



INNOQ Technology Lunch / 08.11.2023

Das erwartet Dich mit JDK 21!



MICHAEL VITZ
SENIOR CONSULTANT

MICHAEL VITZ

Java Champion

Senior Consultant at INNOQ





Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds
Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &
Specification

Review

Compiler

Conformance

Core Libraries

JDK 21

This release is the Reference Implementation of version 21 of the Java SE Platform, as specified by [JSR 396](#) in the Java Community Process.

JDK 21 reached [General Availability](#) on 19 September 2023. Production-ready binaries under the GPL are [available from Oracle](#); binaries from other vendors will follow shortly.

The features and schedule of this release were proposed and tracked via the [JEP Process](#), as amended by the [JEP 2.0 proposal](#). The release was produced using the [JDK Release Process \(JEP 3\)](#).

Features

430: [String Templates \(Preview\)](#)

431: [Sequenced Collections](#)

439: [Generational ZGC](#)

440: [Record Patterns](#)

441: [Pattern Matching for switch](#)

442: [Foreign Function & Memory API \(Third Preview\)](#)

443: [Unnamed Patterns and Variables \(Preview\)](#)

444: [Virtual Threads](#)

Review

Compiler

Conformance

Core Libraries

Governing Board

HotSpot

IDE Tooling & Support

Internationalization

JMX

Members

Networking

Porters

Quality

Security

Serviceability

Vulnerability

Web

Projects

(overview, archive)

Amber

Audio Engine

CRaC

Caciocavallo

Closures

Code Tools

Coin

Common VM

Interface

Compiler Grammar

Detroit

Developers' Guide

Device I/O

Duke

Font Scaler

Galahad

Graal

443: Unnamed Patterns and Variables (Preview)

444: Virtual Threads

445: Unnamed Classes and Instance Main Methods (Preview)

446: Scoped Values (Preview)

448: Vector API (Sixth Incubator)

449: Deprecate the Windows 32-bit x86 Port for Removal

451: Prepare to Disallow the Dynamic Loading of Agents

452: Key Encapsulation Mechanism API

453: Structured Concurrency (Preview)

JDK 21 will be a long-term support (LTS) release from most vendors. For a complete list of the JEPs integrated since the previous LTS release, JDK 17, please see [here](#).

Schedule

2023/06/08	Rampdown Phase One (fork from main line)
2023/07/20	Rampdown Phase Two
2023/08/10	Initial Release Candidate
2023/08/24	Final Release Candidate
2023/09/19	General Availability

Last update: 2023/9/19 10:53 UTC



LTS

 Notifications
  Fork 4.9k
  Star 16.9k
 

[🔗 master](#)
[🔗 249 branches](#)
[🔖 892 tags](#)
[Go to file](#)
[Code](#)

About

JDK main-line development
<https://openjdk.org/projects/jdk>

java jvm openjdk

📖 Readme

📄 GPL-2.0 license

📈 Activity

★ 16.9k stars

👁 312 watching

🍴 4.9k forks

GitHub - openjdk/jdk: JDK main-line development

Product Solutions Open Source Pricing

Search or jump to... / Sign in Sign up

openjdk / jdk Public

Notifications Fork 4.9k Star 16.9k

Code Pull requests 288 Security Insights

master 249 branches 892 tags Go to file Code

About

JDK main-line development
<https://openjdk.org/projects/jdk>

java jvm openjdk

Readme GPL-2.0 license Activity 16.9k stars 312 watching 4.9k forks

JDK mainline governed by OpenJDK under Oracle's leadership

<https://github.com/openjdk/jdk>

Product

Solutions

Open Source

Pricing

Search or jump to...

/

Sign in

Sign up

This repository has been archived by the owner on Sep 19, 2023. It is now read-only.

openjdk / jdk21

Public archive

Notifications

Fork 54

Star 53

<> Code

Pull requests

Security

Insights

master

1 branch

875 tags

Go to file

Code

About

<https://openjdk.org/projects/jdk/21>

released 2023-09-19

Readme

GPL-2.0 license

Activity

53 stars

7 watching

54 forks

Report repository

Daniel D. Daugherty 8314062: P... 5 890adb6 on Aug 9 74,404 commits

.github	8310259: Pin msys2/setup-msys2 github ac...	3 months ago
.jcheck	8296149: Start of release updates for JDK 21	9 months ago
bin	8309501: Remove workaround in bin/idea.s...	3 months ago
doc	8309287: Add fontconfig requirement to bui...	4 months ago
make	8312985: Remove EA from the JDK 21 versi...	last month

<https://github.com/openjdk/jdk21>

GitHub - openjdk/jdk21: https://github.com/openjdk/jdk21

Product Solutions Open Source Pricing Search or jump to... Sign in Sign up

This repository has been archived by the owner on Sep 19, 2023. It is now read-only.

