

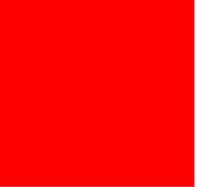
**ORACLE®**



**ORACLE<sup>®</sup>**

**Java 7 and 8**

**Tomas Nilsson, Principal Product Manager Java SE Platform group**



The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

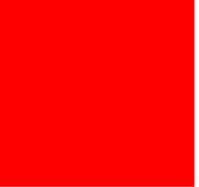
# Agenda

- JDK 7
- JDK 8
- Oracle's Java Strategy
- JVM Convergence



# JDK 7 & 8 Features and Roadmap





Disclaimer: The JDK roadmap contents reflect Oracle's current assumption on what Java SE 7 and 8 will contain. It will be updated to reflect content changes decided by the JCP Expert Groups.

# The best laid plans...

- ~ 2008 – *Java 7*

# Plan B

- *Java 7 = Java 7 + Java 8*

# JDK 7 Highlights - Evolution

- Coin – Small language enhancements (JSR 334)
- Dynamic Language Support (JSR 292)
- Concurrency and Collections updates (JSR 166y)
- Network and File System (JSR 203)
- Security
- Internationalization
- Other Miscellaneous Enhancements
- *JVM Convergence*

# Coin - Small Language Enhancements (JSR 334)

- Small Language changes for Java
- Simplifies every-day coding for most developers
- Six new language features:
  - Strings in Switch
  - Better Integral Literals
  - Improved Exception handling
  - Improved Type Inference for Generic Instance Creation (Diamond Operator)
  - Automatic Resource Management (a.k.a try-with-resources)
  - Simplified Varargs Method Invocation
- Will as a proof of concept be applied to a subset of the core libraries

# Coin (1) Strings in switch

```
String s = ...
switch(s) {
case "foo":
    processFoo(s);
    break;
}
```

# Coin (2) Changes to literals

- `1234_5678`
- `1_2_3_4_5_6_7_8L`
- `0b0001_0010_0100_1000`
- `3.141_592_653_589_793d`
- `0x1.ffff_ffff_ffff_fP1_023 // Double.MAX_VALUE`

# Coin (3) Multi Catch, precise rethrow

```
try {
    doWork(file);
} catch (IOException ex) {
    logger.log(ex);
}
catch (SQLException ex) {
    logger.log(ex);
}
```

```
try {
    doWork(file);
} catch (IOException|SQLException ex) {
    logger.log(ex);
}
```

# Coin (4) Diamond operator

```
Map<String,List<String>> m = new HashMap<String,List<String>>();
```

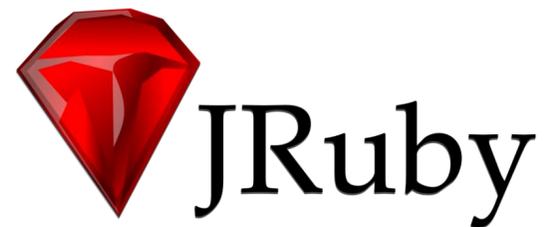
```
Map<String,List<String>> m = new HashMap<>();
```

# Coin (5) Try with resources

```
try (AutoCloseableResource) {  
    // Do stuff  
}  
  
// closed here
```

# Dynamic Language Support (JSR 292)

- Invokedyynamic
  - Email [tomas.nilsson@oracle.com](mailto:tomas.nilsson@oracle.com)

