

24. September 2019
Düsseldorf / JCON

Neues von Java und dem JDK

INOQ



MICHAEL VITZ

**Senior Consultant
bei INNOQ Deutschland GmbH**





Vote

JDK Java <u>Version</u>	
5	
6	
7	
8	
9	
10	
11	

Usage

Module (JPMs)

Ja	
Nein	

Support

Ja	
Nein	

Vendor

Oracle	
OpenJDK	
IBM	
Azul	
2	
OS5	

26.05.2018

6

Java

7

8 ~~||||~~ ~~||||~~ |||

9

10 |

11 |||

12 |

>=8 -> Module?

Ja

Nein ~~||||~~ ~~||||~~

Support?

Ja

Nein ~~||||~~ ~~||||~~

Vendor?

Oracle ||

Adopt Open JDK ||

Open JDK ~~||||~~ ~~||||~~

Azul

SAP

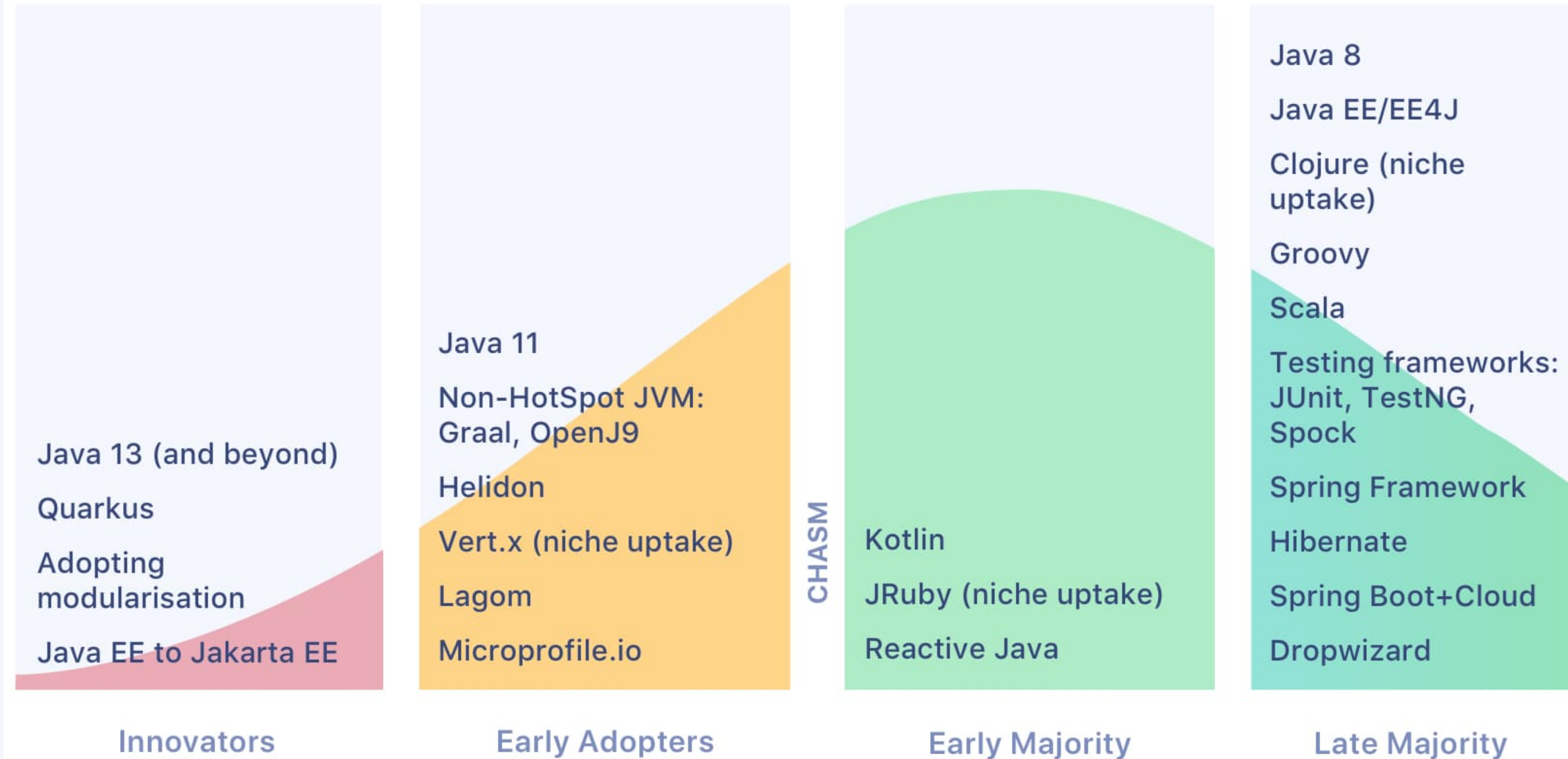
IBM

18.05.2019

Software Development Java 2019 Q3 Graph

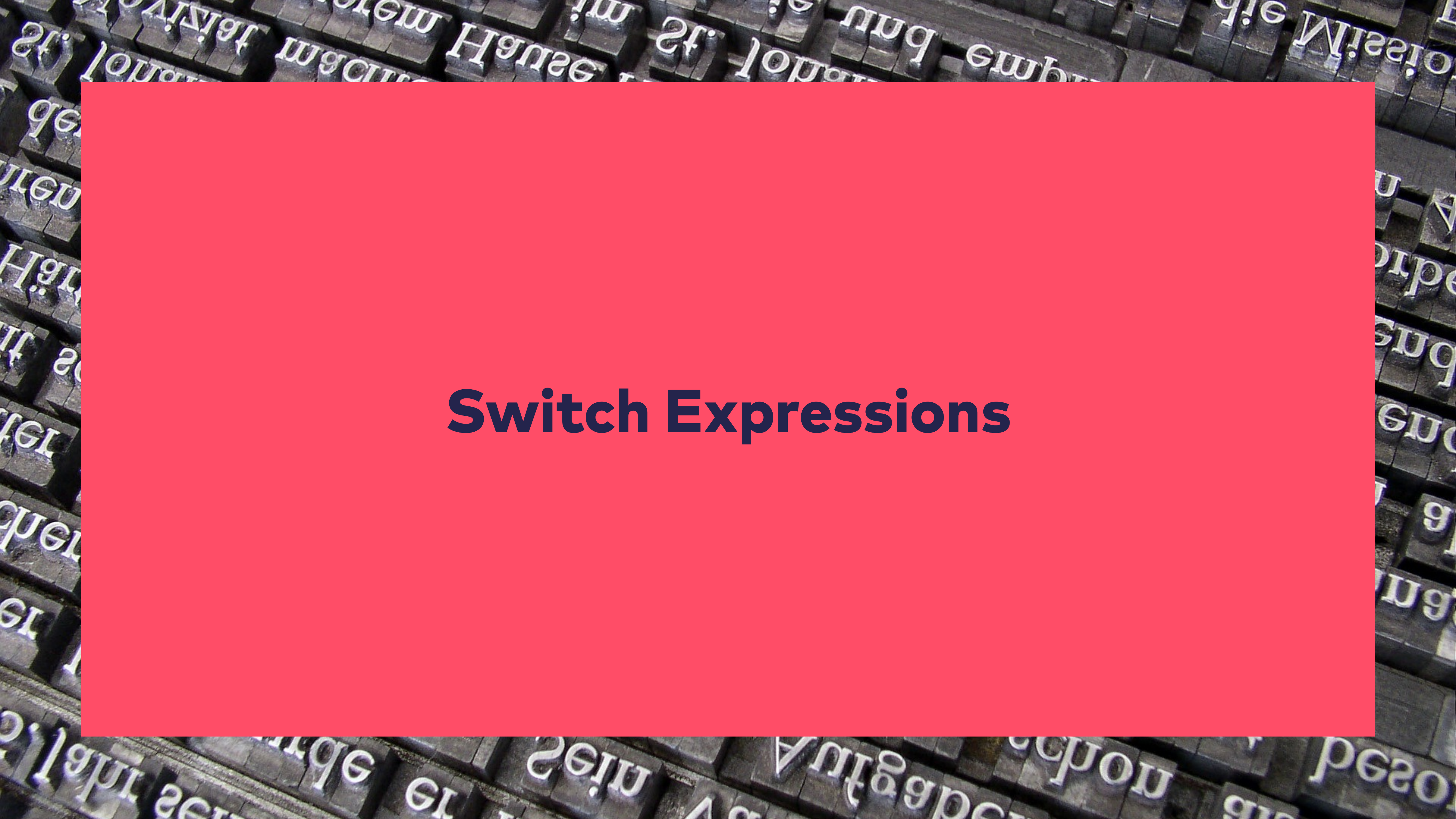
<http://infoq.link/java2019>

InfoQ





Sprachfeatures



Switch Expressions

Switch Expressions

- Preview Feature

<https://blog.codefx.org/java/enable-preview-features/>

