New Features In JDK 9 and JDK 10

Simon Ritter
Deputy CTO, Azul Systems
azul.com
JDK 9: Java Platform Modularity

- The core Java libraries are now a set of modules
  - 97 modules
    - 28 Java SE
    - 8 JavaFX
    - 59 JDK
    - 2 Oracle
- No more rt.jar or tools.jar
JDK 9: Internal API Encapsulation

- Most internal APIs now encapsulated
  - Override with
    - --illegal-access
    - --add-exports
    - --add-opens
- JEP 260 enables access to some via jdk.unsupported
  - sun.misc.Unsafe
JDK 9: jlink

Modular run-time image

bin  conf  lib  ...

jlink
JDK 9: Collection Factory Methods

Old

```java
Set<String> set = new HashSet<>();
set.add("a");
set.add("b");
set.add("c");
set = Collections.unmodifiableSet(set);
```

New

```java
Set<String> set = Set.of("a", "b", "c");
```
JDK 9: Stream Enhancements

- `dropWhile()` and `takeWhile()`
  - Like `skip()` and `limit()`
  - Uses Predicate rather than fixed value
JDK 9: jshell

- Read-Eval-Print Loop

```java
[MBP > .jshell
 | Welcome to JShell -- Version 9-ea
 | Type /help for help

[-> ArrayList<String> l
 | Added variable l of type ArrayList<String>

[-> l.add("hello")
 | java.lang.NullPointerException thrown
 | at (#10:1)

[-> l = new ArrayList<>();
 | Variable l has been assigned the value []

[-> l.add("hello")
 | Expression value is: true
 | assigned to temporary variable $4 of type boolean

[-> l
 | Variable l of type ArrayList<String> has value [hello]
```
JDK 9: Small Language Changes

- No more single underscore as a variable name
- Private methods in interfaces
- Effectively final variables in try-with-resources
JDK 9: Variable Handles

- Replacement for parts of sun.misc.Unsafe

- VarHandle
  - compareAndExchange(), compareAndSet()
  - getAndAdd(), getAndSet()
  - acquireFence(), releaseFence()
JDK 9: AOT Compilation

- Experimental feature
- Only java.base at the moment
  - Faster start-up, no interpreted mode
    - Platform specific
JDK 10: Local Variable Type Inference

- Java gets var!

```java
var list = new ArrayList<String>(); // infers ArrayList<String>
var stream = list.stream(); // infers Stream<String>
```
JDK 10: Application Class Data Sharing

- Share application classes between JVMs
  - Reduces resources used

-XX:+UseAppCDS  -XX:SharedArchiveFile=archive.jsa
JDK 10: Experimental JIT

- Extend use of Graal to JIT compilation
  - From AOT

- Part of Project Metropolis

-XX:+UnlockExperimentalVMOptions -XX:+UseJVMCICompiler
JDK 10: Alternative Heap Devices

- Some Non-Volatile RAM now as fast as DRAM
- -XX:AllocateHeapAt=<path>
JDK 10: New Collections APIs

- Create unmodifiable copies of collections
  
  - `List.copyOf(Collection)`
  - `Set.copyOf(Collection)`
  - `Map.copyOf(Collection)`
JDK 10: New Collectors For Streams

- Collectors
  - toUnmodifiableList
  - toUnmodifiableMap
  - toUnmodifiableSet
JDK 10: Container Support

- JVM now properly CGroup aware
- Initial values correctly calculated
  - Heap sizing
  - Thread pooling
JDK Release Cadence

- New JDK release every six months
  - No more waiting years
- May include features that break compatibility
- More agile development
JDK Updates

- Oracle Long Term Support Release (LTS)
  - Every three years
  - JDK 8, JDK 11, JDK 17...

- Not free for commercial use

- Only six-months for OpenJDK binaries
Zulu JDK

- Azul build of OpenJDK
  - Passes all TCK tests

- Free!

- Commercial support if you want it

www.azul.com/downloads/zulu