Java, IntelliJ IDEA and Maven

https://github.com/heig-vd-dai-course

Web · PDF

L. Delafontaine and H. Louis, with the help of GitHub Copilot.

Objectives

- Learn why Java is a popular programming language
- Manage multiple Java versions with SDKMAN!
- Develop Java apps with IntelliJ IDEA and Maven
- Manage dependencies with Maven
- Develop essential skills for professional Java development

Java

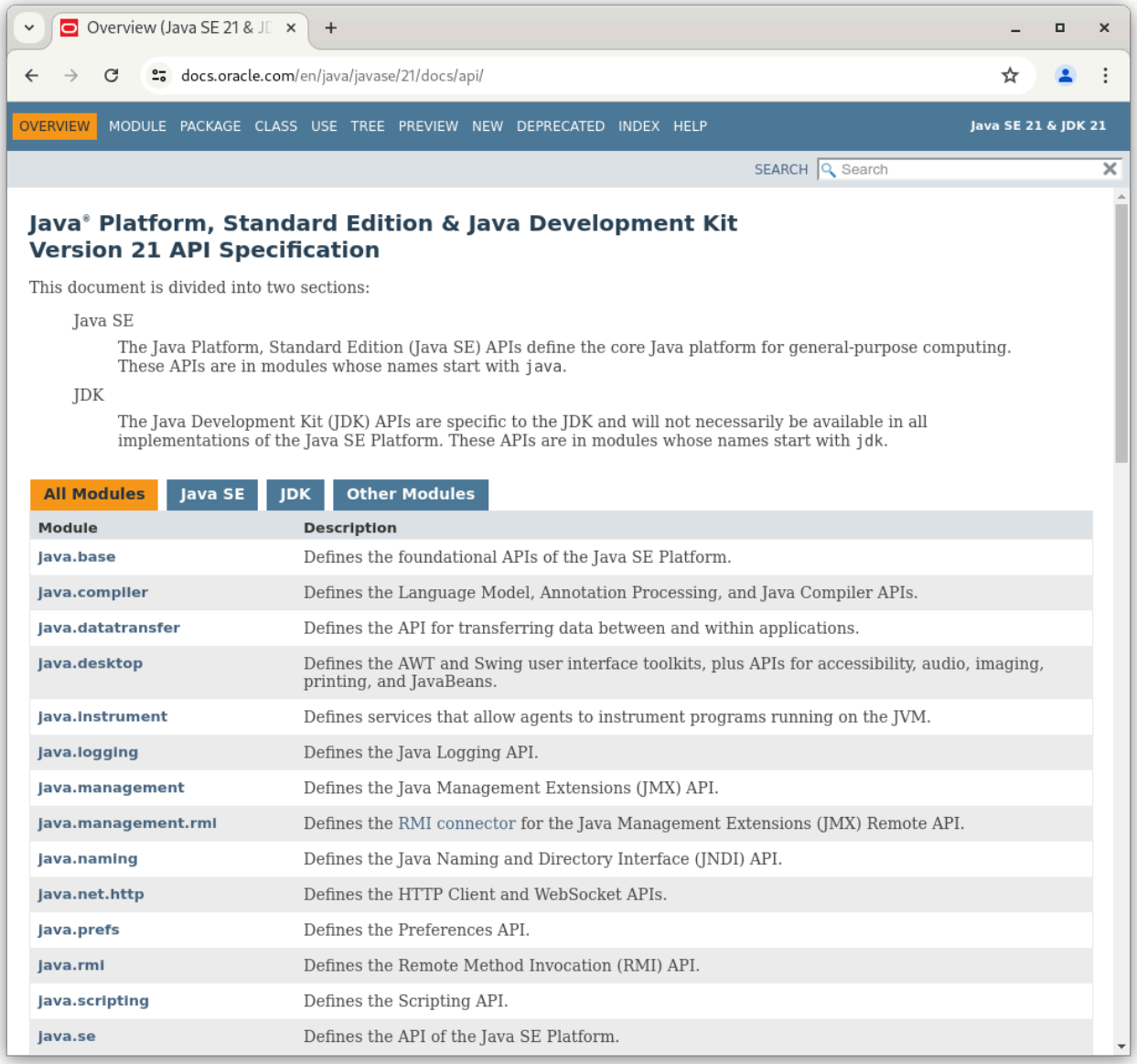
More details for this section in the course material. You can find other resources and alternatives as well.

