	8th May 2000	October 2010	?
J2SE 1.4	48	13th February 2002	October 2008	February 2013
Java SE 5	49	29th September 2004	November 2009	April 2015
Java SE 6	50	11th December 2006	April 2013	December 2018 for Oracle ^[8] December 2026 for Azul ^[11]
Java SE 7	51	28th July 2011	September 2022 for OpenJDK Maintained by Oracle until May 2015 ^[15] , Red Hat until August 2020 ^[16] and Azul until September 2022 ^[17]	July 2022 for Oracle ^[8] June 2020 for Red Hat ^[12] December 2027 for Azul ^[11]
Java SE 8 (LTS)	52	18th March 2014	(OpenJDK currently maintained by Red Hat) ^[18] March 2022 for Oracle (commercial) December 2030 for Oracle (non-commercial) December 2030 for Azul ^[11] May 2026 for IBM Semeru ^[13] At least May 2026 for Eclipse Adoptium ^[9] At least May 2026 for Amazon Corretto ^[10]	December 2030 for Oracle ^[8] November 2026 for Red Hat ^[12]
Java SE 9	53	21st September 2017	March 2018 for OpenJDK	—
Java SE 10	54	20th March 2018	September 2018 for OpenJDK	—
Java SE 11 (LTS)	55	25th September 2018	(OpenJDK currently maintained by Red Hat) ^[19] September 2026 for Azul ^[11] October 2024 for IBM Semeru ^[13] At least October 2024 for Eclipse Adoptium ^[9] At least September 2027 for Amazon Corretto ^[10] At least October 2024 for Microsoft ^{[20][14]}	September 2026 for Oracle ^[8] September 2026 for Azul ^[11] October 2024 for Red Hat ^[12]

Java SE 12	56	19th March 2019	September 2019 for OpenJDK	—
Java SE 13	57	17th September 2019	(OpenJDK currently maintained by Azul) ^[21] March 2023 for Azul ^[11]	—
Java SE 14	58	17th March 2020	September 2020 for OpenJDK	—
Java SE 15	59	16th September 2020	(OpenJDK currently maintained by Azul) ^[22] March 2023 for Azul ^[11]	—
Java SE 16	60	16th March 2021	September 2021 for OpenJDK	—
Java SE 17 (LTS)	61	14th September 2021	(OpenJDK currently maintained by SAP) ^[23] September 2029 for Azul ^[11] October 2027 for IBM Semeru ^[13] At least September 2027 for Microsoft ^[14] At least September 2027 for Eclipse Adoptium ^[9]	September 2029 or later for Oracle ^[8] September 2029 for Azul ^[11] October 2027 for Red Hat ^[12]
Java SE 18	62	22nd March 2022	September 2022 for OpenJDK and Adoptium	—
Java SE 19	63	20th September 2022	March 2023 for OpenJDK	—
Java SE 20	64	21st March 2023	September 2023 for OpenJDK	—
Java SE 21 (LTS)	65	September 2023	September 2028	September 2031 for Oracle ^[8]

Legend: Old version Older version, still maintained Latest version Future release

JDK 1.0

The first version was released on January 23, 1996.^{[24][25]} The first stable version, JDK 1.0.2, is called Java 1.^[25]

JDK 1.0

Released	January 23, 1996
-----------------	------------------

JDK 1.1

Major additions in the release on February 19, 1997 included:^[26]

JDK 1.1

Released	February 19, 1997
-----------------	-------------------

- an extensive retooling of the AWT event model
- inner classes added to the language
- JavaBeans
- JDBC
- RMI and serialization
- reflection which supported Introspection only, no modification at runtime was possible. (The ability to modify objects reflectively was added in J2SE 1.2, by introducing the AccessibleObject (<https://docs.oracle.com/en/java/javase/20/docs/api/java.base/java/lang/reflect/AccessibleObject.html>) class and its subclasses such as the Field (<https://docs.oracle.com/en/java/javase/20/docs/api/java.base/java/lang/reflect/Field.html>) class.)

- JIT (Just In Time) compiler on Microsoft Windows platforms, produced for JavaSoft by Symantec
- Internationalization and Unicode support originating from Telligent^[27]

J2SE 1.2

The release on December 8, 1998 and subsequent releases through J2SE 5.0 were rebranded retrospectively **Java 2** and the version name "J2SE" (Java 2 Platform, Standard Edition) replaced JDK to distinguish the base platform from J2EE (Java 2 Platform, Enterprise Edition) and J2ME (Java 2 Platform, Micro Edition). This was a very significant release of Java as it tripled the size of the Java platform to 1520 classes in 59 packages. Major additions included:^[28]

J2SE 1.2

Codename	Playground
Released	December 8, 1998

- strictfp keyword (by JVM 17 an obsolete keyword, shouldn't be used in new code)
- the Swing graphical API was integrated into the core classes
- Sun's JVM was equipped with a JIT compiler for the first time
- Java plug-in
- Java IDL, an IDL implementation for CORBA interoperability
- Collections framework

J2SE 1.3

The most notable changes in the May 8, 2000 release were:^{[29][30]}

J2SE 1.3

Codename	Kestrel
Released	May 8, 2000

- HotSpot JVM included (the HotSpot JVM was first released in April 1999 for the J2SE 1.2 JVM)
- RMI was modified to support optional compatibility with CORBA
- Java Naming and Directory Interface (JNDI) included in core libraries (previously available as an extension)
- Java Platform Debugger Architecture (JPDA)
- JavaSound
- Synthetic proxy classes

Java 1.3 is the last release of Java to officially support Microsoft Windows 95.^[31]

J2SE 1.4

The February 6, 2002 release was the first release of the Java platform developed under the Java Community Process as JSR 59 (<http://www.jcp.org/en/jsr/detail?id=59>). Major changes included:^{[32][33]}

J2SE 1.4

Codename	Merlin
Released	February 6, 2002
Support ended	
Public	October 2008
Paid	February 2013

- Language changes
 - assert keyword (specified in JSR 41 (<https://web.archive.org/web/20080616233205/http://www.jcp.org/en/jsr/detail?id=41>))
- Library improvements

- Regular expressions modeled after Perl regular expressions
- Exception chaining allows an exception to encapsulate original lower-level exception
- Internet Protocol version 6 (IPv6) support
- Non-blocking I/O (named NIO) (specified in JSR 51 (<http://www.jcp.org/en/jsr/detail?id=51>))
- Logging API (specified in JSR 47 (<http://www.jcp.org/en/jsr/detail?id=47>))
- Image I/O API for reading and writing images in formats like JPEG and PNG
- Integrated XML parser and XSLT processor (JAXP) (specified in JSR 5 (<http://www.jcp.org/en/jsr/detail?id=5>) and JSR 63 (<http://www.jcp.org/en/jsr/detail?id=63>))
- Integrated security and cryptography extensions (JCE, JSSE, JAAS)
- Java Web Start included (Java Web Start was first released in March 2001 for J2SE 1.3) (specified in JSR 56 (<http://www.jcp.org/en/jsr/detail?id=56>))
- Preferences API (`java.util.prefs`)

Public support and security updates for Java 1.4 ended in October 2008. Paid security updates for Oracle customers ended in February 2013.^[34]

Java SE 5

The release on September 30, 2004 was originally numbered 1.5, which is still used as the internal version number. The number was changed to "better reflect the level of maturity, stability, scalability and security of the J2SE".^[35] This version was developed under JSR 176 (<http://www.jcp.org/en/jsr/detail?id=176>).

Java SE 5 entered its end-of-public-updates period on April 8, 2008; updates are no longer available to the public as of November 3, 2009. Updates were available to paid Oracle customers until May 2015.^[3]

Tiger added a number of significant new language features:^{[36][37]}

- Generics: provides compile-time (static) type safety for collections and eliminates the need for most typecasts (type conversion) (specified by JSR 14 (<http://www.jcp.org/en/jsr/detail?id=14>))
- Metadata: also called annotations; allows language constructs such as classes and methods to be tagged with additional data, which can then be processed by metadata-aware utilities (specified by JSR 175 (<http://www.jcp.org/en/jsr/detail?id=175>)))
- Autoboxing/unboxing: automatic conversions between primitive types (such as `int`) and primitive wrapper classes (such as `Integer` (<https://docs.oracle.com/en/java/javase/19/docs/api/java.base/java/lang/Integer.html>))) (specified by JSR 201 (<http://www.jcp.org/en/jsr/detail?id=201>)))
- Enumerations: the `enum` keyword creates a typesafe, ordered list of values (such as `Day.MONDAY`, `Day.TUESDAY`, etc.); previously this could only be achieved by non-typesafe constant integers or manually constructed classes (typesafe enum pattern) (specified by JSR 201 (<http://www.jcp.org/en/jsr/detail?id=201>)))
- Varargs: the last parameter of a method can now be declared using a type name followed by three dots (e.g. `void drawtext(String... lines)`); in the calling code any number of parameters of that type can be used and they are then placed in an array to be passed to the method, or alternatively the calling code can pass an array of that type
- Enhanced for each loop: the `for` loop syntax is extended with special syntax for iterating over each member of either an array or any Iterable (<https://docs.oracle.com/en/java/javase/19/docs/api/java.base/java/lang/Iterable.html>), such as the standard

Java SE 5

Codename	Tiger
Released	September 30, 2004
Support ended	
Public	November 2009
Paid	April 2015

[Collection](https://docs.oracle.com/en/java/javase/19/docs/api/java.base/java/util/Collection.html) (<https://docs.oracle.com/en/java/javase/19/docs/api/java.base/java/util/Collection.html>) classes (specified by [JSR 201](http://www.jcp.org/en/jsr/detail?id=201) (<http://www.jcp.org/en/jsr/detail?id=201>))

- Improved semantics of execution for multi-threaded Java programs; the new [Java memory model](#) addresses issues of complexity, effectiveness, and performance of previous specifications^[38]
- [Static imports](#)

There were also the following improvements to the standard libraries:

- Automatic [stub](#) generation for [RMI](#) objects
- [Swing](#): New [skinnable look and feel](#), called [synth](#)
- The [concurrency utilities](#) (<https://java.sun.com/j2se/1.5.0/docs/guide/concurrency/overview.html>) in package [java.util.concurrent](#) (<https://java.sun.com/j2se/1.5.0/docs/api/java/util/concurrent/package-summary.html>)^[39]
- Scanner class for parsing data from various input streams and buffers

Java 5 is the last release of Java to officially support Microsoft [Windows 98](#) and [Windows ME](#),^[40] while [Windows Vista](#) was the newest version of Windows that Java SE 5 was supported on prior to Java 5 going end-of-life in October of 2009.^[34]

Java 5 Update 5 (1.5.0_05) is the last release of Java to work on [Windows 95](#) (with [Internet Explorer 5.5](#) installed) and [Windows NT 4.0](#).^[41]

Java 5 was first available on Apple Mac OS X 10.4 (Tiger)^[42] and was the default version of Java installed on Apple Mac OS X 10.5 (Leopard).

