

Java SE

State Of The Union



Gil Tene, CTO & co-Founder, Azul Systems
@giltene

Agenda

- Brief introduction
- Some history
- Some more history & future speak
- Some obligatory “What’s in Java 9?” stuff
- Let’s chat

About me: Gil Tene

- co-founder, CTO @Azul Systems
- Have been working on “think different” GC approaches since 2002
- A Long history building Virtual & Physical Machines, Operating Systems, Enterprise apps, etc...
- I also depress people by pulling the wool up from over their eyes...
- JCP EC member, OpenJDK contributor...



* working on real-world trash compaction issues, circa 2004

Java SE

State Of The Union



Gil Tene, CTO & co-Founder, Azul Systems
@giltene

A brief history of Java



By Java. Java is a trademark of Sun Microsystems, Inc. All rights reserved. Java is a registered trademark of Sun Microsystems, Inc. in the USA and other countries.

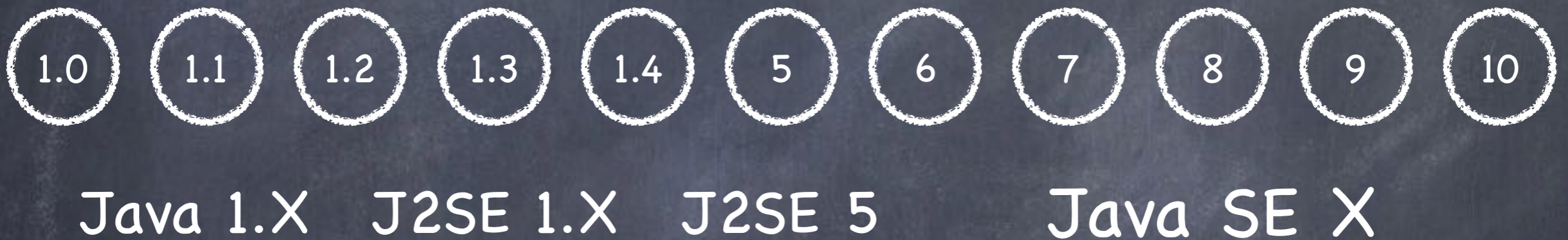
The Java SE Timeline



The Java SE Timeline



The Java SE Timeline



The Java SE Timeline



The Java SE Timeline

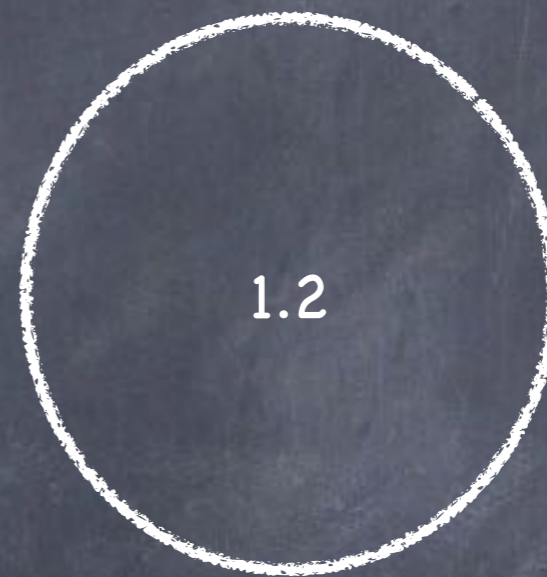
1.0

1.1

1.2

1.3

The Java SE Timeline



Reflection

`java.util.Collection`

Swing

Weak refs

The Java SE Timeline

1.2

`java.util.Collection`

Weak refs

1.3

1.4

NIO

5

The Java SE Timeline

1.4

NIO

5

Generics<>

`java.util.concurrent`

6

7

The Java SE Timeline

7

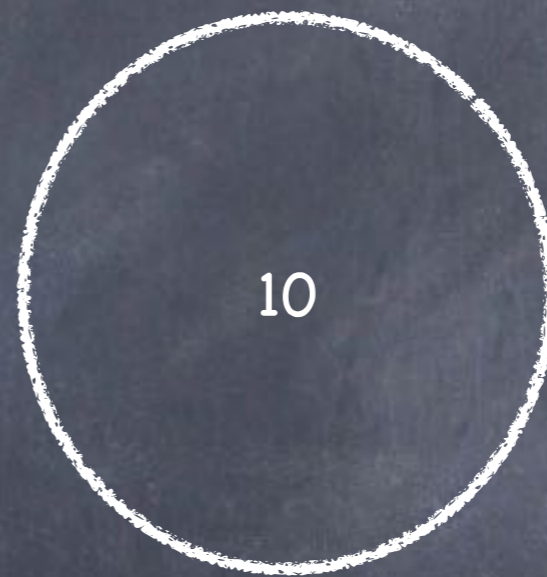
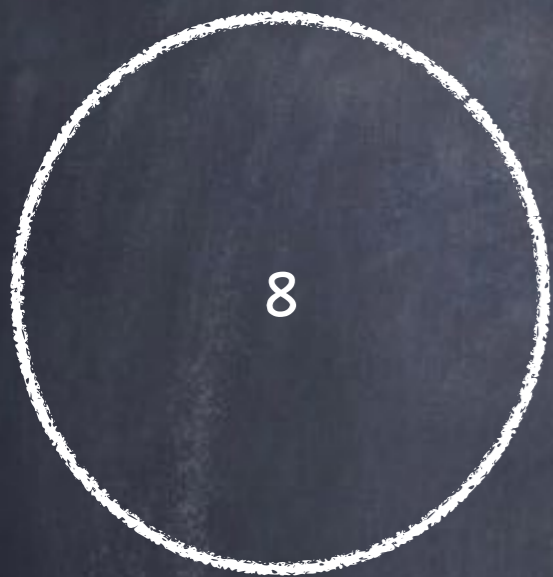
8

9



Stream API

The Java SE Timeline



?
?
?



<>

?

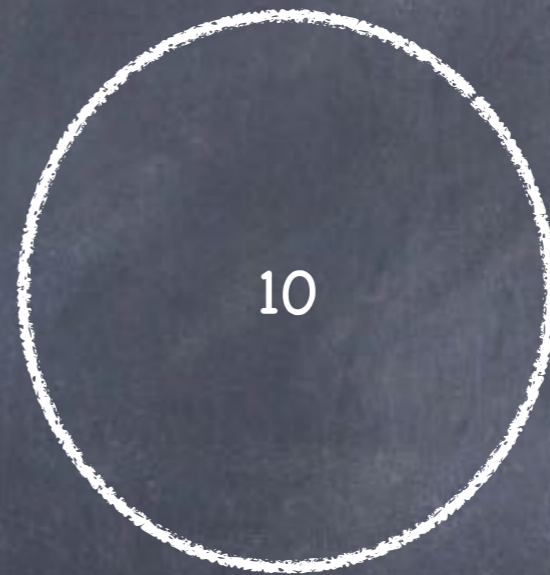
?

?

Stream API



The Java SE Timeline



val? var?

?

?

Arrays 2.0



<>

?

value
types

JNR/FFI

Jigsaw!

HashMap<int>

Stream API



How is Java Doing?



How is Java Doing?

