

Java 17

Die relevanten Features der neuen LTS-Version

FALK SIPPACH // EMBARC

BaselOne 2021

Donnerstag, 21.10.2021



1

Java 17 - Was bringt die neueste LTS-Version?

Seit einigen Jahren kommen nun schon halbjährlich neue Java Major-Releases heraus. Dieses Vorgehen hat sich etabliert und funktioniert erstaunlich gut. Natürlich dürft Ihr nicht den Funktionsumfang von den früheren Versionen (9 und älter) erwarten. Dafür bekommt Ihr als Entwickler aber viel regelmäßiger die aktuellen Änderungen mit. In den Preview-Phasen kann sogar Feedback gegeben und somit die aktive Weiterentwicklung von Java mit gestaltet werden. Alle drei Jahre erscheinen zudem Long-Term-Support-Versionen, die länger mit Updates und Patches versorgt werden. Im September 2021 ist mit der Version 17 wieder ein solches LTS-Release herausgekommen, für welches in den letzten 3 Jahren jede Menge interessante neue Funktionen entwickelt wurden:

- **Switch Expressions**
- **Text Blocks**
- **Pattern Matching für instanceof**
- **Helpful NullPointerExceptions**
- **Records**
- **Sealed Classes**
- **und noch einiges mehr**

In diesem Vortrag betrachten wir außerdem hilfreiche API-Verbesserungen und Änderungen an der JVM, z. B. bei den Garbage Collectoren. Falls Ihr selbst noch nicht bei Java 11 oder höher angekommen seid, bekommt Ihr damit auch einen kompakten Überblick, wie sich Java aktuell anfühlt.

2

Falk Sippach

- Softwarearchitekt, Berater, Trainer bei embarc
- früher bei Orientation in Objects (OIO), Trivadis

Schwerpunkte:

- Architekturberatung und -bewertung
- Cloud- und Java-Technologien



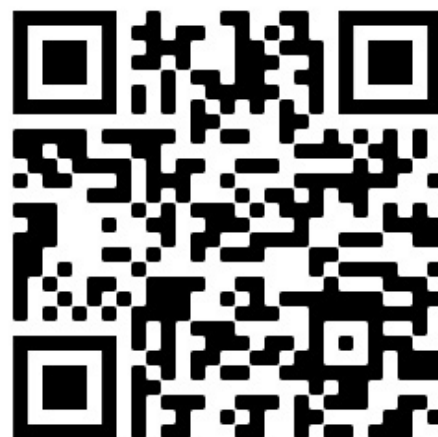
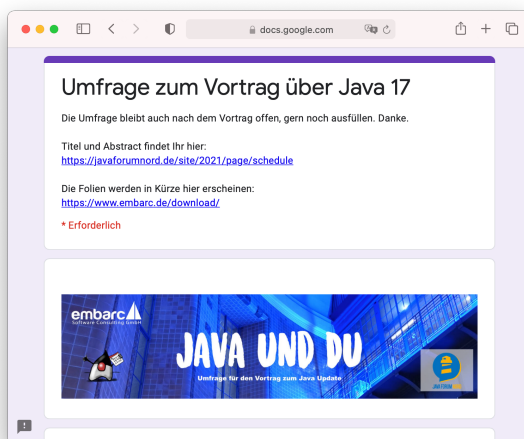
✉ fs@embarc.de

🐦 @sipsack

🔗 → [xing.to/fsi](https://www.xing.com/profile/falk_sippach)

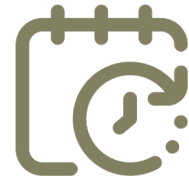


Wer seid Ihr?



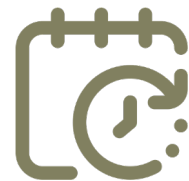
<https://tinyurl.com/bo-java17>

Agenda



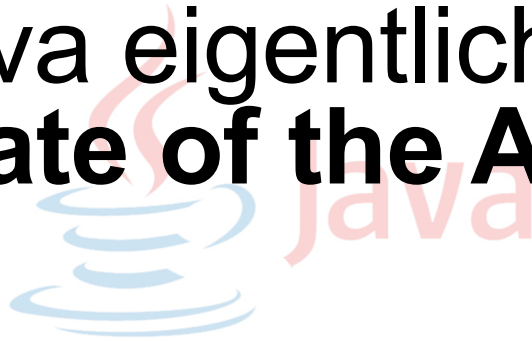
- 1 Einführung
- 2 Was bisher geschah
- 3 Auf dem Weg zu Java 17
- 4 Neue Features im Blick
- 5 Ausblick

Agenda



- 1 Einführung**
- 2 Was bisher geschah
- 3 Auf dem Weg zu Java 17
- 4 Neue Features im Blick
- 5 Ausblick

Ist Java eigentlich noch State of the Art?



"Dieses Foto" von Unbekannter Autor ist lizenziert gemäß [CC BY-SA](#)