Public support and security updates for Java 1.5 ended in November 2009. Paid security updates for Oracle customers ended in April 2015.

Versioning change

This version introduced a new versioning system for the Java language, although the old versioning system continued to be used for developer libraries:

Both version numbers "1.5.0" and "5.0" are used to identify this release of the Java 2 Platform Standard Edition. Version "5.0" is the product version, while "1.5.0" is the developer version. The number "5.0" is used to better reflect the level of maturity, stability, scalability and security of the J2SE.

— "Version 1.5.0 or 5.0?", [Java release notes](#)^[43]

This correspondence continued through later releases (Java 6 = JDK 1.6, Java 7 = JDK 1.7, and so on).

Java SE 6

As of the version released on December 11, 2006, Sun replaced the name "J2SE" with **Java SE** and dropped the ".o" from the version number.^[44] Internal numbering for developers remains 1.6.0.^[45]

Java SE 6

Codename	Mustang
Released	November 11, 2006

This version was developed under JSR 270 (<http://www.jcp.org/en/jsr/detail?id=270>).

Support ended

Public

February 2013

During the development phase, new builds including enhancements and bug fixes were released approximately weekly. Beta versions were released in February and June 2006, leading up to a final release that occurred on December 11, 2006.

Major changes included in this version:^{[46][47]}

- Support for older Win9x versions dropped; unofficially, Java 6 Update 7 was the last release of Java shown to work on these versions of Windows. This is believed to be due to the major changes in Update 10.
- Scripting Language Support (JSR 223): Generic API for tight integration with scripting languages, and built-in Mozilla JavaScript Rhino integration.
- Dramatic performance improvements for the core platform,^{[48][49]} and Swing.
- Improved Web Service support through JAX-WS (JSR 224).
- JDBC 4.0 support (JSR 221).
- Java Compiler API (JSR 199): an API allowing a Java program to select and invoke a Java Compiler programmatically.
- Upgrade of JAXB to version 2.0: Including integration of a StAX parser.
- Support for pluggable annotations (JSR 269).^[50]
- Many GUI improvements, such as integration of SwingWorker in the API, table sorting and filtering, and true Swing double-buffering (eliminating the gray-area effect).
- JVM improvements include: synchronization and compiler performance optimizations, new algorithms and upgrades to existing garbage collection algorithms, and application start-up performance.

Java 6 can be installed to Mac OS X 10.5 (Leopard) running on 64-bit (Core 2 Duo and higher) processor machines.^[51] Java 6 is also supported by both 32-bit and 64-bit machines running Mac OS X 10.6 (Snow Leopard).

Java 6 reached the end of its supported life in February 2013, at which time all public updates, including security updates, were scheduled to be stopped.^{[52][53]} Oracle released two more updates to Java 6 in March and April 2013, which patched some security vulnerabilities.^{[54][55]}

Java SE 7

Java 7 is a major update that was launched on July 7, 2011^[57] and was made available for developers on July 28, 2011.^[58] The development period was organized into thirteen milestones; on June 6, 2011, the last of the thirteen milestones was finished.^{[58][59]} On average, 8 builds (which generally included enhancements and bug fixes) were released per milestone. The feature list at the OpenJDK 7 project (<https://openjdk.java.net/projects/jdk7/features/>) lists many of the changes.

Additions in Java 7 include:^[60]

Java SE 7

Codename	Dolphin ^[56]
Released	July 28, 2011
Support ended	
Public	April 2015
Paid	June 2022

- JVM support for dynamic languages, with the new invokedynamic bytecode under JSR-292,^[61] following the prototyping work currently done on the Multi Language Virtual Machine

- Compressed 64-bit pointers^[62] (available in Java 6 with `-XX:+UseCompressedOops`)^[63]
- These small language changes (grouped under a project named Coin):^[64]
 - Strings in `switch`^[65]
 - Automatic resource management in try-statement aka *try-with-resources statement*^[66]
 - Improved type inference for generic instance creation, aka *the diamond operator <>*^[67]
 - Simplified varargs method declaration^[68]
 - Binary integer literals^[69]
 - Allowing underscores in numeric literals^[70]
 - Catching multiple exception types and rethrowing exceptions with improved type checking^[71]
- Concurrency utilities under JSR 166^[72]
- New file I/O library (defined by JSR 203) adding support for multiple file systems, file metadata and symbolic links. The new packages are `java.nio.file`, `java.nio.file.attribute` and `java.nio.file.spi`^{[73][74]}
- Timsort is used to sort collections and arrays of objects instead of merge sort
- Library-level support for elliptic curve cryptography algorithms
- An XRender pipeline for Java 2D, which improves handling of features specific to modern GPUs
- New platform APIs for the graphics features originally implemented in version 6u10 as unsupported APIs^[75]
- Enhanced library-level support for new network protocols, including SCTP and Sockets Direct Protocol
- Upstream updates to XML and Unicode
- Java deployment rule sets^[76]

Lambda (Java's implementation of lambda functions), Jigsaw (Java's implementation of modules), and part of Coin were dropped from Java 7, and released as part of Java 8 (except for Jigsaw, which was released in Java 9).^{[77][78]}

Java 7 was the default version to download on java.com from April 2012 until Java 8 was released.^[79]

Java 7 updates

Oracle issued public updates to the Java 7 family on a quarterly basis^[80] until April 2015 when the product reached the end of its public availability.^[81] Further updates for JDK 7, which continued until July 2022, are only made available to customers with a support contract.^[82]

Java SE 8

Java 8 was released on March 18, 2014,^{[83][84]} and included some features that were planned for Java 7 but later deferred.^[85]

Work on features was organized in terms of JDK Enhancement Proposals (JEPs).^[86]

Java SE 8	
<u>LTS version</u>	
Released	March 18, 2014
# of JEPs	8

- JSR 335, JEP 126: Language-level support for lambda expressions (officially, lambda expressions; unofficially, closures) under Project Lambda^[87] and default methods (virtual extension methods)^{[88][89][90]} which can be used to add methods to interfaces without breaking existing implementations. There was an ongoing debate in the Java community on whether to add support for lambda expressions.^{[91][92]} Sun later declared that lambda expressions would be included in Java and asked for community input to refine the feature.^[93] Supporting lambda expressions also enables functional-style operations on streams of elements, such as MapReduce-inspired transformations on collections. Default methods can be used by an author of an API to add new methods to an interface without breaking the old code using it. Although it was not their primary intent,^[88] default methods can also be used for multiple inheritance of behavior (but not state).
- JEP 174: Project Nashorn (<https://openjdk.java.net/jeps/174>), a JavaScript runtime which can run JavaScript code embedded within applications
- JEP 104: Annotation on Java types (<https://openjdk.java.net/jeps/104>)
- Unsigned integer arithmetic^[94]
- JEP 120: Repeating annotations (<https://openjdk.java.net/jeps/120>)
- JEP 150: Date and time API (<https://openjdk.java.net/jeps/150>)
- JEP 178: Statically-linked JNI libraries (<https://openjdk.java.net/jeps/178>)
- JEP 153: Launch JavaFX applications (<https://openjdk.java.net/jeps/153>) (direct launching of JavaFX application JARs)
- JEP 122: Remove the permanent generation (<https://openjdk.java.net/jeps/122>)

Java 8 is not supported on Windows XP^[95] but as of JDK 8 update 25, it can still be installed and run under Windows XP.^[96] Previous updates of JDK 8 could be run under XP by downloading archived zip format file and unzipping it for the executable. The last version of Java 8 could run on XP is update 251. But its components compatibility starts to break on unsupported OS in early build during Java 8 updates development.