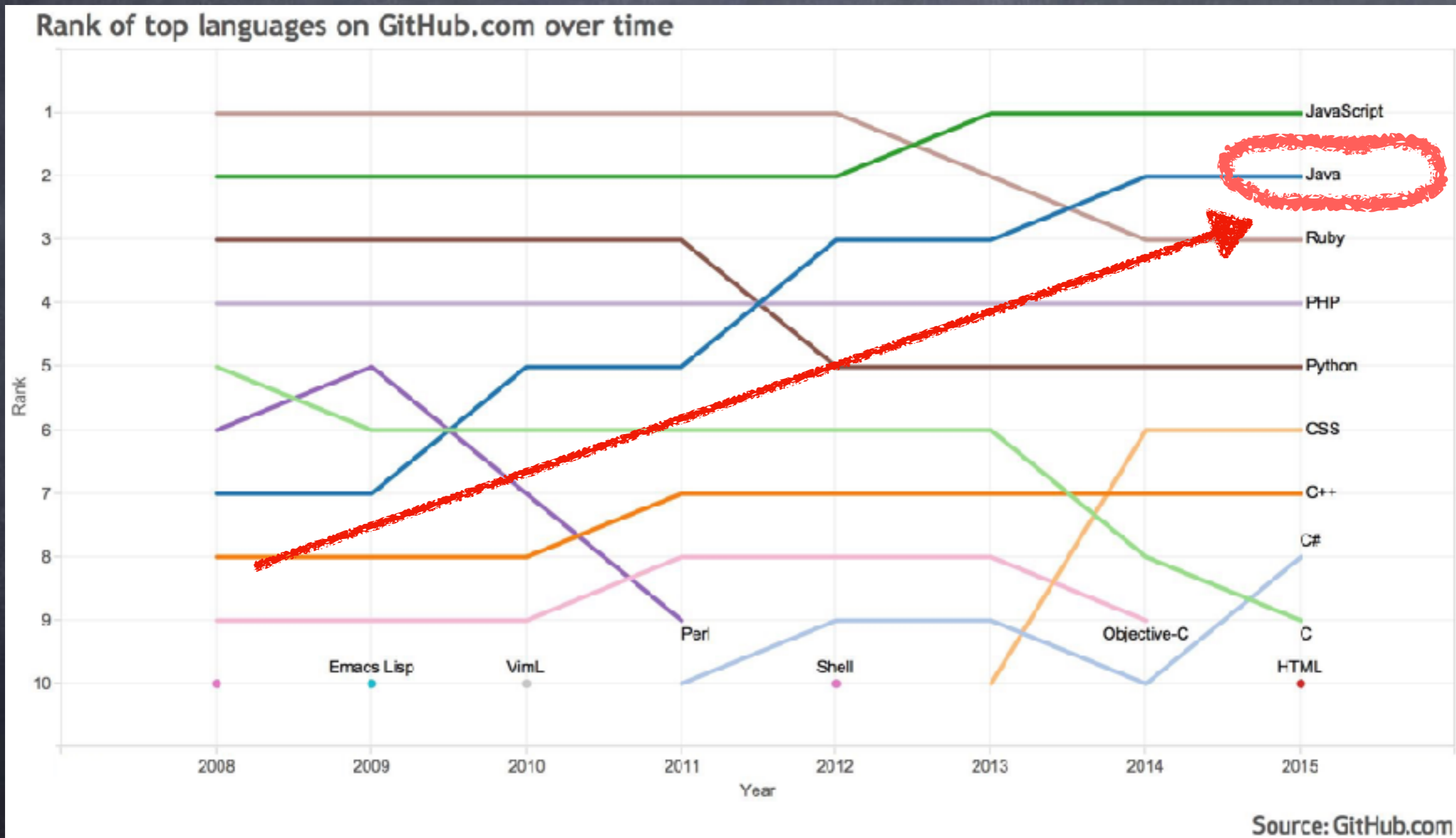
Is Java still popular?

Are other things cooler?

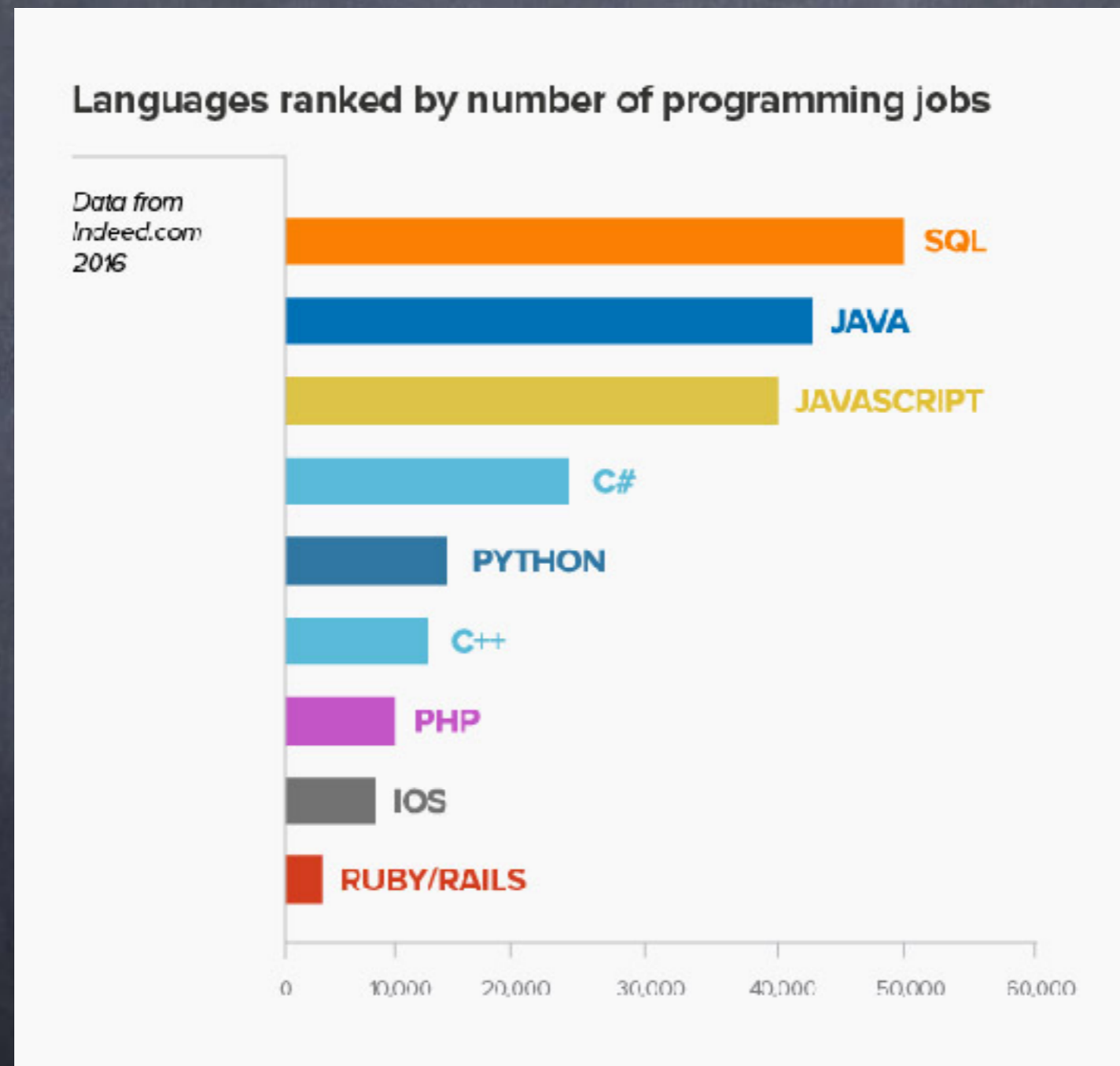
Is it trending away?