1995



1996



heute



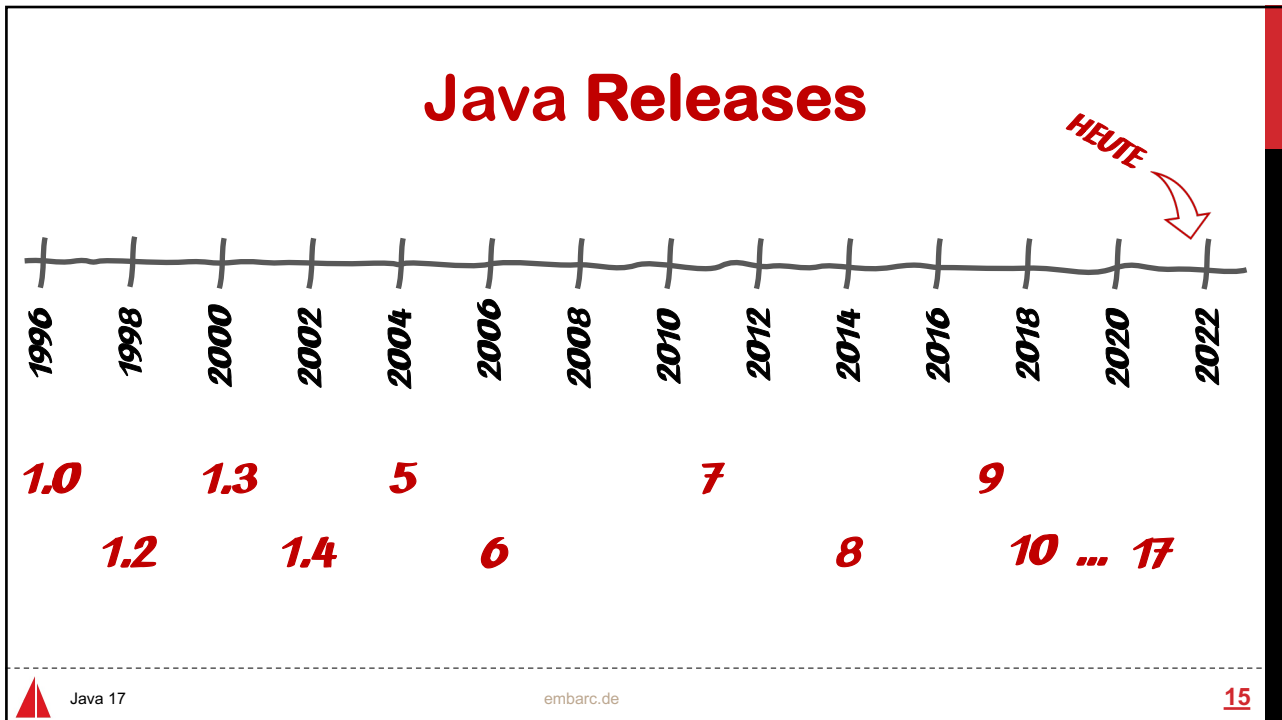
WORA

Write once, run anywhere

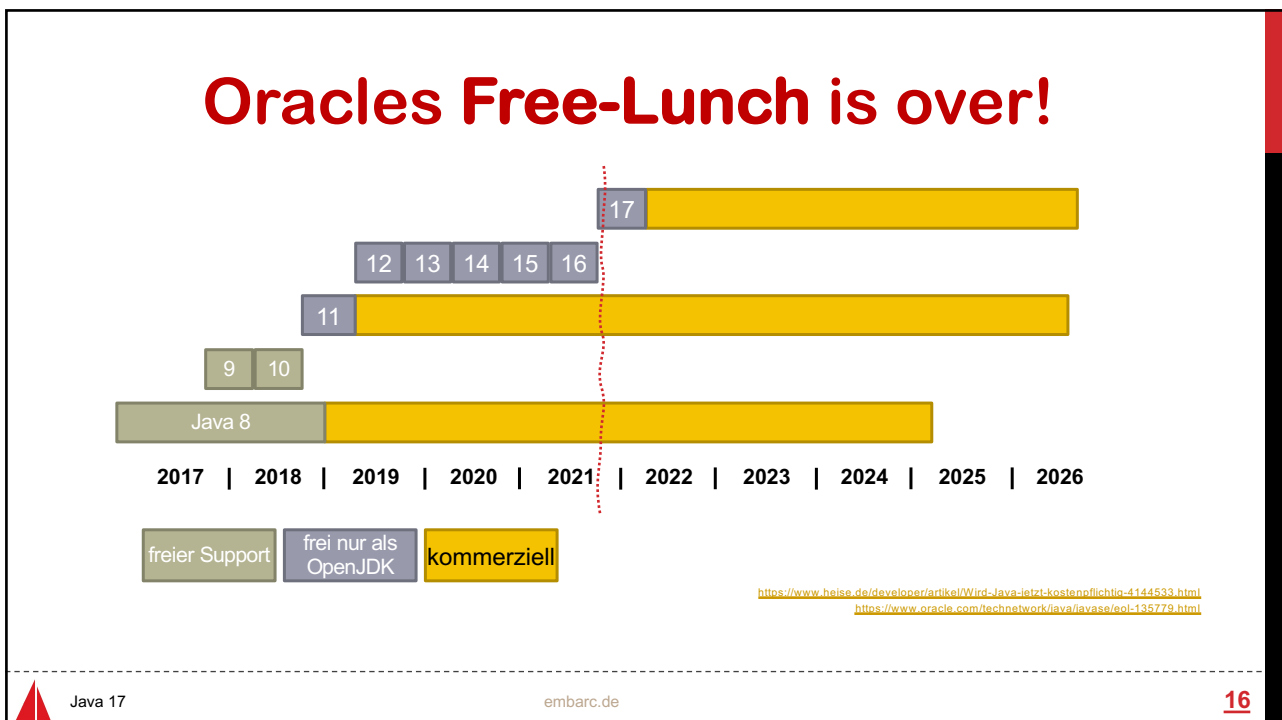
GraalVM™



**CLOUD
NATIVE**

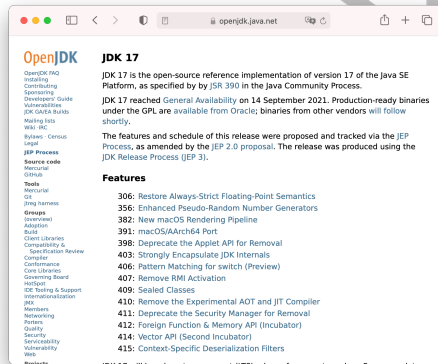


15



16

Kurze oder lange Geschichte



Gerrit Grunwald
@hansolo_

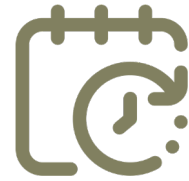


If you just look from one [#jdk](#) to the next, [#jdk17](#) doesn't look that impressive...BUT when you realize that it contains all the features from [#jdk12](#), [#jdk13](#), [#jdk14](#), [#jdk15](#) and [#jdk16](#)...it has a lot of really good stuff to offer...just say'n [#java](#) rocks 😊👍

[Tweet übersetzen](#)

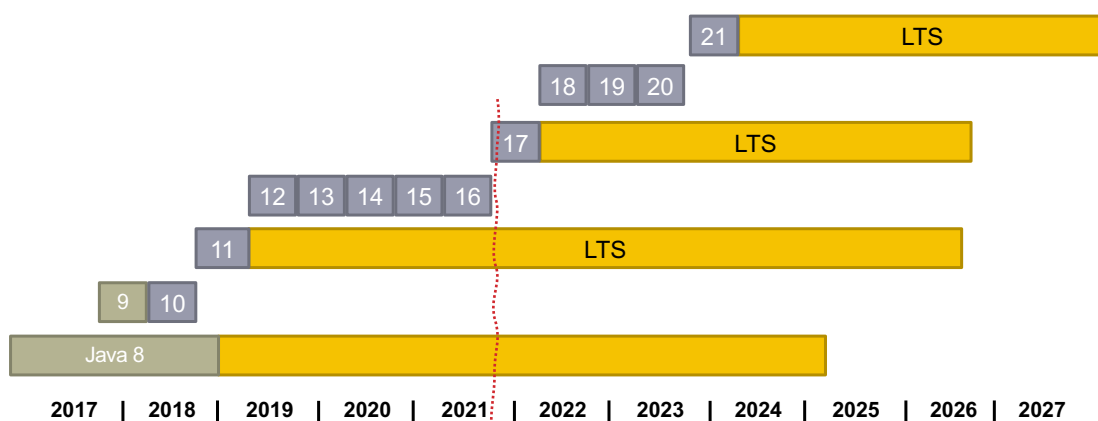
9:47 nachm. · 10. Sep. 2021 · Echofon

Agenda



- 1 Einführung
- 2 Was bisher geschah
- 3 Auf dem Weg zu Java 17
- 4 Neue Features im Blick
- 5 Ausblick

Release Train



Zoo von JDKs

OpenJDK



ADOPTIUM

Amazon Corretto 8
Now Generally Available




OpenJ9








ORACLE



Java 17



Alibaba Dragonwell



embarc.de



21

21


OpenJDK

JDK 17


JDK 17 is the open-source reference implementation of version 17 of the Java SE Platform, as specified by by JSR 390 in the Java Community Process.