From October 2014, Java 8 was the default version to download (and then again the download replacing Java 9) from the official website.^[97] "Oracle will continue to provide Public Updates and auto updates of Java SE 8, Indefinitely for Personal Users".^[4]

Java SE 9

Java SE 9 was made available on September 21, 2017,^[98] due to controversial acceptance of the current implementation of Project Jigsaw by Java Executive Committee,^[99] which led Oracle to fix some open issues and concerns, and to refine some critical technical questions. In the last days of June 2017, Java Community Process expressed nearly unanimous consensus on the proposed Module System scheme.^[100]

- | | |
|------------------|--------------------|
| Java SE 9 | |
| Released | September 21, 2017 |
| # of JEPs | 9 |
- JSR 376: Modularization of the JDK under Project Jigsaw ([Java Platform Module System](#))^[78]
 - JavaDB was removed from JDK^[101]
 - JEP 193: Variable handles (<https://openjdk.java.net/jeps/193>), define a standard means to invoke the equivalents of various `java.util.concurrent.atomic` and `sun.misc.Unsafe` operations
 - JEP 213: Milling Project Coin (<https://openjdk.java.net/jeps/213>), allow `@SafeVarargs` on private instance methods; Allow effectively-final variables to be used as resources in the `try-with-resources` statement; Allow diamond with anonymous classes if the argument type of the

inferred type is denotable; Complete the removal, begun in Java SE 8, of underscore from the set of legal identifier names; Support for private methods in interfaces

- [JEP 222: jshell: The Java Shell \(Read-Eval-Print Loop\)](https://openjdk.java.net/jeps/222) (<https://openjdk.java.net/jeps/222>): JShell is a REPL command-line interface for the Java language.^[102]
- [JEP 254: Compact Strings](https://openjdk.java.net/jeps/254) (<https://openjdk.java.net/jeps/254>)
- [JEP 263: HiDPI graphics: automatic scaling and sizing](https://openjdk.java.net/jeps/263) (<https://openjdk.java.net/jeps/263>)
- [JEP 266: More concurrency updates](https://openjdk.java.net/jeps/266) (<https://openjdk.java.net/jeps/266>), it includes a Java implementation of Reactive Streams,^[103] including a new Flow class^[104] that included the interfaces previously provided by Reactive Streams^[105]
- [JEP 268: XML catalogs](https://openjdk.java.net/jeps/268) (<https://openjdk.java.net/jeps/268>)
- [JEP 282: jlink: The Java Linker](https://openjdk.java.net/jeps/282) (<https://openjdk.java.net/jeps/282>), create a tool that can assemble and optimize a set of modules and their dependencies into a custom run-time image. It effectively allows to produce a fully usable executable including the JVM to run it
- [JEP 295: Ahead-of-Time Compilation](https://openjdk.java.net/jeps/295) (<https://openjdk.java.net/jeps/295>), ahead-of-time compilation provided by GraalVM

The first Java 9 release candidate was released on August 9, 2017.^[106] The first stable release of Java 9 was on September 21, 2017.^[107]

History

At JavaOne 2011, Oracle discussed features they hoped to release for Java 9 in 2016.^[108] Java 9 should include better support for multi-gigabyte heaps, better native code integration, a different default garbage collector (G1, for "shorter response times")^[109] and a self-tuning JVM.^[110] In early 2016, the release of Java 9 was rescheduled for March 2017^[111] and later again postponed four more months to July 2017.^[112]

Java SE 10

OpenJDK 10 was released on March 20, 2018, with twelve new features confirmed.^[113] Among these features were:

- [JEP 286: Local-Variable Type Inference](https://openjdk.java.net/jeps/286) (<https://openjdk.java.net/jeps/286>)
- [JEP 296: Consolidate the JDK Forest into a Single Repository](https://openjdk.java.net/jeps/296) (<https://openjdk.java.net/jeps/296>)
- [JEP 304: Garbage-Collector Interface](https://openjdk.java.net/jeps/304) (<https://openjdk.java.net/jeps/304>)
- [JEP 307: Parallel Full GC for G1](https://openjdk.java.net/jeps/307) (<https://openjdk.java.net/jeps/307>)
- [JEP 310: Application Class-Data Sharing](https://openjdk.java.net/jeps/310) (<https://openjdk.java.net/jeps/310>)
- [JEP 312: Thread-Local Handshakes](https://openjdk.java.net/jeps/312) (<https://openjdk.java.net/jeps/312>)
- [JEP 313: Remove the Native-Header Generation Tool \(javah\)](https://openjdk.java.net/jeps/313) (<https://openjdk.java.net/jeps/313>)
- [JEP 314: Additional Unicode Language-Tag Extensions](https://openjdk.java.net/jeps/314) (<https://openjdk.java.net/jeps/314>)
- [JEP 316: Heap Allocation on Alternative Memory Devices](https://openjdk.java.net/jeps/316) (<https://openjdk.java.net/jeps/316>)
- [JEP 317: Experimental Java-Based JIT Compiler](https://openjdk.java.net/jeps/317) (<https://openjdk.java.net/jeps/317>)
- [JEP 319: Root Certificates](https://openjdk.java.net/jeps/319) (<https://openjdk.java.net/jeps/319>)
- [JEP 322: Time-Based Release Versioning](https://openjdk.java.net/jeps/322) (<https://openjdk.java.net/jeps/322>)

Java SE 10

Released	March 20, 2018
# of JEPs	12

The first of these JEP 286 *Local-Variable Type Inference*, allows the `var` keyword to be used for local variables with the actual type calculated by the compiler. So we can do:

```
var list = new ArrayList<String>(); // infers ArrayList<String>
var stream = list.stream();           // infers Stream<String>
```

Java SE 11

JDK 11 was released on September 25, 2018 and the version is currently open for bug fixes. It offers LTS, or Long-Term Support. Among others, Java 11 includes a number of new features, such as:^[114]

- JEP 181: Nest-Based Access Control (<https://openjdk.java.net/jeps/181>)
- JEP 309: Dynamic Class-File Constants (<https://openjdk.java.net/jeps/309>)
- JEP 315: Improve Aarch64 Intrinsics (<https://openjdk.java.net/jeps/315>)
- JEP 318: Epsilon: A No-Op Garbage Collector (<https://openjdk.java.net/jeps/318>)
- JEP 320: Remove the Java EE and CORBA Modules (<https://openjdk.java.net/jeps/320>)
- JEP 321: HTTP Client (Standard) (<https://openjdk.java.net/jeps/321>)
- JEP 323: Local-Variable Syntax for Lambda Parameters (<https://openjdk.java.net/jeps/323>)
- JEP 324: Key Agreement with Curve25519 and Curve448 (<https://openjdk.java.net/jeps/324>)
- JEP 327: Unicode 10 (<https://openjdk.java.net/jeps/327>)
- JEP 328: Flight Recorder (<https://openjdk.java.net/jeps/328>)
- JEP 329: ChaCha20 and Poly1305 Cryptographic Algorithms (<https://openjdk.java.net/jeps/329>)
- JEP 330: Launch Single-File Source-Code Programs (<https://openjdk.java.net/jeps/330>)
- JEP 331: Low-Overhead Heap Profiling (<https://openjdk.java.net/jeps/331>)
- JEP 332: Transport Layer Security (TLS) 1.3 (<https://openjdk.java.net/jeps/332>)
- JEP 333: ZGC: A Scalable Low-Latency Garbage Collector (Experimental) (<https://openjdk.java.net/jeps/333>)
- JEP 335: Deprecate the Nashorn JavaScript Engine (<https://openjdk.java.net/jeps/335>)
- JEP 336: Deprecate the Pack200 Tools and API (<https://openjdk.java.net/jeps/336>)

Java SE 11	
	<i>LTS version</i>
Released	September 25, 2018
# of JEPs	17
	Removal(s)
Notable	Java applets, Java Web Start, JavaFX, JavaEE, and CORBA modules

A number of features from previous releases were dropped; in particular, Java applets and Java Web Start are no longer available. JavaFX, Java EE and CORBA modules have been removed from JDK.^[115]

Java SE 12

JDK 12 was released on March 19, 2019. Among others, Java 12 includes a number of new features, such as:^[116]

Java SE 12	
Released	March 19, 2019
# of JEPs	8
	Addition(s)

- [JEP 230: Microbenchmark Suite](https://openjdk.java.net/jeps/230) (<https://openjdk.java.net/jeps/230>)
- [JEP 325: Switch Expressions \(Preview\)](https://openjdk.java.net/jeps/325) (<https://openjdk.java.net/jeps/325>)
- [JEP 334: JVM Constants API](https://openjdk.java.net/jeps/334) (<https://openjdk.java.net/jeps/334>)
- [JEP 340: One AArch64 Port, Not Two](https://openjdk.java.net/jeps/340) (<https://openjdk.java.net/jeps/340>)
- [JEP 341: Default CDS Archives](https://openjdk.java.net/jeps/341) (<https://openjdk.java.net/jeps/341>)
- [JEP 344: Abortable Mixed Collections for G1](https://openjdk.java.net/jeps/344) (<https://openjdk.java.net/jeps/344>)
- [JEP 346: Promptly Return Unused Committed Memory from G1](https://openjdk.java.net/jeps/346) (<https://openjdk.java.net/jeps/346>)

Preview(s) Enhanced switch statements

The preview feature JEP 325 extends the `switch` statement so it can also be used as an expression, and adds a new form of case label where the right hand side is an expression. No break statement is needed. For complex expressions a `yield` statement can be used. This becomes standard in Java SE 14.

```
int ndays = switch(month) {
    case JAN, MAR, MAY, JUL, AUG, OCT, DEC -> 31;
    case APR, JUN, SEP, NOV -> 30;
    case FEB -> {
        if (year % 400 == 0) yield 29;
        else if (year % 100 == 0) yield 28;
        else if (year % 4 == 0) yield 29;
        else yield 28; }
};
```

Java SE 13

JDK 13 was released on September 17, 2019. Java 13 includes the following new features, as well as "hundreds of smaller enhancements and thousands of bug fixes".^[117]