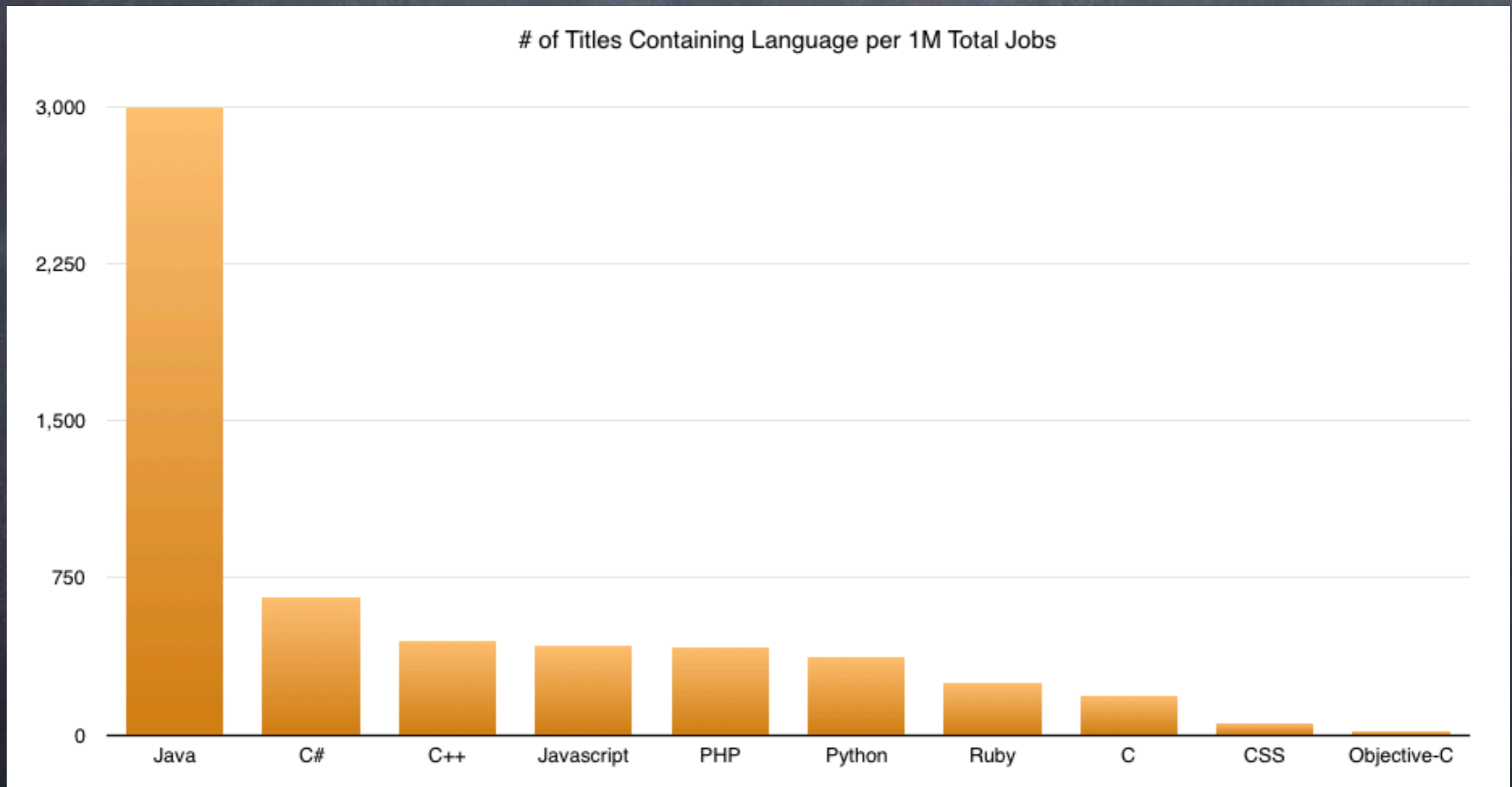
How is Java Doing?



How is Java Doing?



How is Java Doing?



A different historical view: Java leaps and bounds



Java leaps and bounds

1995–2001: Server domination

2001–2010: Applications

2010–2016+: Infrastructure



1995–2001: Server domination

JDBC Pools Servlets JavaEE

2001–2010: Applications

WebLogic WebSphere JBoss Tomcat

Portals Web Services SOA Data Grids

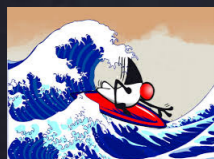
...

2010–2016+: Infrastructure

Hadoop Cassandra Lucene/Solr/Elastic

Kafka Spark Storm Zookeeper

...



The state of the union is
GOOD



What's in Java 9



What's in Java 9?

Modules!



What's in Java 9?

Modules!
and

some other stuff...



What's in Java 9?

Modules

some other stuff...



What's in Java 9?

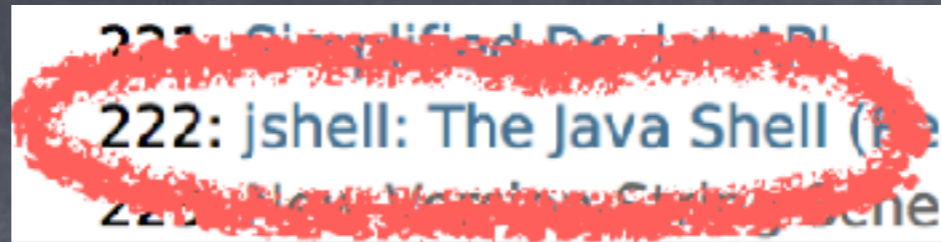
Modules

of the
some other stuff...