JDK 17 reached **General Availability** on 14 September 2021. Production-ready binaries under the GPL are **available from Oracle**; binaries from other vendors **will follow shortly**.

<http://openjdk.java.net/projects/jdk/17/>



Java 17



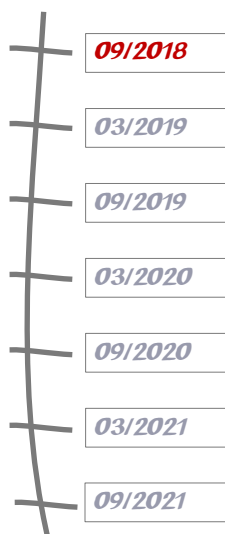
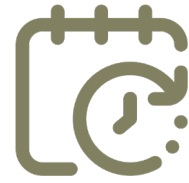
embarc.de

22

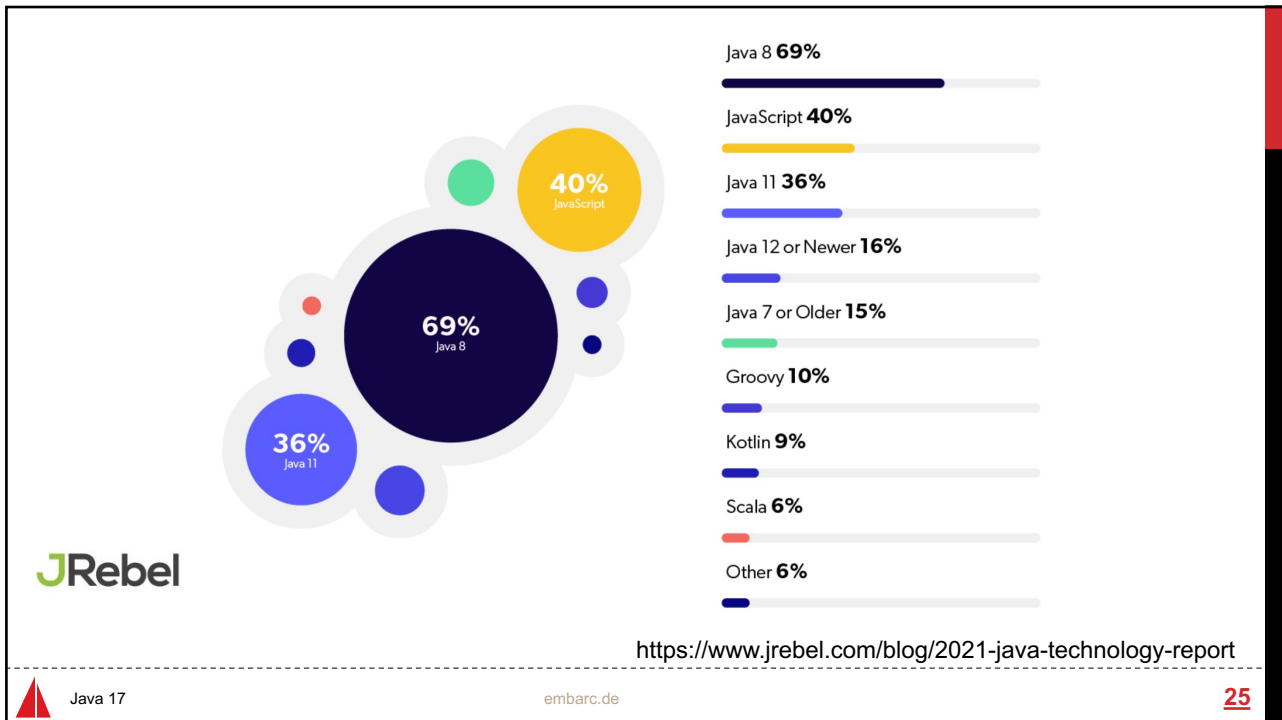
22

Agenda

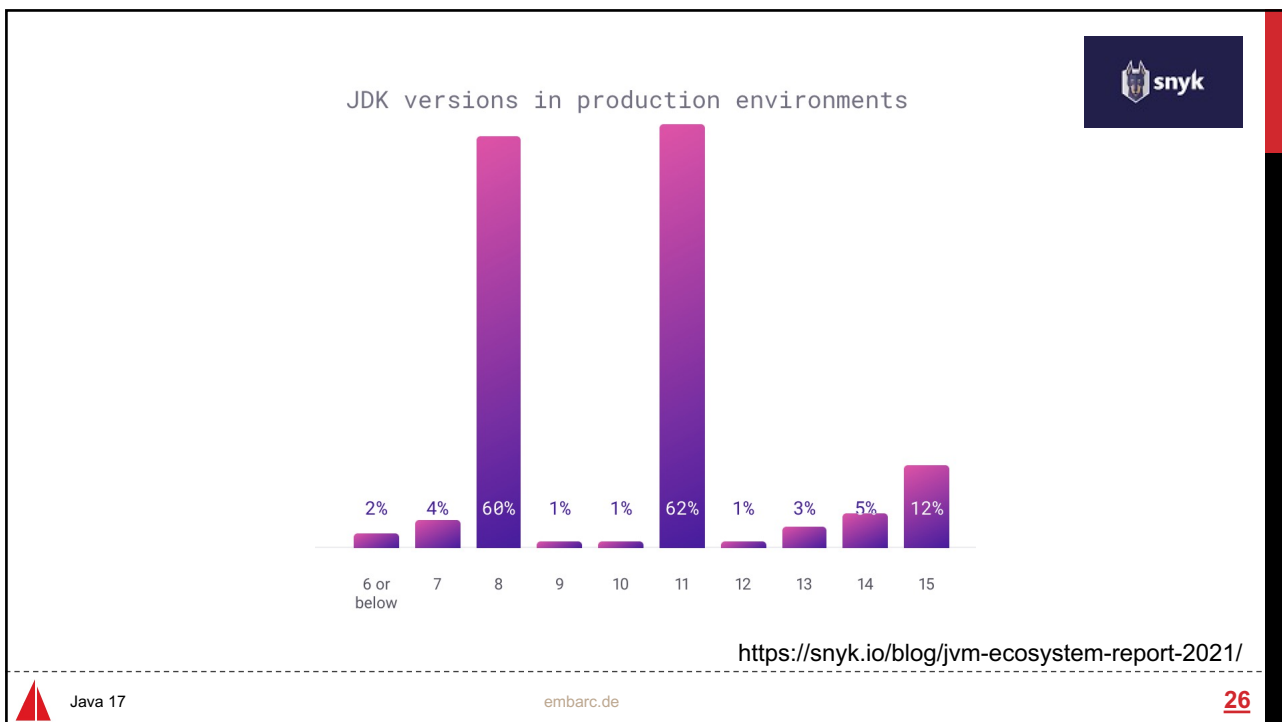
- 1 Einführung
- 2 Was bisher geschah
- 3 Auf dem Weg zu Java 17**
- 4 Neue Features im Blick
- 5 Ausblick