- [JEP 350: Dynamic CDS Archives](https://openjdk.java.net/jeps/350) (<https://openjdk.java.net/jeps/350>)
- [JEP 351: ZGC: Uncommit Unused Memory](https://openjdk.java.net/jeps/351) (<https://openjdk.java.net/jeps/351>)
- [JEP 353: Reimplement the Legacy Socket API](https://openjdk.java.net/jeps/353) (<https://openjdk.java.net/jeps/353>)
- [JEP 354: Switch Expressions \(Preview\)](https://openjdk.java.net/jeps/354) (<https://openjdk.java.net/jeps/354>)
- [JEP 355: Text Blocks \(Preview\)](https://openjdk.java.net/jeps/355) (<https://openjdk.java.net/jeps/355>)

Java SE 13

Released	September 17, 2019
# of JEPs	5
Addition(s)	
Preview(s)	Enhanced <u>switch statements</u> , <u>text blocks</u>

JEP 355 *Text Blocks* allows multiline string literals:

```
String html = """
<html lang="en">
  <body>
    <p>Hello, world</p>
  </body>
</html>
""";
```

Java SE 14

JDK 14 was released on March 17, 2020. Java 14 includes the following new features, as well as "hundreds of smaller enhancements and thousands of bug fixes".^[118]

- [JEP 305: Pattern Matching for instanceof \(Preview\)](https://openjdk.java.net/jeps/305) ([http://openjdk.java.net/jeps/305](https://openjdk.java.net/jeps/305))
- [JEP 343: Packaging Tool \(Incubator\)](https://openjdk.java.net/jeps/343) (<https://openjdk.java.net/jeps/343>)
- [JEP 345: NUMA-Aware Memory Allocation for G1](https://openjdk.java.net/jeps/345) (<https://openjdk.java.net/jeps/345>)
- [JEP 349: JFR Event Streaming](https://openjdk.java.net/jeps/349) (<https://openjdk.java.net/jeps/349>)
- [JEP 352: Non-Volatile Mapped Byte Buffers](https://openjdk.java.net/jeps/352) (<https://openjdk.java.net/jeps/352>)
- [JEP 358: Helpful NullPointerExceptions](https://openjdk.java.net/jeps/358) (<https://openjdk.java.net/jeps/358>)
- [JEP 359: Records \(Preview\)](https://openjdk.java.net/jeps/359) (<https://openjdk.java.net/jeps/359>)
- [JEP 361: Switch Expressions \(Standard\)](https://openjdk.java.net/jeps/361) (<https://openjdk.java.net/jeps/361>)
- [JEP 362: Deprecate the Solaris and SPARC Ports](https://openjdk.java.net/jeps/362) (<https://openjdk.java.net/jeps/362>)
- [JEP 363: Remove the Concurrent Mark Sweep \(CMS\) Garbage Collector](https://openjdk.java.net/jeps/363) (<https://openjdk.java.net/jeps/363>)
- [JEP 364: ZGC on macOS](https://openjdk.java.net/jeps/364) (<https://openjdk.java.net/jeps/364>)
- [JEP 365: ZGC on Windows](https://openjdk.java.net/jeps/365) (<https://openjdk.java.net/jeps/365>)
- [JEP 366: Deprecate the ParallelScavenge + SerialOld GC Combination](https://openjdk.java.net/jeps/366) (<https://openjdk.java.net/jeps/366>)
- [JEP 367: Remove the Pack200 Tools and API](https://openjdk.java.net/jeps/367) (<https://openjdk.java.net/jeps/367>)
- [JEP 368: Text Blocks \(Second Preview\)](https://openjdk.java.net/jeps/368) (<https://openjdk.java.net/jeps/368>)
- [JEP 370: Foreign-Memory Access API \(Incubator\)](https://openjdk.java.net/jeps/370) (<https://openjdk.java.net/jeps/370>)

Java SE 14

Released	March 17, 2020
# of JEPs	16
Addition(s)	
Notable	Helpful NullPointerExceptions , enhanced switch statements
Preview(s)	Pattern matching for instanceof , records , text blocks
Incubating	jpackager, Foreign memory access
Removal(s)	
Notable	Remove Concurrent Mark Sweep garbage collector

JEP 305, *Pattern Matching for instanceof* simplifies the common case of an `instanceof` test being immediately followed by cast, replacing

```
if (obj instanceof String) {
    String s = (String) obj;
    System.out.println( s.length() );
}
```

with

```
if (obj instanceof String s) {
    System.out.println( s.length() );
}
```

JEP 359 *Records* allows easy creation of simple immutable [Tuple](#)-like classes.^[119]

```
record Point(int x, int y) { }
Point p = new Point(3,4);
System.out.println( p.x() );
```

Java SE 15

JDK 15 was released on September 15, 2020. Java 15 adds e.g. support for multi-line string literals (aka Text Blocks). The Shenandoah and Z garbage collectors (latter sometimes abbreviated ZGC) are now ready for use in production (i.e. no longer marked experimental). Support for Oracle's Solaris operating system (and SPARC CPUs) is dropped (while still available in e.g. Java 11). The Nashorn JavaScript Engine is removed. Also removed some root CA certificates.

- [JEP 339: Edwards-Curve Digital Signature Algorithm \(EdDSA\)](https://openjdk.java.net/jeps/339) (<https://openjdk.java.net/jeps/339>)
- [JEP 360: Sealed Classes \(Preview\)](https://openjdk.java.net/jeps/360) (<https://openjdk.java.net/jeps/360>)
- [JEP 371: Hidden Classes](https://openjdk.java.net/jeps/371) (<https://openjdk.java.net/jeps/371>)
- [JEP 372: Remove the Nashorn JavaScript Engine](https://openjdk.java.net/jeps/372) (<https://openjdk.java.net/jeps/372>)
- [JEP 373: Reimplement the Legacy DatagramSocket API](https://openjdk.java.net/jeps/373) (<https://openjdk.java.net/jeps/373>)
- [JEP 374: Disable and Deprecate Biased Locking](https://openjdk.java.net/jeps/374) (<https://openjdk.java.net/jeps/374>)
- [JEP 375: Pattern Matching for instanceof \(Second Preview\)](https://openjdk.java.net/jeps/375) (<https://openjdk.java.net/jeps/375>)
- [JEP 377: ZGC: A Scalable Low-Latency Garbage Collector](https://openjdk.java.net/jeps/377) (<https://openjdk.java.net/jeps/377>)
- [JEP 378: Text Blocks](https://openjdk.java.net/jeps/378) (<https://openjdk.java.net/jeps/378>)
- [JEP 379: Shenandoah: A Low-Pause-Time Garbage Collector](https://openjdk.java.net/jeps/379) (<https://openjdk.java.net/jeps/379>)
- [JEP 381: Remove the Solaris and SPARC Ports](https://openjdk.java.net/jeps/381) (<https://openjdk.java.net/jeps/381>)
- [JEP 383: Foreign-Memory Access API \(Second Incubator\)](https://openjdk.java.net/jeps/383) (<https://openjdk.java.net/jeps/383>)
- [JEP 384: Records \(Second Preview\)](https://openjdk.java.net/jeps/384) (<https://openjdk.java.net/jeps/384>)
- [JEP 385: Deprecate RMI Activation for Removal](https://openjdk.java.net/jeps/385) (<https://openjdk.java.net/jeps/385>)

JEP 360 *Sealed Classes* adds sealed classes and interfaces that restrict which other classes or interfaces may extend or implement them. Only those classes specified in a `permits` clause may extend the class or interface.

```
package com.example.geometry;

public abstract sealed class Shape
    permits Circle, Rectangle, Square {...}
```

Together with records, sealed classes are sum types. They work well with other recent features like records, switch expressions, and pattern matching for instance-of. They all form part of a system for "Pattern matching in Java" first discussed by Gavin Bierman and Brian Goetz, in September 2018.^[120]

Java SE 16

Java SE 16

JDK 16 was released on March 16, 2021. Java 16 removes Ahead-of-Time compilation (and Graal JIT) options.^[121] The Java implementation itself was and is still written in C++, while as of Java 16, more recent C++14 (but still not e.g. C++17 or C++20) is allowed. The code was also moved to GitHub, dropping Mercurial as the source control system.

- [JEP 338: Vector API \(Incubator\)](https://openjdk.java.net/jeps/338) (<https://openjdk.java.net/jeps/338>)
- [JEP 347: Enable C++14 Language Features](https://openjdk.java.net/jeps/347) (<https://openjdk.java.net/jeps/347>)
- [JEP 357: Migrate from Mercurial to Git](https://openjdk.java.net/jeps/357) (<https://openjdk.java.net/jeps/357>)
- [JEP 369: Migrate to GitHub](https://openjdk.java.net/jeps/369) (<https://openjdk.java.net/jeps/369>)
- [JEP 376: ZGC: Concurrent Thread-Stack Processing](https://openjdk.java.net/jeps/376) (<https://openjdk.java.net/jeps/376>)
- [JEP 380: Unix-Domain Socket Channels](https://openjdk.java.net/jeps/380) (<https://openjdk.java.net/jeps/380>)
- [JEP 386: Alpine Linux Port](https://openjdk.java.net/jeps/386) (<https://openjdk.java.net/jeps/386>) – not yet stable
- [JEP 387: Elastic Metaspace](https://openjdk.java.net/jeps/387) (<https://openjdk.java.net/jeps/387>)
- [JEP 388: Windows/AArch64 Port](https://openjdk.java.net/jeps/388) (<https://openjdk.java.net/jeps/388>)
- [JEP 389: Foreign Linker API \(Incubator\)](https://openjdk.java.net/jeps/389) (<https://openjdk.java.net/jeps/389>)
- [JEP 390: Warnings for Value-Based Classes](https://openjdk.java.net/jeps/390) (<https://openjdk.java.net/jeps/390>)
- [JEP 392: Packaging Tool](https://openjdk.java.net/jeps/392) (<https://openjdk.java.net/jeps/392>)
- [JEP 393: Foreign-Memory Access API \(Third Incubator\)](https://openjdk.java.net/jeps/393) (<https://openjdk.java.net/jeps/393>)
- [JEP 394: Pattern Matching for instanceof](https://openjdk.java.net/jeps/394) (<https://openjdk.java.net/jeps/394>)
- [JEP 395: Records](https://openjdk.java.net/jeps/395) (<https://openjdk.java.net/jeps/395>)
- [JEP 396: Strongly Encapsulate JDK Internals by Default](https://openjdk.java.net/jeps/396) (<https://openjdk.java.net/jeps/396>)
- [JEP 397: Sealed Classes \(Second Preview\)](https://openjdk.java.net/jeps/397) (<https://openjdk.java.net/jeps/397>)

Released	March 16, 2021
# of JEPs	17
Addition(s)	
Notable	Windows/AArch64 Port, jpackager, pattern matching for instanceof, records
Preview(s)	Sealed classes
Incubating	Foreign linker, Foreign-memory access

Java SE 17

JDK 17 is the current long-term support (LTS) release since September 2021.^[122] Java 17 is the 2nd long-term support (LTS) release since switching to the new 6-month release cadence (the first being Java 11).