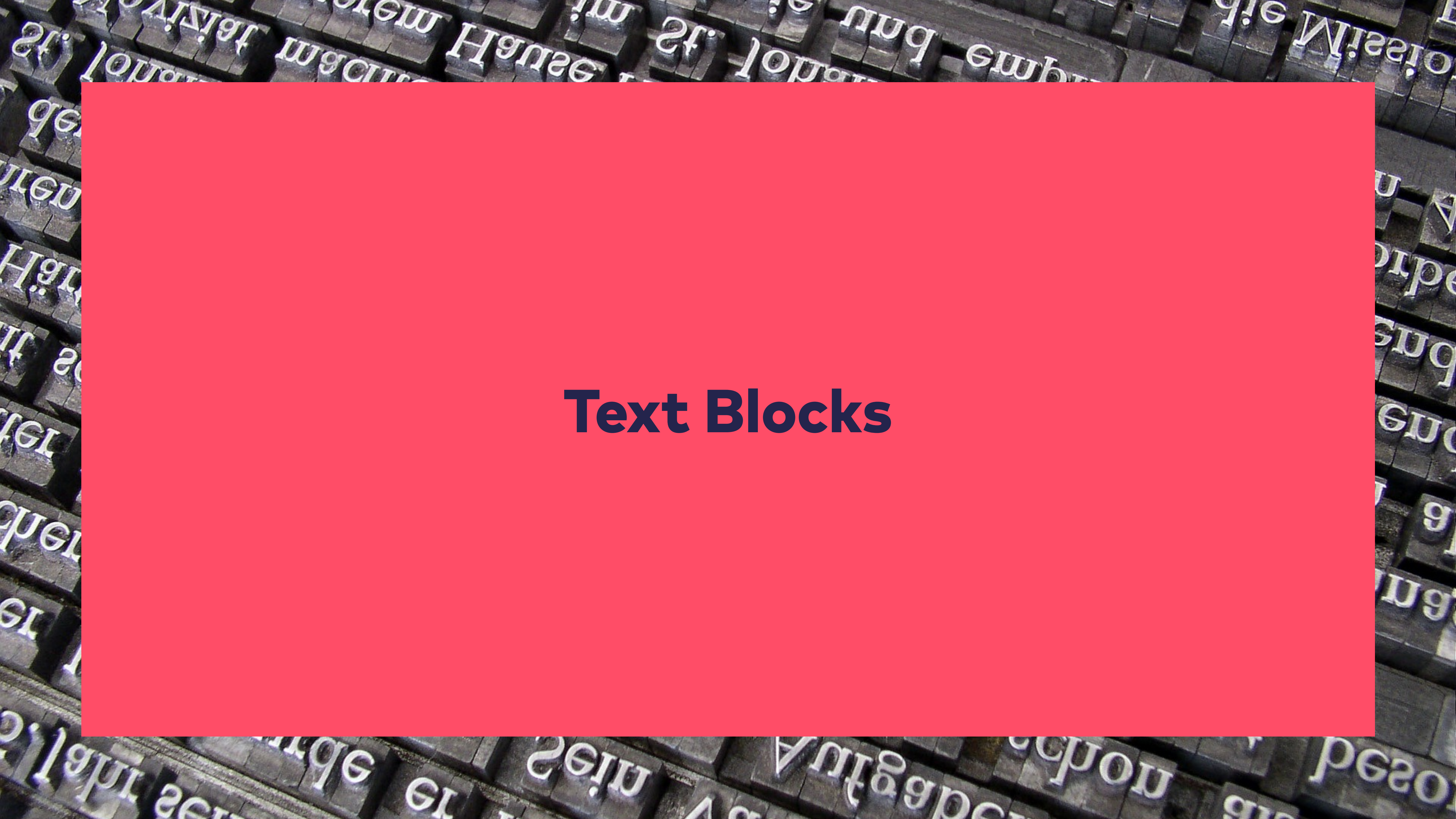
- JEP 325 in JDK 12

<http://openjdk.java.net/jeps/325>

- JEP 354 in JDK 13

<http://openjdk.java.net/jeps/354>


```
int numLetters = switch (day) {  
    case MONDAY, FRIDAY, SUNDAY -> {  
        System.out.println("Whoop");  
        yield 6;  
    }  
    case TUESDAY -> 7;  
    case THURSDAY, SATURDAY -> 8;  
    case WEDNESDAY -> 9;  
};
```

