Keeping up with Java

By Sander Mak

luminis
Conversing worlds

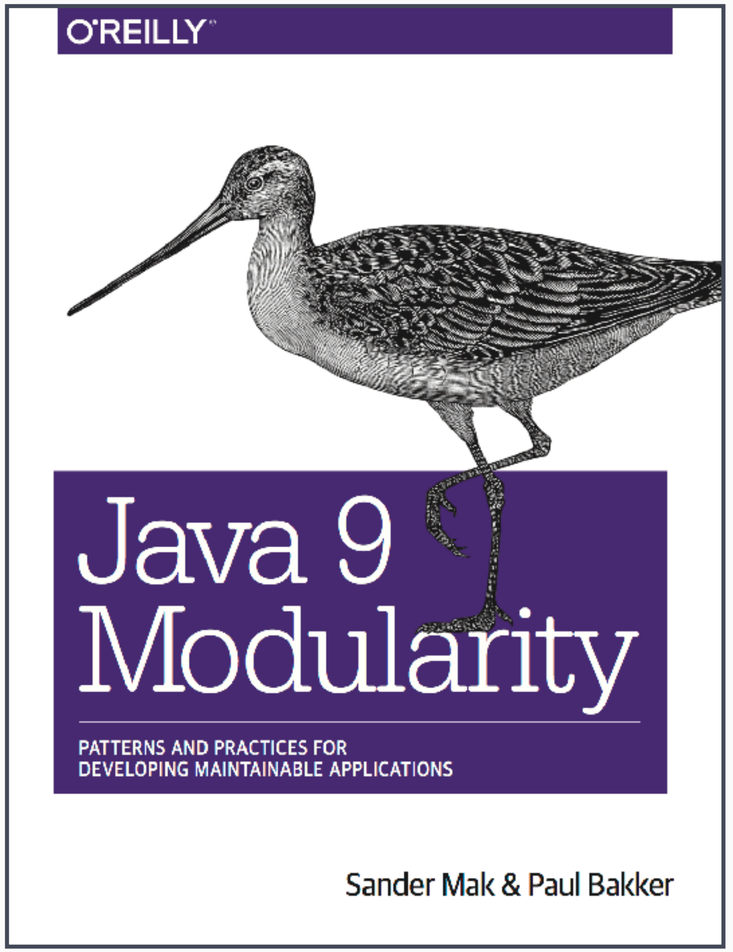
@Sander_Mak

About Sander

Fellow, Software Architect
@ **Luminis**

Experience in modular development stacks

Conference Speaker & Author





PLURALSIGHT

Java



(Sept. 2017)

Quick Java Module Overview



module-info.java

```
module main {

}
```

Module Descriptors

main

```
main.web
main.persistence
main.integration
```

module-info.java

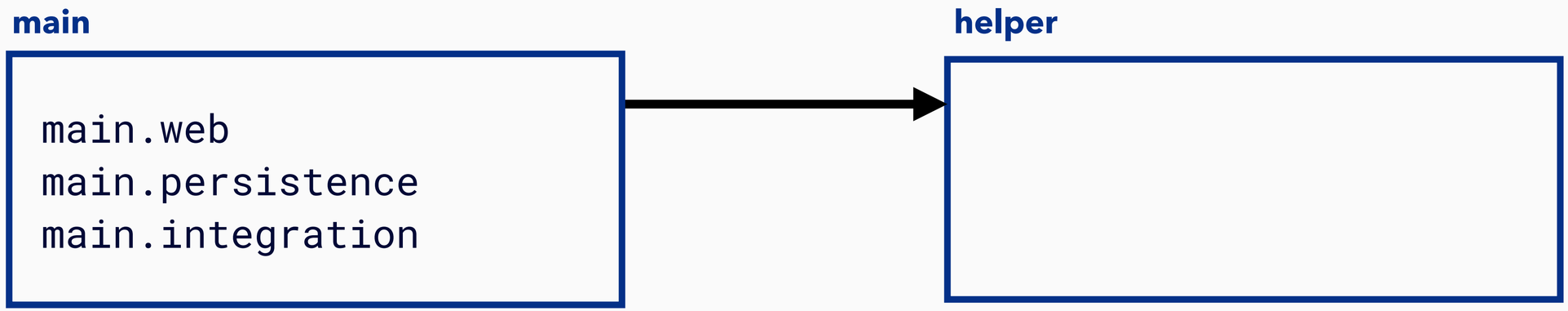
```
module main {
    requires helper;
}
```

