

SE 350 - Object-Oriented Software Development

Environment Setup

Instructor: Stefan Mitsch



Learning Objectives

- Understand the difference between Java editions
- Understand the difference between JVM, JRE, JDK
- Set up a Java programming environment

Why Java?

Ecosystem

- Java Runtime Environment: execute Java programs
- Java Development Kit: develop Java programs (compile, debug, profile, ...)
- Integrated Development Environment: develop Java programs more comfortably

History

- 1990: start (Sun Microsystems)
- Goal: better C++
- 2010: Oracle purchases Sun
- 2023: Java SE 21

Features

- Object-oriented programming
- High-level language, extensive library
- Platform-independent
- Stability, safety, massive community



Java Editions

- Java SE (standard edition, used in this course): for developing desktop applications, libraries for networking, security, database access, graphical user interface (GUI) development, and XML parsing
- Jakarta EE (enterprise edition): for large-scale, multi-tiered, scalable, reliable, and secure distributed computing and web services
- Java ME (micro edition): small-footprint virtual machine for small devices (mobile phones, printers etc.)
- Java Card: applets on smart cards



Programming Environment Terminology

- Java Code (`.java`)
- Java Virtual Machine (JVM): interprets compiled Java programs (`.class`)
- Java Runtime Environment (JRE): JVM + library packages
- Java Development Kit (JDK): JRE + development tools (compiler etc.)
- Bytecode: JVM machine language
- Platform: where the JVM runs
- JVM, JRE, and JDK are platform-dependent, but Java programs are independent
- Integrated Development Environment: more convenient development with syntax highlighting, compiling, debugging
- Oracle JDK vs. Open JDK



JVM-based Languages

- BeanShell: lightweight scripting language for Java
- Ceylon: object-oriented, strongly statically typed language, immutability
- Clojure: functional dialect of Lisp
- Groovy: fully Java interoperable, syntax-compatible, static and dynamic language with features from Python, Ruby, Perl, and Smalltalk
- JRuby: Ruby interpreter
- Jython: Python interpreter
- Kotlin: industrial programming language for JVM with full Java interoperability
- Rhino: JavaScript interpreter
- Scala: multi-paradigm language (object-oriented + functional)



Integrated Development Environments

Desktop applications

- IntelliJ IDEA
- Visual Studio Code
- Eclipse
- Netbeans

Online tools

- [repl.it](#)
- [compilejava](#)
- [codiva](#)
- [jdoodle](#)
- [browxy](#)
- [onlinegdb](#)
- [IDEOne](#)



Summary

- Java is a platform-independent, statically-typed, object-oriented language with a large library and community
- JVM executes programs, JRE provides necessary libraries for running, JDK provides tools for developing programs
- Select the appropriate Java edition for a task
- Many languages build on top of the JVM



Exercises

- Download and install Java SE 21 LTS
- Download and install VS Code
 - If you do not already have VS Code: download **Coding Pack for Java**
 - If you already have VS Code: install plugins **Extension Pack for Java**
- Implement and run `hello, world` in Java
- Test-driven development
 - Implement a function that computes the Fibonacci numbers
 - Write and run unit tests
- Detailed instructions are in [Worksheet 1: Environment Setup](#)