


# JCP EC Update: Musings from DevRel

October 16<sup>th</sup>, 2024

**Chad Arimura**

Vice President, Java Developer Relations

 @chadarimura



# Java DevRel

- Perception: “Java is your parents boring language, and nobody uses it anymore.”
- Reality: “Java powers half the world’s GDP across billions of devices and even with its strong commitment to compatibility, language evolution is outpacing all other major languages.”
- Our job is to communicate this reality sprinkled with a bit of future direction
- “Perception == Reality”
- The existential threat to Java is not each other, but this perception growing
- So what can we do.....

# Java SWOT

## Strengths

- Java code is everywhere
- Sprawling ecosystem of libraries
- Educators don't want to learn other languages
- Designed for teams, large projects, and enterprise development

## Weaknesses

- Old Java code is everywhere
- Sprawling ecosystem of unmaintained libraries
- Educators don't want to learn Java beyond 8
- Not well suited for hackers, hobbyists, small programs

## Opportunities

- Platform momentum
- Java's 30<sup>th</sup> anniversary
- JavaOne
- JUGs
- JCP
- Younger gen, clean slate, contrarian

## Threats

- Python
- Gravity of perception too strong
- Infighting
- Perception that Oracle is big bad wolf

# Turning Weaknesses Around

- Old Java code is everywhere
  - Create and release new code (Hackathons, grants, open up projects)
  - Fix the Internet (joke, but.. How? StackOverflow? → ...feeds LLMs)
- Sprawling ecosystem of unmaintained libraries
  - Encourage Tip & Tail model
  - Phase out unmaintained libraries
- Educators don't want to learn Java beyond 8
  - Heather Stephens has entire presentations on this... 😊
- Not well suited for hackers, hobbyists, small programs
  - Amber, paving the onramp, JShell, multi-file source code launcher, etc.

# Java Launch Briefings

# JDK 23

12 JDK Enhancement Proposals (JEPs) delivered

*Plus thousands of performance, stability and security updates*

## Language Improvements

- 455: Primitive Types in Patterns, instanceof, and switch (Preview)
- 476: Module Import Declarations (Preview)
- 477: Implicitly Declared Classes and Instance Main Methods (Third Preview)
- 482: Flexible Constructor Bodies (Second Preview)

## Performance and Runtime

- 474: ZGC: Generational Mode by Default

## Tools

- 467: Markdown Document Comments

## Libraries

- 466: Class-File API (Second Preview)
- 469: Vector API (Eighth Incubator)
- 473: Stream Gatherers (Second Preview)
- 480: Structured Concurrency (Third Preview)
- 481: Scoped Values (Third Preview)

## Integrity

- 471: Deprecate the Memory-Access Methods in sun.misc.unsafe for Removal

[openjdk.org/projects/jdk/23](https://openjdk.org/projects/jdk/23)