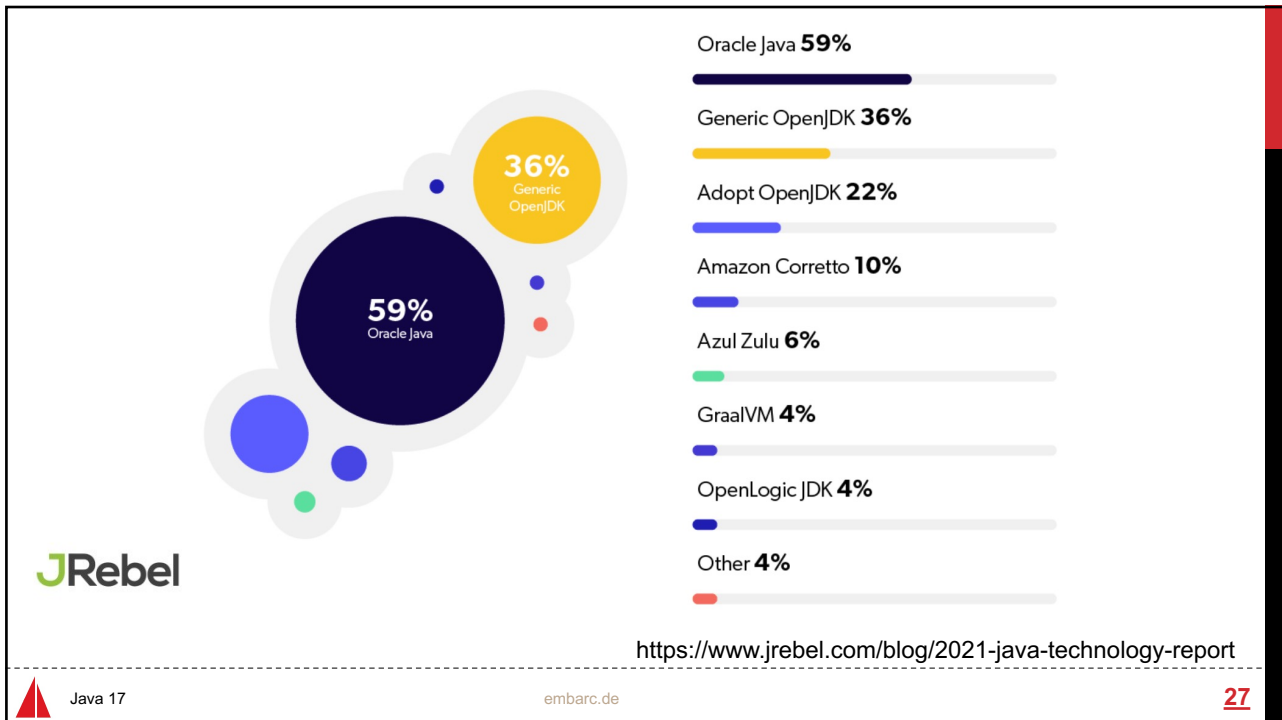
11



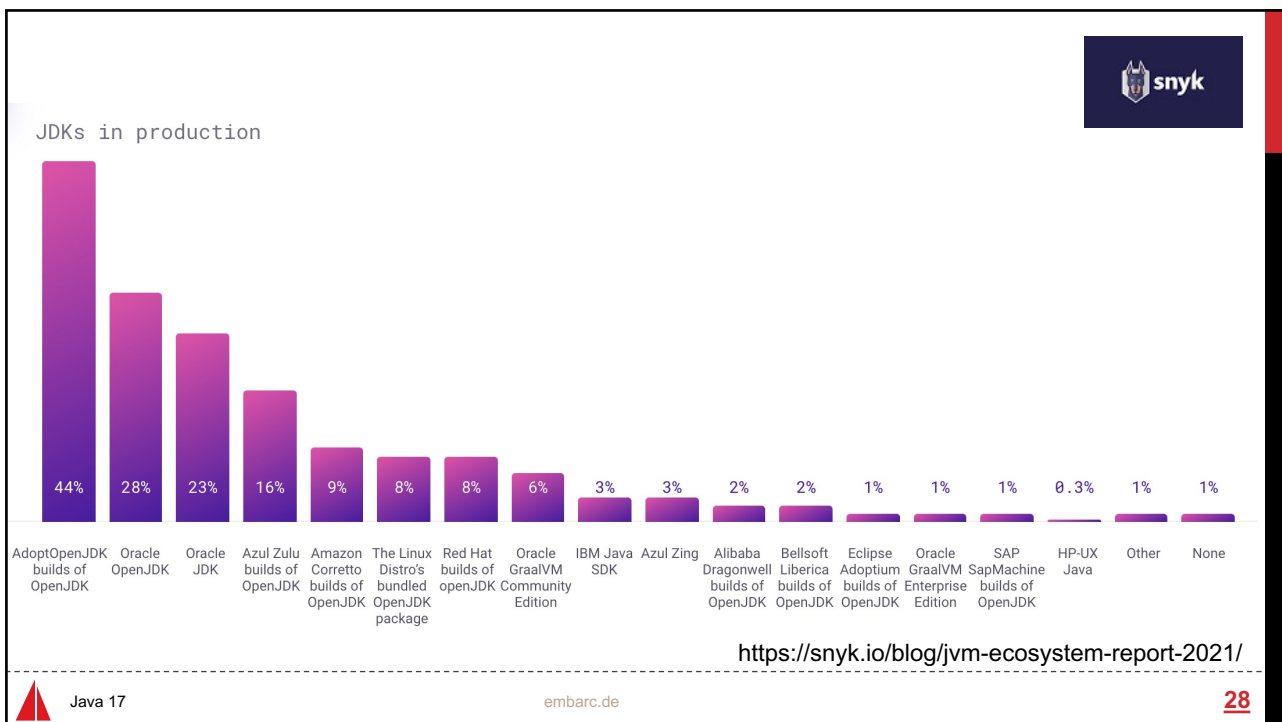
25



26



27



28



03/2019

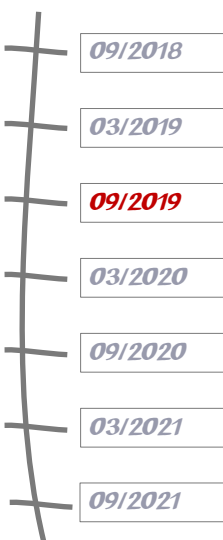
Features

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



12


 Java 17 embarc.de 29




09/2019

Features

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



13

 Java 17 embarc.de 30

Features

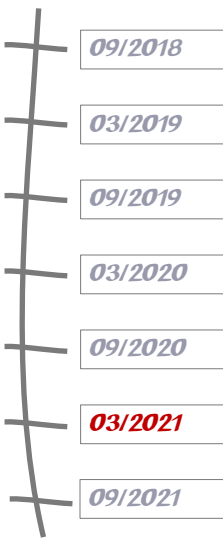
- 305: Pattern Matching for instanceof (Preview)
- 343: Packaging Tool (Incubator)
- 345: NUMA-Aware Memory Allocation for G1
- 349: JFR Event Streaming
- 352: Non-Volatile Mapped Byte Buffers
- 358: Helpful NullPointerExceptions
- 359: Records (Preview)
- 361: Switch Expressions (Standard)
- 362: Deprecate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

