

Java Features

Azat Satklichov

azats@seznam.cz,

<http://sahet.net/htm/java.html>,

<https://github.com/azatsatklichov/Java-Features>



JAVA 9

Stream API Improvements

```
//prevent NullPointerException
Stream.ofNullable(), Stream.empty();

//more
takeWhile()[while true], dropWhile()[drops until true]

//more
.iterate()[hasNext predicate as parameter]

@Deprecated(forRemoval = true, since = "9")
e.g java.lang.Object#finalize,
javax.net.ssl.HandshakeCompletedEvent#getPeerCertificateChain
```

```
List<String> ll = getList();
Stream<String> stream = ll == null ? Stream.empty(): ll.stream();

int count = Stream.ofNullable(ll).count();
System.out.println(count); //0

//or Apache CollectionUtils
org.apache.commons.collections4.CollectionUtils.emptyIfNull(list).stream()

//Java8 Optional.ofNullable
stream = Optional.ofNullable(ll).orElse(List.of()).stream();

Stream.of("a", "b", "c", " ", "d", "", "z").takeWhile(s ->
!s.isEmpty()).forEach(System.out::print); //abc d

Stream.of("a", "b", "c", " ", "d", "", "z").dropWhile(s ->
!s.isEmpty()).forEach(System.out::print); //z

Stream.of("a", "b", "c", " ", "d", "", "z").takeWhile(s ->
!s.isBlank()).forEach(System.out::print); //abc

Stream.of("a", "b", "c", " ", "d", "", " ", "z").dropWhile(s ->
!s.isBlank()).forEach(System.out::print); // d z

IntStream.iterate(0, x -> x < 100, i -> i + 4).forEach(System.out::print);
//04812162024
```

Optional & other Improvements

```
stream(), ifPresentOrElse(), or()
```

-- NEW Java9 Javadoc HTML5 way

```
javadoc -d C:/JAVA_doc_new -html5 _IntroJava9.java
```

- Jshell

- **Diagnostic tools:** Jcmd, jinfo, jmap, jstat

- <https://visualvm.github.io>

- Multirelease:

```
> javac --release 7 JRelease7.java
```

```
➤ javac --release 9 JRelease9.java
```

```
List<Optional<String>> list = Arrays.asList(Optional.empty(),  
Optional.of("A"), Optional.empty(), Optional.of("B"));
```

```
List<String> filteredListJava9 = list.stream()  
.flatMap(Optional::stream).collect(Collectors.toList()); //[A, B]
```

```
optional.ifPresentOrElse(x -> System.out.println("Value: " + x), () ->  
System.out.println("Not Present.)); //[A, B]
```

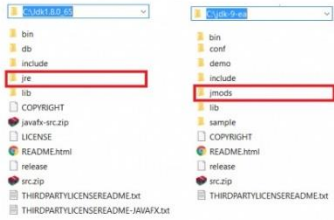
```
Optional<String> optional1 = Optional.of("Rimini");  
Supplier<Optional<String>> supplierString = () -> Optional.of("Not Present");  
optional1 = optional1.or(supplierString);  
optional1.ifPresent(x -> System.out.println("Value: " + x)); //Value: Rimini
```

```
//Java 9//11 - modular, JEE removed
➤ java --list-modules
```

Compare JDK 8 and JDK 9

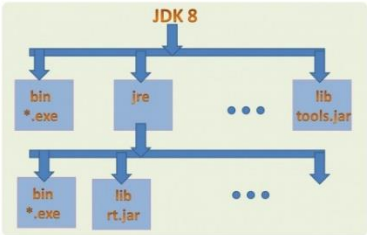
We know what a JDK software contains. After installing JDK 8 software, we can see a couple of directories like bin, jre, lib etc in Java Home folder.

However, Oracle Corp has changed this folder structure a bit differently as shown below.

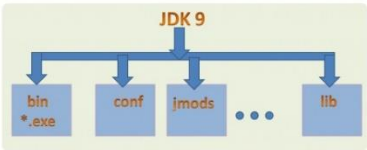


JDK 8 Vs JDK 9

JDK 8 Folder Structure:



JDK 9 Folder Structure:



Here JDK 9 does NOT contain JRE. In JDK 9, JRE is separated into a separate distribution folder. JDK 9 software contains a new folder “jmods”. It contains a set of Java 9 Modules as shown below.

JAVA 10 – Last free version of JDK

Root certificate - **cacerts**: With Java 10, Oracle has open-sourced the root certificates in Oracle

Optional class improvement:

orElseThrow - java.util.Optional, java.util.OptionalDouble, java.util.OptionalInt and java.util.OptionalLong each got a new method orElseThrow()

Unmodifiable collections - **copyOf()** on List, Set, Map and Collectors.**toUnmodifiableList()**

```
TRY: > cd cd C:\apps\Java\jdk-14\lib\security
> keytool -list -keystore cacerts
* changeit
```

```
OptionalInt opInt = OptionalInt.of(12345);
System.out.println("int Value via " + " orElseThrow() = " +
opInt.orElseThrow()); //NoSuchElementException
```

```
//existing one
IntStream.of(1, 2, 3).average().orElseThrow(() -> new
IllegalArgumentException("Array is empty"));
```

```
//MAP.put(entry.getKey(), ImmutableSet.copyOf(entry.getValue()));
MAP.put(entry.getKey(), Set.copyOf(entry.getValue()));
```

```
List<Integer> evenList = ints.stream()
.filter(i -> i % 2 == 0).collect(Collectors.toUnmodifiableList());
evenList.add(4); // UnsupportedOperationException
```

JAVA 11 – LTS Release after Java 8(2019), next is 17

Oracle JDK was licensed under Oracle Binary Code License Agreement. Stable and more performant.

OpenJDK has the GNU General Public License (GNU GPL) version 2 with a linking exception. OpenJDK, in contrast, will deliver releases more often.

There is no real technical difference between the two since the build process for the Oracle JDK is based on that of OpenJDK.

- ***Launch Single Source File without explicitly compiling***
- ***Run Shebang file***
- ***Java Flight Recorder (Events, Dataflows)***
- ***String methods***
- ***Files readString, writeString***
- ***New TLS crypto - ChaCha20, ..***

```
>java A_LaunchingSingleFileSourceFiles.java
```

For more about Java 1-15 Features:

<https://github.com/azatsatklichov/Java-Features>



THANK YOU