openjdk / jdk21 Public archive Notifications Fork 54 Star 53

Code Pull requests Security Insights

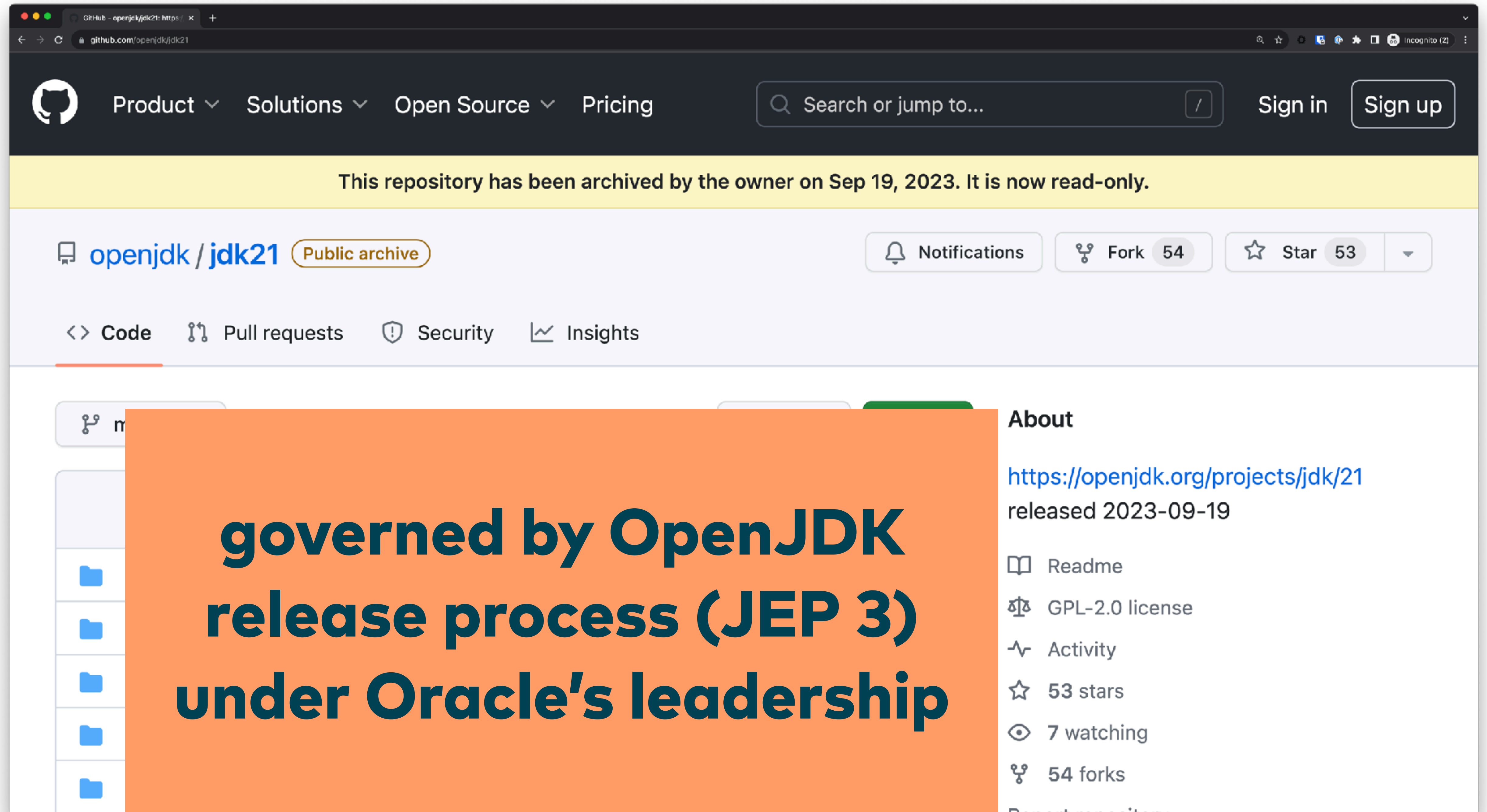
forked from mainline ~3 months before release

