Overview, Changes, Outlook, Suggestions

Java 8 to Java 17

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

BIS Seminar Paper – Lukas Artwohl

Contents

Introduction to Java

- Created in 1995
- Key Features
  - Object-oriented
  - Platform-independent
  - Simplicity
- Key Use Cases
  - Applications (web & mobile)
  - Big Data
  - Embedded Systems
- Current version: Java 18

Java 8

- First Long-time support (LTS) version
- Lambda expressions
- Type inference
- Date Time API
- Already very expansive language

Changes up to Java 17

- Java Modular System (Java 9)
- Var data type (Java 10)
- Start of licensing model (Java 11)
- Removal of JavaFX, Java Mission Control from the JDK
- Improved error messages (NullPointer, Java 17)
- General performance and security improvements

Key Differences Between Java 8 & Java 17

- Introduction of licensing model
- Performance improvements (significant!)
- Security enhancements
- Java 8 could do enough for most programmers
- New features needed for Bitcoin, 4K video

Outlook up to Java 21

- Java 18
  - UTF-8 as default
  - Vector API introduced
  - Improved garbage collectors (ZGC, SerialGC, ParallelGC)
- Outlook beyond Java 18
  - Use code outside Java runtime
- Suggestions
  - Improve to compete with Kotlin
  - Establish Java in blockchain

Conclusio

- Prevalent in Programming
  - Big data, applications, embedded systems
  - Object-oriented, platform independent, high performance (i.e. vs. Python)
- Java 8
  - No licensing
  - Lambda expressions, type inference
- Java 17
  - Licensing (since Java 11)
  - Much better performance, security
  - Project Jigsaw, JMS
- Java in the future
  - Compete with Kotlin
  - Become prevalent with blockchains