module-info.java

```
module helper {


}
```

Explicit Dependencies



module-info.java

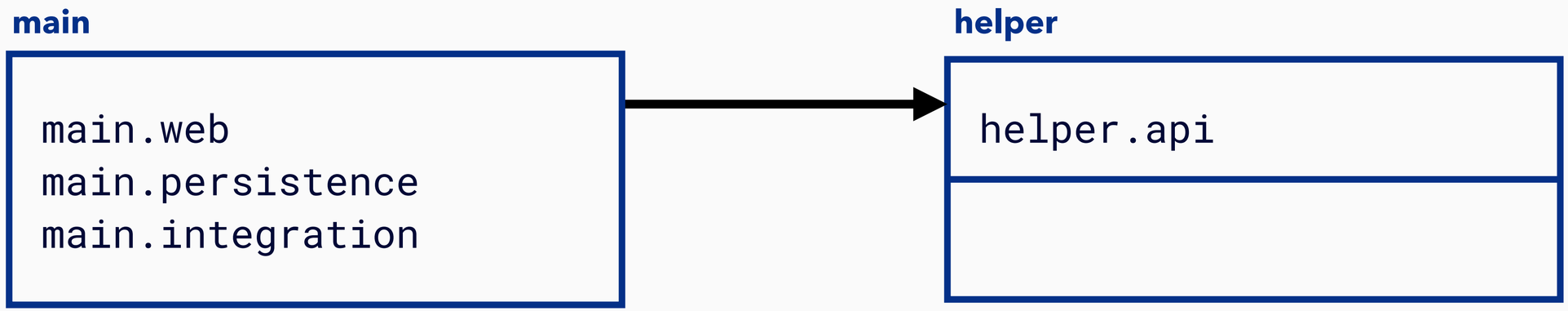
```
module main {
    requires helper;
}
```