About

<https://openjdk.org/projects/jdk/21>
released 2023-09-19

Readme
GPL-2.0 license
Activity
53 stars
7 watching
54 forks
Report repository

<https://github.com/openjdk/jdk21>



**governed by OpenJDK
release process (JEP 3)
under Oracle's leadership**

About

<https://openjdk.org/projects/jdk/21>
released 2023-09-19

📖 Readme

📄 GPL-2.0 license

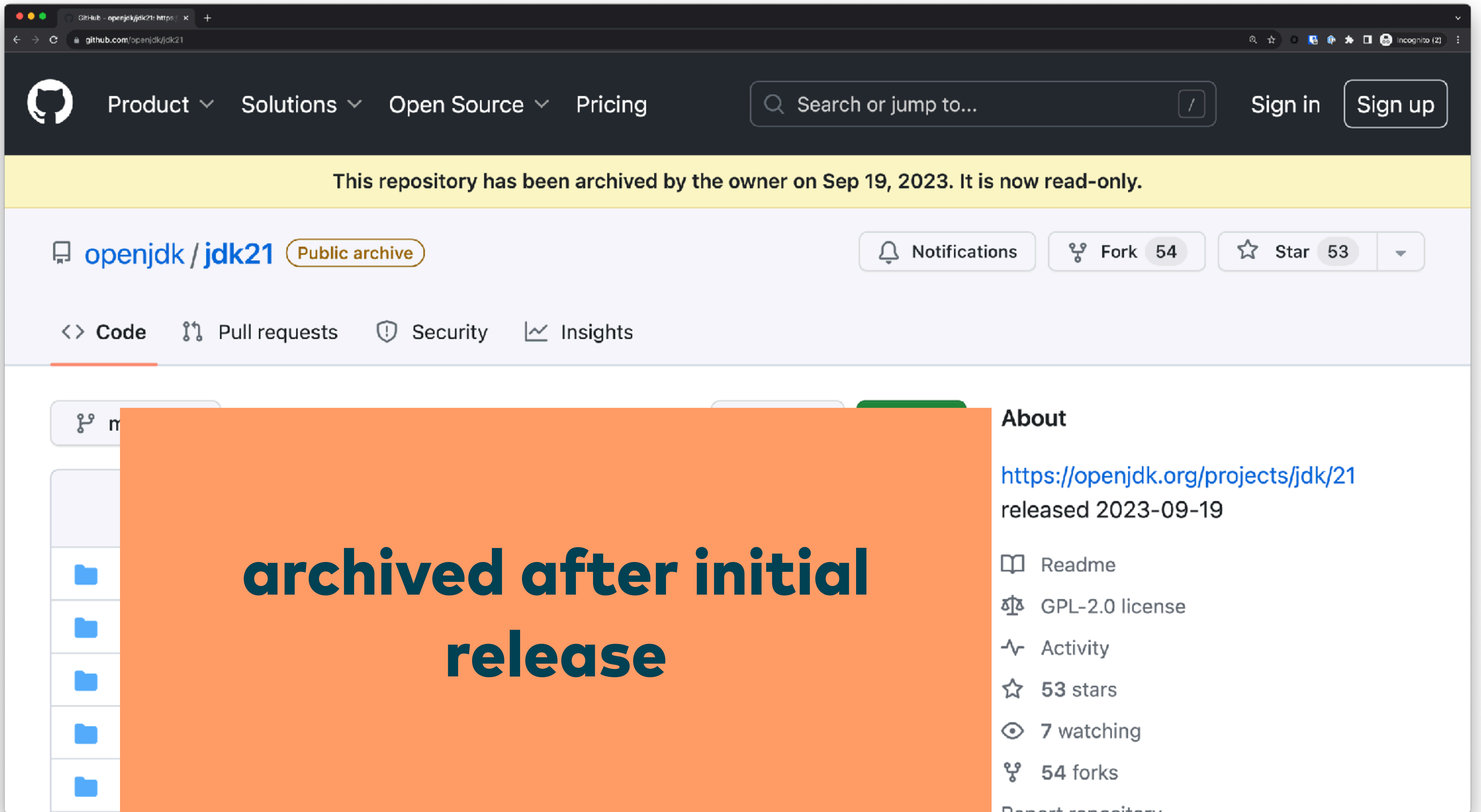
📈 Activity

★ 53 stars

👁 7 watching

🍴 54 forks

Report repository




**archived after initial
release**




About


<https://openjdk.org/projects/jdk/21>
released 2023-09-19

- 📖 Readme
- 📄 GPL-2.0 license
- 📈 Activity
- ★ 53 stars
- 👁 7 watching
- 🍴 54 forks

Report repository




Product  Solutions  Open Source  Pricing

 Search or jump to...


/


Sign in



Sign up


 **openjdk / jdk21u**


Public


 Notifications


 Fork 50



 Star 19 


 **Code**


 Pull requests 20

 Security


 Insights

 master 


 20 branches


 875 tags

Go to file


Code 







Man Cao and Chuck Rasbold...



 1


ea00fca 18 hours ago


 74,577 commits


 .github	8315863: [GHA] Update checkout action to ...	2 weeks ago
 .jcheck	8315688: Update jdk21u fix version to 21.0.2	3 weeks ago
 bin	8309501: Remove workaround in bin/idea.s...	3 months ago
 doc	8309287: Add fontconfig requirement to bui...	4 months ago
 make	8313244: NM flags handling in configure pr...	5 days ago
 src	8316468: os::write incorrectly handles parti...	18 hours ago


About


<https://openjdk.org/projects/jdk-updates>


 Readme

 GPL-2.0 license

 Activity

 19 stars

 5 watching

 50 forks

Report repository

<https://github.com/openjdk/jdk21u>

GitHub - openjdk/jdk21u: https://github.com/openjdk/jdk21u

Product Solutions Open Source Pricing Search or jump to... Sign in Sign up

openjdk / jdk21u Public Notifications Fork 50 Star 19

Code Pull requests 20 Security Insights

master 20 branches 875 tags Go to file Code About

forked from release few weeks before release date

<https://openjdk.org/projects/jdk-updates>

- Readme
- GPL-2.0 license
- Activity
- 19 stars
- 5 watching
- 50 forks
- Report repository

GitHub - openjdk/jdk21u: https://github.com/openjdk/jdk21u

Product Solutions Open Source Pricing Search or jump to... Sign in Sign up

openjdk / jdk21u Public Notifications Fork 50 Star 19

Code Pull requests 20 Security Insights

master 20 branches 875 tags Go to file Code About

Oracle leads the community effort for two security releases (= 6 months)

<https://openjdk.org/projects/jdk-updates>

- Readme
- GPL-2.0 license
- Activity
- 19 stars
- 5 watching
- 50 forks
- Report repository

GitHub - openjdk/jdk21u: https://github.com/openjdk/jdk21u

Product Solutions Open Source Pricing Search or jump to... Sign in Sign up

openjdk / jdk21u Public Notifications Fork 50 Star 19

Code Pull requests 20 Security Insights

master 20 branches 875 tags Go to file Code About


afterwards another trustworthy community member can take over





<https://openjdk.org/projects/jdk-updates>

- Readme
- GPL-2.0 license
- Activity
- 19 stars
- 5 watching
- 50 forks
- Report repository

blogs.oracle.com/java/post/moving-the-jdk-to-a-two-year-lts-cadence

Java


Follow: 



Java Technology

Moving the JDK to a Two Year LTS Cadence

September 14, 2021 | 2 minute read



Donald Smith

Vice President of Product Management

Executive Summary:

- Oracle proposes to shift the cadence of JDK LTS releases from every three years to every two years. If accepted, this means that the next JDK LTS release after JDK 17 will be JDK 21, rather than JDK 25.
- Duration of the Oracle JDK LTS support remains unchanged, starting at a minimum of eight years for each LTS release.

Article

The release of JDK 17 is a perfect time to reflect on the release cadence introduced several years ago and make any course corrections for the future. Developers and organizations have had time to adjust from the “massive change every few years” model to the more

<https://blogs.oracle.com/java/post/moving-the-jdk-to-a-two-year-lts-cadence>



JDK 21

JDK 21

openjdk.org/projects/jdk/21/

JEP Process

Source code

Mercurial

GitHub

Tools

Git

jtreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility & Specification

Review

Compiler

Conformance

Core Libraries

Governing Board

HotSpot

IDE Tooling & Support

Internationalization

JMX

Members

Networking

Porters

Quality

Security

Serviceability

Vulnerability

Web

Projects

(overview, archive)

Amber

JDK Release Process (JEP 3).

Features

430: String Templates (Preview)

431: Sequenced Collections

439: Generational ZGC

440: Record Patterns

441: Pattern Matching for switch

442: Foreign Function & Memory API (Third Preview)

443: Unnamed Patterns and Variables (Preview)

444: Virtual Threads

445: Unnamed Classes and Instance Main Methods (Preview)

446: Scoped Values (Preview)

448: Vector API (Sixth Incubator)

449: Deprecate the Windows 32-bit x86 Port for Removal

451: Prepare to Disallow the Dynamic Loading of Agents

452: Key Encapsulation Mechanism API

453: Structured Concurrency (Preview)

JDK 21 will be a long-term support (LTS) release from most vendors. For a complete list of the JEPs integrated since the previous LTS release, JDK 17, please see [here](#).



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &
Specification

Review

Compiler

Conformance

Core Libraries

Governing Board

JEPs in JDK 21 integrated since JDK 17

Here are all of the JEPs integrated since JDK 17, which was the previous long-term-support (LTS) release from most vendors. Preview and Incubator JEPs that were superseded by later JEPs in JDKs 18 through 21 are not included. The release in which a JEP was integrated is shown in parentheses after the JEP's title.