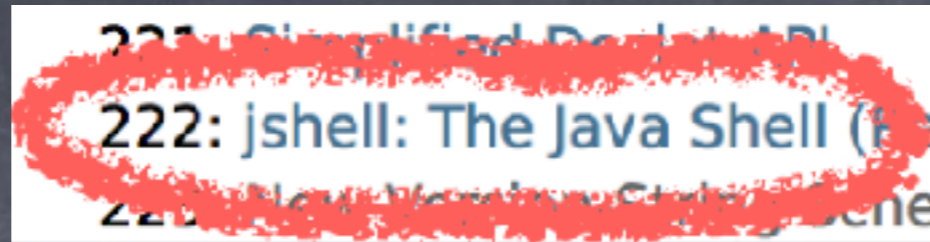
JEPs targeted to JDK 9, so far

- 102: Process API Updates
- 110: HTTP 2 Client
- 145: Improved G1 Garbage Logging
- 158: Unified JVM Logging
- 177: Fragmented G1 Cache
- 193: Variable Handles
- 199: Smart Java Compilation, Phase Two
- 200: The Modular JDK
- 201: Modular Source Code
- 211: Elide Deprecation Waivers
- 212: Resolve Lint and Doclint
- 213: Milling Project Coin
- 214: Remove GC Combinators
- 215: Tiered Attribution for GC
- 216: Process Import Statement
- 217: Annotations Pipeline
- 219: Datagram Transport Layer
- 220: Modular Run-Time Image
- 221: Simplified Desktop API
- 222: jshell: The Java Shell (REPL)
- 223: New Windows Start Menu
- 224: HTML5 Javadoc
- 225: Javadoc Search
- 226: UTF-8 Property Files
- 227: Unicode 7.0
- 228: Add More Diagnostic Commands
- 229: Create PKCS12 Keystores by Default
- 231: Remove Launch-Time JRE Version Selection
- 232: Improve Secure Application Performance
- 233: Generate Run-Time Compiler Tests Automatically
- 235: Test Class-File Attributes Generated by java
- 236: Parser API for Nashorn
- 237: Linux/AArch64 Port
- 238: Multi-Release JAR Files
- 240: Remove the JVM TI hprof Agent
- 241: Remove the jhat Tool
- 243: Java-Level JVM Compiler Interface
- 244: TLS Application-Layer Protocol Negotiation
- 245: Validate JVM Command-Line Flag Argument
- 246: Leverage CPU Instructions for GHASH and SHA-256
- 247: Compile for Older Platform Versions
- 248: Make G1 the Default Garbage Collector
- 249: OCSP Stapling for TLS
- 250: Store Interned Strings in CDS Archives
- 251: Multi-Resolution Images
- 252: Use CLDR Locale Data by Default
- 253: Prepare JavaFX UI Controls & CSS APIs for Modularization
- 254: Compact Strings
- 255: Merge Selected Xerces 2.11.0 Updates into JAXP
- 256: BeanInfo Annotations
- 257: Update JavaFX/Media to Newer Version of GStreamer
- 258: HarfBuzz Font-Layout Engine
- 259: Stack-Walking API
- 260: Encapsulate Most Internal APIs
- 261: Module System
- 262: TIFF Image I/O
- 263: HiDPI Graphics on Windows and Linux
- 264: Platform Logging API and Service
- 265: Marlin Graphics Renderer
- 266: More Concurrency Updates
- 267: Unicode 8.0
- 268: XML Catalogs
- 269: Convenience Factory Methods for Collections
- 270: Reserved Stack Areas for Critical Sections
- 271: Unified GC Logging
- 272: Platform-Specific Desktop Features
- 273: DRBG-Based SecureRandom Implementations
- 274: Enhanced Method Handles
- 275: Modular Java Application Packaging
- 276: Dynamic Linking of Language-Defined Object Models
- 277: Enhanced Deprecation
- 278: Additional Tests for Humongous Objects in G1
- 279: Improve Test-Failure Troubleshooting
- 280: Indify String Concatenation
- 281: HotSpot C++ Unit-Test Framework
- 282: jlink: The Java Linker
- 283: Enable GTK 3 on Linux
- 284: New HotSpot Build System
- 285: Spin-Wait Hints
- 287: SHA-2 Hash Algorithms
- 288: Disable SHA-1 Certificates
- 289: Deprecate the Applet API
- 290: Filter Incoming Serialization Data
- 292: Implement Selected ECMAScript 6 Features in Nashorn



Java has a REPL!

(read... eval... print... loop...)

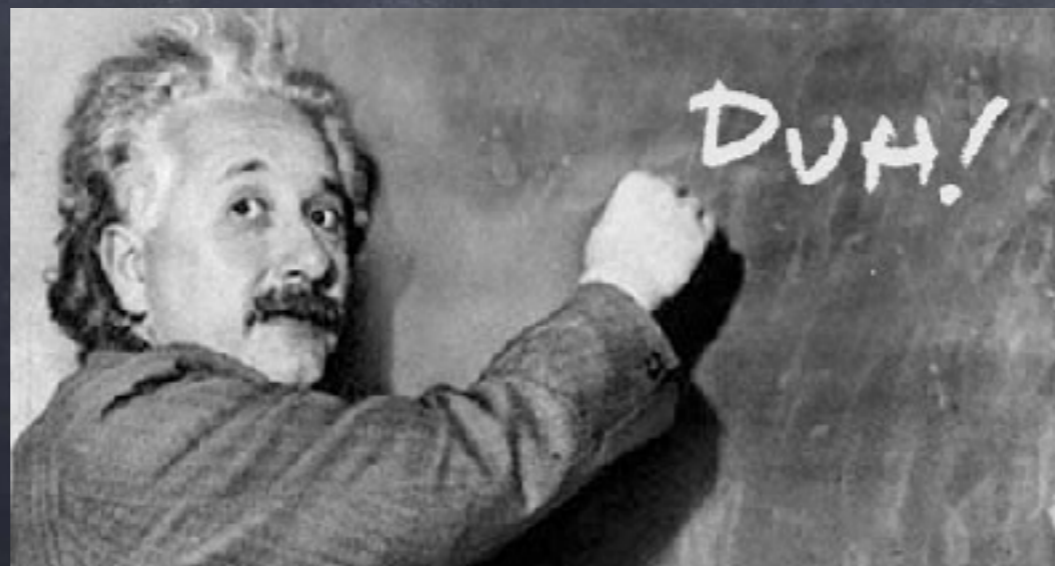


Java has a REPL!

Why is this important?

Because virtually every other modern environment has one...

***(where "modern" starts with "LISP")



JEPs targeted to JDK 9, so far

- 102: Process API Updates
- 110: HTTP 2 Client
- 145: Improved Concurrent Locking
- 158: Unified JVM Logging
- 187: Enhanced G1 Cache
- 193: Variable Handles
- 197: Enhanced G1 Cache
- 199: Smart Java Compilation, Phase Two
- 200: The Modular JDK
- 201: Modular Source Code
- 211: Elide Deprecation Wa
- 212: Resolve Lint and Doct
- 213: Milling Project Coin
- 214: Remove GC Combinat
- 215: Tiered Attribution for
- 216: Process Import Stater
- 217: Annotations Pipeline
- 219: Datagram Transport L
- 220: Modular Run-Time Im
- 221: Simplified Desktop API
- 222: jshell: The Java Shell (B
- 223: New Windows System Gene
- 224: HTML5 Javadoc
- 225: Javadoc Search
- 226: UTF-8 Property Files
- 227: Unicode 7.0
- 228: Add More Diagnostic Commands
- 229: Create PKCS12 Keystores by Default
- 231: Remove Launch-Time JRE Version Selection
- 232: Improve Secure Application Performance
- 233: Generate Run-Time Compiler Tests Automal
- 235: Test Class-File Attributes Generated by java
- 236: Parser API for Nashorn
- 237: Linux/AArch64 Port
- 238: Multi-Release JAR Files
- 240: Remove the JVM TI hprof Agent
- 241: Remove the jhat Tool
- 243: Java-Level JVM Compiler Interface
- 244: TLS Application-Layer Protocol Negotiation
- 245: Validate JVM Command-Line Flag Argument
- 246: Leverage CPU Instructions for GHASH and
- 247: Compile for Older Platform Versions
- 248: Make G1 the Default Garbage Collector
- 249: OCSP Stapling for TLS
- 250: Store Interned Strings in CDS Archives
- 251: Multi-Resolution Images
- 252: Use CLDR Locale Data by Default
- 253: Prepare JavaFX UI Controls & CSS APIs for Modularization
- 254: Compact Strings
- 255: Merge Selected Xerces 2.11.0 Updates into JAXP
- 256: BeanInfo Annotations
- 257: Update JavaFX/Media to Newer Version of GStreamer
- 258: HarfBuzz Font-Layout Engine
- 259: Stack-Walking API
- 260: Encapsulate Most Internal APIs
- 261: Module System
- 262: TIFF Image I/O
- 263: HiDPI Graphics on Windows and Linux
- 264: Platform Logging API and Service
- 265: Marlin Graphics Renderer
- 266: More Concurrency Updates
- 267: Unicode 8.0
- 268: XML Catalogs
- 269: Convenience Factory Methods for Collections
- 270: Reserved Stack Areas for Critical Sections
- 271: Unified GC Logging
- 272: Platform-Specific Desktop Features
- 273: DRBG-Based SecureRandom Implementations
- 274: Enhanced Method Handles
- 275: Modular Java Application Packaging
- 276: Dynamic Linking of Language-Defined Object Models
- 277: Enhanced Deprecation
- 278: Additional Tests for Humongous Objects in G1
- 279: Improve Test-Failure Troubleshooting
- 280: Indify String Concatenation
- 281: HotSpot C++ Unit-Test Framework
- 282: jlink: The Java Linker
- 283: Enable GTK 3 on Linux
- 284: New HotSpot Build System
- 285: Spin-Wait Hints
- 287: SHA-2 Hash Algorithms
- 288: Disable SHA-1 Certificates
- 289: Deprecate the Applet API
- 290: Filter Incoming Serialization Data
- 292: Implement Selected ECMAScript 6 Features in Nashorn