module-info.java

```
module helper {
    exports helper.api;

}
```

Well-defined Interfaces



module-info.java

```
module main {
    requires helper;
}
```

module-info.java

```
module helper {
    exports helper.api;

}
```

Strong Encapsulation

main

```
main.web
main.persistence
main.integration
```

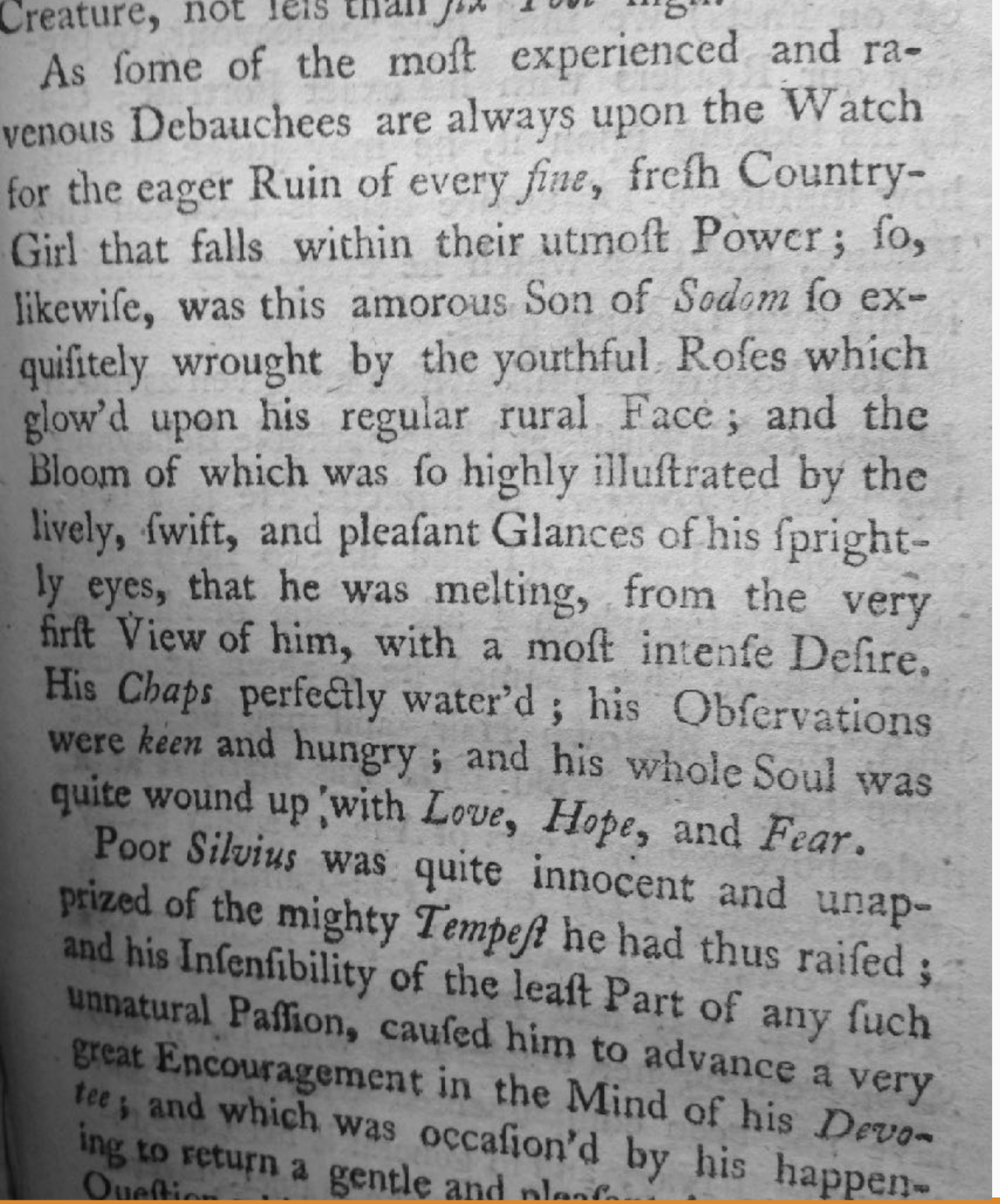


helper

```
helper.api
```

```
helper.impl
```

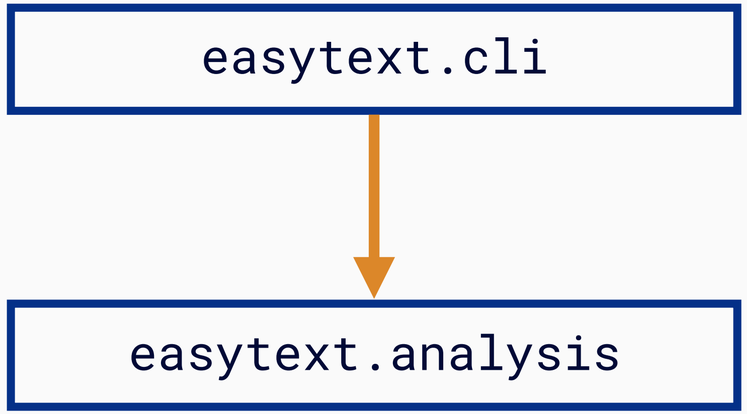



Creature, not leſs than *ſix Foot* high.

As ſome of the moſt experienced and ravenous Debauchees are always upon the Watch for the eager Ruin of every *fine*, freſh Country-Girl that falls within their utmoſt Power; ſo, likewiſe, was this amorous Son of *Sodom* ſo exquiſitely wrought by the youthful Roſes which glow'd upon his regular rural Face; and the Bloom of which was ſo highly illuſtrated by the lively, ſwift, and pleaſant Glances of his ſprightly eyes, that he was melting, from the very firſt View of him, with a moſt intenſe Deſire. His *Chaps* perfectly water'd; his Obſervations were *keen* and hungry; and his whole Soul was quite wound up with *Love*, *Hope*, and *Fear*.

Poor *Silvius* was quite innocent and unapprized of the mighty *Tempeſt* he had thus raiſed; and his Inſenſibility of the leaſt Part of any ſuch unnatural Paſſion, cauſed him to advance a very great Encouragement in the Mind of his *Devotee*; and which was occaſion'd by his happening to return a gentle and pleaſ