Text Blocks

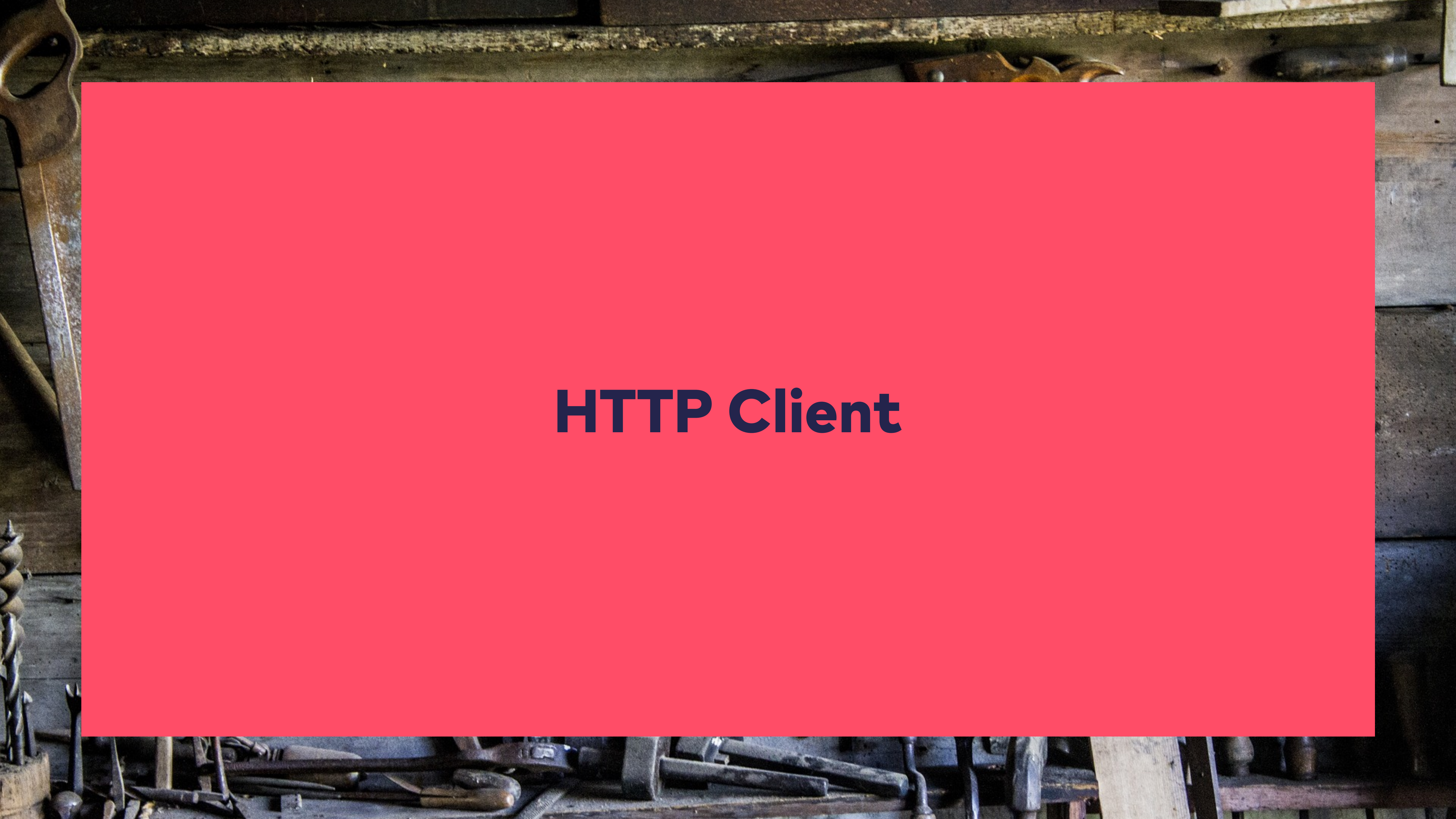
Text Blocks

- Preview Feature
- JEP 355 in JDK 13
<https://openjdk.java.net/jeps/355>
- Detaillierter Artikel
<https://blog.codefx.org/java/text-blocks/>


