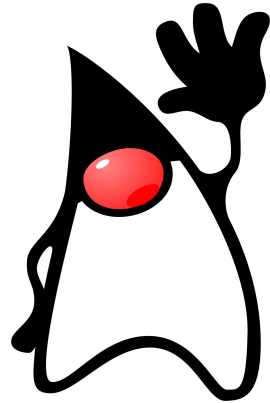


Introduction to Java 8: JVM, Language, and Platform (UGF11077)



Timothy Fagan, NYCJava JUG, Lab49
Dario Laverde, NYCJava JUG, HTC

This session

In this session, you'll learn about the past, present, and future of the Java platform. You'll see plenty of running code samples of Java 8 features, including lambda expressions, streams, JavaFX, CompletableFuture, and the new Date API.

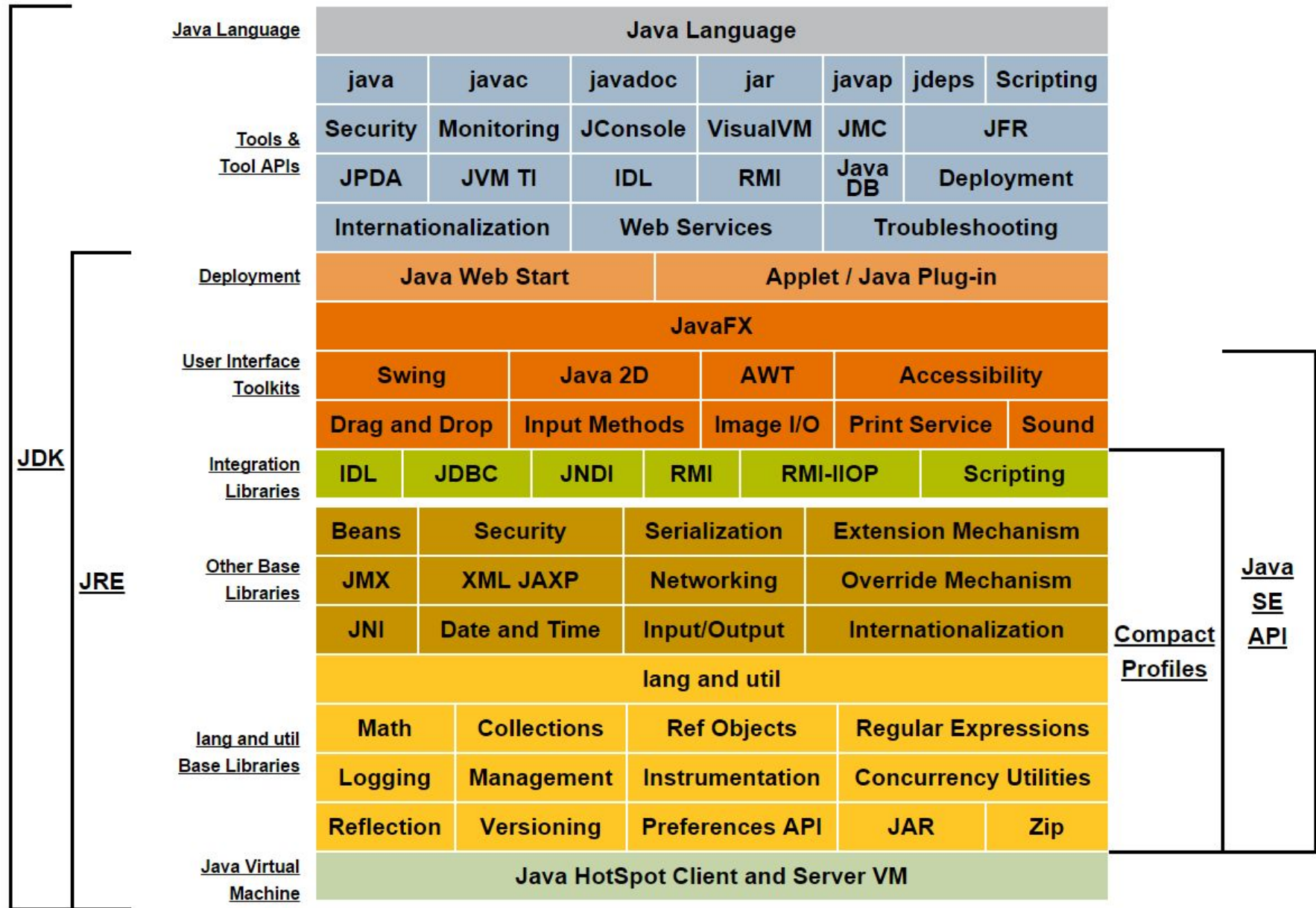
Agenda

- Recap
- The Java 8 Platform
- The Java 8 Virtual Machine
- The Java 8 Language and API
- The Java 8 Demo
- The Future of Java
- Q&A

Recap

- Developed by Sun Microsystems
- Familiar syntax, inspired by C and C++
- Provided garbage collection, like Smalltalk and LISP
- Java 1.0 was released in 1995
- Available on multiple platforms at no cost
- The first JavaOne conference was held 1996

The Java Platform



The Java Virtual Machine

- Programs are compiled into *byte codes* and stored in class files
- Byte code is interpreted by JVM or compiled just-in-time (JIT) into native code
- Responsible for garbage collection (GC)
- Provides a secure sandbox that restricts what programs can do
- Removal of PermGen in Java 8 JVM

