JDK

The first version was released on January 23, 1996.[15][16] The first stable version, JDK 1.0.2, is called Java 1.[16]

JDK 1.1

Major additions in the release on February 19, 1997 included:[17]

- extensive retooling of the Abstract Window Toolkit (AWT) event model
- inner classes added to the language
- JavaBeans
- Java Database Connectivity (JDBC)
- Java remote method invocation (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 class and its subclasses such as the Field class.)
- Just-in-time compilation (JIT) on Microsoft Windows platforms, produced for JavaSoft by Symantec
- Internationalization and Unicode support originating from Taligent[18]

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:[19]

- `strictfp` keyword (by JVM 17 an obsolete keyword, should not 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:[20][21]

- 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.[22]

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. Major changes included:[23][24]

- Language changes
  - `assert` keyword (specified in JSR 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)
  - Logging API (specified in JSR 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 and JSR 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)
  - Preferences API (`java.util.prefs`)

Java SE 6

As of the version released on December 11, 2006, Sun replaced the name "J2SE" with **Java SE** and dropped the ".0" from the version number.[35] Internal numbering for developers remains 1.6.0.[36]

This version was developed under JSR 270.

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:[37][38]

- 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.[*citation needed*] This is believed[*by whom?*] 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,[39][40] 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).[41]
- 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 SE 7

Java 7 was a major update that launched on July 7, 2011[85] and was made available for developers on July 28, 2011.[86] The development period was organized into thirteen milestones; on June 6, 2011, the last of the thirteen milestones was finished.[86][87] On average, 8 builds (which generally included enhancements and bug fixes) were released per milestone. The feature list at the OpenJDK 7 project lists many of the changes.

- These small language changes (grouped under a project named Coin):[92]
  - Strings in switch[93]
  - Automatic resource management in try-statement aka *try-with-resources statement*[94]

- Improved type inference for generic instance creation, aka *the diamond operator* <>[95]
- Simplified varargs method declaration[96]
- Binary integer literals[97]
- Allowing underscores in numeric literals[98]
- Catching multiple exception types and rethrowing exceptions with improved type checking[99]

Java SE 8

Java 8 was released on March 18, 2014,[146][147] and included some features that were planned for Java 7 but later deferred.[148]

Work on features was organized in terms of JDK Enhancement Proposals (JEPs).[149]

- JEP 153: Launch JavaFX applications (direct launching of JavaFX application JARs)
- JEP 122: Remove the permanent generation

Java 8 is not supported on Windows XP[158] but as of JDK 8 update 25, it can still be installed and run under Windows XP.[159] 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.[*citation needed*]

From October 2014, Java 8 was the default version to download (and then again the download replacing Java 9) from the official website.[160] "Oracle will continue to provide Public Updates and auto updates of Java SE 8, Indefinitely for Personal Users".[161]

Java SE 9

Java SE 9 was made available on September 21, 2017[238] due to controversial acceptance of the current implementation of Project Jigsaw by Java Executive Committee[239] 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.[240]

- JEP 295: Ahead-of-Time Compilation, ahead-of-time compilation provided by GraalVM

The first Java 9 release candidate was released on August 9, 2017.[246] The first stable release of Java 9 was on September 21, 2017.[247]

Java SE 10

OpenJDK 10 was released on March 20, 2018, with twelve new features confirmed.[258] Among these features were:

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

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

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.[265]

Java SE 12

JDK 12 was released on March 19, 2019. Among others, Java 12 includes a number of new features, such as:[312]

- JEP 189: Shenandoah: A Low-Pause-Time Garbage Collector (Experimental)
- JEP 230: Microbenchmark Suite
- JEP 325: Switch Expressions (Preview)
- JEP 334: JVM Constants API
- JEP 340: One AArch64 Port, Not Two
- JEP 341: Default CDS Archives
- JEP 344: Abortable Mixed Collections for G1
- JEP 346: Promptly Return Unused Committed Memory from G1

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".[318]

- JEP 350: Dynamic CDS Archives
- JEP 351: ZGC: Uncommit Unused Memory
- JEP 353: Reimplement the Legacy Socket API
- JEP 354: Switch Expressions (Preview)
- JEP 355: Text Blocks (Preview)

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".[323]

- JEP 305: Pattern Matching for instanceof (Preview)
- JEP 343: Packaging Tool (Incubator)
- JEP 345: NUMA-Aware Memory Allocation for G1
- JEP 349: JFR Event Streaming
- JEP 352: Non-Volatile Mapped Byte Buffers
- JEP 358: Helpful NullPointerExceptions
- JEP 359: Records (Preview)
- JEP 361: Switch Expressions (Standard)
- JEP 362: Deprecate the Solaris and SPARC Ports
- JEP 363: Remove the Concurrent Mark Sweep (CMS) Garbage Collector

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.