193: Variable Handles

1.7: Concurrent Collections

VarHandles

“Define a standard means to invoke the equivalents of `java.util.concurrent.atomic` and `sun.misc.Unsafe` operations upon object fields and array elements.”

Unsafe?

Unconfused?

Safe Safe?

Unsafe Safe

Unsafe?

Safe Unsafe

Unsafe Unsafe

Safe Safe?

Unsafe Safe

~~Unsafe?~~

Safe Unsafe

Unsafe Unsafe

~~Unsafe?~~

Is unsafe really going away?

No.

193: Variable Handles

1.7. Customized Cache

VarHandles

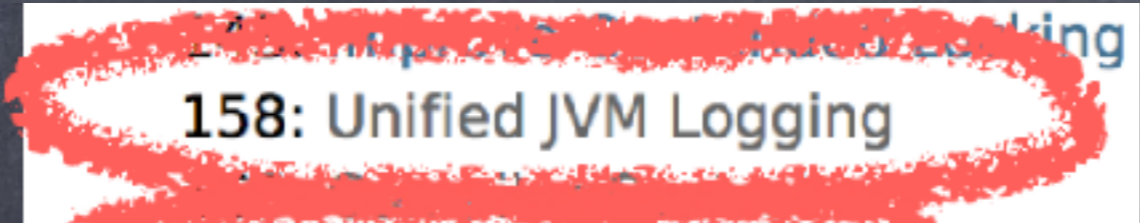
Also (finally) provides a fencing API

Supports ordering fences, but also:

```
try {
    <method body>
} finally {
    Reference.reachabilityFence(this);
}
```

JEPs targeted to JDK 9, so far

- 102: Process API Updates
- 110: HTTP 2 Client
- 111: Improved Error Reporting
- 158: Unified JVM Logging
- 193: Variable Handles
- 197: Concurrent Copy Cache
- 199: Smart Java Compilation, Phase Two
- 200: The Modular JDK
- 201: Modular Source Code
- 211: Elide Deprecation Warnings
- 212: Resolve Lint and Doclint
- 213: Milling Project Coin
- 214: Remove GC Combinators
- 215: Tiered Attribution for GC
- 216: Process Import Statement
- 217: Annotations Pipeline
- 219: Datagram Transport Layer
- 220: Modular Run-Time Image
- 221: Simplified Deployment API
- 222: jshell: The Java Shell (BETA)
- 223: New Windows System Launcher
- 224: HTML5 Javadoc
- 225: Javadoc Search
- 226: UTF-8 Property Files
- 227: Unicode 7.0
- 228: Add More Diagnostic Commands
- 229: Create PKCS12 Keystores by Default
- 231: Remove Launch-Time JRE Version Selection
- 232: Improve Secure Application Performance
- 233: Generate Run-Time Compiler Tests Automatically
- 235: Test Class-File Attributes Generated by java
- 236: Parser API for Nashorn
- 237: Linux/AArch64 Port
- 238: Multi-Release JAR Files
- 240: Remove the JVM TI hprof Agent
- 241: Remove the jhat Tool
- 243: Java-Level JVM Compiler Interface
- 244: TLS Application-Layer Protocol Negotiation
- 245: Validate JVM Command-Line Flag Argument
- 246: Leverage CPU Instructions for GHASH and SHA-2
- 247: Compile for Older Platform Versions
- 248: Make G1 the Default Garbage Collector
- 249: OCSP Stapling for TLS
- 250: Store Interned Strings in CDS Archives
- 251: Multi-Resolution Images
- 252: Use CLDR Locale Data by Default
- 253: Prepare JavaFX UI Controls & CSS APIs for Modularization
- 254: Compact Strings
- 255: Merge Selected Xerces 2.11.0 Updates into JAXP
- 256: BeanInfo Annotations
- 257: Update JavaFX/Media to Newer Version of GStreamer
- 258: HarfBuzz Font-Layout Engine
- 259: Stack-Walking API
- 260: Encapsulate Most Internal APIs
- 261: Module System
- 262: TIFF Image I/O
- 263: HiDPI Graphics on Windows and Linux
- 264: Platform Logging API and Service
- 265: Marlin Graphics Renderer
- 266: More Concurrency Updates
- 267: Unicode 8.0
- 268: XML Catalogs
- 269: Convenience Factory Methods for Collections
- 270: Reserved Stack Areas for Critical Sections
- 271: Unified GC Logging
- 272: Platform-Specific Desktop Features
- 273: DRBG-Based SecureRandom Implementations
- 274: Enhanced Method Handles
- 275: Modular Java Application Packaging
- 276: Dynamic Linking of Language-Defined Object Models
- 277: Enhanced Deprecation
- 278: Additional Tests for Humongous Objects in G1
- 279: Improve Test-Failure Troubleshooting
- 280: Indify String Concatenation
- 281: HotSpot C++ Unit-Test Framework
- 282: jlink: The Java Linker
- 283: Enable GTK 3 on Linux
- 284: New HotSpot Build System
- 285: Spin-Wait Hints
- 287: SHA-2 Hash Algorithms
- 288: Disable SHA-1 Certificates
- 289: Deprecate the Applet API
- 290: Filter Incoming Serialization Data
- 292: Implement Selected ECMAScript 6 Features in Nashorn



