Taste of new in Java 9
Agenda

- A bit of history
- jshell
- JDK Benchmarks / jmh
- small stuff...
- jigsaw
- what’s beyond 9?
Bit of history

1991: Green (Oak)

1994: JDK EAs

Jan 1996: JDK 1.0

Feb 1997: JDK 1.1

Dec 1998: J2SE 1.2

May 2000: J2SE 1.3

Feb 2002: J2SE 1.4

Sep 2004: J2SE 5.0

Dec 2006: Java SE 6

Jul 2011: Java SE 7

Mar 2014: Java SE 8

Dec 2015: Java SE 9 feature complete

Sep 2016: Java SE 9 release
New versioning scheme

● Current scheme:
  ○ Minor releases containing changes beyond security fixes are multiples of 20
  ○ Security releases based on the previous minor release are odd numbers incremented by five, or by six if necessary in order to keep the update number odd
New versioning scheme

- New scheme:
  - MAJOR.MINOR.SECURITY
  - 1.8.0 => 8.0.0
  - 1.8.0_05 => 8.0.1
  - 1.8.0_11 => 8.0.2
  - 1.8.0_20 => 8.1.2
  - 1.8.0_25 => 8.1.3
JDK Benchmark Suite

- based on JMH
- stable, tuned benchmarks
- targeted for continuous performance check
- lot of benchmarks for standard libraries
- comparison with JDK 8
HttpResponse response = HttpRequest
    .create(new URI("http://www.ocado.com"))
    .body(noBody())
    .GET().send();

int responseCode = response.responseCode();
String responseBody = response.body(asString());

System.out.println(responseBody);
HttpRequest req = HttpRequest
    .create(new URI("http://www.ocado.com"))
    .body(noBody())
    .GET();
CompletableFuture<HttpResponse> asyncResp = req.sendAsync();
Thread.sleep(10);
if (!asyncResp.isDone()) {
    asyncResp.cancel(true);
    System.err.println("timeout");
    return;
}
HttpResponse response = asyncResp.get();
JVM logging

-Xlog[:option]

option := [what][:[<output>][:<decorators>][:<output-options>]]

'help'
'disable'

what := <selector>[,...]

selector := <tag-set>[*][=<level>]

tag-set := <tag>[+...]

'tag'
'all'

tag := name of tag

level := trace
debug
info
warning
error

output := 'stderr'
'stdout'
[file=]<filename>

decorators := <decorator>[,...]

'decorator' := time
uptime
timemillis
uptimemillis
timenanos
uptimenanos
pid
tid
level
tags

output-options := <output_option>[,...]

'output-option' := filecount=<file count>
filesize=<file size in kb>
parameter=value
Coin project - language adjustments

- Allow `@SaveVargs` on private instance methods
- Allow effectively-final variables in `try-with-resources`
- More type inference with generics
- Underscore will be no more a valid identifier
- Private methods in interfaces
Hotspot / JDK diagnostics

- `print_class_summary`
- `print_codegen_list`
- `print_utf8pool`
- `dump_codelist`
- `print_codeblocks`
- `set_vmflags`
Other...

- improved contented locking
- variable handles (Unsafe!)
- sjavac
- fixed imports processing by javac
- Security (Datagram / Application transport layer)
- HTML5 javadoc
- Unicode 7.0
Other...

- String internal representation altered
- Java-Level JVM Compiler Interface
- Parser API for Nashorn
- Support for AArch64 (ARMv8) architecture
- Compile for Older Platform Versions
- G1 will be default garbage collector
- Store Interned Strings in CDS archives
Jigsaw - modules for Java

- Motivation:
  - JAR/classpath hell reduction
  - unexpressed/transitive dependencies
  - manual dependencies management
  - hide some (private) packages; com.sun.misc.*
  - system instead of manual security
  - JRE size reduction
Jigsaw - piece of action...
module com.foo.app {
    requires com.foo.bar;
    requires java.sql;
}

module java.sql {
    requires java.logging;
    requires java.xml;
    exports java.sql;
    exports javax.sql;
    exports javax.transaction.xa;
}
Jigsaw - one more example
Jigsaw - implied readability

String url = ...;
Properties props = ...;
Driver d = DriverManager.getDriver(url);
Connection c = d.connect(url, props);
d.getParentLogger().info("Connection acquired");
Jigsaw - implied readability

module java.sql {
    requires public java.logging;
    requires public java.xml;
    exports java.sql;
    exports javax.sql;
    exports javax.transaction.xa;
}

Jigsaw - implied readability
module com.mysql.jdbc {
    requires java.sql;
    requires org.slf4j;
    exports com.mysql.jdbc;
}
module java.sql {
    requires public java.logging;
    requires public java.xml;
    exports java.sql;
    exports javax.sql;
    exports javax.transaction.xa;
    uses java.sql.Driver;
}
module com.mysql.jdbc {
    requires java.sql;
    requires org.slf4j;
    exports com.mysql.jdbc;
    provides java.sql.Driver with com.mysql.jdbc.Driver;
}
Jigsaw - services
Jigsaw - qualified exports

module java.base {
    ...
    exports sun.reflect to
        java.corba,
        java.logging,
        java.sql,
        java.sql.rowset,
        jdk.scripting.nashorn;
}
Jigsaw - qualified exports
Java 10: value classes

value class Point {
    int x;
    int y;
    Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

Java 10: generics

- Reified Generics
- Generics with values

```java
List<int> intList = new ArrayList<>();
...
int val = intList.get(0);
class List<Point> { Point get(Point index); }
```
Thank you!

...questions?