- [JEP 306: Restore Always-Strict Floating-Point Semantics](https://openjdk.java.net/jeps/306) (<https://openjdk.java.net/jeps/306>)
- [JEP 356: Enhanced Pseudo-Random Number Generators](https://openjdk.java.net/jeps/356) (<https://openjdk.java.net/jeps/356>)
- [JEP 382: New macOS Rendering Pipeline](https://openjdk.java.net/jeps/382) (<https://openjdk.java.net/jeps/382>)
- [JEP 391: macOS/AArch64 Port](https://openjdk.java.net/jeps/391) (<https://openjdk.java.net/jeps/391>)
- [JEP 398: Deprecate the Applet API for Removal](https://openjdk.java.net/jeps/398) (<https://openjdk.java.net/jeps/398>)
- [JEP 403: Strongly Encapsulate JDK Internals](https://openjdk.java.net/jeps/403) (<https://openjdk.java.net/jeps/403>)
- [JEP 406: Pattern Matching for switch \(Preview\)](https://openjdk.java.net/jeps/406) (<https://openjdk.java.net/jeps/406>)

Java SE 17

LTS version	
Released	September 14, 2021
Addition(s)	
Notable	macOS/AArch64 Port, sealed classes
Preview(s)	
Incubating	Vector API, Foreign function & memory API
Removal(s)	
Notable	AOT compiler, RMI activation, strictfp keyword made obsolete (PEP 306)

- [JEP 407: Remove RMI Activation](https://openjdk.java.net/jeps/407) (<https://openjdk.java.net/jeps/407>)
- [JEP 409: Sealed Classes](https://openjdk.java.net/jeps/409) (<https://openjdk.java.net/jeps/409>)
- [JEP 410: Remove the Experimental AOT and JIT Compiler](https://openjdk.java.net/jeps/410) (<https://openjdk.java.net/jeps/410>)
- [JEP 411: Deprecate the Security Manager for Removal](https://openjdk.java.net/jeps/411) (<https://openjdk.java.net/jeps/411>)
- [JEP 412: Foreign Function & Memory API \(Incubator\)](https://openjdk.java.net/jeps/412) (<https://openjdk.java.net/jeps/412>)
- [JEP 414: Vector API \(Second Incubator\)](https://openjdk.java.net/jeps/414) (<https://openjdk.java.net/jeps/414>)
- [JEP 415: Context-Specific Deserialization Filters](https://openjdk.java.net/jeps/415) (<https://openjdk.java.net/jeps/415>)

JEP 406 extends the pattern matching syntax used in instanceof operations to switch statements and expressions. It allows cases to be selected based on the type of the argument, null cases and refining patterns

```
Object o = ...;
return switch (o) {
    case null      -> "Null";
    case String s  -> String.format("String %s", s);
    case Long l    -> String.format("long %d", l);
    case Double d  -> String.format("double %f", d);
    case Integer i && i > 0           // refining patterns
        -> String.format("positive int %d", i);
    case Integer i && i == 0
        -> String.format("zero int %d", i);
    case Integer i && i < 0
        -> String.format("negative int %d", i);
    default         -> o.toString();
};
};
```

Java SE 18

JDK 18 was released on March 22, 2022.

- [JEP 400: UTF-8 by Default](https://openjdk.java.net/jeps/400) (<https://openjdk.java.net/jeps/400>)
- [JEP 408: Simple Web Server](https://openjdk.java.net/jeps/408) (<https://openjdk.java.net/jeps/408>)
- [JEP 413: Code Snippets in Java API Documentation](https://openjdk.java.net/jeps/413) ([http://openjdk.java.net/jeps/413](https://openjdk.java.net/jeps/413))
- [JEP 416: Reimplement Core Reflection with Method Handles](https://openjdk.java.net/jeps/416) (<https://openjdk.java.net/jeps/416>)
- [JEP 417: Vector API \(Third Incubator\)](https://openjdk.java.net/jeps/417) (<https://openjdk.java.net/jeps/417>)
- [JEP 418: Internet-Address Resolution SPI](https://openjdk.java.net/jeps/418) (<https://openjdk.java.net/jeps/418>)
- [JEP 419: Foreign Function & Memory API \(Second Incubator\)](https://openjdk.java.net/jeps/419) (<https://openjdk.java.net/jeps/419>)
- [JEP 420: Pattern Matching for switch \(Second Preview\)](https://openjdk.java.net/jeps/420) (<https://openjdk.java.net/jeps/420>)
- [JEP 421: Deprecate Finalization for Removal](https://openjdk.java.net/jeps/421) (<https://openjdk.java.net/jeps/421>)

Java SE 18

Released	March 22, 2022
# of JEPs	9
Addition(s)	
Notable	UTF by default Javadoc code snippets
Preview(s)	Switch pattern matching
Incubating	Vector API Foreign function & memory API
Removal(s)	
Notable	Deprecated finalization for removal

Java SE 19

JDK 19 was released on 20 September 2022.^[123]

- [JEP 405: Record Patterns \(Preview\)](https://openjdk.java.net/jeps/405) (<https://openjdk.java.net/jeps/405>)
- [JEP 422: Linux/RISC-V Port](https://openjdk.java.net/jeps/422) (<https://openjdk.java.net/jeps/422>)
- [JEP 424: Foreign Function & Memory API \(Preview\)](https://openjdk.java.net/jeps/424) ([http s://openjdk.java.net/jeps/424](https://openjdk.java.net/jeps/424))
- [JEP 425: Virtual Threads \(Preview\)](https://openjdk.java.net/jeps/425) ([https://openjdk.java.n et/jeps/425](https://openjdk.java.net/jeps/425))
- [JEP 426: Vector API \(Fourth Incubator\)](https://openjdk.java.net/jeps/426) ([https://openjdk.jav a.net/jeps/426](https://openjdk.java.net/jeps/426))
- [JEP 427: Pattern Matching for switch \(Third Preview\)](https://openjdk.java.net/jeps/427) ([http s://openjdk.java.net/jeps/427](https://openjdk.java.net/jeps/427))
- [JEP 428: Structured Concurrency \(Incubator\)](https://openjdk.java.net/jeps/428) (<https://openjdk.java.net/jeps/428>)

JEP 405 allows record patterns, extending the pattern matching capabilities of instanceof operators, and switch expressions, to include record patterns that explicitly refer to the components of the record.

```
record Rectangle(int x, int y, int w, int h) {}

int area(Object o) {
    if (o instanceof Rectangle(int x, int y, int w, int h)) {
        return w * h;
    }
    return 0;
}
```

Such patterns can include nested patterns, where the components of records are themselves records, allowing patterns to match more object graphs.

Java SE 20

Java 20 was released on 21 March 2023.^[124]

- [JEP 429: Scoped Values \(Incubator\)](https://openjdk.java.net/jeps/429) (<https://openjdk.java.net/jeps/429>)
- [JEP 432: Record Patterns \(Second Preview\)](https://openjdk.java.net/jeps/432) (<https://openj dk.java.net/jeps/432>)
- [JEP 433: Pattern Matching for switch \(Fourth Preview\)](https://openjdk.java.net/jeps/433) ([htt ps://openjdk.java.net/jeps/433](https://openjdk.java.net/jeps/433))
- [JEP 434: Foreign Function & Memory API \(Second Preview\)](https://openjdk.java.net/jeps/434) (<https://openjdk.java.net/jeps/434>)
- [JEP 436: Virtual Threads \(Second Preview\)](https://openjdk.java.net/jeps/436) (<https://openjdk.java.net/jeps/436>)
- [JEP 437: Structured Concurrency \(Second Incubator\)](https://openjdk.java.net/jeps/437) (<https://openjdk.java.net/jeps/437>)
- [JEP 438: Vector API \(Fifth Incubator\)](https://openjdk.java.net/jeps/438) (<https://openjdk.java.net/jeps/438>)

Java SE 20

Releasing	March 21, 2023
# of JEPs	7
Addition(s)	
Incubating	Scoped values

Future features

- [Project Valhalla](https://openjdk.java.net/projects/valhalla/) (<https://openjdk.java.net/projects/valhalla/>): *Value types*, objects without identity but with an efficient memory layout and leading to better results of *escape analysis*.
- [Project Panama](https://openjdk.java.net/projects/panama/) (<https://openjdk.java.net/projects/panama/>): *Improved interoperability with native code*, to enable Java source code to call functions and use data types from other

languages, in a way that is easier and has better performance than today. Vector API (a portable and relatively low-level abstraction layer for SIMD programming) is also developed under Project Panama umbrella.