The Java 8 Language Enhancements

- Lambda expressions
 - Treat functionality as a method argument, or code as data
 - Single-method interfaces (functional interfaces)
- Method references
 - Easy-to-read lambda expressions for methods that already have a name
- Default methods
 - Enable new functionality to be added to the interfaces of libraries and ensure binary compatibility with code written for older versions of those interfaces

The Java 8 Language Enhancements

- Repeating Annotations
 - Provide the ability to apply the same annotation type more than once to the same declaration or type use
- Type Annotations
 - Apply an annotation anywhere a type is used, not just on a declaration
 - Used with a pluggable type system, this feature enables improved type checking of your code
- Improved type inference
- Method parameter reflection

Lambda Expressions

Before Java 8:

```
Thread t = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        System.out.println("wow");  
    }  
});
```

Java 8:

```
new Thread(() -> System.out.println("wow"));
```

The Java 8 API

- Stream
- CompletableFuture
- Parallel processing: Arrays and Streams
- Java Date
- Map enhancements
- Optional
- Base64
- Logger enhancements
- JavaFX - 3D and WebView

The Java 8 Language Demo

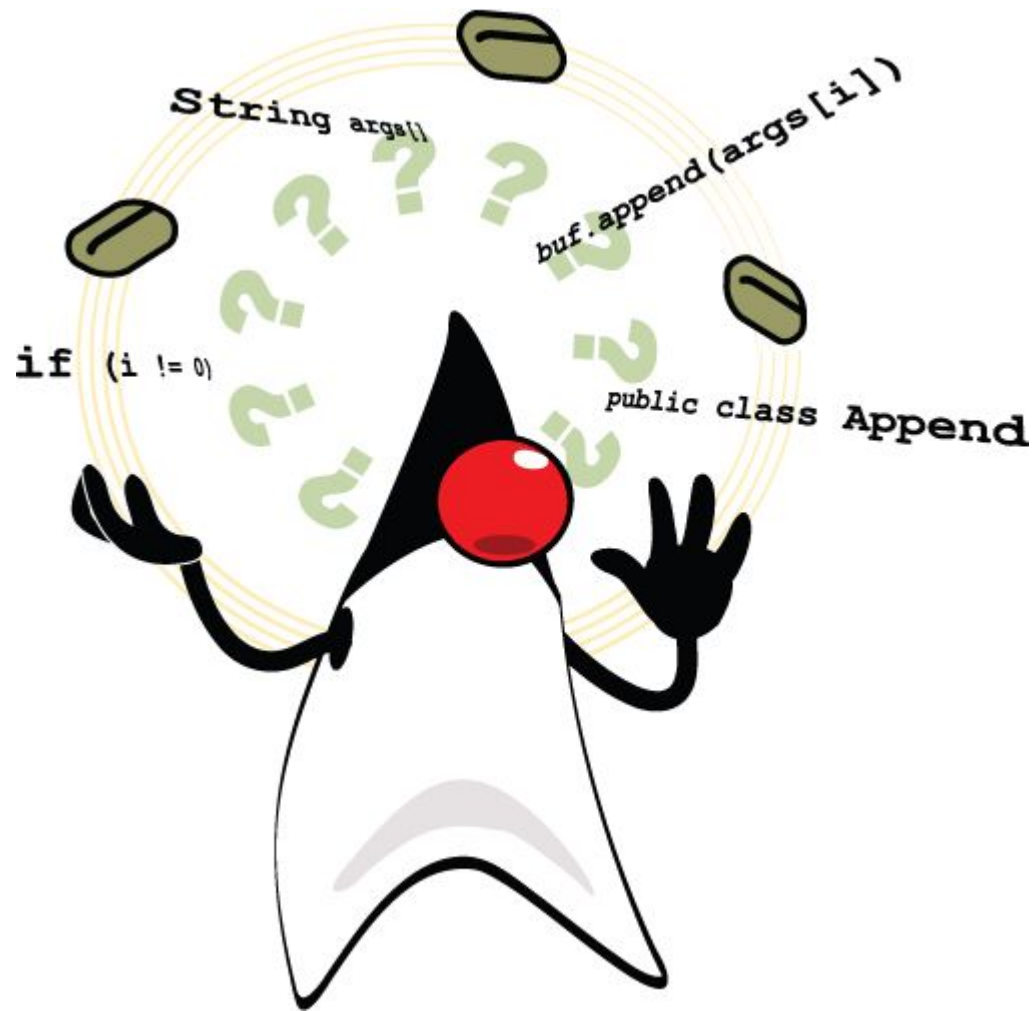
Planned features for Java 9

- jshell
- A HTTP 2 Client (HTTP/2 and websockets)
- Process API Improvements (native process management)
- Smart Java Compilation, Part 2 (parallel compilation)
- Modular Source Code (not project Jigsaw)
- Make G1 the default GC

Candidate features for Java 10+

- Better support for multi-gigabyte heaps
- Better native code integration
- Self-tuning JVM
- Project Jigsaw
- Money and Currency API
- Project Sumatra

Questions and Answers



References and Resources

- What's new in JDK 8

<http://www.oracle.com/technetwork/java/javase/8-whats-new-2157071.html>

- Java tutorials

<http://docs.oracle.com/javase/tutorial>

- Planned Java 9 Features

<http://openjdk.java.net/projects/jdk9>

- Demo source code

<https://github.com/NYCJava/JavaOne/tree/master/JavaOne2015>

- Contact us

dario@nycjava.net

timothy@nycjava.net