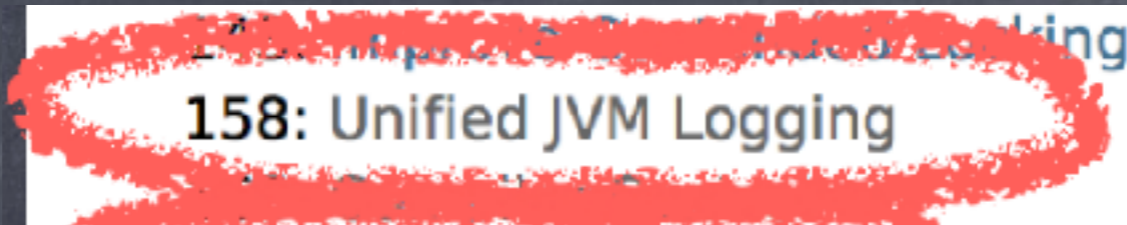
Unified JVM Logging

Current Logging stuff:

Logging Flags

```
-XX:+TraceClassResolution      -XX:+TraceExceptions
-XX:+TraceClassUnloading      -XX:+PrintGCDetails
-XX:+PrintGC                  -XX:+TraceMonitorInflation
-XX:+PrintVtables             -XX:+TraceClassLoaderData
-XX:+TraceVMOperation         -XX:+PrintGCCause
-XX:+TraceClassInitialization -XX:+TraceMonitorMismatch
-XX:+TraceDynamicGCThreads    -XX:+TraceSafepoint      -XX:+TraceRedefineClasses
-XX:+VerboseVerification     -XX:+TraceStartupTime
-XX:+PrintCMSStatistics       -XX:+TraceDefaultMethods -XX:+PrintTaskQueue
-XX:+TraceBiasedLocking
-XX:+PrintJNIgcStalls         -XX:+PrintCompressedOopsMode -XX:+PrintTLAB
-XX:+PrintPLAB                -XX:+TraceReferenceGC
-XX:+TraceClassResolution
-XX:+TraceMonitorInflation
```





Unified JVM Logging

New world order for logging: -Xlog

Look up Marcus Larsson's excellent
"Unified JVM Logging in JDK 9"
from JavaOne 2016 [CON6225]

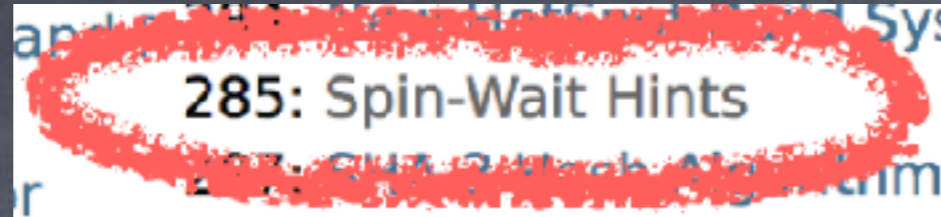
JEPs targeted to JDK 9, so far

- 102: Process API Updates
- 110: HTTP 2 Client
- 122: Improved Class-File Loading
- 158: Unified JVM Logging
- 193: Variable Handles
- 197: Compact and G1 Cache
- 199: Smart Java Compilation, Phase Two
- 200: The Modular JDK
- 201: Modular Source Code
- 211: Elide Deprecation Warnings
- 212: Resolve Lint and Doclint
- 213: Milling Project Coin
- 214: Remove GC Combinator
- 215: Tiered Attribution for GC
- 216: Process Import Statement
- 217: Annotations Pipeline
- 219: Datagram Transport Layer
- 220: Modular Run-Time Image
- 221: Simplified Desktop API
- 222: jshell: The Java Shell (REPL)
- 223: New Windows System Launcher
- 224: HTML5 Javadoc
- 225: Javadoc Search
- 226: UTF-8 Property Files
- 227: Unicode 7.0
- 228: Add More Diagnostic Commands
- 229: Create PKCS12 Keystores by Default
- 231: Remove Launch-Time JRE Version Selection
- 232: Improve Secure Application Performance
- 233: Generate Run-Time Compiler Tests Automatically
- 235: Test Class-File Attributes Generated by java
- 236: Parser API for Nashorn
- 237: Linux/AArch64 Port
- 238: Multi-Release JAR Files
- 240: Remove the JVM TI hprof Agent
- 241: Remove the jhat Tool
- 243: Java-Level JVM Compiler Interface
- 244: TLS Application-Layer Protocol Negotiation
- 245: Validate JVM Command-Line Flag Argument
- 246: Leverage CPU Instructions for GHASH and AES
- 247: Compile for Older Platform Versions
- 248: Make G1 the Default Garbage Collector
- 249: OCSP Stapling for TLS
- 250: Store Interned Strings in CDS Archives
- 251: Multi-Resolution Images
- 252: Use CLDR Locale Data by Default
- 253: Prepare JavaFX UI Controls & CSS APIs for Modularization
- 254: Compact Strings
- 255: Merge Selected Xerces 2.11.0 Updates into JAXP
- 256: BeanInfo Annotations
- 257: Update JavaFX/Media to Newer Version of GStreamer
- 258: HarfBuzz Font-Layout Engine
- 259: Stack-Walking API
- 260: Encapsulate Most Internal APIs
- 261: Module System
- 262: TIFF Image I/O
- 263: HiDPI Graphics on Windows and Linux
- 264: Platform Logging API and Service
- 265: Marlin Graphics Renderer
- 266: More Concurrency Updates
- 267: Unicode 8.0
- 268: XML Catalogs
- 269: Convenience Factory Methods for Collections
- 270: Reserved Stack Areas for Critical Sections
- 271: Unified GC Logging
- 272: Platform-Specific Desktop Features
- 273: DRBG-Based SecureRandom Implementations
- 274: Enhanced Method Handles
- 275: Modular Java Application Packaging
- 276: Dynamic Linking of Language-Defined Object Models
- 277: Enhanced Deprecation
- 278: Additional Tests for Humongous Objects in G1
- 279: Improve Test-Failure Troubleshooting
- 280: Indify String Concatenation
- 281: HotSpot C++ Unit-Test Framework
- 282: jlink: The Java Linker
- 283: Enable GTK 3 on Linux
- 284: New Mac OS Build System
- 285: Spin-Wait Hints
- 287: SHA-2/3 Hash Algorithms
- 288: Disable SHA-1 Certificates
- 289: Deprecate the Applet API
- 290: Filter Incoming Serialization Data
- 292: Implement Selected ECMAScript 6 Features in Nashorn



Spin-Wait Hints

My Favorite...



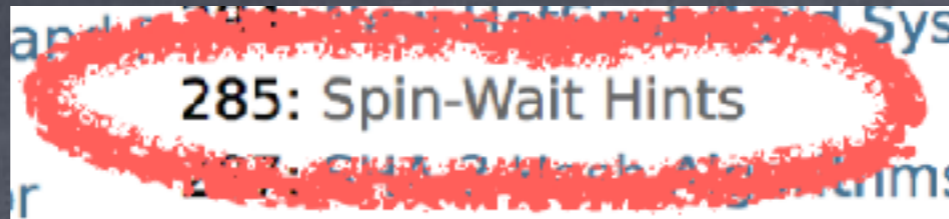
Spin-Wait Hints

Adds a single method to the JDK:

```
java.lang.Thread.onSpinWait()
```

Which does absolutely nothing...

but it does nothing faster...