Java 17
embarc.de
31

Features


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


Java 17
embarc.de
32

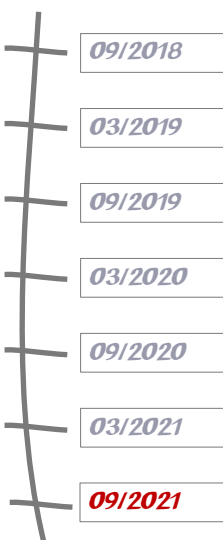


Features

- 338: Vector API (Incubator)
- 347: Enable C++14 Language Features
- 357: Migrate from Mercurial to Git
- 369: Migrate to GitHub
- 376: ZGC: Concurrent Thread-Stack Processing
- 380: Unix-Domain Socket Channels
- 386: Alpine Linux Port
- 387: Elastic Metaspace
- 388: Windows/AArch64 Port
- 389: Foreign Linker API (Incubator)
- 390: Warnings for Value-Based Classes
- 392: Packaging Tool
- 393: Foreign-Memory Access API (Third Incubator)
- 394: Pattern Matching for instanceof
- 395: Records
- 396: Strongly Encapsulate JDK Internals by Default
- 397: Sealed Classes (Second Preview)



16


 Java 17
embarc.de
33



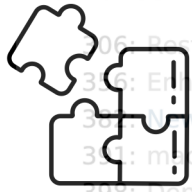
Features

- 306: Restore Always-Strict Floating-Point Semantics
- 356: Enhanced Pseudo-Random Number Generators
- 382: **New macOS Rendering Pipeline**
- 391: macOS/AArch64 Port
- 398: Deprecate the Applet API for Removal
- 403: Strongly Encapsulate JDK Internals
- 406: Pattern Matching for switch (Preview)
- 407: Remove RMI Activation
- 409: Sealed Classes
- 410: Remove the Experimental AOT and JIT Compiler
- 411: Deprecate the Security Manager for Removal
- 412: Foreign Function & Memory API (Incubator)
- 414: Vector API (Second Incubator)
- 415: **Context-Specific Deserialization Filters**



17

 Java 17
embarc.de
34


Features




Neue Features




Internas

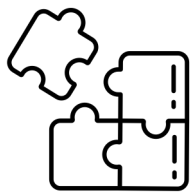


Garbage Collectoren




API-Änderungen (JDK)


 Java 17
embarc.de
35




Neue Features




Internas




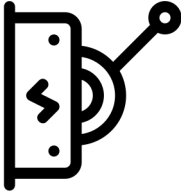

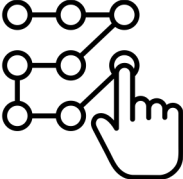



Garbage Collectoren



API-Änderungen (JDK)


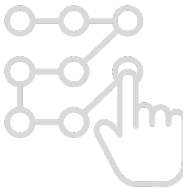

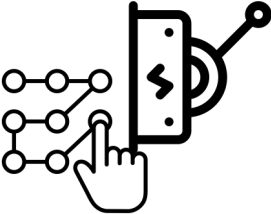

 Java 17
embarc.de
36



<p>12, 13, 14</p>  <p>Switch Expression</p>	<p>13, 14, 15</p>  <p>Text Blocks</p>	<p>14, 15, 16</p>  <p>Pattern Matching for instanceof</p>
<p>15, 16</p>  <p>Helpful NullPointerExceptions</p>	<p>14, 15, 16</p>  <p>Records</p>	<p>15, 16, 17</p>  <p>Sealed Classes</p>


Java 17 embarc.de 37

37


 <p>Switch Expression</p>	 <p>Pattern Matching for instanceof</p>	
 <p>Records</p>	 <p>Pattern Matching for switch</p>	 <p>Sealed Classes</p>

Java 17 embarc.de 38


38




16




**Mercurial => Git,
Umzug nach Github**

 Java 17 embarc.de 39


39



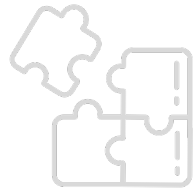
16



**Packaging
Tool**

 Java 17 embarc.de 40

40



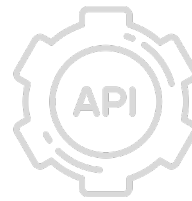
Neue Features



Garbage Collectoren



Internas



API-Änderungen (JDK)

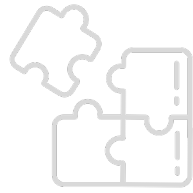


Garbage Collectoren

Neue Generation von Low-Latency Garbage Collectoren

moderne Architekturen: Multi-Core + TB RAM

Ziel: kurze GC-Pausen im ms-Bereich



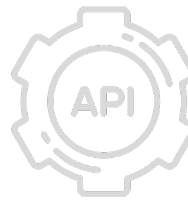
Neue Features



Garbage Collectoren



Internas



API-Änderungen (JDK)

14, 15, 16, 17

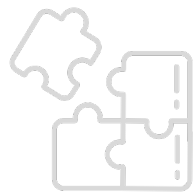


Foreign Linker +
Foreign-Memory
Access API



16, 17 $\begin{Bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{Bmatrix}$

Vector API



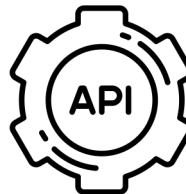
Neue Features




Garbage Collectoren



Internas



API-Änderungen (JDK)



JDK

The Java Version Almanac
javaalmanac.io

The Java Version Almanac / JDK Releases / Java 17 / Feedback on this page?


New APIs in Java 17

Comparing **Java 17** (17+35-2724-open) with **Java 11** (11.0.12+7-tem).

Element	Modification
java.base	
java.io	
@ Serial	added
CharArrayReader	
• read(CharBuffer)	added
Console	
• charset()	added
FileInputStream	
• finalize()	removed
• readAllBytes()	added
• readNBytes(int)	added
FileOutputStream	
• finalize()	removed

<https://javaalmanac.io/jdk/17/apidiff/11/>

47



Java 17

embarc.de

47


```

var s = Stream.of(1, 2, 3, 4, 5);


var l1 = s.collect(Collectors.toList());

var l2 = s.toList();


```



JDK



16

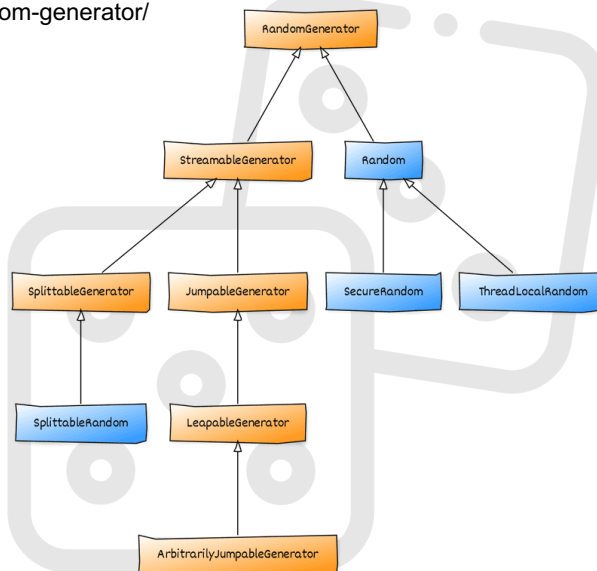
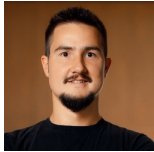


Java 17

embarc.de

48

<https://nipafx.dev/java-random-generator/>



JDK



CREATED WITH YUML

Auf Wiedersehen ...