Demo: EasyText



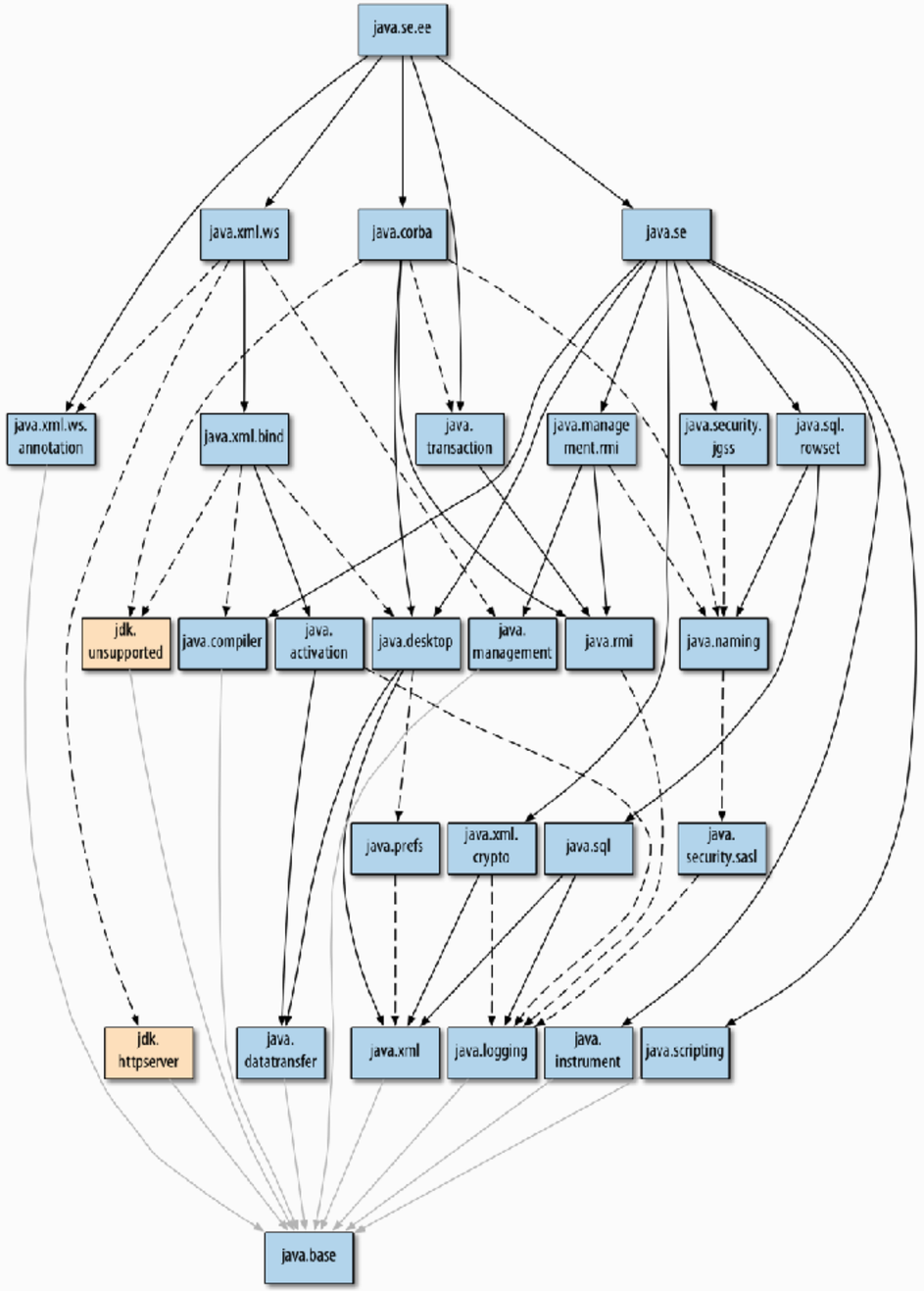
Why?