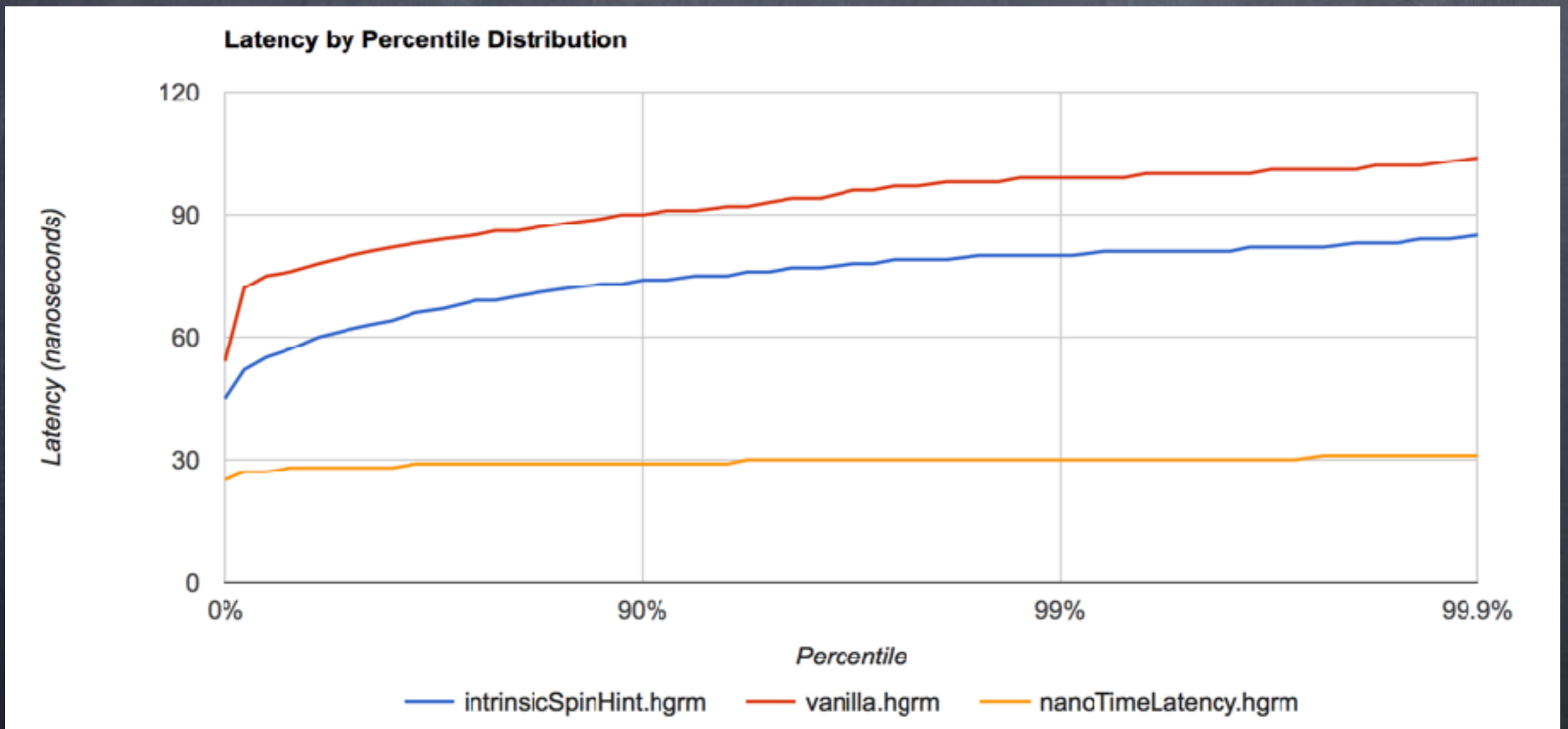


java.lang.Thread.onSpinWait()

```
while (spinData >= 0) {  
    while ((spinData & 0x1) == 0) {  
        // busy spin until ready to consume  
        ThreadHints.onSpinWait();  
    }  
    spinData++; // consume  
}
```


285: Spin-Wait Hints

java.lang.Thread.onSpinWait()



JEPs targeted to JDK 9, so far

- 102: Process API Updates
- 110: HTTP 2 Client
- 122: Simplified Desktop API
- 158: Unified JVM Logging
- 193: Variable Handles
- 197: Compact and G1 Cache
- 199: Smart Java Compilation, Phase Two
- 200: The Modular JDK
- 201: Modular Source Code
- 211: Elide Deprecation Waivers
- 212: Resolve Lint and Doclint
- 213: Milling Project Coin
- 214: Remove GC Combinator
- 215: Tiered Attribution for GC
- 216: Process Import Statement
- 217: Annotations Pipeline
- 219: Datagram Transport Layer
- 220: Modular Run-Time Image
- 221: Simplified Desktop API
- 222: jshell: The Java Shell (BETA)
- 223: Java-Level JVM Compiler Interface
- 224: HTML5 Javadoc
- 225: Javadoc Search
- 226: UTF-8 Property Files
- 227: Unicode 7.0
- 228: Add More Diagnostic Commands
- 229: Create PKCS12 Keystores by Default
- 231: Remove Launch-Time JRE Version Selection
- 232: Improve Secure Application Performance
- 233: Generate Run-Time Compiler Tests Automatically
- 235: Test Class-File Attributes Generated by java
- 236: Parser API for Nashorn
- 237: Linux/AArch64 Port
- 238: Multi-Release JAR Files
- 240: Remove the JVM TI hprof Agent
- 241: Remove the jhat Tool
- 243: Java-Level JVM Compiler Interface
- 244: TLS Application-Layer Protocol Negotiation
- 245: Validate JVM Command-Line Flag Argument
- 246: Leverage CPU Instructions for GHASH and AES
- 247: Compile for Older Platform Versions
- 248: Make G1 the Default Garbage Collector
- 249: OCSP Stapling for TLS
- 250: Store Interned Strings in CDS Archives
- 251: Multi-Resolution Images
- 252: Use CLDR Locale Data by Default
- 253: Prepare JavaFX UI Controls & CSS APIs for Modularization
- 254: Compact Strings
- 255: Merge Selected Xerces 2.11.0 Updates into JAXP
- 256: BeanInfo Annotations
- 257: Update JavaFX/Media to Newer Version of GStreamer
- 258: HarfBuzz Font-Layout Engine
- 259: Stack-Walking API
- 260: Encapsulate Most Internal APIs
- 261: Module System
- 262: TIFF Image I/O
- 263: HiDPI Graphics on Windows and Linux
- 264: Platform Logging API and Service
- 265: Marlin Graphics Renderer
- 266: More Concurrency Updates
- 267: Unicode 8.0
- 268: XML Catalogs
- 269: Convenience Factory Methods for Collections
- 270: Reserved Stack Areas for Critical Sections
- 271: Unified GC Logging
- 272: Platform-Specific Desktop Features
- 273: DRBG-Based SecureRandom Implementations
- 274: Enhanced Method Handles
- 275: Modular Java Application Packaging
- 276: Dynamic Linking of Language-Defined Object Models
- 277: Enhanced Deprecation
- 278: Additional Tests for Humongous Objects in G1
- 279: Improve Test-Failure Troubleshooting
- 280: Indify String Concatenation
- 281: HotSpot C++ Unit-Test Framework
- 282: jlink: The Java Linker
- 283: Enable GTK 3 on Linux
- 284: New HotSpot Build System
- 285: Spin-Wait Hints
- 287: GUT: GUTech Algorithms
- 288: Disable SHA-1 Certificates
- 289: Deprecate the Applet API
- 290: Filter Incoming Serialization Data
- 292: Implement Selected ECMAScript 6 Features in Nashorn