Additions

HotSpot JVM

422: [Linux/RISC-V Port](#) (19)

Garbage Collectors

439: [Generational ZGC](#) (21)

Serviceability

451: [Prepare to Disallow the Dynamic Loading of Agents](#) (21)

Language

441: [Pattern Matching for switch](#) (21)

440: [Record Patterns](#) (21)

Libraries

444: [Virtual Threads](#) (21)

Client Libraries

Compatibility & Specification

Review

Compiler

Conformance

Core Libraries

Governing Board

HotSpot

IDE Tooling & Support

Internationalization

JMX

Members

Networking

Porters

Quality

Security

Serviceability

Vulnerability

Web

Projects

(overview, archive)

Amber

Audio Engine

CRaC

Caciocavallo

Closures

Code Tools

Coin

Common VM

Interface

Compiler Grammar

Detroit

Developers' Guide

Device I/O

Duke

Fast Compiler

441: Pattern Matching for Switch (21)

440: Record Patterns (21)

Libraries

444: Virtual Threads (21)

Collections

431: Sequenced Collections (21)

Cryptography

452: Key Encapsulation Mechanism API (21)

I/O

400: UTF-8 by Default (18)

Networking

418: Internet-Address Resolution SPI (18)

408: Simple Web Server (18)

Reflection & Method Handles

416: Reimplement Core Reflection with Method Handles (18)

Tools

JavaDoc

413: Code Snippets in Java API Documentation (18)

Preview & Incubating

JEPs in JDK 21 integrated since JDK 17

openjdk.org/projects/jdk/21/jeps-since-jdk-17

Code Tools

Coin

Common VM Interface

Compiler Grammar

Detroit

Developers' Guide

Device I/O

Duke

Font Scaler

Galahad

Graal

Graphics Rasterizer

IcedTea

JDK 7

JDK 8

JDK 8 Updates

JDK 9

JDK (... , 21, 22)

JDK Updates

JavaDoc.Next

Jigsaw

Kona

Kulla

Lambda

Lanai

Leyden

Lilliput

Locale Enhancement

Loom

Memory Model Update

Metropolis

Mission Control

Modules

Multi-Language VM

Nekean

416: Reimplement Core Reflection with Method Handles (18)

Tools

JavaDoc

413: Code Snippets in Java API Documentation (18)

Preview & Incubating

Language

430: String Templates (Preview) (21)

445: Unnamed Classes and Instance Main Methods (Preview) (21)

443: Unnamed Patterns and Variables (Preview) (21)

Libraries

442: Foreign Function & Memory API (Third Preview) (21)

446: Scoped Values (Preview) (21)

453: Structured Concurrency (Preview) (21)

448: Vector API (Sixth Incubator) (21)

Deprecations

HotSpot JVM

449: Deprecate the Windows 32-bit x86 Port for Removal (21)

<https://openjdk.org/projects/jdk/21/jeps-since-jdk-17>

Device I/O

Duke

Font Scaler

Galahad

Graal

Graphics Rasterizer

IcedTea

JDK 7

JDK 8

JDK 8 Updates

JDK 9

JDK (... , 21, 22)

JDK Updates

JavaDoc.Next

Jigsaw

Kona

Kulla

Lambda

Lanai

Leyden

Lilliput

Locale Enhancement

Loom

Memory Model

Update

Metropolis

Mission Control

Modules

Multi-Language VM

Nashorn

New I/O

OpenJFX

Panama

Penrose

Port: AArch32

Port: AArch64

Preview & Incubating

Language

430: [String Templates \(Preview\) \(21\)](#)

445: [Unnamed Classes and Instance Main Methods \(Preview\) \(21\)](#)

443: [Unnamed Patterns and Variables \(Preview\) \(21\)](#)

Libraries

442: [Foreign Function & Memory API \(Third Preview\) \(21\)](#)

446: [Scoped Values \(Preview\) \(21\)](#)

453: [Structured Concurrency \(Preview\) \(21\)](#)

448: [Vector API \(Sixth Incubator\) \(21\)](#)

Deprecations

HotSpot JVM

449: [Deprecate the Windows 32-bit x86 Port for Removal \(21\)](#)

Libraries

421: [Deprecate Finalization for Removal \(18\)](#)



Features



JEP 11: Incubator Modules

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds
Mailing lists
Wiki · IRC
Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtrex harness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

Authors Chris Hegarty, Alex Buckley

Owner Chris Hegarty

Type Informational

Scope JDK

Status Active

Discussion [jdk dash dev at openjdk dot java dot net](#)

Effort S

Duration S

Reviewed by Alan Bateman, Alex Buckley, Brian Goetz, Paul Sandoz

Endorsed by Brian Goetz

Created 2016/11/16 09:17

Updated 2019/12/05 17:14

Issue [8169768](#)

Summary

Incubator modules are a means of putting non-final APIs and non-final tools in the hands of developers, while the APIs/tools progress towards either finalization or removal in a future release.

Goals



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 12: Preview Features

Owner Alex Buckley

Type Informational

Scope SE

Status Active

Discussion [jdk dash dev at openjdk dot java dot net](#)

Effort M

Duration M

Reviewed by Alan Bateman, Brian Goetz, Mark Reinhold

Endorsed by Mark Reinhold

Created 2018/01/19 01:27

Updated 2023/09/21 21:14

Issue [8195734](#)

Summary

A *preview feature* is a new feature of the Java language, Java Virtual Machine, or Java SE API that is fully specified, fully implemented, and yet impermanent. It is available in a JDK feature release to provoke developer feedback based on real world use; this may lead to it becoming permanent in a future Java SE Platform.