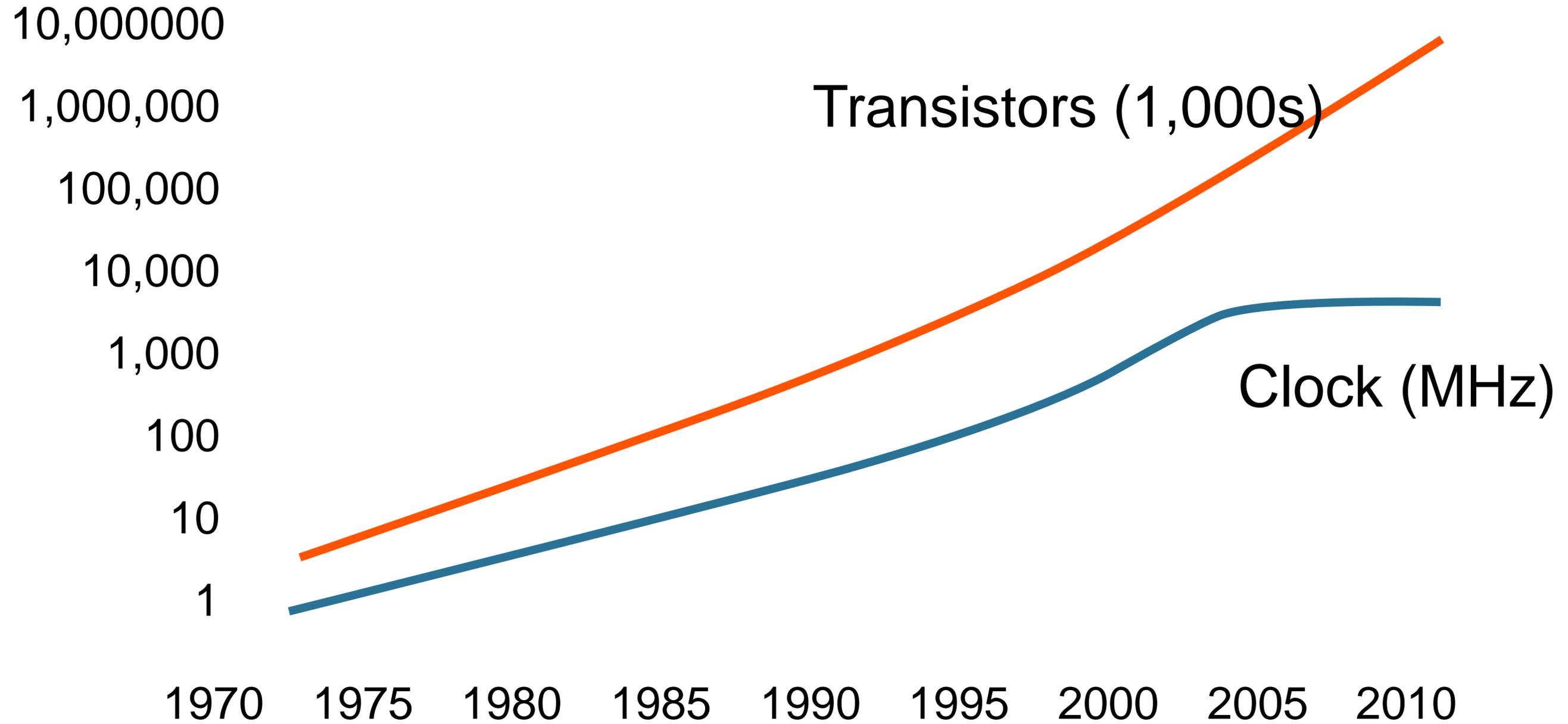


- Lambda (JDK 8)

(\*) Source: <http://www.cs.iastate.edu/~design/vmil/2010/papers/p06-kaewkasi.pdf>

# Concurrency and Collections Updates (JSR 166y)

## Lightweight Fork/Join Framework



# Network and File System

## NIO.2 (JSR 203)

- NIO.2
- Zip/Jar
- SCTP (Solaris)
- SDP (Infiniband)
- Vista IPv6 stack
  
- (new) Enhanced JMX agent and MBeans

# Security

- Native ECC
- TLS (Transport Layer Security) 1.2

# Internationalization

- Unicode 6.0
- IETF BCP47 and UTR35
  - One language many character codes (Kanji, Hiragana, Katakana, Romaji)
  - Three-letter base language codes; three-digit region codes
- Separate user locale and user interface locale
- Currency data enhancements
  - Switch from hard coded data to a model similar to the one used for time zone data.
  - Easier to keep currency data up to date

# Other Enhancements

## Client & Graphics

- Added Nimbus L&F to the standard
  - Much better –modern- look than what was previously available
- Platform APIs for Java 6u10 Graphics Features
  - Shaped and translucent windows
- JXLayer core included in the standard
  - Formerly SwingLabs component
  - Allows easier layering inside of a component
- Optimized Java2D Rendering Pipeline for X-Windows
  - Allows hardware accelerated remote X

# Other Enhancements

- Better font configuration on Unix
  - Now uses standard Unix mechanism to find fonts.
- JAXP 1.4.4, JAXWS 2.2 and JAXB 2.2
- JDBC 4.1, Rowset 1.1
- Upgrade class-loader architecture
  - Modifications to the ClassLoader API and implementation to avoid deadlocks in non-hierarchical class-loader topologies
- Close URLClassloaders
  - Allows applications to proactively clean up classloaders, freeing up native resources and unlocking JAR files
- Javadoc support for stylesheets