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




Java API



HTTP Client

HTTP Client

- JEP 321 in JDK 11

<http://openjdk.java.net/jeps/321>

- Unterstützt HTTP 2
- Synchrones und Asynchrones Modell
- WebSocket Support

- Intro

<http://openjdk.java.net/groups/net/httpclient/intro.html>


```
HttpClient client = HttpClient.newHttpClient();
HttpRequest request = HttpRequest.newBuilder()
    .uri(URI.create("http://openjdk.java.net/"))
    .build();
client.sendAsync(request, ofString())
    .thenApply(HttpResponse::body)
    .thenAccept(System.out::println)
    .join();
```




java.lang.String

java.lang.String in JDK 11

- String::repeat(int)

<https://bugs.openjdk.java.net/browse/JDK-8197594>

- String::lines

<https://bugs.openjdk.java.net/browse/JDK-8200380>

- String::strip, String::stripLeading, String::stripTrailing

<https://bugs.openjdk.java.net/browse/JDK-8200377>

- String::isBlank

<https://bugs.openjdk.java.net/browse/JDK-8200436>

java.lang.String in JDK 12

- `String::indent(int)`

<https://bugs.openjdk.java.net/browse/JDK-8200435>

- `String::transform(Function)`

<https://bugs.openjdk.java.net/browse/JDK-8203703>

java.lang.String in JDK 13

- String::formatted

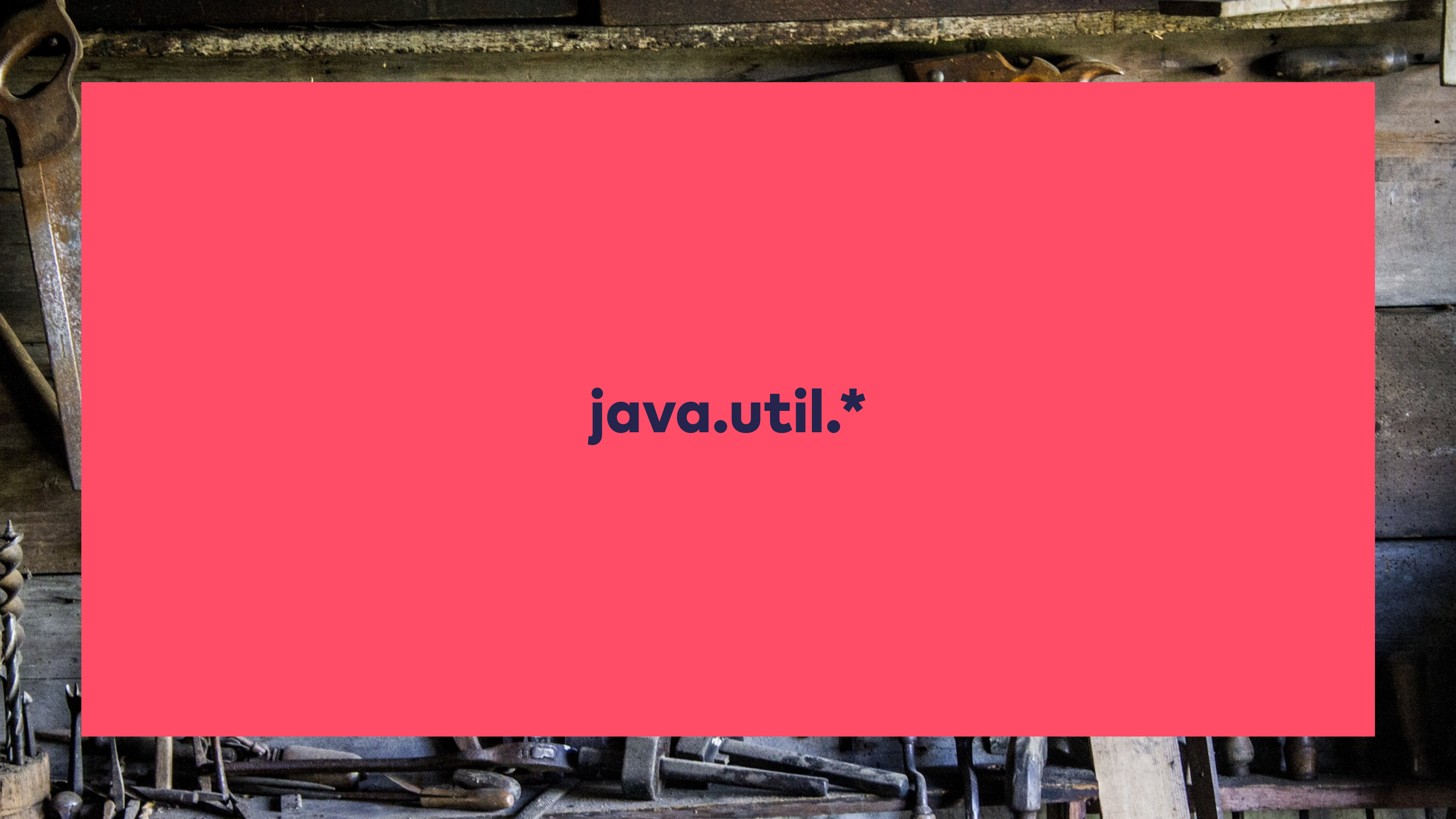
<https://bugs.openjdk.java.net/browse/JDK-8203444>