- Project Loom (<https://openjdk.java.net/projects/loom/>): *Virtual threads*, a lightweight user-mode scheduled alternative to standard OS managed threads. Virtual threads are mapped to OS threads in a many-to-many relationship, in contrast to the many-to-one relationship from the original green threads implementation in early versions of Java.

Implementations

The officially supported Java platform, first developed at Sun and now stewarded by Oracle, is Java SE. Releases are based on the OpenJDK project, a free and open-source project with an open development model. Other Java implementations exist, however—in part due to Java's early history as proprietary software. In contrast, some implementations were created to offer some benefits over the standard implementation, often the result of some area of academic or corporate-sponsored research. Many Linux distributions include builds of OpenJDK through the IcedTea project started by Red Hat, which provides a more straightforward build and integration environment.

Visual J++ and the Microsoft Java Virtual Machine were created as incompatible implementations. After the *Sun v. Microsoft* lawsuit, Microsoft abandoned it and began work on the .NET platform. In 2021, Microsoft started distributing compatible "Microsoft Build of OpenJDK" for Java 11 first then also for Java 17. Their builds support not only Windows, but also Linux and macOS.

Other proprietary Java implementations are available, such as Azul's Zing. Azul offers certified open source OpenJDK builds under the Zulu moniker.

Prior to the release of OpenJDK, while Sun's implementation was still proprietary, the GNU Classpath project was created to provide a free and open-source implementation of the Java platform. Since the release of JDK 7, when OpenJDK became the official reference implementation, the original motivation for the GNU Classpath project almost completely disappeared, and its last release was in 2012.

The Apache Harmony project was started shortly before the release of OpenJDK. After Sun's initial source code release, the Harmony project continued, working to provide an implementation under a lax license, in contrast to the protective license chosen for OpenJDK. Google later developed Android and released it under a lax license. Android incorporated parts of the Harmony project, supplemented with Google's own Dalvik virtual machine and ART. Apache Harmony has since been retired, and Google has switched its Harmony components with equivalent ones from OpenJDK.

Both Jikes and Jikes RVM are open-source research projects that IBM developed.

Several other implementations exist that started as proprietary software but are now open source. IBM initially developed OpenJ9 as the proprietary J9^[125] but has since relicensed the project and donated it to the Eclipse Foundation. JRockit is a proprietary implementation that was acquired by Oracle and incorporated into subsequent OpenJDK versions.

References

1. Reinhold, Mark (2017-09-06). "Moving Java Forward Faster" (<https://mreinhold.org/blog/forward-faster>). Retrieved 2017-09-16.

2. "Calling 'all aboard' on the six-month Java release train" (<http://www.theserverside.com/news/450426185/Calling-all-aboard-on-the-six-month-Java-release-train>). theserverside.com. 2017-09-12. Retrieved 2017-09-16.
3. "Oracle Java SE Support Roadmap" (<https://www.oracle.com/technetwork/java/java-se-support-roadmap.html>). www.oracle.com.
4. "Oracle Java SE Support Roadmap" (<https://www.oracle.com/technetwork/java/javase/eol-135779.html>). Retrieved 2021-02-17.
5. Reinhold, Mark (2018-08-17). "What does LTS mean for OpenJDK?" (<https://mail.openjdk.java.net/pipermail/jdk-dev/2018-August/001824.html>). Retrieved 2018-08-28.
6. "JDK 17" (<http://openjdk.java.net/projects/jdk/17/>). openjdk.java.net. Retrieved 2021-08-26.
7. "Chapter 4. The class File Format" (<https://docs.oracle.com/javase/specs/jvms/se19/html/jvms-4.html#jvms-4.1-200-B.2>).
8. "Oracle Java SE Support Roadmap" (<https://www.oracle.com/technetwork/java/java-se-support-roadmap.html>). www.oracle.com.
9. "Support | AdoptOpenJDK" (<https://adoptopenjdk.net/support.html#roadmap>). adoptopenjdk.net.
10. "Amazon Corretto 8 & 11 support extended" (<https://aws.amazon.com/about-aws/whats-new/2020/08/amazon-corretto-8-11-support-extended/>).
11. "Azul Support Roadmap" (https://www_azul_com_products_azul-support-roadmap/). Azul | Better Java Performance, Superior Java Support. Retrieved 2021-07-08.
12. "OpenJDK Life Cycle and Support Policy" (<https://access.redhat.com/articles/1299013>). Retrieved 2022-09-15.
13. "Semeru Runtimes support" (https://www_ibm_com_support_pages_semeru-runtimes-support). www.ibm.com. 2021-09-28. Retrieved 2021-10-15.
14. "Support roadmap for the Microsoft Build of OpenJDK" (<https://docs.microsoft.com/en-us/java/openjdk/support>). docs.microsoft.com. Retrieved 2021-08-03. "Support for OpenJDK 17 on Windows x86 32 bit will end on January 10th, 2023 following the End of Life support of Windows 32 bit versions."
15. "Maintainer Resignation" (<https://mail.openjdk.org/pipermail/jdk7u-dev/2015-May/010307.html>). 2015-05-22. Retrieved 2022-09-07.
16. "JDK 7u Project Lead" (<https://mail.openjdk.org/pipermail/jdk7u-dev/2020-August/011040.html>). 2020-08-24. Retrieved 2022-09-07.
17. "Resigning as JDK 7u Project Lead" (<https://mail.openjdk.org/pipermail/jdk7u-dev/2022-September/011231.html>). 2022-09-05. Retrieved 2022-09-07.
18. "New lead for the JDK 8 Update Releases Project: Andrew Haley" (<https://mail.openjdk.org/pipermail/jdk8u-dev/2019-February/008503.html>). 2019-02-01. Retrieved 2022-09-15.
19. "New Lead Maintainer for the JDK11 Updates repository: Andrew Haley" (<https://mail.openjdk.org/pipermail/jdk-updates-dev/2019-February/000450.html>). 2019-02-18. Retrieved 2022-09-15.
20. "Announcing General Availability of Microsoft Build of OpenJDK" (<https://devblogs.microsoft.com/java/announcing-general-availability-of-microsoft-build-of-openjdk/>). Java at Microsoft. 2021-05-25. Retrieved 2021-08-03.
21. "[13u Communication] New Lead Maintainer for the JDK13 Updates repository: Yuri Nesterenko" (<https://mail.openjdk.org/pipermail/jdk-updates-dev/2020-March/002669.html>). 2020-03-04. Retrieved 2022-09-07.
22. "[15u Communication] Future Lead Maintainer for the JDK15 Updates repository: Yuri Nesterenko" (<https://mail.openjdk.org/pipermail/jdk-updates-dev/2021-February/005084.html>). 2021-02-22. Retrieved 2022-09-07.
23. "[17u-communication] Future Lead Maintainer for the JDK17 Updates repository: Goetz Lindenmaier" (<https://mail.openjdk.org/pipermail/jdk-updates-dev/2021-December/010796.html>). 2021-12-13. Retrieved 2022-09-15.

24. "JavaSoft ships Java 1.0" (<https://web.archive.org/web/20070310235103/http://www.sun.com/sm/Press/sunflash/1996-01/sunflash.960123.10561.xml>). Archived from the original (<http://www.sun.com/sm/Press/sunflash/1996-01/sunflash.960123.10561.xml>) on March 10, 2007. Retrieved 2008-02-05.
25. Ortiz, C. Enrique; Giguère, Éric (2001). *Mobile Information Device Profile for Java 2 Micro Edition: Developer's Guide* (<https://archive.org/details/mobileinformatio0000orti>). John Wiley & Sons. ISBN 978-0471034650. Retrieved May 30, 2012.
26. Version 1.1 press release (<https://web.archive.org/web/20080210044125/http://www.sun.com/sm/Press/sunflash/1997-02/sunflash.970219.0001.xml>), Sun.
27. Tennant, Don (March 15, 1997). "Telligent prepares internationalisation technology for the big time" (<http://www2.computerworld.co.nz/article/669327/twitter-makes-global-changes-comply-privacy-laws/>). *Computerworld*. IDG. Retrieved January 16, 2021.
28. Version 1.2 press release (<https://web.archive.org/web/20070816170028/http://www.sun.com/sm/Press/sunflash/1998-12/sunflash.981208.9.xml>), Sun.
29. Version 1.3 press release (<https://web.archive.org/web/20070817053430/http://www.sun.com/sm/Press/sunflash/2000-05/sunflash.20000508.3.xml>), Sun.
30. "Version 1.3 full list of changes" (<https://web.archive.org/web/20061107051825/https://java.sun.com/j2se/1.3/docs/relnotes/features.html>). Archived from the original (<https://java.sun.com/j2se/1.3/docs/relnotes/features.html>) on November 7, 2006.
31. "How do I run Java on a Windows 95 computer?" (<https://www.java.com/en/download/help/win95.html>).
32. Version 1.4 press release (<https://web.archive.org/web/20070815095726/http://www.sun.com/sm/Press/sunflash/2002-02/sunflash.20020206.5.xml>).
33. "Version full 1.4 list of changes" (<https://web.archive.org/web/20070108015942/https://java.sun.com/j2se/1.4.2/docs/relnotes/features.html>). Archived from the original (<https://java.sun.com/j2se/1.4.2/docs/relnotes/features.html>) on January 8, 2007.
34. "Java 2 Platform 5.0 and Java for Business 5.0" (<https://www.oracle.com/technetwork/java/javase/system-configurations-139801.html>). Oracle Corporation. 2010. Retrieved October 22, 2012.
35. "Version 1.5.0 or 5.0?" (<http://docs.oracle.com/javase/1.5.0/docs/relnotes/version-5.0.html>). Oracle.
36. Version 1.5 press release (<https://web.archive.org/web/20080207083457/http://www.sun.com/sm/Press/sunflash/2004-09/sunflash.20040930.1.xml>).
37. Version 1.5 full list of changes (<https://java.sun.com/j2se/1.5/docs/relnotes/features.html>).
38. "JSR 133, 2.4 Why isn't this need met by existing specifications?" (<https://www.jcp.org/en/jsr/detail?id=133>). Oracle.
39. Goetz, Brian (2006). *Java Concurrency in Practice* (<https://archive.org/details/javaconcurrency00goet>). Addison-Wesley. p. xvii. ISBN 0-321-34960-1.
40. "Java 5.0 is no longer available on Java.com" (http://www.java.com/en/download/windows98me_manual.jsp). Java.com. 2009-11-03. Retrieved 2016-09-30.
41. Lineback, Nathan. "Misc Windows 2 – Nathan's Toasty Technology page" (<http://toastytech.com/guis/misccb2.html>). Retrieved 2016-09-30.
42. Yank, Kebin (May 3, 2005). "Java 5 available for Mac OS X" (<https://www.sitepoint.com/java-5-available-for-mac-os-x/>). Sitepoint. Retrieved September 30, 2016.
43. "Version 1.5.0 or 5.0?" (<http://docs.oracle.com/javase/1.5.0/docs/relnotes/version-5.0.html>). Oracle.com. Retrieved April 18, 2016.
44. Java brand naming (<http://www.java.com/en/about/brand/naming.jsp>).
45. Version 6 (<https://java.sun.com/javase/6/webnotes/version-6.html>), Java webnotes, Sun.
46. Version 1.6 press release (<http://www.sun.com/sm/Press/sunflash/2006-12/sunflash.20061211.1.xml>).