# Paving the On-Ramp (JEP 477 | Third Preview)

Before

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

Classes for modeling  
and organization

Static vs instance  
behavior

Command line  
interaction, arrays

Magic method name

Access control

Static fields

After

```
void main() {  
    println("Hello!");  
}
```

# Java plug-in for VS Code

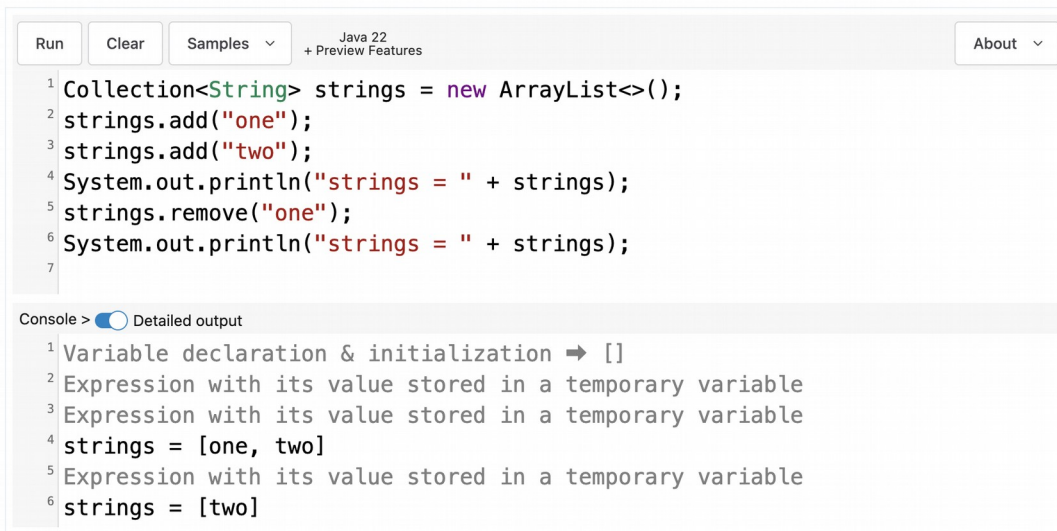
- Over **1M downloads** (5.0 out of 5.0 rating)
- Java Platform plugin availability in the VS Code marketplace
- Exists with other Oracle plugins in the VS Code marketplace
- Oracle's VS Code extension for Java is based on the OpenJDK's `javac` compiler for code editing and compilation and on the OpenJDK's debugger interface for debugging
- Allows Oracle to offer VS Code IDE support for new JDK features as soon as they are introduced, even during Early Access of the JDK
- Oracle's VS Code extension supports the current JDK releases as well as the next upcoming JDK version





# Java Playground Inline Tutorials

You can run the following example. Here, you create an instance of the [Collection](#) interface using the [ArrayList](#) implementation. The generics used tells the Java compiler that you want to store [String](#) objects in this collection. [ArrayList](#) is not the only implementation of [Collection](#) you may use. More on that later.



The screenshot shows a Java Playground interface. At the top, there are buttons for 'Run', 'Clear', and 'Samples', along with 'Java 22 + Preview Features' and an 'About' dropdown. The code editor contains the following code:

```
1 Collection<String> strings = new ArrayList<>();
2 strings.add("one");
3 strings.add("two");
4 System.out.println("strings = " + strings);
5 strings.remove("one");
6 System.out.println("strings = " + strings);
7
```

Below the code editor, the console output is displayed with 'Detailed output' checked. The output shows the state of the 'strings' variable at each step:

```
1 Variable declaration & initialization → []
2 Expression with its value stored in a temporary variable
3 Expression with its value stored in a temporary variable
4 strings = [one, two]
5 Expression with its value stored in a temporary variable
6 strings = [two]
```

Running the previous code should print the following:

```
1 | strings = [one, two]
2 | strings = [two]
```

# Headlines

InfoWorld

Home - Artificial Intelligence - Can Java rival Python in AI development?


by Paul Krill  
Editor at Large

## Can Java rival Python in AI development?

Feature  
Sep 18, 2024 - 4 mins

Artificial Intelligence Generative AI Java

Java proponents see the language gaining traction in AI and machine learning as AI becomes incorporated into business logic.



CREDIT: FOTOKAZA / SHUTTERSTOCK

Can **Java** give **Python** a run for its money in the burgeoning, trendy AI space? While **Python still gets top billing when it comes to developing for AI**, Java proponents see the nearly 30-year-old Java programming language ready to take charge in the AI field.

ITProToday

NEWSLETTER SIGN-UP


Cloud OS IT Mgmt Career Storage Security Dev DX Infrastructure More

## Java 23 Modernizes Open Source Language to Better Compete With Python, Rust

The latest version of Java adds incremental preview capabilities as the open source programming language continues to modernize and improve.

Sean Michael Kermer, Contributor  
September 17, 2024

4 Min Read Editor's Choice



ALAMY

in f X e ↻

Java is among the most widely used enterprise programming languages, and in some circles it has a reputation for being a less modern language like Python and Rust.

THE NEW STACK

PODCASTS EBOOKS EVENTS NEWSLETTER CONTRIBUTE

ARCHITECTURE ENGINEERING OPERATIONS

## Oracle Unveils Java 23: Simplicity Meets Enterprise Power

Oracle unveiled its latest Java 23 update, promising simpler development, better performance, and performance enhancements for enterprise applications.