Applet API deprecated for Removal

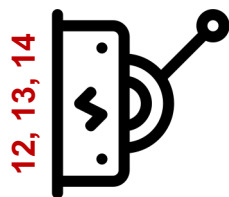
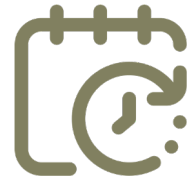
Security Manager deprecated for Removal

Experimental AOT and JIT Compiler

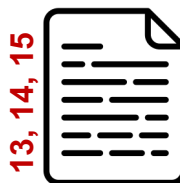


Agenda

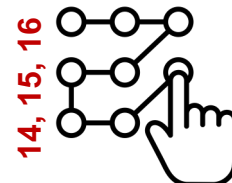
- 1 Einführung
- 2 Was bisher geschah
- 3 Auf dem Weg zu Java 17
- 4 **Neue Features im Blick**
- 5 Ausblick



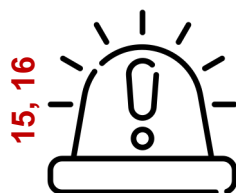
Switch Expression



Text Blocks



Pattern Matching
for instanceof



Helpful NullPointerExceptions



Records



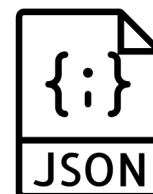
Sealed Classes



```
private static String developerRating( int numberOfChildren ) {  
    return switch (numberOfChildren) {  
        case 0 -> "open source contributor";  
        case 1, 2 -> "junior";  
        case 3 -> "senior";  
        default -> {  
            if (numberOfChildren < 0)  
                throw new IndexOutOfBoundsException( numberOfChildren );  
            yield "manager";  
        }  
    };  
}
```



Switch
Expression



Text Blocks

```
var sql = "select t.name, m.name, m.year\n" +
          "from mascots m\n" +
          "left join technology t\n" +
          "on t.mascot_fk = m.id\n" +
          "where m.year = '1996'";
```

```
@Query("select u\n" +
        "from User u\n" +
        "where u.username == :1")
```



Text Blocks

```
var sql = """
select t.name, m.name, m.year
from mascots m
left join technology t
on t.mascot_fk = m.id
where m.year = '1996'""";
```

```
@Query("""
select u
from User u
where u.username == :1""")
```



Text Blocks

```
private static boolean isEmpty(Object o) {
    return o == null ||
           o instanceof String && ((String) o).isBlank() ||
           o instanceof Collection && ((Collection) o).isEmpty();
}
```



```
private static boolean isEmpty(Object o) {
    return o == null ||
           o instanceof String s && s.isBlank() ||
           o instanceof Collection c && c.isEmpty();
}
```



```
System.out.println(isEmpty(null)); // true
System.out.println(isEmpty("")); // true
System.out.println(isEmpty(List.of(1, 2, 3))); // false
```



Pattern
Matching for
instanceof



Michael Simons
@rotnroll666

...

Unrelated: How could I ever lived without instanceof-pattern-matching? #Java17

[Tweet übersetzen](#)

10:07 vorm. · 20. Okt. 2021 · Twitter Web App



Pattern
Matching for
instanceof


```
System.out.println(contact.person()
    .socialInsuranceNumber()
    .placeOfBirth()
    .toUpperCase());
```

Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.toUpperCase()" because the return value of "Demo\$SocialInsuranceNumber.placeOfBirth()" is null
at Demo.nullPointerExceptions(Demo.java:55)
at Demo.main(Demo.java:39)

Exception in thread "main" java.lang.NullPointerException: Cannot invoke "Demo\$Person.socialInsuranceNumber()" because the return value of "Demo\$Contact.person()" is null
at Demo.nullPointerExceptions(Demo.java:58)
at Demo.main(Demo.java:39)



Helpful
NullPointerExceptions

```
package de.sippsack.records;

import java.math.BigDecimal;
import java.util.Currency;

public record MonetaryAmount(BigDecimal value, Currency currency) {}
```

→ records javap MonetaryAmount.class
Compiled from "MonetaryAmount.java"

```
public final class de.sippsack.records.MonetaryAmount extends java.lang.Record {
    public de.sippsack.records.MonetaryAmount(java.math.BigDecimal, java.util.Currency);
    public final java.lang.String toString();
    public final int hashCode();
    public final boolean equals(java.lang.Object);
    public java.math.BigDecimal value();
    public java.util.Currency currency();
}
```



Records

```

public record MonetaryAmount(BigDecimal value,
    Currency currency) {

    public MonetaryAmount(BigDecimal value) {
        this(value, Currency.getInstance("EUR"));
    }

    public MonetaryAmount times(BigDecimal factor) {
        return new MonetaryAmount(value.multiply(factor),
            currency);
    }
}

```



Records

```

MonetaryAmount tenEuro = new MonetaryAmount(BigDecimal.TEN,
    Currency.getInstance("EUR"));
MonetaryAmount anotherTenEuro = new MonetaryAmount(BigDecimal.TEN);

System.out.println(tenEuro);
System.out.println(tenEuro.times(BigDecimal.TEN));

System.out.println(tenEuro.equals(anotherTenEuro)); // true
System.out.println(tenEuro.equals(new MonetaryAmount(BigDecimal.TEN,
    Currency.getInstance("USD")))); // false

```



Records

feingranulare Steuerung der Vererbungshierarchie

Algebraische Datentypen

Prüfung der “Exhaustiveness” im switch



Sealed
Classes

```
public sealed interface JsonValue
    permits JsonValue.JsonArray, JsonValue.JsonObject {

    non-sealed class JsonArray implements JsonValue {}

    sealed class JsonObject implements JsonValue
        permits JsonString, JsonInteger {}

    final class JsonString extends JsonObject {}

    final class JsonInteger extends JsonObject {}

    public static void main(String[] args) {

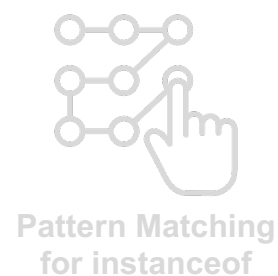
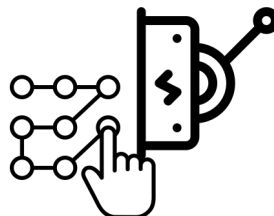
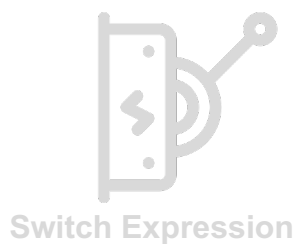
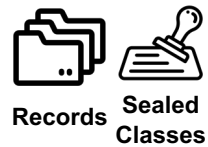
        JsonValue root = null;
        System.out.println(new JsonArray());

    }
}
```