Goals



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 394: Pattern Matching for instanceof

Owner Gavin Bierman

Type Feature

Scope SE

Status Closed / Delivered

Release 16

Component specification / language

Discussion amber dash dev at openjdk dot java dot net

Relates to [JEP 305: Pattern Matching for instanceof \(Preview\)](#)
[JEP 375: Pattern Matching for instanceof \(Second Preview\)](#)

Reviewed by Alex Buckley, Brian Goetz, Maurizio Cimadamore

Endorsed by Brian Goetz

Created 2020/07/27 13:05

Updated 2022/06/10 16:12

Issue [8250623](#)

Summary

Enhance the Java programming language with *pattern matching* for the instanceof operator. [Pattern matching](#) allows common logic in a program, namely the conditional extraction of components from objects, to be expressed more concisely and safely.



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 441: Pattern Matching for switch

Owner Gavin Bierman

Type Feature

Scope SE

Status Closed / Delivered

Release 21

Component specification / language

Discussion [amber dash dev at openjdk dot org](#)

Relates to [JEP 433: Pattern Matching for switch \(Fourth Preview\)](#)

Reviewed by Brian Goetz

Endorsed by Brian Goetz

Created 2023/01/18 14:43

Updated 2023/09/19 13:38

Issue [8300542](#)

Summary

Enhance the Java programming language with pattern matching for switch expressions and statements. Extending pattern matching to switch allows an expression to be tested against a number of patterns, each with a specific action, so that complex data-oriented queries can be expressed concisely and safely.



JEP 440: Record Patterns

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

<i>Owner</i>	Gavin Bierman
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Closed / Delivered
<i>Release</i>	21
<i>Component</i>	specification / language
<i>Discussion</i>	amber dash dev at openjdk dot org
<i>Relates to</i>	JEP 432: Record Patterns (Second Preview)
<i>Reviewed by</i>	Brian Goetz
<i>Endorsed by</i>	Brian Goetz
<i>Created</i>	2023/01/18 14:38
<i>Updated</i>	2023/08/28 16:51
<i>Issue</i>	8300541

Summary

Enhance the Java programming language with *record patterns* to deconstruct record values. Record patterns and type patterns can be nested to enable a powerful, declarative, and composable form of data navigation and processing.

History



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 443: Unnamed Patterns and Variables (Preview)

<i>Owner</i>	Angelos Bimpoudis
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Closed / Delivered
<i>Release</i>	21
<i>Component</i>	specification / language
<i>Discussion</i>	amber dash dev at openjdk dot org
<i>Effort</i>	S
<i>Duration</i>	S
<i>Reviewed by</i>	Alex Buckley
<i>Endorsed by</i>	Brian Goetz
<i>Created</i>	2022/09/26 08:00
<i>Updated</i>	2023/08/28 18:58
<i>Issue</i>	8294349

Summary

Enhance the Java language with *unnamed patterns*, which match a record component without stating the component's name or type, and *unnamed variables*, which can be initialized but not used. Both are denoted by an underscore character, `_`. This is a [preview language feature](#).



JEP 444: Virtual Threads

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

Author Ron Pressler & Alan Bateman

Owner Alan Bateman

Type Feature

Scope SE

Status Closed / Delivered

Release 21

Component core-libs

Discussion [loom dash dev at openjdk dot org](#)

Relates to [JEP 436: Virtual Threads \(Second Preview\)](#)

Reviewed by Alex Buckley

Endorsed by Brian Goetz

Created 2023/03/06 18:00

Updated 2023/09/22 16:54

Issue [8303683](#)

Summary

Introduce *virtual threads* to the Java Platform. Virtual threads are lightweight threads that dramatically reduce the effort of writing, maintaining, and observing high-throughput concurrent applications.



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 446: Scoped Values (Preview)

Author Andrew Haley & Andrew Dinn

Owner Andrew Haley

Type Feature

Scope SE

Status Closed / Delivered

Release 21

Component core-libs

Discussion loom dash dev at openjdk dot org

Reviewed by Alan Bateman, Mark Reinhold

Endorsed by Brian Goetz

Created 2023/03/16 16:01

Updated 2023/09/06 16:45

Issue [8304357](#)

Summary

Introduce *scoped values*, values that may be safely and efficiently shared to methods without using method parameters. They are preferred to thread-local variables, especially when using large numbers of virtual threads. This is a [preview API](#).

In effect, a scoped value is an *implicit method parameter*. It is "as if" every method



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 453: Structured Concurrency (Preview)

Author Ron Pressler & Alan Bateman

Owner Alan Bateman

Type Feature

Scope SE

Status Closed / Delivered

Release 21

Component core-libs

Discussion [loom dash dev at openjdk dot org](#)

Relates to [JEP 437: Structured Concurrency \(Second Incubator\)](#)

Reviewed by Brian Goetz

Endorsed by Brian Goetz

Created 2023/04/21 06:18

Updated 2023/09/26 08:29

Issue [8306641](#)

Summary

Simplify concurrent programming by introducing an API for *structured concurrency*. Structured concurrency treats groups of related tasks running in different threads as a single unit of work, thereby streamlining error handling and cancellation, improving reliability, and enhancing observability. This is a [preview API](#).



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 431: Sequenced Collections

Owner Stuart Marks

Type Feature

Scope SE

Status Closed / Delivered

Release 21

Component core-libs / java.util:collections

Discussion core dash libs dash dev at openjdk dot org

Reviewed by Brian Goetz

Endorsed by Brian Goetz

Created 2022/01/27 22:13

Updated 2023/09/07 01:51

Issue [8280836](#)

Summary

Introduce new interfaces to represent collections with a defined encounter order. Each such collection has a well-defined first element, second element, and so forth, up to the last element. It also provides uniform APIs for accessing its first and last elements, and for processing its elements in reverse order.

"Life can only be understood backwards; but it must be lived forwards."

Kierkegaard



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 413: Code Snippets in Java API Documentation

Authors Jonathan Gibbons, Pavel Rappo

Owner Pavel Rappo

Type Feature

Scope JDK

Status Closed / Delivered

Release 18

Component tools / javadoc(tool)

Discussion javadoc dash dev at openjdk dot java dot net

Reviewed by Alex Buckley

Created 2018/04/13 10:54

Updated 2022/02/10 15:47

Issue [8201533](#)

Summary

Introduce an @snippet tag for JavaDoc's Standard Doclet, to simplify the inclusion of example source code in API documentation.