Modular JDK

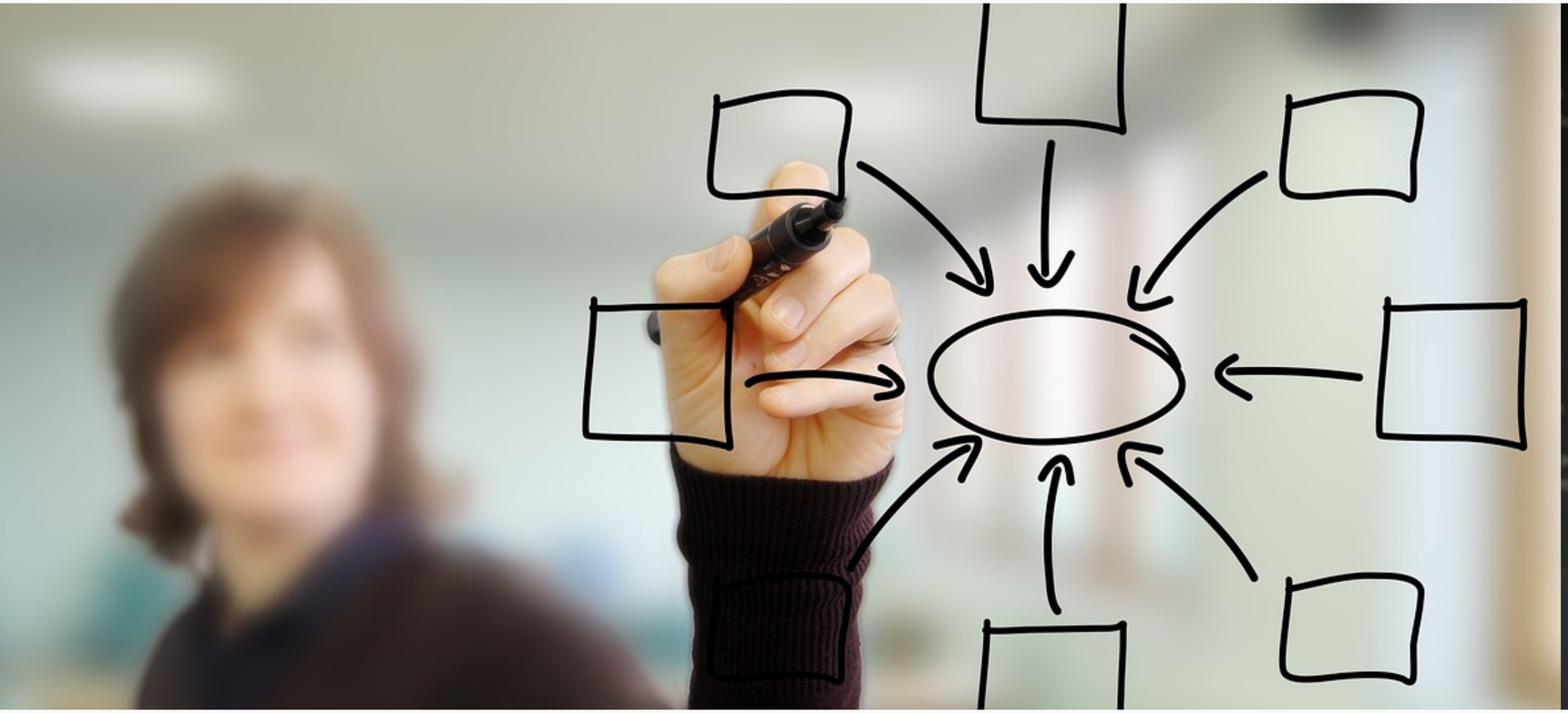
Maintainability

Flexibility

Reliable composition



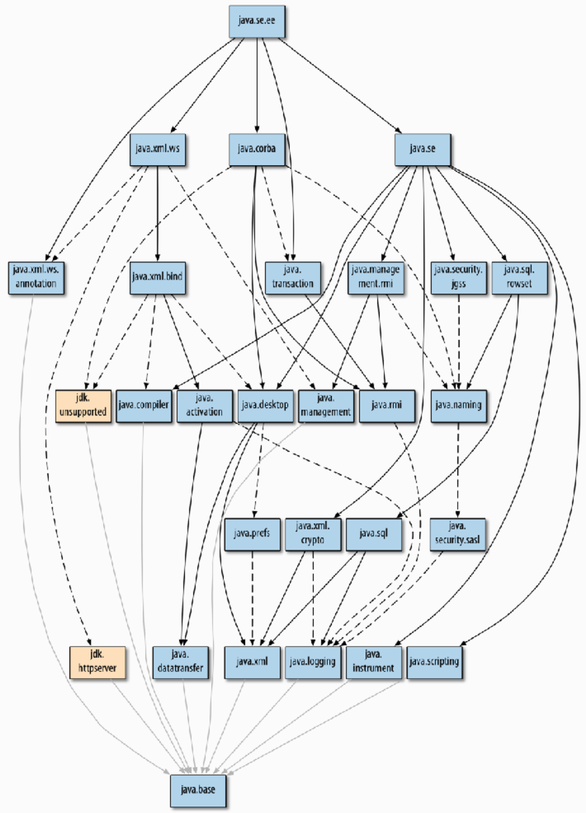
Why?





Decrease the model-code gap

jlink



jdk

~300mb

+

Custom Run-time Image

main

lib1

lib2

java.logging

java.xml

java.base

JVM

application

~25mb

~2mb

@Sander_Mak

Impact on Java 8 codebases

Use of encapsulated JDK types:

- Run-time warnings
- Compile-time errors

Use of enterprise APIs in JDK:

- Won't resolve by default
- Gone in Java 11!

```
java.corba
java.xml.bind
java.xml.ws.*
java.activation
java.transaction
```

jshell

Read → **Input code**

Eval → **Run code**

Print → **See results**

Loop → **Iteratively refine**

jshell

DEMO

Collection Factory Methods

```
List<String> books = new ArrayList<>();
books.add("Java 9 Modularity");
books.add("Designing Data-Intensive Applications");
books.add("Java 8 Lambdas");
```



Collection Factory Methods

```
List<String> books = List.of("Java 9 Modularity",
  "Designing Data-Intensive Applications",
  "Java 8 Lambdas");
```

DEMO

`Set.of` `Map.of`

Java



(March 2018)

var

```
public void aMethod() {
  var name = "Sander";
}
```

var

```
URL url = new URL("https://javamodularity.com");
URLConnection connection = url.openConnection();
BufferedInputStream inputStream =
   new BufferedInputStream(connection.getInputStream());
```

DEMO

```
var bookurl = new URL("https://javamodularity.com");
var connection = bookurl.openConnection();
var bookStream = new BufferedInputStream(connection.getInputStream());
```

