

What's up in Java 9 ?

Author: **Piotr Hejwowski**

Who am I ?

Piotrek

- Java developer in Codete
- <http://github.com/Hejwo>
- piotr.hejwowski@codete.com

Code of Conduct

- To Lightning Talk == brief overview of topic
- Just digging into Java 9
- More of knowledge base :) - **tons of links approaching**
- It will be full time conference at some point ;)

Short story

- JDK 1.0 -> 1996
- Java 5 -> 2004
- Java 6 -> 2006

Short story - Java 7

- JDK 1.0 -> 1996
- Java 5 -> 2004
- Java 6 -> 2006
- **Java 7 -> 2011**

What happened ?

- Multi-Catch -> **`catch(Exception1 | Exception2 ex)`**
- Diamond operators -> **`new ArrayList<>();`**
- NIO.2 -> **`Path src = Path.get("/etc/hosts");`**
- **`ForkJoinPool();` + *Tasks***
- Tons of minor stuff
- ...

<https://www.slideshare.net/boulderjug/55-things-in-java-7>

Short story - Java 8

- JDK 1.0 -> 1996
- Java 5 -> 2004
- Java 6 -> 2006
- Java 7 -> 2011
- **Java 8 -> 2014**



Java 8 -> Features, features everywhere

- JDK 1.0 -> 1996
- Java 5 -> 2004
- Java 6 -> 2006
- Java 7 -> 2011
- **Java 8 -> 2014**

What happened ?

- * Lambdas ! -> **Runnable** r1 = () -> **System.out.println("aaa");**
- * Extension methods -> **default Collections.<T>setReducer;**
- Annotation on Types -> **process(@nonnull List data) { ... }**
- * **Stream API**
- Permgen -> Metaspace
- Concurrency improvements
- * **DateTime API**
- JavaFX direct launch of JavaFX apps
- Getting ready for da Jigsaw
- ...

<https://www.slideshare.net/SimonRitter/javase8-55thingsv2-sritter>

Abandon all hope ! Java 9

- JDK 1.0 -> 1996
- Java 5 -> 2004
- Java 6 -> 2006
- Java 7 -> 2011
- Java 8 -> 2014
- **Java 9 -> 2017 (21 September)**



Java 9 - What we get for free ? \$\$\$

- Backward compatibility as always ! :)
- Better Use of Memory
- Better performance of :
 - Locking
 - Secure Apps
 - Graphics
- Better Use of Hardware
- Better documentation
- Faster Compilation
- Prettier Graphics

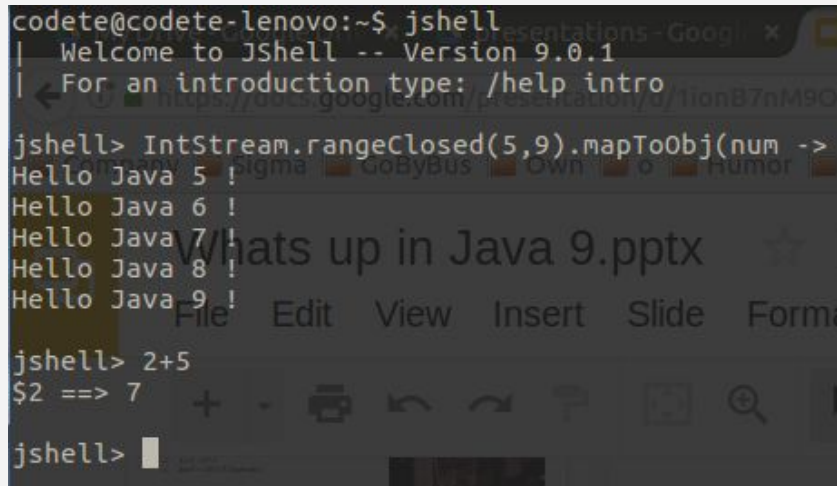
<https://www.slideshare.net/SimonRitter/55-new-features-in-jdk-9>

Let's dive into features !!!



JShell - meet interactive Java

- REPL at last ! Only after “a few” years
- No need for sample projects to run simple things
- Working autocomplete



```
codete@codete-lenovo:~$ jshell
| Welcome to JShell -- Version 9.0.1
| For an introduction type: /help intro

jshell> IntStream.rangeClosed(5,9).mapToObj(num ->
Hello Java 5 !
Hello Java 6 !
Hello Java 7 !
Hello Java 8 !
Hello Java 9 !

jshell> 2+5
$2 ==> 7

jshell>
```

<https://docs.oracle.com/javase/9/jshell/introduction-jshell.htm>

Reactive Streams as a standard

- Basically only a few common interfaces
- NO IMPLEMENTATION in raw Java 9
- But still we could join JavaRx 2, Akka Streams and Spring's Reactor

```
Flow.Publisher<String> publisher = new SubmissionPublisher<>();  
Flow.Subscriber<String> subscriber = null;  
Flow.Processor<String, Integer> processor = null;  
Flow.Subscription subscription = null;
```

<http://www.reactive-streams.org/>
<http://www.baeldung.com/java-9-reactive-streams>

Jigsaw



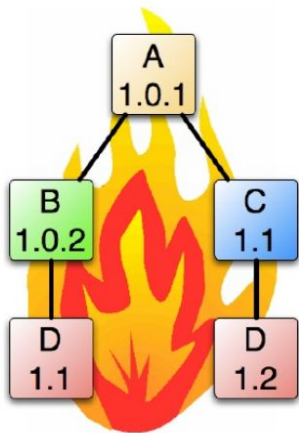
<https://www.youtube.com/watch?v=QnMDsl2GbOc>

<http://www.baeldung.com/project-jigsaw-java-modularity>

What's the puzzle ?