- String::stripIndent

<https://bugs.openjdk.java.net/browse/JDK-8223775>

- String::translateEscapes

<https://bugs.openjdk.java.net/browse/JDK-8223780>



java.util.*

java.util.Optional

- JDK 11: `Optional::isEmpty`
<https://bugs.openjdk.java.net/browse/JDK-8184693>
- Bald Deprecated? `Optional::get`
<https://bugs.openjdk.java.net/browse/JDK-8160606>

java.util.Predicate

- JDK 11: Predicate#not

<https://bugs.openjdk.java.net/browse/JDK-8050818>

java.util.stream.Collectors

- JDK 12: Collectors#teeing(Collector, Collector, BiFunction)
<https://bugs.openjdk.java.net/browse/JDK-8209685>

java.util.regex.Pattern

- JDK 11: Pattern::asMatchPredicate
<https://bugs.openjdk.java.net/browse/JDK-8201308>



java.(n)io.*

java.io.*

- `java.io.FileReader(java.lang.String, java.nio.charset.Charset)`
- `java.io.FileWriter(java.lang.String, java.nio.charset.Charset)`
- `java.io.InputStream#nullInputStream`
- `java.io.OutputStream#nullOutputStream`
- `java.io.Reader#nullReader`
- `java.io.Writer#nullWriter`

java.nio.file.Files

- JDK 11: Files#isSameContent

<https://bugs.openjdk.java.net/browse/JDK-8202302>

- JDK 11: Files#readString, Files#writeString

<https://bugs.openjdk.java.net/browse/JDK-8202055>

- JDK 12: Files#mismatch(Path, Path)

<https://bugs.openjdk.java.net/browse/JDK-8202302>

java.nio.file.Path

- JDK 11: Path#**of**(URI), Path#**of**(String, String...)

<https://bugs.openjdk.java.net/browse/JDK-8199485>

java.nio.file.FileSystems

- JDK 13: FileSystems#`newFileSystem`(Path, ...)
<https://bugs.openjdk.java.net/browse/JDK-8219793>

A workshop background with various tools like pliers, a hammer, and a drill bit.

java.lang.Thread

java.lang.Thread in JDK 11

- Thread::stop(Throwable), Thread::destroy

<https://bugs.openjdk.java.net/browse/JDK-8204243>



JVM

Single-File Source-Code Programs

- JEP 330 in JDK 11
<https://openjdk.java.net/jeps/330>
- `java Foo.java`
- `#!/usr/bin/java -source 11`

Garbage Collectors

- JDK 11: Epsilon: A No-Op GC

<https://openjdk.java.net/jeps/318>

- JDK 11: ZGC: A Scalable Low-Latency GC (Experimental)

<https://openjdk.java.net/jeps/333>

- JDK 12: Shenandoah: A Low-Pause-Time GC (Experimental)

<https://openjdk.java.net/jeps/189>

Garbage Collectors

- JDK 12: Abortable Mixed Collections for G1

<https://openjdk.java.net/jeps/344>

- JDK 12: Promptly Return Unused Committed Memory from G1

<https://openjdk.java.net/jeps/346>

- JDK 13: ZGC: Uncommit Unused Memory

<https://openjdk.java.net/jeps/351>

Class Data Sharing

- JDK 12: Default CDS Archives

<https://openjdk.java.net/jeps/341>

- JDK 13: Dynamic CDS Archives

<https://openjdk.java.net/jeps/350>



Zukunft

1

24

JDK 14



Workshop

[OpenJDK FAQ](#)

[Installing](#)

[Contributing](#)

[Sponsoring](#)

[Developers' Guide](#)

[Vulnerabilities](#)

[Mailing lists](#)

[IRC](#) · [Wiki](#)

[Bylaws](#) · [Census](#)

[Legal](#)

JEP Process

Source code

[Mercurial](#)

[Bundles \(6\)](#)

Groups

[\(overview\)](#)

JDK 14

This release will be the Reference Implementation of version 14 of the Java SE Platform, as specified by [JSR 389](#) in the Java Community Process.