Goals

- Facilitate the validation of source code fragments, by providing API access to those fragments. Although correctness is ultimately the responsibility of



JEP 408: Simple Web Server

[Installing](#)
[Contributing](#)
[Sponsoring](#)
[Developers' Guide](#)
[Vulnerabilities](#)
[JDK GA/EA Builds](#)

[Mailing lists](#)
[Wiki · IRC](#)

[Bylaws · Census](#)
[Legal](#)

Workshop

JEP Process

Source code

[Mercurial](#)
[GitHub](#)

Tools

[Git](#)
[jreg harness](#)

Groups

[\(overview\)](#)

[Adoption](#)

[Build](#)

[Client Libraries](#)

[Compatibility &](#)

[Specification](#)

[Review](#)

[Compiler](#)

[Conformance](#)

[Core Libraries](#)

Owner Julia Boes

Type Feature

Scope JDK

Status Closed / Delivered

Release 18

Component core-libs / java.net

Discussion net dash dev at openjdk dot java dot net

Effort S

Duration S

Reviewed by Alex Buckley, Brian Goetz, Chris Hegarty, Daniel Fuchs

Endorsed by Brian Goetz

Created 2021/01/27 12:47

Updated 2022/03/07 10:20

Issue [8260510](#)

Summary

Provide a command-line tool to start a minimal web server that serves static files only. No CGI or servlet-like functionality is available. This tool will be useful for prototyping, ad-hoc coding, and testing purposes, particularly in educational contexts.



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 400: UTF-8 by Default

Authors Alan Bateman, Naoto Sato

Owner Naoto Sato

Type Feature

Scope SE

Status Closed / Delivered

Release 18

Component core-libs / java.nio.charsets

Discussion core dash libs dash dev at openjdk dot java dot net

Effort XS

Duration XS

Reviewed by Alex Buckley, Brian Goetz

Endorsed by Brian Goetz

Created 2017/08/31 13:16

Updated 2023/06/12 13:46

Issue [8187041](#)

Summary

Specify UTF-8 as the default charset of the standard Java APIs. With this change, APIs that depend upon the default charset will behave consistently across all implementations, operating systems, locales, and configurations.



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 421: Deprecate Finalization for Removal

Authors Brent Christian, Stuart Marks

Owner Brent Christian

Type Feature

Scope SE

Status Closed / Delivered

Release 18

Component core-libs / java.lang

Discussion core dash libs dash dev at openjdk dot java dot net

Effort S

Duration S

Reviewed by Alex Buckley, Brian Goetz, Kim Barrett

Endorsed by Brian Goetz, Mikael Vidstedt

Created 2021/09/30 20:24

Updated 2022/09/02 17:48

Issue [8274609](#)

Summary

Deprecate finalization for removal in a future release. Finalization remains enabled by default for now, but can be disabled to facilitate early testing. In a future release it will be disabled by default, and in a later release it will be removed.

OpenJDK

Installing

Contributing

Sponsoring

Developers' Guide

Vulnerabilities

JDK GA/EA Builds

Mailing lists

Wiki · IRC

Bylaws · Census

Legal

Workshop

JEP Process

Source code

Mercurial

GitHub

Tools

Git

jtreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility & Specification

Review

Compiler

Conformance

Core Libraries

Governing Board

HotSpot

JEP 439: Generational ZGC

Owner

Stefan Karlsson

Type

Feature

Scope

Implementation

Status

Completed

Release

21

Component

hotspot / gc

Discussion

hotspot dash gc dash dev at openjdk dot org

Effort

XL

Duration

XL

Relates to

JEP 377: ZGC: A Scalable Low-Latency Garbage Collector (Production)

Reviewed by

Erik Helin, Erik Österlund, Vladimir Kozlov

Endorsed by

Vladimir Kozlov

Created

2021/08/25 12:01

Updated

2023/08/14 20:24

Issue

8272979

Summary

Improve application performance by extending the Z Garbage Collector (ZGC) to maintain separate generations for young and old objects. This will allow ZGC to collect young objects — which tend to die young — more frequently.



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 430: String Templates (Preview)

<i>Owner</i>	Jim Laskey
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Closed / Delivered
<i>Release</i>	21
<i>Component</i>	specification / language
<i>Discussion</i>	amber dash dev at openjdk dot org
<i>Effort</i>	M
<i>Duration</i>	M
<i>Reviewed by</i>	Alex Buckley, Brian Goetz, Maurizio Cimadamore
<i>Endorsed by</i>	Brian Goetz
<i>Created</i>	2021/09/17 13:41
<i>Updated</i>	2023/09/06 22:45
<i>Issue</i>	8273943

Summary

Enhance the Java programming language with *string templates*. String templates complement Java's existing string literals and text blocks by coupling literal text with embedded expressions and *template processors* to produce specialized results. This is a [preview language feature and API](#).



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 445: Unnamed Classes and Instance Main Methods (Preview)

<i>Author</i>	Ron Pressler
<i>Owner</i>	Jim Laskey
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Closed / Delivered
<i>Release</i>	21
<i>Component</i>	specification / language
<i>Discussion</i>	amber dash dev at openjdk dot org
<i>Effort</i>	S
<i>Reviewed by</i>	Alex Buckley, Brian Goetz
<i>Endorsed by</i>	Brian Goetz
<i>Created</i>	2023/02/13 13:58
<i>Updated</i>	2023/08/16 16:34
<i>Issue</i>	8302326

Summary

Evolve the Java language so that students can write their first programs without needing to understand language features designed for large programs. Far from using a separate dialect of Java, students can write streamlined declarations for



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 442: Foreign Function & Memory API (Third Preview)

Owner Maurizio Cimadamore

Type Feature

Scope SE

Status Closed / Delivered

Release 21

Component core-libs

Discussion panama dash dev at openjdk dot org

Relates to [JEP 434: Foreign Function & Memory API \(Second Preview\)](#)

Reviewed by Alex Buckley, Jorn Vernee

Endorsed by Mark Reinhold

Created 2023/02/01 14:58

Updated 2023/08/29 22:29

Issue [8301625](#)

Summary

Introduce an API by which Java programs can interoperate with code and data outside of the Java runtime. By efficiently invoking foreign functions (i.e., code outside the JVM), and by safely accessing foreign memory (i.e., memory not managed by the JVM), the API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI. This is a [preview API](#).