# JDK 7 Platform Support

- Windows x86
  - Server 2008, Server 2008 R2, 7 & 8 (when it GAs)
  - Windows Vista, XP
- Linux x86
  - Oracle Linux 5.5+, 6.x
  - Red Hat Enterprise Linux 5.5+, 6.x
  - SuSE Linux Enterprise Server 10.x, 11.x
  - Ubuntu Linux 10.04 LTS, 11.04
- Solaris x86/SPARC
  - Solaris 10.9+, 11.x
- Apple OSX x86
  - will be supported post-GA, detailed plan TBD

*Note: JDK 7 should run on pretty much any Windows/Linux/Solaris. These configurations are the ones primarily tested by Oracle, and for which we provide commercial support.*

# JDK 7 Project Plan

- From 1/21 OpenJDK Schedule
  - 2010/12/16 Feature Complete
  - 2011/04/12 Rampdown start: P1-P3 bugs only
  - 2011/04/28 API/interface changes: Showstoppers only
  - 2011/05/11 All targeted bugs addressed  
First release candidate built
  - 2011/05/18 Bug fixes: Showstoppers only
  - 2011/06/08 Final test cycle starts
  - 2011/07/28 General Availability

# JDK 8 – Fall/Winter 2012

## Features from “Plan B”

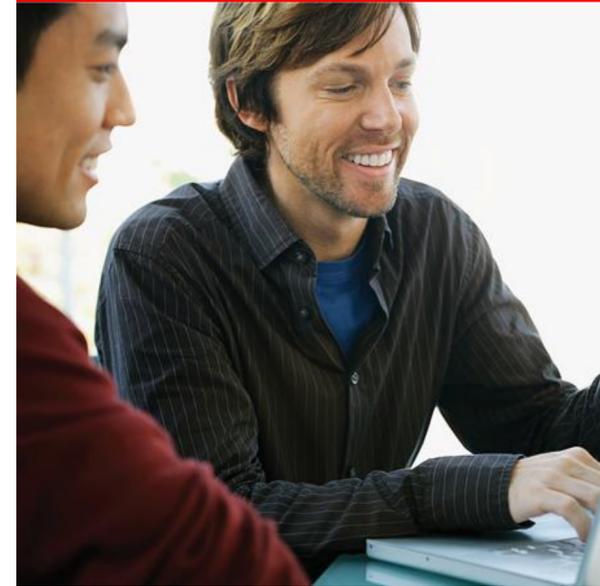
- Modularization
  - Language and VM Support
  - Platform Modularization
- Project Lambda
  - Lambda Expressions
  - Default Methods
  - Bulk Data Operations
- Annotations on Java types (JSR 308)
- More Small Language Enhancements
  - Project Coin part 2

## Other Things On Oracle’s Wish List\*

- Serialization fixes
- Multicast improvements
- Java APIs for accessing location, compass and other “environmental” data (partially exists in ME)
- Improved language interop
- Faster startup/warmup
- Dependency injection (JSR 330)
- Include select enhancements from Google Guava
- Small Swing enhancements
- More security/crypto features, improved support for x.509-style certificates etc
- Internationalization: non-Gregorian calendars, more configurable sorting
- Date and Time (JSR 310)
- Process control API

\* Many of these will undoubtedly NOT make JDK 8.

# Strategy and JVM Convergence



# Oracle's Java Strategy

## In Order of Priority

### 1. A Vibrant Java Ecosystem

- Oracle depends on Java for its software business. Keeping Java vibrant and competitive ensures business continuity.
- In other words: Our goals are aligned with the community – make Java successful!

### 2. Generate Revenue

- Primarily indirectly by making Oracle software and hardware more competitive, but also directly through support revenue etc.

### 3. Lower Costs

- Improve language, libraries and tools in such a way that it makes Oracle's thousands of Java developers more effective.
- This will of course also help goal #1 in most cases

# JVM Convergence

## Project “HotRokit”

🎯 = Premium Feature

### JDK 7 GA

- Java SE 7 Support
- Rebranding
- No PermGen (stretch goal)
- Improved JMX Agent
- Command line servicability tool (jrcmd)
- JRockit Mission Control Console support 🎯

### JDK 7 Update X

- Performance
- Initial JRockit Flight Recorder Support 🎯

### JDK 7 Update Y

- More performance
- Improved command line servicability (jrcmd)
- G1 complete as CMS replacement
- Non-contiguous heaps
- Complete JRockit Flight Recorder Support 🎯
- JRockit Virtual Edition Support 🎯
- Soft Real Time GC 🎯

### JDK 8 GA

- Java SE 8 Support
- All performance features from JRockit ported
  - I/O Performance
  - JIT Optimizations
- All servicability features from JRockit ported
  - Compiler controls
  - Verbose logging
- JRockit Mission Control Memleak Tool Support 🎯

# **Hardware and Software Engineered to Work Together**