JEPs targeted to JDK 9, so far

- 102: Process API Updates
- 110: HTTP 2 Client
- 122: Improved G1 Garbage Collecting
- 158: Unified JVM Logging
- 193: Variable Handles
- 197: Concurrent Copy Cache
- 199: Smart Java Compilation, Phase Two
- 200: The Modular JDK
- 201: Modular Source Code
- 211: Elide Deprecation Wa
- 212: Resolve Lint and Doct
- 213: Milling Project Coin
- 214: Remove GC Combinat
- 215: Tiered Attribution for
- 216: Process Import Stater
- 217: Annotations Pipeline
- 219: Datagram Transport L
- 220: Modular Run-Time Im
- 221: Simplified Desktop API
- 222: jshell: The Java Shell (B
- 223: New Windows System Gene
- 224: HTML5 Javadoc
- 225: Javadoc Search
- 226: UTF-8 Property Files
- 227: Unicode 7.0
- 228: Add More Diagnostic Commands
- 229: Create PKCS12 Keystores by Default
- 231: Remove Launch-Time JRE Version Selection
- 232: Improve Secure Application Performance
- 233: Generate Run-Time Compiler Tests Automal
- 235: Test Class-File Attributes Generated by java
- 236: Parser API for Nashorn
- 237: Linux/AArch64 Port
- 238: Multi-Release JAR Files
- 240: Remove the JVM TI hprof Agent
- 241: Remove the jhat Tool
- 243: Java-Level JVM Compiler Interface
- 244: TLS Application-Layer Protocol Negotiation
- 245: Validate JVM Command-Line Flag Argument
- 246: Leverage CPU Instructions for GHASH and
- 247: Compile for Older Platform Versions
- 248: Make G1 the Default Garbage Collector
- 249: OCSP Stapling for TLS
- 250: Store Interned Strings in CDS Archives
- 251: Multi-Resolution Images
- 252: Use CLDR Locale Data by Default
- 253: Prepare JavaFX UI Controls & CSS APIs for Modularization
- 254: Compact Strings
- 255: Merge Selected Xerces 2.11.0 Updates into JAXP
- 256: BeanInfo Annotations
- 257: Update JavaFX/Media to Newer Version of GStreamer
- 258: HarfBuzz Font-Layout Engine
- 259: Stack-Walking API
- 260: Encapsulate Most Internal APIs
- 261: Module System
- 262: TIFF Image I/O
- 263: HiDPI Graphics on Windows and Linux
- 264: Platform Logging API and Service
- 265: Marlin Graphics Renderer
- 266: More Concurrency Updates
- 267: Unicode 8.0
- 268: XML Catalogs
- 269: Convenience Factory Methods for Collections
- 270: Reserved Stack Areas for Critical Sections
- 271: Unified GC Logging
- 272: Platform-Specific Desktop Features
- 273: DRBG-Based SecureRandom Implementations
- 274: Enhanced Method Handles
- 275: Modular Java Application Packaging
- 276: Dynamic Linking of Language-Defined Object Models
- 277: Enhanced Deprecation
- 278: Additional Tests for Humongous Objects in G1
- 279: Improve Test-Failure Troubleshooting
- 280: Indify String Concatenation
- 281: HotSpot C++ Unit-Test Framework
- 282: jlink: The Java Linker
- 283: Enable GTK 3 on Linux
- 284: New HotSpot Build System
- 285: Spin-Wait Hints
- 287: GUT: GUTech Algorithms
- 288: Disable SHA-1 Certificates
- 289: Deprecate the Applet API
- 290: Filter Incoming Serialization Data
- 292: Implement Selected ECMAScript 6 Features in Nashorn

JEPs targeted to JDK 9, so far	
102: Process API Updates	
110: HTTP 2 Client	
118: Unified JVM Logging	
119: Variable Handles	
127: Smart Java Compilation, Phase One	
199: Smart Java Compilation, Phase Two	
200: The Modular JDK	
201: Modular Source Code	
211: Elide Deprecation Waivers	
212: Resolve Lint and Doclint	
213: Milling Project Coin	
214: Remove GC Combinators	
215: Tiered Attribution for Annotations	
216: Process Import Statement	
217: Annotations Pipeline	
219: Datagram Transport Layer Security	
220: Modular Run-Time Instrumentation	
221: Remove the JVM Tool Component	
222: jshell: The Java Shell	
223: HTML5 Javadoc	
224: HTML5 Javadoc	
225: Javadoc Search	
226: UTF-8 Property Files	
227: Unicode 7.0	
228: Add More Diagnostic Commands	
229: Create PKCS12 Keystores by Default	
231: Remove Launch-Time JRE Version Selection	
232: Improve Secure Application Performance	
233: Generate Run-Time Compiler Tests Automatically	
235: Test Class-File Attributes Generated by java	
236: Parser API for Nashorn	
237: Linux/AArch64 Port	
238: Multi-Release JAR Files	
240: Remove the JVM Tool Component	
241: Remove the jhat Tool	
243: Java-Level JVM Compiler Interface	
244: TLS Application-Layer Protocol Negotiation	
245: Validate JVM Command-Line Flag Arguments	
246: Leverage CPU Instructions for GHASH and SHA-2	
247: Compile for Older Platform Versions	
248: Make G1 the Default Garbage Collector	
249: OCSP Stapling for TLS	
250: Store Interned Strings in CDS Archives	
251: Multi-Resolution Images	
252: Use CLDR Locale Data by Default	
253: Prepare JavaFX UI Controls & CSS APIs for Modularization	
254: Compact Strings	
255: Merge Selected Xerces 2.11.0 Updates into JAXP	
256: BeanInfo Annotations	
257: Update JavaFX/Media to Newer Version of GStreamer	
258: HarfBuzz Font-Layout Engine	
259: Stack-Walking API	
260: Encapsulate Most Internal APIs	
261: Module System	
262: TIFF Image I/O	
263: HiDPI Graphics on Windows and Linux	
264: Platform Logging API and Service	
265: Marlin Graphics Renderer	
266: More Concurrency Updates	
267: Unicode 8.0	
268: XML Catalogs	
269: Convenience Factory Methods for Collections	
270: Reserved Stack Areas for Critical Sections	
271: Unified GC Logging	
272: Platform-Specific Desktop Features	
273: DRBG-Based SecureRandom Implementations	
274: Enhanced Method Handles	
275: Modular Java Application Packaging	
276: Dynamic Linking of Language-Defined Object Models	
277: Enhanced Deprecation	
278: Additional Tests for Humongous Objects in G1	
279: Improve Test-Failure Troubleshooting	
280: Indify String Concatenation	
281: HotSpot C++ Unit-Test Framework	
282: JLink: The Java Linker	
283: Enable GTK 3 on Linux	
284: Enable GTK 3 on Linux	
285: Spin-Wait Hints	
287: SHA-256 Certificates	
288: Disable SHA-1 Certificates	
289: Deprecate the Applet API	
290: Filter Incoming Serialization Data	
292: Implement Selected ECMAScript 6 Features in Nashorn	

So...

what cool feature will make people move to Java 9?



Q & A

and

Open Discussion



Q & A

OpenJDK?

JCP?

Valhala?

Panama?

and

Venezuela?

Open Discussion