Sep 17th, 2024 1:01pm by Darryl K. Taft

ADT MAG  
APPLICATION DEVELOPMENT TRENDS

AI	ALM/TESTING	CITIZEN DEVELOPER
CLOUD	DATA	DEVOPS
ENTERPRISE	IOT	JAVA
LOW CODE	MOBILE	OPEN SOURCE
SECURITY	WEB DEV	WATERS WORKS

## WatersWorks

by John K. Waters  
Blog Home

## JDK 23 Arrives with AI, Performance, and Security Enhancements

Oracle announced the general availability of JDK 23 as the work's feature release with enhanced capabilities focused on cloud-native applications, enterprise performance, and the growing demands of AI.

Available now, this release of the Java dev kit includes 12 new JDK enhancement proposals (JEPs), as well as thousands of smaller performance, stability, and security updates. It builds on its predecessors, of course, with updates in areas such as concurrency, memory management, and language simplicity, ensuring that Java—which just passed the three-decade mark—remains one of the most popular and widely-used programming languages in the world.

## AppTrends

Sign up for our newsletter.

Email Address\*

Country\*

I agree to this site's Privacy Policy

Please type the letters/numbers you see above.

SUBMIT

# DevRel

# Q1 By the Numbers

- JVM Language Summit
- We were at JJUG, KCDC, DevBCN, JNation, CSTA, CRA, JavaZone, and 21 other events/JUGs
- YouTube hours watched up 40% YoY
- YouTube impressions up 40% YoY
- Dev.java & Inside.java monthly unique visitors up 14% YoY
- Inside Java Newsletter 10% open rate

# JVM Language Summit

## JVM Language Summit

**August 5-7, 2024**

**Oracle Auditorium**

**Santa Clara, California, USA**

We are pleased to announce the 2024 JVM Language Summit to be held at Oracle's Santa Clara campus on August 5-7, 2024.

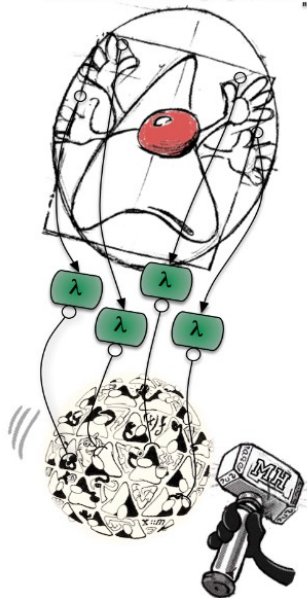
The JVM Language Summit is an open technical collaboration among language designers, compiler writers, tool builders, runtime engineers, and VM architects. We will share our experiences as creators of both the JVM and programming languages for the JVM. We also welcome non-JVM developers of similar technologies to attend or speak on their runtime, VM, or language of choice.

Presentations will be recorded and made available to the public.

This event is being organized by language and JVM engineers—no marketers involved! So bring your slide rules and be prepared for some seriously geeky discussions.

The Summit will be followed by the [OpenJDK Committers' Workshop](#) on August 8-9.

```
implementors().stream()
    .filter(i -> i.projects().any(():isLanguage))
    .forEach(i -> i.hail("
Hello language implementors!
Join us in person August 5-7, 2024
for the 14th JVM Language Summit..."))
```



## Project Leyden

JVM Language Summit 2024

## JVM Language Summit 2024

Java

Public

12 videos 3,175 views Updated today



▶ Play all

↻ Shuffle

The JVM Language Summit is an open technical collaboration among language designers, compiler writers, tool builders, runtime engineers, and VM architects. We share our experiences as creators of both the JVM and programming languages for the JVM. We also welcome non-JVM developers of similar technologies to attend or speak on their runtime, VM, or language of choice.