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

JEP 448: Vector API (Sixth Incubator)

<i>Owner</i>	Paul Sandoz
<i>Type</i>	Feature
<i>Scope</i>	JDK
<i>Status</i>	Closed / Delivered
<i>Release</i>	21
<i>Component</i>	core-libs
<i>Discussion</i>	panama dash dev at openjdk dot org
<i>Effort</i>	S
<i>Duration</i>	S
<i>Relates to</i>	JEP 438: Vector API (Fifth Incubator)
<i>Reviewed by</i>	Vladimir Ivanov
<i>Endorsed by</i>	John Rose
<i>Created</i>	2023/04/11 20:18
<i>Updated</i>	2023/08/08 03:44
<i>Issue</i>	8305868

Summary

Introduce an API to express vector computations that reliably compile at runtime to optimal vector instructions on supported CPU architectures, thus achieving performance superior to equivalent scalar computations.



Future

OpenJDK

Installing

Contributing

Sponsoring

Developers' Guide

Vulnerabilities

JDK GA/EA Builds

Mailing lists

Wiki · IRC

Bylaws · Census

Legal

Workshop

JEP Process

Source code

Mercurial

GitHub

Tools

Git

jtreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility & Specification

Review

Compiler

Conformance

Core Libraries

Governing Board

HotSpot

IDE Tooling & Support

Internationalization

JMX

Members

Networking

JDK 22

This release will be the Reference Implementation of version 22 of the Java SE Platform, as specified by [JSR 397](#) in the Java Community Process.

Status

The [main-line code repository](#) is open for bug fixes, small enhancements, and JEPs as proposed and tracked via the [JEP Process](#).

Schedule

2023/12/07	Rampdown Phase One (fork from main line)
2024/01/18	Rampdown Phase Two
2024/02/08	Initial Release Candidate
2024/02/22	Final Release Candidate
2024/03/19	General Availability

Features

JEPs proposed to target JDK 22

454: [Foreign Function & Memory API](#)

456: [Unnamed Variables & Patterns](#)

460: [Vector API \(Seventh Incubator\)](#)

review ends

https://openjdk.java.net/projects/jdk/22/



JEP 454: Foreign Function & Memory API

[Installing](#)
[Contributing](#)
[Sponsoring](#)
[Developers' Guide](#)
[Vulnerabilities](#)
[JDK GA/EA Builds](#)

[Mailing lists](#)
[Wiki · IRC](#)

[Bylaws · Census](#)
[Legal](#)

Workshop

JEP Process

Source code

[Mercurial](#)
[GitHub](#)

Tools

[Git](#)
[jtreg harness](#)

Groups

[\(overview\)](#)
[Adoption](#)
[Build](#)
[Client Libraries](#)
[Compatibility & Specification](#)
[Review](#)
[Compiler](#)
[Conformance](#)
[Core Libraries](#)

Owner [Maurizio Cimadamore](#)

Type [Feature](#)

Scope [SE](#)

Status [Integrated](#)

Release [22](#)

Component [core-libs / java.lang.foreign](#)

Discussion [panama dash dev at openjdk dot org](#)

Relates to [JEP 442: Foreign Function & Memory API \(Third Preview\)](#)

Reviewed by [Alex Buckley, Jorn Vernee](#)

Endorsed by [Alan Bateman](#)

Created [2023/06/22 09:36](#)

Updated [2023/11/02 21:39](#)

Issue [8310626](#)

Summary

Introduce an API by which Java programs can interoperate with code and data outside of the Java runtime. By efficiently invoking foreign functions (i.e., code outside the JVM), and by safely accessing foreign memory (i.e., memory not managed by the JVM), the API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI.



JEP 456: Unnamed Variables & Patterns

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)

Adoption

Build

Client Libraries

Compatibility &

Specification

Review

Compiler

Conformance

Core Libraries

Owner Angelos Bimpoudis

Type Feature

Scope SE

Status Integrated

Release 22

Component specification / language

Discussion amber dash dev at openjdk dot org

Effort S

Duration S

Reviewed by Brian Goetz

Endorsed by Brian Goetz

Created 2023/07/10 16:17

Updated 2023/11/03 15:15

Issue [8311828](#)

Summary

Enhance the Java programming language with unnamed variables and unnamed patterns, which can be used when variable declarations or nested patterns are required but never used. Both are denoted by the underscore character, `_`.



JEP 460: Vector API (Seventh Incubator)

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

<i>Owner</i>	Paul Sandoz
<i>Type</i>	Feature
<i>Scope</i>	JDK
<i>Status</i>	Targeted
<i>Release</i>	22
<i>Component</i>	core-libs
<i>Discussion</i>	panama dash dev at openjdk dot org
<i>Effort</i>	XS
<i>Duration</i>	XS
<i>Relates to</i>	JEP 448: Vector API (Sixth Incubator)
<i>Reviewed by</i>	Vladimir Ivanov
<i>Endorsed by</i>	John Rose
<i>Created</i>	2023/09/08 17:29
<i>Updated</i>	2023/11/06 13:40
<i>Issue</i>	8315945

Summary

Introduce an API to express vector computations that reliably compile at runtime to optimal vector instructions on supported CPU architectures, thus achieving performance superior to equivalent scalar computations.



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

JEP 401: Value Classes and Objects (Preview)

<i>Owner</i>	Dan Smith
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Submitted
<i>Discussion</i>	valhalla dash dev at openjdk dot java dot net
<i>Effort</i>	XL
<i>Duration</i>	XL
<i>Reviewed by</i>	Brian Goetz
<i>Created</i>	2020/08/13 19:31
<i>Updated</i>	2023/10/09 23:59
<i>Issue</i>	8251554

Summary

Enhance the Java object model with *value objects*, class instances that have only final instance fields and lack object identity. This is a [preview language and VM feature](#).