47. [Version 1.6 full list of changes](https://java.sun.com/javase/6/features.jsp) (<https://java.sun.com/javase/6/features.jsp>).
48. [Java Lobby](http://www.javalobby.org/java/forums/t66270.html) (<http://www.javalobby.org/java/forums/t66270.html>) Archived (<https://web.archive.org/web/20081013044209/http://www.javalobby.org/java/forums/t66270.html>) 2008-10-13 at the Wayback Machine.
49. "Mustang's HotSpot" (https://web.archive.org/web/20070102103432/http://weblogs.java.net/blog/opinali/archive/2005/11/mustangs_hotspo.html). Archived from the original (http://weblogs.java.net/blog/opinali/archive/2005/11/mustangs_hotspo.html) on January 2, 2007.
50. Darcy, Joe (2008-08-03). "An apt replacement" (https://blogs.oracle.com/darcy/entry/an_apt_replacement). Retrieved 2009-07-29.
51. "Install Java 6 on Mac OS X Leopard | Gephi, open source graph visualization software" (<https://gephi.org/users/install-java-6-mac-os-x-leopard/>). Gephi.org. Retrieved 2016-09-30.
52. "Oracle Java SE Support Roadmap" (<https://www.oracle.com/technetwork/java/eol-135779.html>). Oracle Corporation. September 19, 2012. Retrieved October 22, 2012.
53. "Auto-update and update through Java Control Panel of JRE 6 will replace JRE 6 with JRE 7" (<https://www.oracle.com/technetwork/java/javase/7u15-relnotes-1907738.html>). Oracle Corporation. February 19, 2013. Retrieved March 2, 2013.
54. "Java SE Development Kit 6, Update 43 Release notes" (<https://www.oracle.com/technetwork/java/javase/6u43-relnotes-1915290.html>). Oracle Corporation. March 4, 2013. Retrieved March 4, 2013.
55. "Java SE Development Kit 6, Update 45 Release notes" (<https://www.oracle.com/technetwork/java/javase/6u43-relnotes-1915290.html>). Oracle Corporation. March 4, 2013. Retrieved March 4, 2013.
56. [JDK7](http://jdk7.java.net/) (<http://jdk7.java.net/>) Archived (<https://web.archive.org/web/20161215071733/https://jdk7.java.net/>) 2016-12-15 at the Wayback Machine, Sun.
57. "Introducing Java 7 Webcast: Moving Java Forward" (https://www.oracle.com/webapps/events/ns/EventsDetail.jsp?p_eventId=134208&src=7299693&src=7299693&Act=18). Oracle Corporation. July 7, 2011. Retrieved May 30, 2012.
58. "JDK 7" (<https://openjdk.java.net/projects/jdk7/>). openjdk.java.net.
59. "JDK 7 Milestones" (<https://openjdk.java.net/projects/jdk7/milestones/>). OpenJDK. Oracle Corporation. Retrieved May 30, 2012.
60. Miller, Alex. "Java 7" (<http://tech.puredanger.com/java7>). Retrieved 2008-05-30.
61. "JSR 292: Supporting Dynamically Typed Languages on the Java Platform" (<http://jcp.org/en/jsr/detail?id=292>). Retrieved August 25, 2013.
62. "Compressed oops in the Hotspot JVM" (<https://wikis.oracle.com/display/HotSpotInternals/CompressedOops>). OpenJDK. Retrieved 2012-08-01.
63. "Java HotSpot VM Options" (<https://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html>). Oracle. Retrieved 2013-04-11.
64. "Java Programming Language Enhancements" (<http://download.oracle.com/javase/7/docs/technotes/guides/language/enhancements.html#javase7>). Download.oracle.com. Retrieved 2013-01-15.
65. "Strings in switch Statements" (<http://download.oracle.com/javase/7/docs/technotes/guides/language/strings-switch.html>). Download.oracle.com. Retrieved 2013-01-15.
66. "The try-with-resources Statement" (<http://download.oracle.com/javase/7/docs/technotes/guides/language/try-with-resources.html>). Download.oracle.com. Retrieved 2013-01-15.
67. "Type Inference for Generic Instance Creation" (<http://download.oracle.com/javase/7/docs/technotes/guides/language/type-inference-generic-instance-creation.html>). Download.oracle.com. Retrieved 2013-01-15.
68. "Improved Compiler Warnings When Using Non-Reifiable Formal Parameters with Varargs Methods" (<http://download.oracle.com/javase/7/docs/technotes/guides/language/non-reifiable-varargs.html>). Download.oracle.com. Retrieved 2013-01-15.

69. "Binary Literals" (<http://download.oracle.com/javase/7/docs/technotes/guides/language/binary-literals.html>). Download.oracle.com. Retrieved 2013-01-15.
70. "Underscores in Numeric Literals" (<http://download.oracle.com/javase/7/docs/technotes/guides/language/underscores-literals.html>). Download.oracle.com. Retrieved 2013-01-15.
71. "Catching Multiple Exception Types and Rethrowing Exceptions with Improved Type Checking" (<http://download.oracle.com/javase/7/docs/technotes/guides/language/catch-multiple.html>). Download.oracle.com. Retrieved 2013-01-15.
72. "Concurrency JSR-166" (<http://gee.cs.oswego.edu/dl/concurrency-interest/index.html>). Retrieved 2010-04-16.
73. "File I/O (Featuring NIO.2)" (<https://java.sun.com/docs/books/tutorial/essential/io/fileio.html>). Java.sun.com. 2008-03-14. Retrieved 2013-01-15.
74. "Legacy File I/O Code" (<https://java.sun.com/docs/books/tutorial/essential/io/legacy.html>). Java.sun.com. 2012-02-28. Retrieved 2013-01-15.
75. "JDK 7 Features" (<https://openjdk.java.net/projects/jdk7/features/#f650>). OpenJDK. 2011-07-28. Retrieved 2013-03-15.
76. "Introducing Deployment Rule Sets" (https://blogs.oracle.com/java-platform-group/entry/introducing_deployment_rule_sets). Oracle. 2013-08-21. Retrieved 2019-01-22.
77. "JavaOne 2011 Keynote" (https://web.archive.org/web/20111026122903/http://blogs.oracle.com/javaone/resource/java_keynote/slide_16_full_size.gif). Oracle. Archived from the original (http://blogs.oracle.com/javaone/resource/java_keynote/slide_16_full_size.gif) on 2011-10-26.
78. "Project Jigsaw" (<https://openjdk.java.net/projects/jigsaw/>). OpenJDK.
"Java Module-System Requirements — DRAFT 12" (<https://openjdk.java.net/projects/jigsaw/doc/draft-java-module-system-requirements-12>). Oracle.
Krill, Paul (July 18, 2012). "Project Jigsaw delayed until Java 9" (<https://www.infoworld.com/article/2078598/project-jigsaw-delayed-until-java-9.html>). *InfoWorld*. Retrieved 2020-07-15.
79. "Java 7 Auto-Update and Java 6" (<https://www.oracle.com/technetwork/java/javase/documentation/autoupdate-1667051.html>). Oracle.
80. Critical Patch Update table shows four dates per year (<https://www.oracle.com/technetwork/topics/security/alerts-086861.html>)
81. "End of Java 7 Public Updates" (https://java.com/en/download/faq/java_7.xml). "After April 2015, Oracle will no longer post updates of Java SE 7 to its public download sites. Existing Java SE 7 downloads already posted as of April 2015 will remain accessible in the Java Archive on the Oracle Technology Network. Developers and end-users are encouraged to update to more recent Java SE versions that remain available for public download in order to continue receiving public updates and security enhancements. [...] July 2015: Updates for Java 7 are no longer available to the public. Oracle offers updates to Java 7 only for customers who have purchased Java support or have Oracle products that require Java 7."
82. "Oracle Java SE Support Roadmap" (<https://www.oracle.com/technetwork/java/javase/downloads/eol-135779.html>). Oracle Corporation. 2014-12-19. Retrieved 2015-01-07.
83. "Proposed new schedule for Java 8" (<http://mail.openjdk.java.net/pipermail/jdk8-dev/2013-April/002336.html>). 2013-04-18. Retrieved 2013-04-19.
84. "JDK 8" (<https://openjdk.java.net/projects/jdk8/>). OpenJDK. 2013-04-18. Retrieved 2014-01-28.
85. "JDK 8 features" (<https://openjdk.java.net/projects/jdk8/features>). OpenJDK. 2014-01-28. Retrieved 2014-01-28.
86. "JSR 337" (<https://jcp.org/en/jsr/detail?id=337>). Retrieved 2014-01-30.
87. "Java 7 Features" (<https://openjdk.java.net/projects/jdk7/features/#f700>). Sun Microsystems. 2010-02-09. Retrieved 2010-04-16.
88. "Interface evolution via virtual extension methods" (http://wiki.jvmlangsummit.com/images/7/71/2011_Goetz.Extension.pdf) (PDF). Brian Goetz. 2011-07-20. Retrieved 2014-03-27.
89. "Lambda Expressions for the Java Programming Language" (<https://www.jcp.org/en/jsr/proposalDetails?id=335>). Brian Goetz. 2012-10-23. Retrieved 2014-03-27.