Sort



### Project Leyden Update #JVMLS

Java • 8K views • 4 weeks ago



### Project Babylon - Code Reflection #JVMLS

Java • 8K views • 1 month ago



### Valhalla - Where Are We? #JVMLS

Java • 20K views • 1 month ago



### An Opinionated Overview on Static Analysis for Java #JVMLS

Java • 6.6K views • 1 month ago



### Rethinking Java String Concatenation #JVMLS

Java • 11K views • 1 month ago



### A New Model for Java Object Initialization #JVMLS

Java • 10K views • 3 weeks ago

# JUGs



# Java 23 Live Stream

## Deep Dive into Java 23's Features! 🖱️

- 13:00: Welcome and event agenda 📅
- 13:05: Java Language Update by Gavin Bierman 😊
- 13:20: Java Language Features in Action 🎬
- 13:35: Integrity by Default with Ron Pressler 🌐
- 13:55: Markdown Documentation Comments w Jonathan Gibbons 🗨️
- 14:10: ZGC: Generational Mode by Default w Stefan Karlsson 🤖

## Java Outreach in 2024 📶

- 14:25: Java Playground Updates with Denys Makogon and Heather Stephens 📺
- 14:40: Java Community Report from Sharat Chander and Jim Grisanzio 🚀

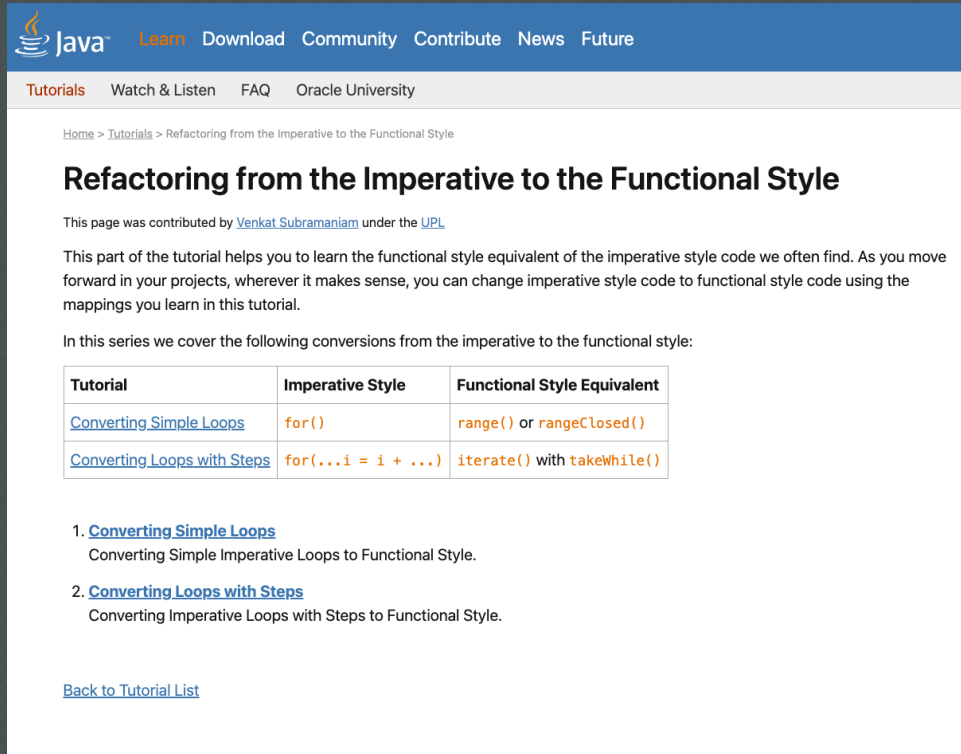
## Java Preview Features and Future work 🌐

- 15:10: Sneak Peek at GraalVM Updates with Alina Yurenko and Fabio Niephaus 🐰
- 15:30: Class-File API with Brian Goetz 🌟
- 15:45: State of Project Leyden with John Rose ✨
- 16:15: Enjoy Your Evening! 🌍