Goals

- Allow developers to "opt in" to a programming model for simple values in which objects are distinguished solely by their field values, much as the



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

JEP 447: Statements before super(...) (Preview)

Author Archie Cobbs & Gavin Bierman

Owner Archie Cobbs

Type Feature

Scope SE

Status Candidate

Release 22

Component specification / language

Discussion [amber dash dev at openjdk dot java dot net](#)

Reviewed by Brian Goetz

Endorsed by Brian Goetz

Created 2023/01/20 17:33

Updated 2023/11/03 14:12

Issue [8300786](#)

Summary

In constructors in the Java programming language, allow statements that do not reference the instance being created to appear before an explicit constructor invocation. This is a [preview language feature](#).

Goals



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtregharness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

JEP 455: Primitive types in Patterns, instanceof, and switch (Preview)

<i>Owner</i>	Angelos Bimpoudis
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Candidate
<i>Component</i>	specification / language
<i>Discussion</i>	amber dash dev at openjdk dot org
<i>Effort</i>	M
<i>Duration</i>	M
<i>Created</i>	2022/06/15 10:05
<i>Updated</i>	2023/10/31 15:50
<i>Issue</i>	8288476

Summary

Enhance pattern matching by allowing primitive type patterns to be used in all pattern contexts, align the semantics of primitive type patterns with that of instanceof, and extend switch to allow primitive constants as case labels. This is a [preview language feature](#).

Goals



JEP 461: Stream Gatherers (Preview)

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtregharness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

<i>Owner</i>	Viktor Klang
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Candidate
<i>Component</i>	core-libs / java.util.stream
<i>Discussion</i>	core dash libs dash dev at openjdk dot org
<i>Reviewed by</i>	Alex Buckley, Paul Sandoz
<i>Endorsed by</i>	Paul Sandoz
<i>Created</i>	2023/10/11 13:08
<i>Updated</i>	2023/11/02 16:09
<i>Issue</i>	8317955

Summary

Enhance the [Stream API](#) to support custom intermediate operations. This will allow stream pipelines to transform data in ways that are not easily achievable with the existing built-in intermediate operations. This is a [preview API](#).

Goals

- Make stream pipelines more flexible and expressive.



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

JEP 462: Structured Concurrency (Second Preview)

Author Ron Pressler & Alan Bateman

Owner Alan Bateman

Type Feature

Scope SE

Status Candidate

Component core-libs

Discussion loom dash dev at openjdk dot org

Reviewed by Paul Sandoz

Created 2023/09/29 09:37

Updated 2023/10/27 14:31

Issue [8317302](#)

Summary

Simplify concurrent programming by introducing an API for *structured concurrency*. Structured concurrency treats groups of related tasks running in different threads as a single unit of work, thereby streamlining error handling and cancellation, improving reliability, and enhancing observability. This is a [preview API](#).

History

Structured Concurrency was proposed by [JEP 428](#) and delivered in [JDK 19](#) as an



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jtreg harness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

JEP 463: Implicitly Declared Classes and Instance Main Methods (Second Preview)

Author Ron Pressler & Jim Laskey

Owner Jim Laskey

Type Feature

Scope SE

Status Candidate

Component specification / language

Discussion amber dash dev at openjdk dot org

Effort S

Duration S

Reviewed by Gavin Bierman

Endorsed by Brian Goetz

Created 2023/08/30 18:07

Updated 2023/11/02 20:36

Issue [8315398](#)

Summary

Evolve the Java programming language so that students can write their first programs without needing to understand language features designed for large programs. Far from using a separate dialect of the language, students can write



Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

Workshop

JEP Process

Source code

Mercurial
GitHub

Tools

Git
jreg harness

Groups

(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries

JEP draft: Ahead of Time Compilation for the Java Virtual Machine

Author Julian Waters

Type Feature

Scope JDK

Status Draft

Component hotspot / compiler

Created 2023/07/28 02:38

Updated 2023/09/29 01:56

Issue [8313278](https://openjdk.org/jeps/8313278)

Summary

Enhance the Java Virtual Machine with the ability to load Java applications and libraries compiled to native code for faster startup and baseline execution.

Goals

Enable the Java Virtual Machine to load Java code compiled Ahead of Time to native code, if it was compiled by a compiler from a matching Java Virtual Machine. Allow C1, C2, and JVMCI Compilers to code native code Ahead of Time for a Java Virtual Machine to use. "Profiling" code is code compiled by C1 in tiers 2 and 3 for use as baseline execution and profiling, while "Optimized" code is compiled

JEP draft: Computed Constants (Preview)

<i>Authors</i>	Per Minborg, Maurizio Cimadamore
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Submitted
<i>Component</i>	core-libs / java.lang
<i>Effort</i>	S
<i>Duration</i>	S
<i>Reviewed by</i>	Alan Bateman
<i>Created</i>	2023/07/24 15:11
<i>Updated</i>	2023/10/27 06:35
<i>Issue</i>	8312611

Summary

Introduce *computed constants*, which are immutable value holders that are initialized at most once. Computed constants offer the performance and safety benefits of final fields, while offering greater flexibility as to the timing of initialization. This is a [preview API](#).

Goals

- Decouple the initialization of computed constants from the initialization of



More

More Ressources

- German article by me:
<https://www.innoq.com/de/articles/2023/09/java-21/>
- German podcast with me:
<https://www.innoq.com/de/podcast/135-java-21/>
- Data Oriented Programming in Java:
<https://www.infoq.com/articles/data-oriented-programming-java/>
- JavaDoc Snippets with Maven:
<https://nipafx.dev/javadoc-snippets-maven/>
- API Changes from 17 to 21:
<https://javaalmanac.io/jdk/21/apidiff/17/>
- Help for choosing a JDK distribution:
<https://whichjdk.com/>

More Ressources

- Presentation of Gatherers:
<https://www.youtube.com/watch?v=8fMFa6OqIY8>

Thanks! Questions?



Michael Vitz

Mail michael.vitz@innoq.com

X [@michaelvitz](https://twitter.com/michaelvitz)

Mastodon [@michaelvitz@innoq.social](https://mastodon.social/@michaelvitz)

LinkedIn [michaelvitz](https://www.linkedin.com/in/michaelvitz)



<https://www.innoq.com/en/talks/2023/11/das-erwartet-dich-mit-jdk-21/>

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Wendenstraße 130
20537 Hamburg

Spichernstraße 44
50672 Köln

Königstorgraben 11
90402 Nürnberg