Status

The [development repositories](#) are open for bug fixes, small enhancements, and JEPs as proposed and tracked via the [JEP Process](#).

Features

JEPs targeted to JDK 14, so far

352: [Non-Volatile Mapped Byte Buffers](#)

Last update: 2019/7/18 18:48 UTC



Project Amber

Pattern Matching for instanceof

- Preview
- JEP 305
<http://openjdk.java.net/jeps/305>

```
if (obj instanceof Foo s) {  
    // can use s here  
} else {  
    // can't use s here  
}
```

```
@Override  
public boolean equals(Object o) {  
    return (o instanceof Foo f) &&  
        f.doSomething(s);  
}
```


Records

- Preview

```
record Point(int x, int y) { }
```

- JEP 359

<http://openjdk.java.net/jeps/359>

- Aktueller Stand

<https://cr.openjdk.java.net/~briangoetz/amber/datum.html>

Pattern matching for switch

- Preview

- Draft

<https://openjdk.java.net/jeps/8213076>

String formatted;

switch (obj) {

case Integer i: formatted = String.format("int %d", i); break;

case Byte b: formatted = String.format("byte %d", b); break;

case Long l: formatted = String.format("long %d", l); break;

case Double d: formatted = String.format("double %f", l); break;

case String s: formatted = String.format("String %s", s); break;

default: formatted = obj.toString();

}

Concise Method Bodies

- Draft

<http://openjdk.java.net/jeps/8209434>

```
// -> is "single expression form"  
int length(String s) -> s.length();
```

```
// = is "method reference form"  
int length(String s) = String::length;
```




Project Loom

Project Loom

Fibers and Continuations



Project Loom is intended to explore, incubate and deliver Java VM features and APIs built on top of them for the purpose of supporting easy-to-use, high-throughput lightweight concurrency and new programming models on the Java platform. This is accomplished by the addition of the following constructs:

- Fibers (lightweight user-mode threads)
- Delimited continuations
- Tail-call elimination

This **OpenJDK** project is sponsored by the **HotSpot Group**.



Source Code

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



Early Access Binaries

<http://jdk.java.net/loom/>



Talks

JVMLS 2019 - [Video](#)

A person in a black hooded robe, resembling the Grim Reaper, stands in a stone archway. They are holding a scythe. A long, thin pole extends from the right side of the frame towards the person. The setting is a dark, stone-walled tunnel or cave with multiple arches leading to a bright light at the end.

Project Valhalla

Value Objects

- JEP 169
<https://openjdk.java.net/jeps/169>

Summary

Provide JVM infrastructure for working with immutable and reference-free objects, in support of efficient by-value computation with non-primitive types.

Goals

- Support user-defined and library-defined abstract data types with performance profiles similar to Java primitive types.
- Bring the semantics of `int` and `java.lang.Integer` closer together.
- Allow representation of ubiquitous types currently not well supported on the JVM, including complex numbers, vector values, and tuples.
- Increase shareability of Java data structures.
- Provide clear and explicit semantics for shared read-only array data.
- Enable functional-style computation with pure data, for optimized parallel computations.
- Increase safety and security and decrease "defensive copying" in applications which must share structured data across trust boundaries.

Non-Goals

This work is not intended to change the Java Language Specification or JVM bytecode instruction set.

Generics over Primitive Types

- JEP 218
<http://openjdk.java.net/jeps/218>

```
class Box<T> {  
}
```

```
new Box<int>( );
```


Danke! Fragen?



Michael Vitz
michael.vitz@innoq.com
+49 151 19116015
@michaelvitz

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin
Germany
+49 2173 3366-0

Ludwigstr. 180E
63067 Offenbach
Germany
+49 2173 3366-0

Kreuzstr. 16
80331 München
Germany
+49 2173 3366-0

Hermannstrasse 13
20095 Hamburg
Germany
+49 2173 3366-0

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
+41 41 743 0116