Java



(Sept. 2018)

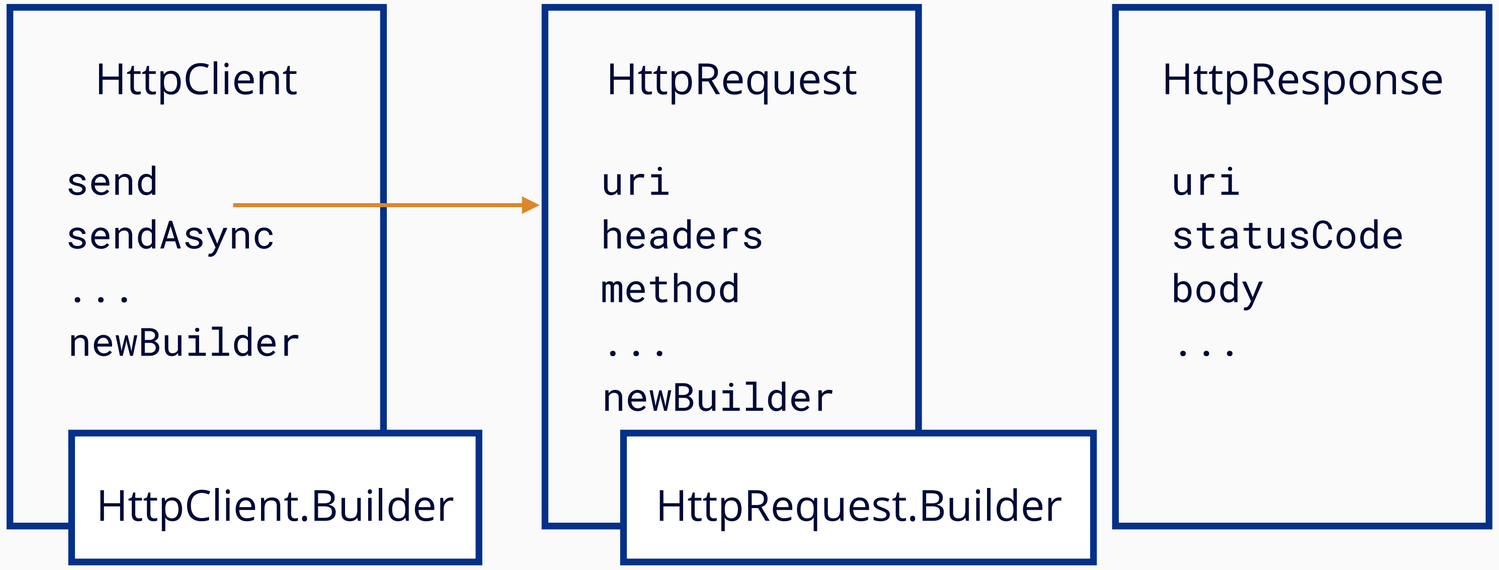
HttpClient

~~HttpURLConnection~~

HTTP/2 & WebSocket

Reactive Streams Support

HttpClient


HttpClient
send
sendAsync
...
newBuilder
HttpClient.Builder
HttpRequest
uri
headers
method
...
newBuilder
HttpRequest.Builder
HttpResponse
uri
statusCode
body
...

HttpClient

```
HttpClient httpClient = HttpClient.newHttpClient();

HttpRequest req = HttpRequest.newBuilder(URI.create("https://google.com"))
                      .GET()
                      .build();

HttpResponse<String> response = httpClient.send(req, BodyHandlers.ofString())
```

Let's Talk About Adoption

What Java version are you currently using?

Java 8 (82%, 243 Votes)

Java 9 (8%, 23 Votes)

Java 7 (7%, 20 Votes)

Java 6 (3%, 8 Votes)

Java 5 (1%, 3 Votes)