Which D version will be loaded ?

JAR HELL



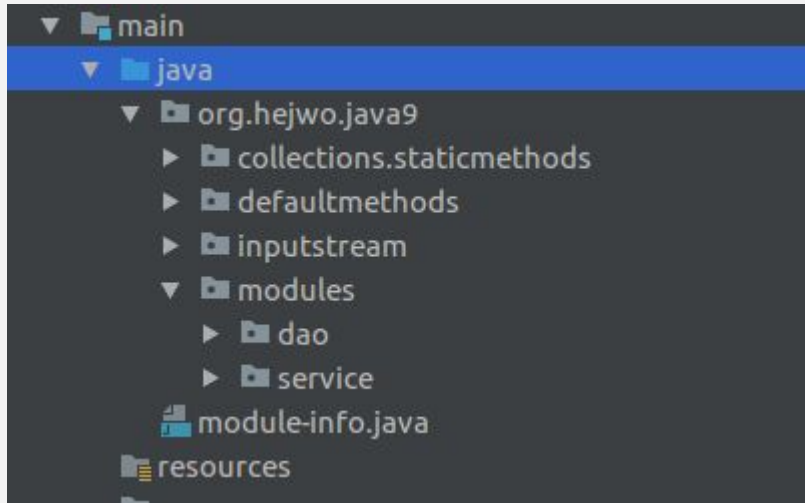
<https://www.youtube.com/watch?v=QnMDsl2GbOc>
<http://www.baeldung.com/project-jigsaw-java-modularity>

My package visibility (and Java system packages also) be like :

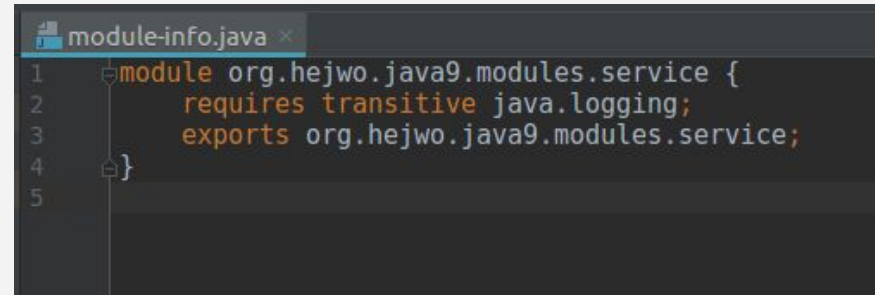


Simple accessors (*public*, *protected*, *<package>*, *private*)
are not enough !

What's the ANSWER ?



-->



<https://www.youtube.com/watch?v=QnMDsl2GbOc>

<http://www.baeldung.com/project-jigsaw-java-modularity>

Small things I

- Optionals are monads. At last.
- Factory methods for collections.
- Private methods in interfaces
- Private static methods also

```
// Immutable collection !!!  
List<Integer> list = List.of(1, 2, 3, 4);  
  
String str = "Hello !!!";  
Integer stringLength = Optional.ofNullable(str)  
    .map(String::length).orElse(0);  
System.out.println(stringLength);
```

```
public interface HighAbstraction {  
    private void foo() {  
        System.out.println("This will never change");  
    }  
  
    private static void foo2() {  
        System.out.println("I'm static !");  
    }  
}
```

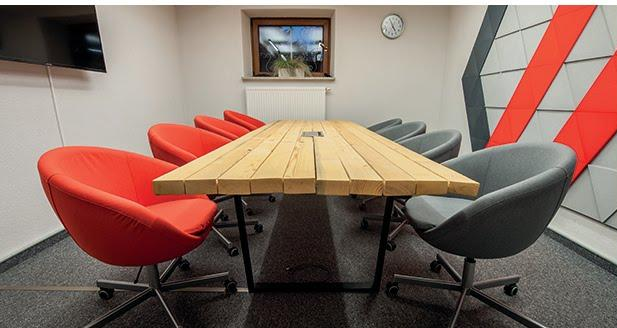

Small things II - Performance improvements

- G1 Garbage collector improvements
 - Concurrent class unloading
 - Lower pause
 - More mature in development
- More compacted Strings

Java 10 - Over horizon

- Java 10 is planned on March 2010 !!!
- Modules helped themselves
- Plans :
 - Pattern matching !
 - Underscore ready to be implemented ! _

```
String _ = "not valid";  
String __ = "valid";  
String STILL_OK = "valid";
```



Codete Przystalski Olechowski Śmiałek Spółka Jawna

ul. Olszańska 7, Kraków
ul. Wojciechowska 7e, Lublin
Bouchéstraße 12, Berlin

mail@codete.com

www.codete.com

Jigsaw - add ons

```
RichDao.java x
1 package org.hejwo.java9.modules.dao;
2
3 import sun.misc.Unsafe;
4
5 public class RichDao {
6
7     public static void main(String[] args) {
8         Unsafe unsafe = Unsafe.getUnsafe();
9
10    }
11
12 }
13
```

```
module-info.java x
1 module org.hejwo.java9.modules.service {
2     requires transitive jdk.unsupported;
3 }
4
5
```