90. "The Java Tutorials: Default Methods" (<https://web.archive.org/web/20170523042436/http://docs.oracle.com/javase/tutorial/java/landl/defaultmethods.html>). Oracle. Archived from the original (<http://docs.oracle.com/javase/tutorial/java/landl/defaultmethods.html>) on 2017-05-23. Retrieved 2014-03-27.
91. Gafter, Neal (2006-08-18). "Closures for Java" (<http://gafter.blogspot.com/2006/08/closures-for-java.html>). Retrieved 2008-03-09.
92. Gosling, James (2008-01-31). "Closures" (<https://web.archive.org/web/20110722180952/http://blogs.oracle.com/jag/entry/closures>). Archived from the original (<http://blogs.oracle.com/jag/entry/closures>) on 2011-07-22. Retrieved 2008-03-09.
93. Reinhold, Mark (2009-11-28). "Closures for Java" (<http://mreinhold.org/blog/closures>). Retrieved 2009-11-24.
94. Darcy, Joe. "Unsigned Integer Arithmetic API now in JDK 8" (<https://blogs.oracle.com/darcy/unsigned-integer-arithmetic-api-now-in-jdk-8>). *blogs.oracle.com*.
95. "Oracle JDK 8 and JRE 8 Certified System Configurations" (<https://www.oracle.com/technetwork/java/javase/certconfig-2095354.html#os>). Oracle Corporation. Retrieved 2014-04-15.
96. Stahl, Henrik (2014-07-11). "Updated: The future of Java on Windows XP" (https://web.archive.org/web/20141111174328/https://blogs.oracle.com/henrik/entry/the_future_of_java_on). Oracle Corporation. Archived from the original (https://blogs.oracle.com/henrik/entry/the_future_of_java_on) on 2014-11-11. Retrieved 2014-11-11. "JDK 8 is not supported on Windows XP. Early versions of JDK 8 had known issues with the installer on Windows XP that prevented it from installing without manual intervention. This was resolved in JDK 8 Update 25. The important point here is that we can no longer provide complete guarantees for Java on Windows XP, since the OS is no longer being updated by Microsoft. We strongly recommend that users upgrade to a newer version of Windows that is still supported by Microsoft in order to maintain a stable and secure environment."
97. Wieldt, Tori (October 27, 2014). "Java SE 8 on Java.com" (https://web.archive.org/web/20141129113353/https://blogs.oracle.com/java/entry/java_se_8_on_java). *blog.oracle.com*. Archived from the original (https://blogs.oracle.com/java/entry/java_se_8_on_java) on November 29, 2014. Retrieved November 24, 2014.
98. "JDK 9" (<https://openjdk.java.net/projects/jdk9/>). Retrieved 2017-06-16.
99. "Java modularity specification opposed by Red Hat, IBM is voted down" (<http://www.infoworld.com/article/3195180/java/java-modularity-specification-opposed-by-red-hat-ibm-is-voted-down.html>). *InfoWorld*. 2017-05-09. Retrieved 2017-06-16.
100. Chirgwin, Richard (July 2, 2017). "Java 9 release back on track, community votes 'yes'" (https://www.theregister.co.uk/2017/07/02/java_9_release_back_on_track/). *theresregister.co.uk*. Retrieved 2017-07-29.
101. "Java Platform, Standard Edition Oracle JDK 9 Migration Guide" (<https://docs.oracle.com/javase/9/migrate/toc.htm>). *docs.oracle.com*. Retrieved 15 May 2018.
102. "OpenJDK: Project Kulla" (<https://openjdk.java.net/projects/kulla/>).
103. Lea, Doug (January 15, 2015). "[concurrency-interest] jdk9 Candidate classes Flow and Submission Publisher" (<https://web.archive.org/web/20150120055013/http://cs.oswego.edu/pipermail/concurrency-interest/2015-January/013641.html>). Archived from the original (<http://cs.oswego.edu/pipermail/concurrency-interest/2015-January/013641.html>) on January 20, 2015. Retrieved December 24, 2015.
104. "Flow (Java SE 9 & JDK 9)" (<https://docs.oracle.com/javase/9/docs/api/java/util/concurrent/Flow.html>). *docs.oracle.com*.
105. Reactive Streams Releases First Stable Version for JVM (<http://www.infoq.com/news/2015/06/Reactive-Streams-JVM-Version>)
106. "JDK 9: First Release Candidate" (<http://mail.openjdk.java.net/pipermail/jdk9-dev/2017-August/005940.html>). 2017-08-09. Retrieved 2017-08-21.

107. "Java 9: Release date and new features" (<https://www.techworld.com/news/developers/java-9-release-date-new-features-3660988/>). techworld.com. 2017-07-21. Retrieved 2017-09-16.
"JDK 9" (<https://openjdk.java.net/projects/jdk9/>). Oracle Corporation. Retrieved 2017-09-16.
108. "Mark your calendar: Java 9 finally has a release date" (<http://www.networkworld.com/article/2920183/microsoft-subnet/mark-your-calendar-java-9-finally-has-a-release-date.html>). 2015-05-07.
109. "Java 9's new garbage collector: What's changing? What's staying?" (<https://jaxenter.com/java-9s-new-garbage-collector-whats-changing-whats-staying-118313.html>). June 26, 2015.
110. "JavaOne: JavaFX 2, Java on iOS" (<http://drdobbs.com/blogs/java/231900029>).
111. "Java 9 Release Date Now March 2017" (<https://news.filehippo.com/2016/01/java-9-release-date-now-march-2017/>). 2016-01-05.
112. "JDK 9 release delayed another four months" (https://www.theregister.co.uk/2016/09/14/jdk_9_release_delay/). The Register. Retrieved 2016-10-14.
113. OpenJDK. "JDK 10" (<https://openjdk.java.net/projects/jdk/10/>).
114. OpenJDK. "JDK 11" (<https://openjdk.java.net/projects/jdk/11/>).
115. "Oracle JDK Migration Guide" (<https://docs.oracle.com/en/java/javase/11/migrate/index.html>). Oracle Help Center. Retrieved 27 September 2018.
116. OpenJDK. "12" (<https://openjdk.java.net/projects/jdk/12/>).
117. oracle.com, mark reinhold at (September 17, 2019). "Java 13 / JDK 13: General Availability" (<https://mail.openjdk.java.net/pipermail/announce/2019-September/000274.html>).
118. oracle.com, mark reinhold at (March 17, 2020). "Java 14 / JDK 14: General Availability" (<https://mail.openjdk.java.net/pipermail/announce/2020-March/000282.html>).
119. Evans, Ben (January 10, 2020). "Records Come to Java" (<https://blogs.oracle.com/javamagazine/records-come-to-java>). Java Magazine. Oracle. Retrieved 10 July 2021.
120. Gavin Bierman; Brian Goetz (September 2018). "Pattern Matching for Java" (<https://cr.openjdk.java.net/~briangoetz/amber/pattern-match.html>). cr.openjdk.java.net. Retrieved 2021-04-28.
121. "[JDK-8255616] Removal of experimental features AOT and Graal JIT - Java Bug System" (<https://bugs.openjdk.java.net/browse/JDK-8255616>). bugs.openjdk.java.net. Retrieved 2021-02-19.
122. "JDK 17" (<https://openjdk.java.net/projects/jdk/17/>). openjdk.java.net. Retrieved 2021-09-17.
123. "JDK 19" (<https://openjdk.org/projects/jdk/19/>). openjdk.org. Retrieved 2022-09-19.
124. "JDK 20" (<https://openjdk.org/projects/jdk/20/>). openjdk.org. Retrieved 2023-04-21.
125. "Downloads - Overview" (<https://developer.ibm.com/javasdk/downloads/>). July 18, 2016.

External links

- [Official Java SE Downloads](https://www.oracle.com/java/technologies/downloads/) (<https://www.oracle.com/java/technologies/downloads/>)
 - [Sun Java Supported versions and EOL](https://www.oracle.com/java/technologies/java-se-support-roadmap.html) (<https://www.oracle.com/java/technologies/java-se-support-roadmap.html>)
 - [Downloads archive for older version of Java](https://www.oracle.com/java/technologies/downloads/archive/) (<https://www.oracle.com/java/technologies/downloads/archive/>)
 - [JDK Releases](https://www.java.com/releases/) (<https://www.java.com/releases/>)
 - [The Java Version Almanac](https://javaalmanac.io/) (<https://javaalmanac.io/>)
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Java_version_history&oldid=1153374605"