Java 8

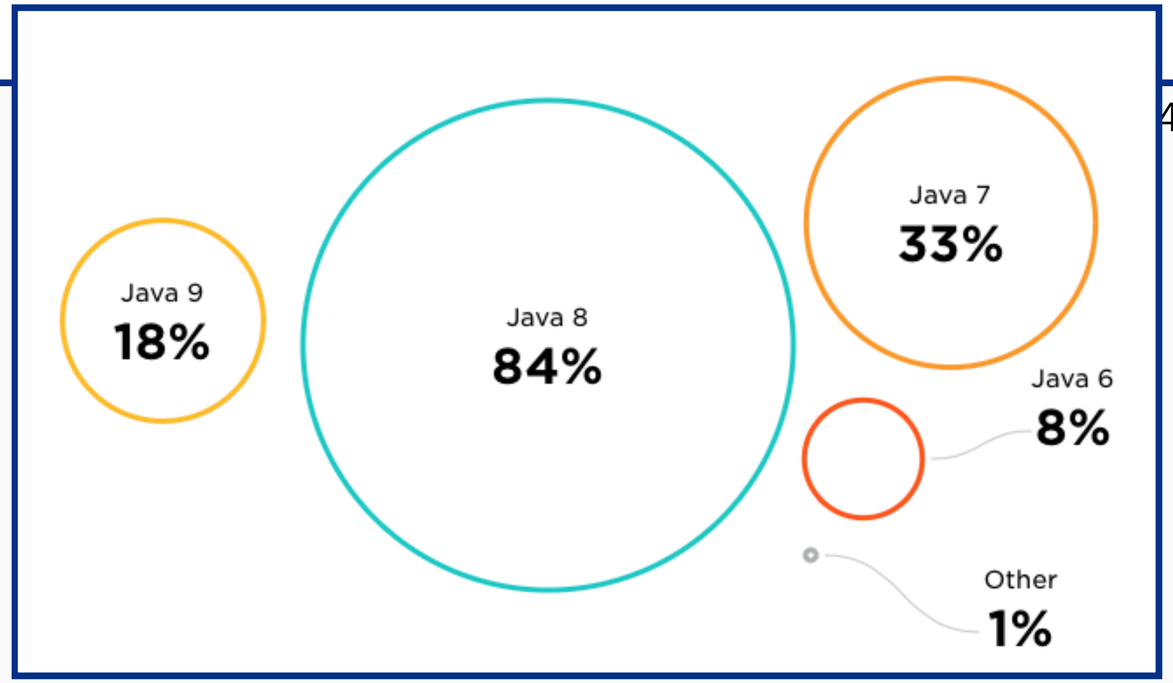
8 in 10 developers use Java SE version 8 in production

Java Magazine & Snyk

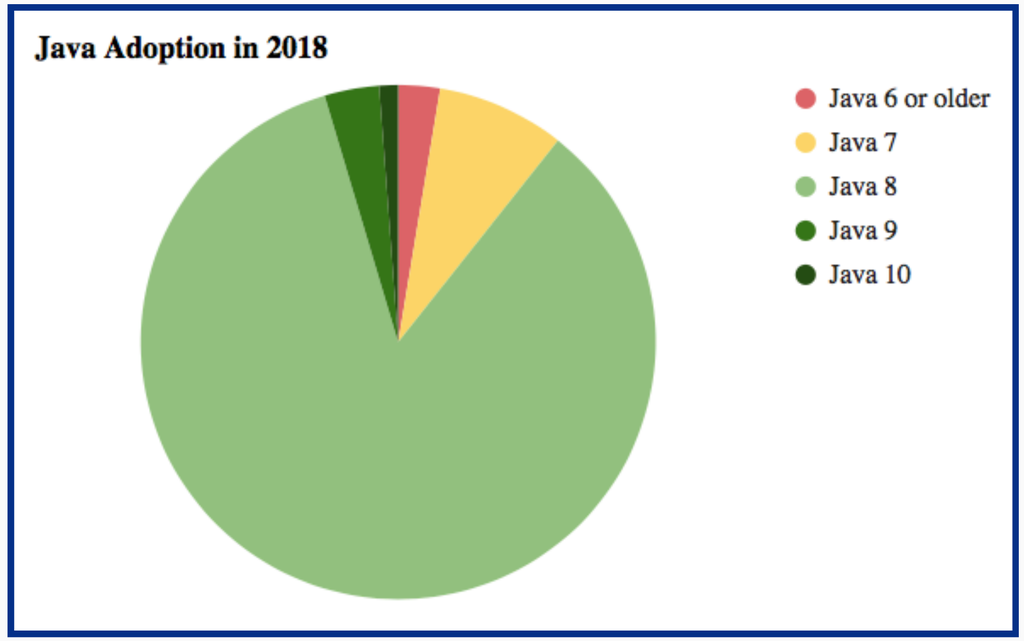
Is Red Hat releasing OpenJDK 9, 10, or 11?

Red Hat will skip Java SE 9 and 10, and ship an OpenJDK distribution based on Java SE 11.