Java SE 16

JDK 16 was released on March 16, 2021. Java 16 removes Ahead-of-Time compilation (and Graal JIT) options.[336] 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.

Java SE 17

JDK 17 was released in September 2021.[342] 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
- JEP 356: Enhanced Pseudo-Random Number Generators
- JEP 382: New macOS Rendering Pipeline
- JEP 391: macOS/AArch64 Port
- JEP 398: Deprecate the Applet API for Removal
- JEP 403: Strongly Encapsulate JDK Internals
- JEP 406: Pattern Matching for switch (Preview)
- JEP 407: Remove RMI Activation
- JEP 409: Sealed Classes
- JEP 410: Remove the Experimental AOT and JIT Compiler
- JEP 411: Deprecate the Security Manager for Removal
- JEP 412: Foreign Function & Memory API (Incubator)
- JEP 414: Vector API (Second Incubator)
- JEP 415: Context-Specific Deserialization Filters

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

Java SE 18

JDK 18 was released on March 22, 2022.[367]

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

Java SE 19

JDK 19 was released on 20 September 2022.[376]

- JEP 405: Record Patterns (Preview)

- JEP 422: Linux/RISC-V Port
- JEP 424: Foreign Function & Memory API (Preview)
- JEP 425: Virtual Threads (Preview)
- JEP 426: Vector API (Fourth Incubator)
- JEP 427: Pattern Matching for switch (Third Preview)
- JEP 428: Structured Concurrency (Incubator)

Java SE 20

Java 20 was released on 21 March 2023.[382] All JEPs were either incubators or previews.

- JEP 429: Scoped Values (Incubator)
- JEP 432: Record Patterns (Second Preview)
- JEP 433: Pattern Matching for switch (Fourth Preview)
- JEP 434: Foreign Function & Memory API (Second Preview)
- JEP 436: Virtual Threads (Second Preview)
- JEP 437: Structured Concurrency (Second Incubator)
- JEP 438: Vector API (Fifth Incubator)

Java SE 21

Java 21 was released on 19 September 2023.[388] There are eight JEPs that are not in preview or incubating, compared to Java 20 which only had previewing and incubating JEPs. Java 21 introduces features first previewed in Java 17 (pattern matching for switch statements) and Java 19 (record patterns). The 32-bit version of Java for Windows on x86 has been deprecated for removal.

- JEP 430: String Templates (Preview)
- JEP 431: Sequenced Collections
- JEP 439: Generational ZGC
- JEP 440: Record Patterns
- JEP 441: Pattern Matching for switch
- JEP 442: Foreign Function & Memory API (Third Preview)
- JEP 443: Unnamed Patterns and Variables (Preview)
- JEP 444: Virtual Threads
- JEP 445: Unnamed Classes and Instance Main Methods (Preview)
- JEP 446: Scoped Values (Preview)
- JEP 448: Vector API (Sixth Incubator)
- JEP 449: Deprecate the Windows 32-bit x86 Port for Removal
- JEP 451: Prepare to Disallow the Dynamic Loading of Agents
- JEP 452: Key Encapsulation Mechanism API
- JEP 453: Structured Concurrency (Preview)

JEP 445, previewing unnamed classes, allows for a barebones Main class without boilerplate code:

Java SE 22

Java 22 was released on March 19, 2024.[394][395]

- JEP 423: Region Pinning for G1
- JEP 447: Statements before super(...) (Preview)
- JEP 454: Foreign Function & Memory API
- JEP 456: Unnamed Variables & Patterns
- JEP 457: Class-File API (Preview)

- JEP 458: Launch Multi-File Source-Code Programs
- JEP 459: String Templates (Second Preview)
- JEP 460: Vector API (Seventh Incubator)
- JEP 461: Stream Gatherers (Preview)
- JEP 462: Structured Concurrency (Second Preview)
- JEP 463: Implicitly Declared Classes and Instance Main Methods (Second Preview)
- JEP 464: Scoped Values (Second Preview)

At least one API has been removed from Java; i.e. a very rarely-used API (for threading) was removed from Java 22.[3][396]

Java SE 23

As of January 2024, the specification for Java 23 has not yet been finalized. Java 23 is scheduled for release in September 2024.[397]