Sealed
Classes

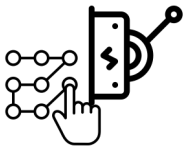
```
public sealed interface Expr
    permits ConstantExpr, PlusExpr, TimesExpr, NegExpr {
}
public record ConstantExpr(int i) implements Expr {}
public record PlusExpr(Expr a, Expr b) implements Expr {}
public record TimesExpr(Expr a, Expr b) implements Expr {}
public record NegExpr(Expr e) implements Expr {}
```



JEP 406: Pattern Matching for switch (Preview)

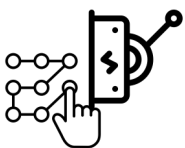
Type Patterns im switch

```
String evaluateTypeWithSwitch( Object o ) {
    return switch(o) {
        case String s -> "String: " + s;
        case Collection c -> "Collection: " + c;
        default -> "Something else: " + o;
    };
}
```



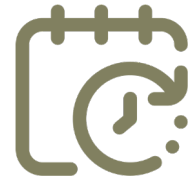
Guarded Patterns und Null-Check

```
boolean isNullOrEmptyWithSwitch( Object o ) {
    return switch(o) {
        case null -> true;
        case String s && s.isBlank() -> true;
        case String s -> false;
        case Collection c && c.isEmpty() -> true;
        default -> false;
    };
}
```

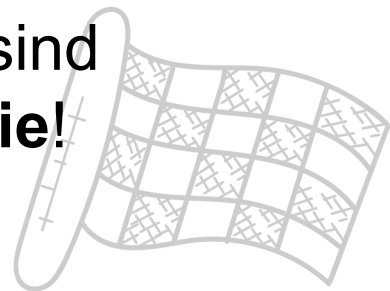


Agenda

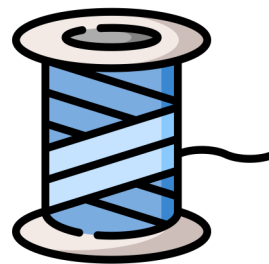
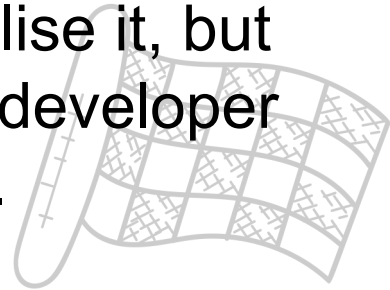
- 1 Einführung
- 2 Was bisher geschah
- 3 Auf dem Weg zu Java 17
- 4 Neue Features im Blick
- 5 **Ausblick**



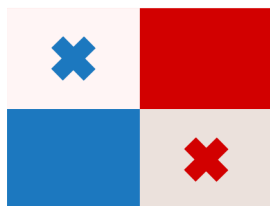
Die Java-Welt und
das Ökosystem sind
lebendig wie nie!



Every feature is not as good as a good developer can utilise it, but as bad as the average developer can misuse it.



Was kommt?



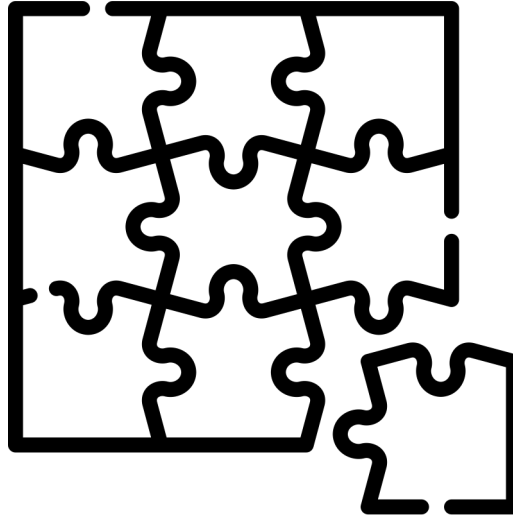
Project Amber



The goal of Project Amber is to explore and incubate smaller, productivity-oriented Java language features that have been accepted as candidate JEPs under the OpenJDK JEP process.

<https://openjdk.java.net/projects/amber/>

Einführung von Pattern Matching in Java



- JEP 325: Switch Expressions (Preview) – Java 12
- JEP 354: Switch Expressions (Second Preview) – Java 13
- JEP 361: Switch Expressions – Java 14
- JEP 359: Records (Preview) – Java 14
- JEP 384: Records (Second Preview) – Java 15
- JEP 395: Records – Java 16
- JEP 305: Pattern Matching for instanceof (Preview) – Java 14
- JEP 375: Pattern Matching for instanceof (Second Preview) – Java 15
- JEP 394: Pattern Matching for instanceof – Java 16
- JEP 360: Sealed Classes (Preview) – Java 15
- JEP 397: Sealed Classes (Second Preview) – Java 16
- JEP 409: Sealed Classes – Java 17
- JEP 406: Pattern Matching for switch (Preview) – Java 17

Switch Expression

Pattern Matching for instanceof

Records

Pattern Matching for switch

Sealed Classes

Java 17

embarc.de

78

JEP 405: Record Patterns & Array Patterns

18?

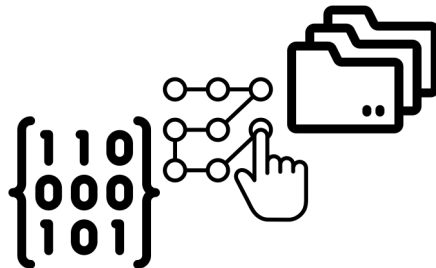
<https://openjdk.java.net/jeps/405>

Java 17

embarc.de

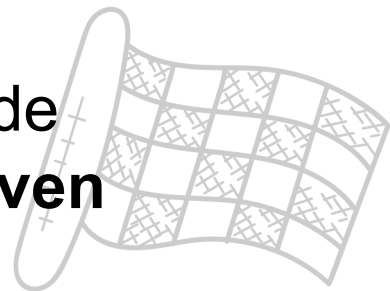
79

```
static void printSumOfFirstTwoXCoords(Object o) {
    if (o instanceof Point[] { Point(var x1, var y1),
        Point(var x2, var y2), ... }) {
        System.out.println(x1 + x2);
    }
}
```