8k live views  
16k views as of 9/23

# Community Contributions



The screenshot shows the Java website navigation bar with links for Learn, Download, Community, Contribute, News, and Future. Below the navigation bar, the breadcrumb trail reads 'Home > Tutorials > Refactoring from the Imperative to the Functional Style'. The main heading is 'Refactoring from the Imperative to the Functional Style'. A sub-heading states 'This page was contributed by Venkat Subramaniam under the UPL'. The text explains that the tutorial helps learn the functional style equivalent of imperative code. A list of conversions is provided, followed by a table with two columns: 'Imperative Style' and 'Functional Style Equivalent'. The table lists two conversions: 'Converting Simple Loops' (for() to range() or rangeClosed()) and 'Converting Loops with Steps' (for(...i = i + ...) to iterate() with takeWhile()). Below the table, two numbered items describe the conversions. A link 'Back to Tutorial List' is at the bottom.

Home > Tutorials > Refactoring from the Imperative to the Functional Style

## Refactoring from the Imperative to the Functional Style

This page was contributed by [Venkat Subramaniam](#) under the [UPL](#).

This part of the tutorial helps you to learn the functional style equivalent of the imperative style code we often find. As you move forward in your projects, wherever it makes sense, you can change imperative style code to functional style code using the mappings you learn in this tutorial.

In this series we cover the following conversions from the imperative to the functional style:

Tutorial	Imperative Style	Functional Style Equivalent
<a href="#">Converting Simple Loops</a>	<code>for()</code>	<code>range()</code> or <code>rangeClosed()</code>
<a href="#">Converting Loops with Steps</a>	<code>for(...i = i + ...)</code>	<code>iterate()</code> with <code>takeWhile()</code>

- [1. Converting Simple Loops](#)  
Converting Simple Imperative Loops to Functional Style.
- [2. Converting Loops with Steps](#)  
Converting Imperative Loops with Steps to Functional Style.

[Back to Tutorial List](#)