42.html

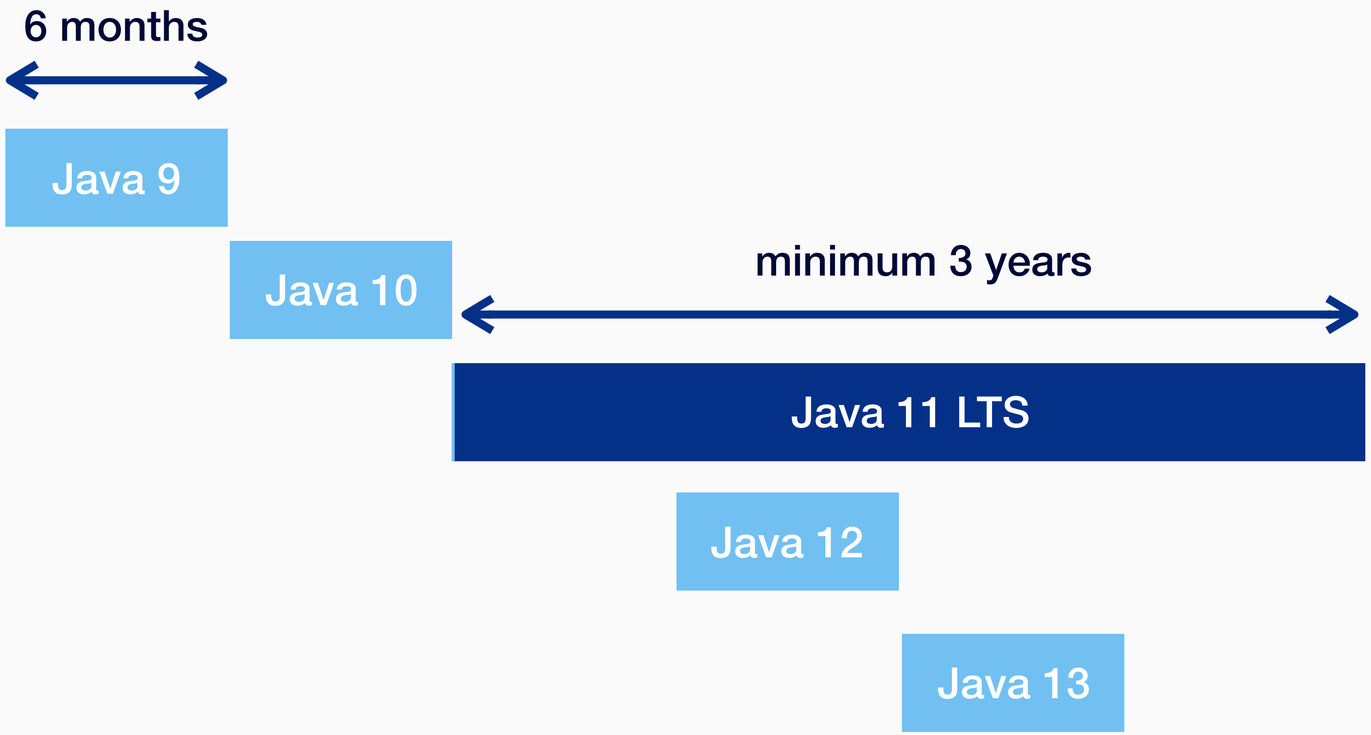


https://www.jetbrains.com/research/devecosystem-2018/java/



https://www.baeldung.com/java-in-2018

Java 11


6 months
Java 9
Java 10
minimum 3 years
Java 11 LTS
Java 12
Java 13


THESE GO
Long
Term
Support

Which JDK?

OpenJDK

2 in 10 developers use OpenJDK in production

ORACLE

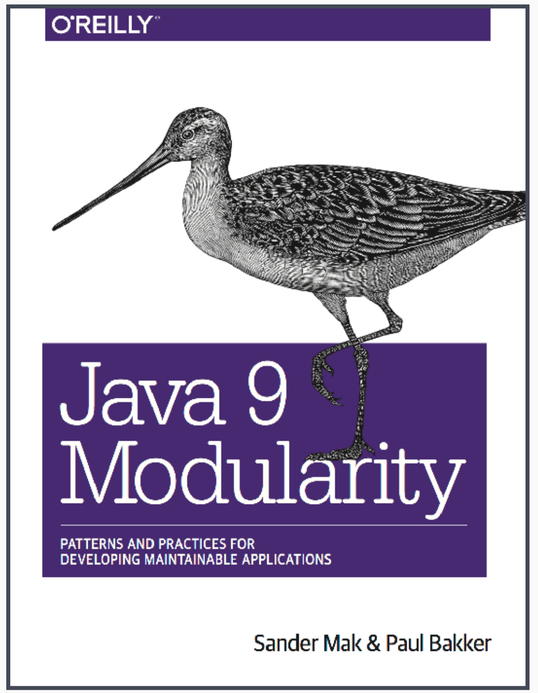
7 in 10 developers use the Oracle JDK in production

JDK 11 changes the game

@Sander_Mak

Thanks. Read More:

javamodularity.com



bit.ly/ps-sander

PLURALSIGHT