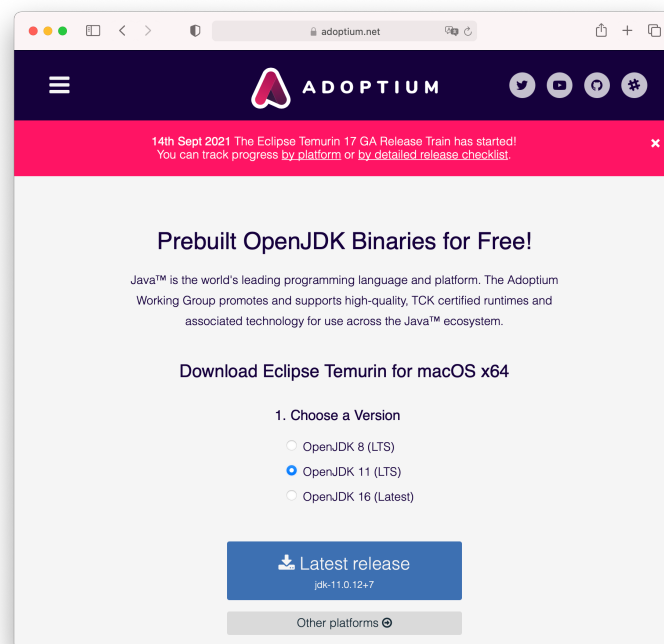
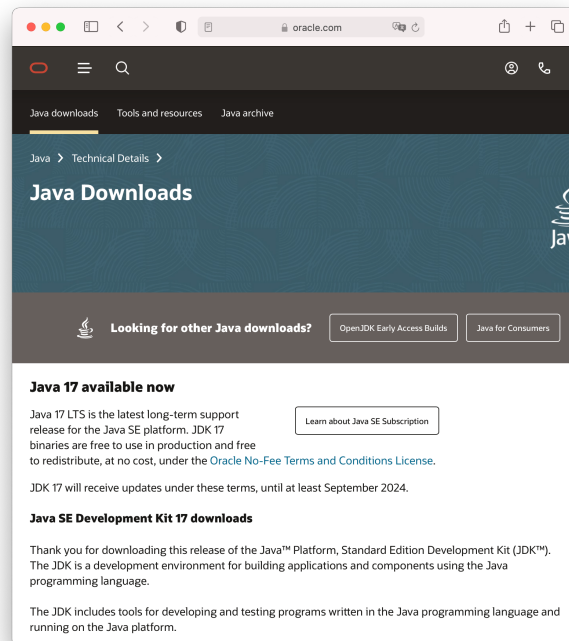


Oracle JDK ist ~~nicht~~
~~wieder~~ ~~mehr~~ kostenlos

~~Außerdem~~ ~~Aber~~, es gibt jede
Menge Alternativen



“Java 17 LTS is the latest long-term support release for the Java SE platform. **JDK 17 binaries are free to use in production** and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions License.



Home / Projects / Eclipse Adoptium

Eclipse Adoptium

Overview Downloads Who's Involved Developer Resources

Governance Contact Us

The mission of the Eclipse Adoptium Top-Level Project is to produce high-quality runtimes and associated technology for use within the Java ecosystem. We achieve this through a set of Projects under the Adoptium PMC and a close working partnership with external projects, most notably OpenJDK for providing the Java SE runtime implementation. Our goal is to meet the needs of both the Eclipse community and broader runtime users by providing a comprehensive set of technologies around runtimes for Java applications that operate alongside existing standards, infrastructures, and cloud platforms.

The AdoptOpenJDK project was established in 2017 following years of discussions about the general lack of an open and reproducible build and test system for OpenJDK source across multiple platforms. Since then it has grown to become a leading provider of high-quality OpenJDK-based binaries used by enterprises in embedded systems, desktops, traditional servers, modern cloud platforms, and large mainframes. The Eclipse Adoptium project is the continuation of the original AdoptOpenJDK mission.

Licenses:
 Apache License, Version 2.0
 Eclipse Distribution License 1.0 (BSD)
 Eclipse Public License 2.0
 -- (Secondary) GNU General Public License, version 2 with OpenJDK Assembly Exception
 -- (Secondary) GNU General Public License, version 2 with the GNU Classpath

PROJECT LINKS

- Adoptium Top Level Project Charter

RELATED PROJECTS

Project Hierarchy:

- Eclipse Adoptium
 - Eclipse Adoptium Incubator
 - Eclipse AQAwt
 - Eclipse Mission Control
 - Eclipse Temurin Compliance
 - Eclipse Temurin*

Java 17 embarc.de 87

87

Migration/Upgrade auf Java 11+ nicht zu lange aufschieben.

Das nächste LTS Release ist nicht fern

...

Java 17 embarc.de 88

88

Java 17 finalisiert angefangene Arbeiten aus Java 12 bis 16, damit es als **LTS Release** für die nächsten ~~drei~~ ^{zwei} Jahre gut da steht.

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

The Arrival of Java 17

Used by over 10 million developers and running on 56 billion devices globally, the Java Platform truly moves the world forward, and now even faster with Java 17. Download or try out Java here!

[Download Java 17](#) [Learn Java](#)

Currently Java 17 · [Previous releases](#)

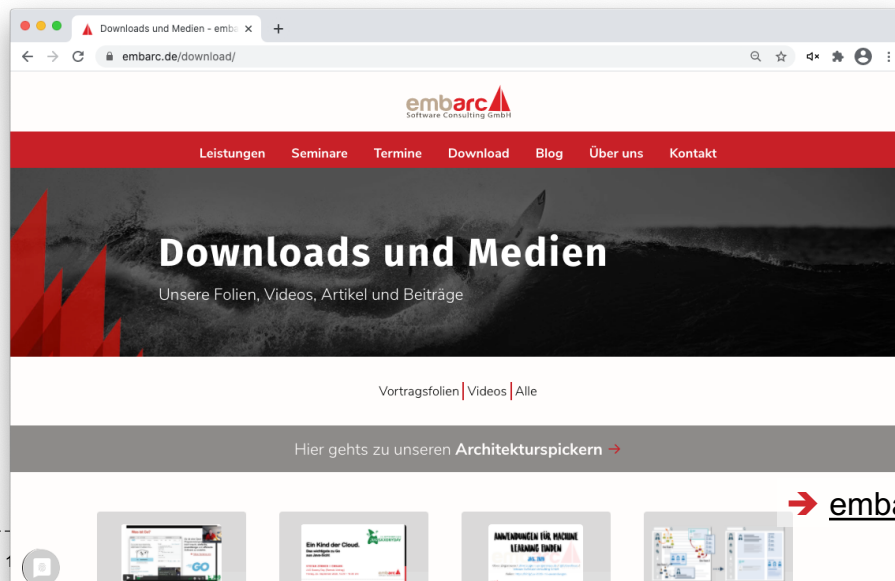
Try Java

Try Java's new features right in a web UI, coming soon!

```
1 var name = "Duke";
2
3 System.out.println("Hello, " + name);
4
```

Soon!

Folien von heute als PDF zum Download



→ embarc.de/download/

91

91

Vielen Dank.

Ich freue mich auf Eure Fragen!



Falk Sippach

✉ fs@embarc.de

🐦 [@sipsack](https://twitter.com/sipsack)

🔗 → [xing.to/fsi](https://www.xing.com/profile/falk_sippach)

92