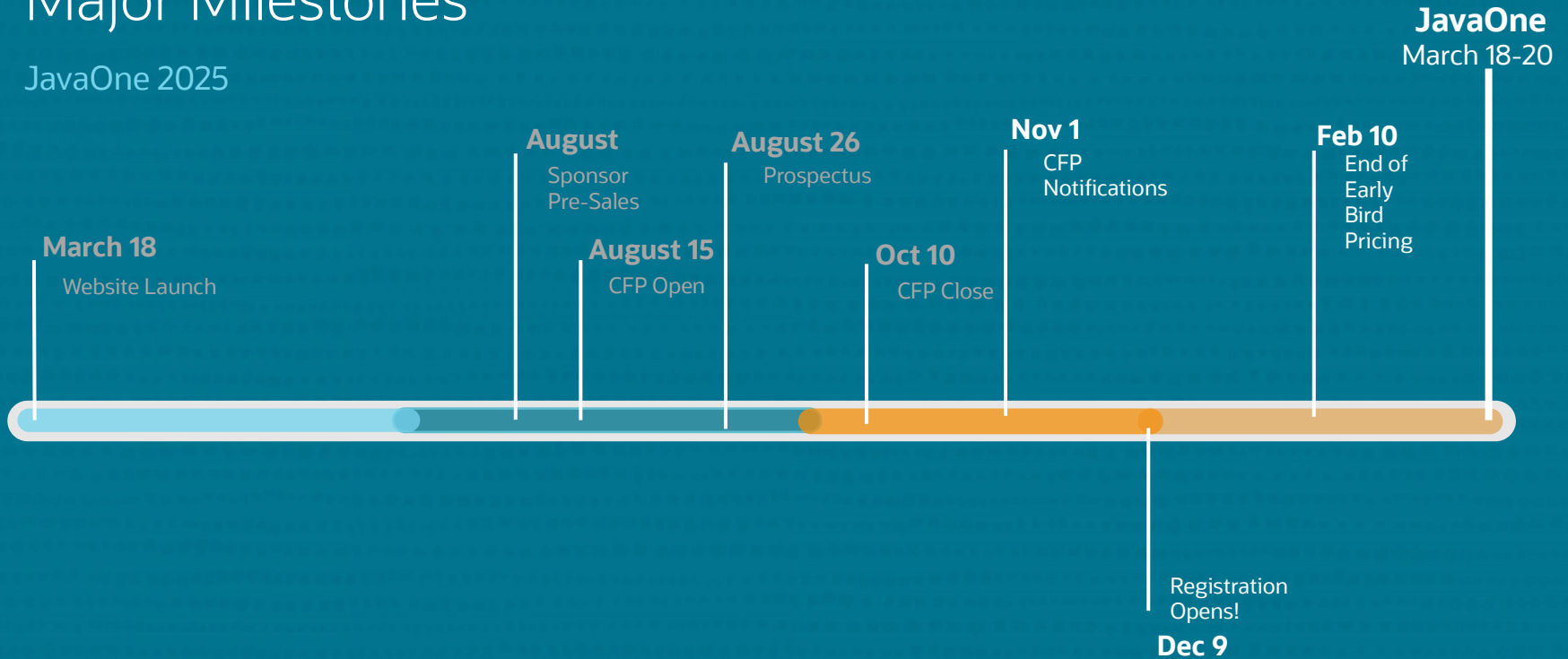




# JavaOne Update

# Major Milestones

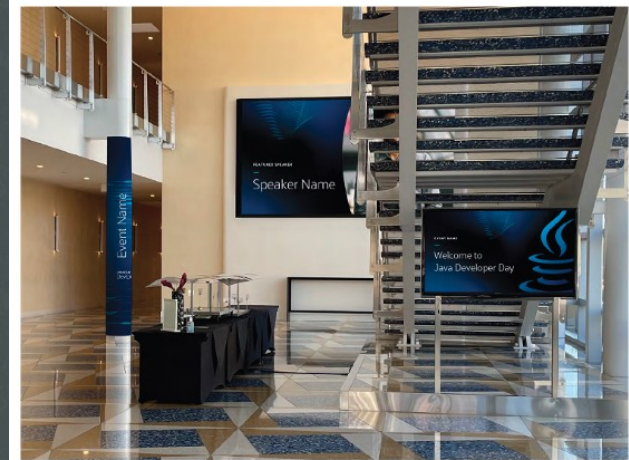
JavaOne 2025



# Venue: Oracle Conference Center



Oracle Conference Center exterior entryway



Oracle Conference lobby

Mon. 3/17

Tues. 3/18

Wed. 3/19

Thur. 3/20

7:00

8:00

9:00

10:00

11:00

12:00

1:00

2:00

3:00

4:00

5:00

6:00

7:00

8:00

Ancillary events

Breakfast  
7:30-8:30am

Keynote  
8:30am-9:30am

Breakouts  
10:00-12:00pm

Lunch  
12:00pm—1:00pm

Breakouts  
1:00-6:00pm

Happy Hour  
6:00-7:00pm

Pavilion

Pavilion

Breakfast  
7:30-8:30am

Keynote  
8:30am-9:30am

Breakouts  
10:00-12:00pm

Lunch  
12:00pm—1:00pm

Breakouts  
1:00-6:00pm

Party  
7:00-10:00pm

Pavilion

Pavilion

Breakfast  
7:30-8:30am

Keynote  
8:30am-9:30am

Breakouts  
10:00-12:00pm

Lunch  
12:00pm—1:00pm

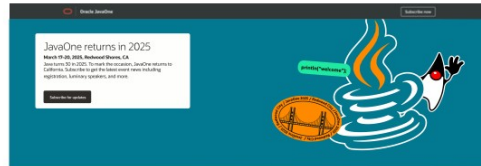
Breakouts  
1:00-4:00pm

# Tracks

1. Core Java Platform
2. UI, Client, and Frontend
3. Enterprise Java, Cloud, and Database
4. Machine Learning and Artificial Intelligence
5. Application Manageability and Tooling
6. The Next Generation: Community, People, and Culture



# Concept



# Last Thoughts

1. Will stream Java 24 launch live from event
2. Over 700 submissions for ~30 open slots
3. Very limited space
4. Very limited sponsorships



# Thank you

<https://openjdk.org>

<https://dev.java>

<https://inside.java>

<https://youtube.com/java>

<https://github.com/java>

 @Java | @OpenJDK