Java

- General-purpose, object-oriented language
- Write once, run anywhere (WORA)
- Created by James Gosling, 1995 at Sun Microsystems
- The documentation seems scary but you will get used to it and it is very useful

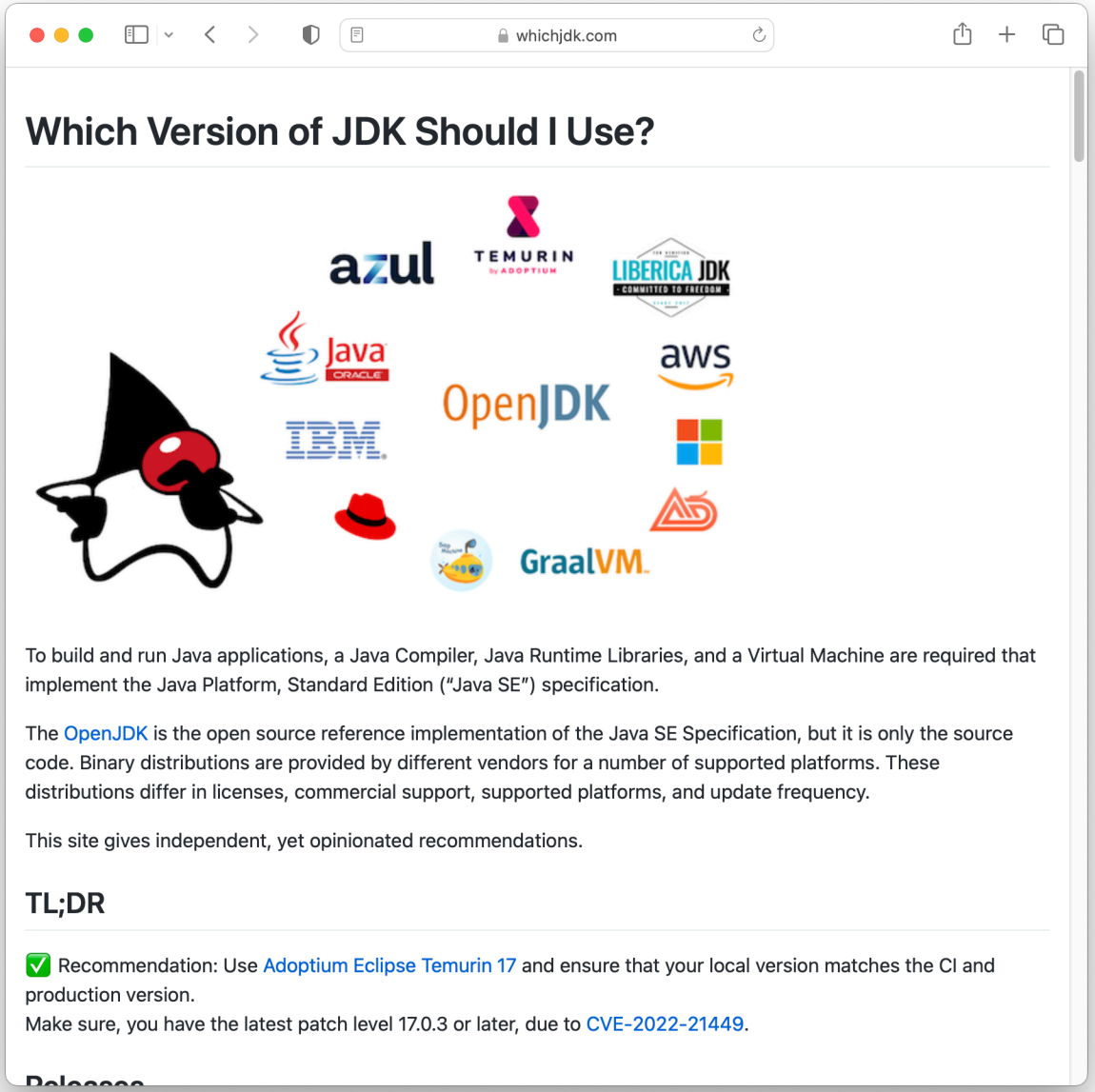
Java virtual machine

- Compiles source code to bytecode
- Executes in Java virtual machine (JVM)
- Where a JVM exists, Java can run (most of the time)



JVM versions

- Multiple implementations exist
- Can target different platforms and/or specific features
- JDK for development, JRE for running
- Eclipse Temurin is recommended



Java versions and version managers

- Java 21 is the latest LTS
- You can use SDKMAN! to manage multipe versions of Java
- Match versions for project consistency

Compiling and running Java programs

- Compile manually with `javac` command

```
javac HelloWorld.java
```

- Execute with `java` command

```
java HelloWorld
```

- Modern way: package into JAR files with the help of Maven

```
java -Xmx1024M -Xms1024M -jar minecraft_server.1.20.1.jar nogui
```

Summary

- Java is a general-purpose, class-based, object-oriented programming language.
- Java is compiled to bytecode, which is then executed by a Java virtual machine (JVM).
- Java is intended to be portable, thanks to the JVM.
- Java has various versions, each with its own set of features and improvements.
- Versions managers allow you to install and switch between different versions of Java.

IntelliJ IDEA

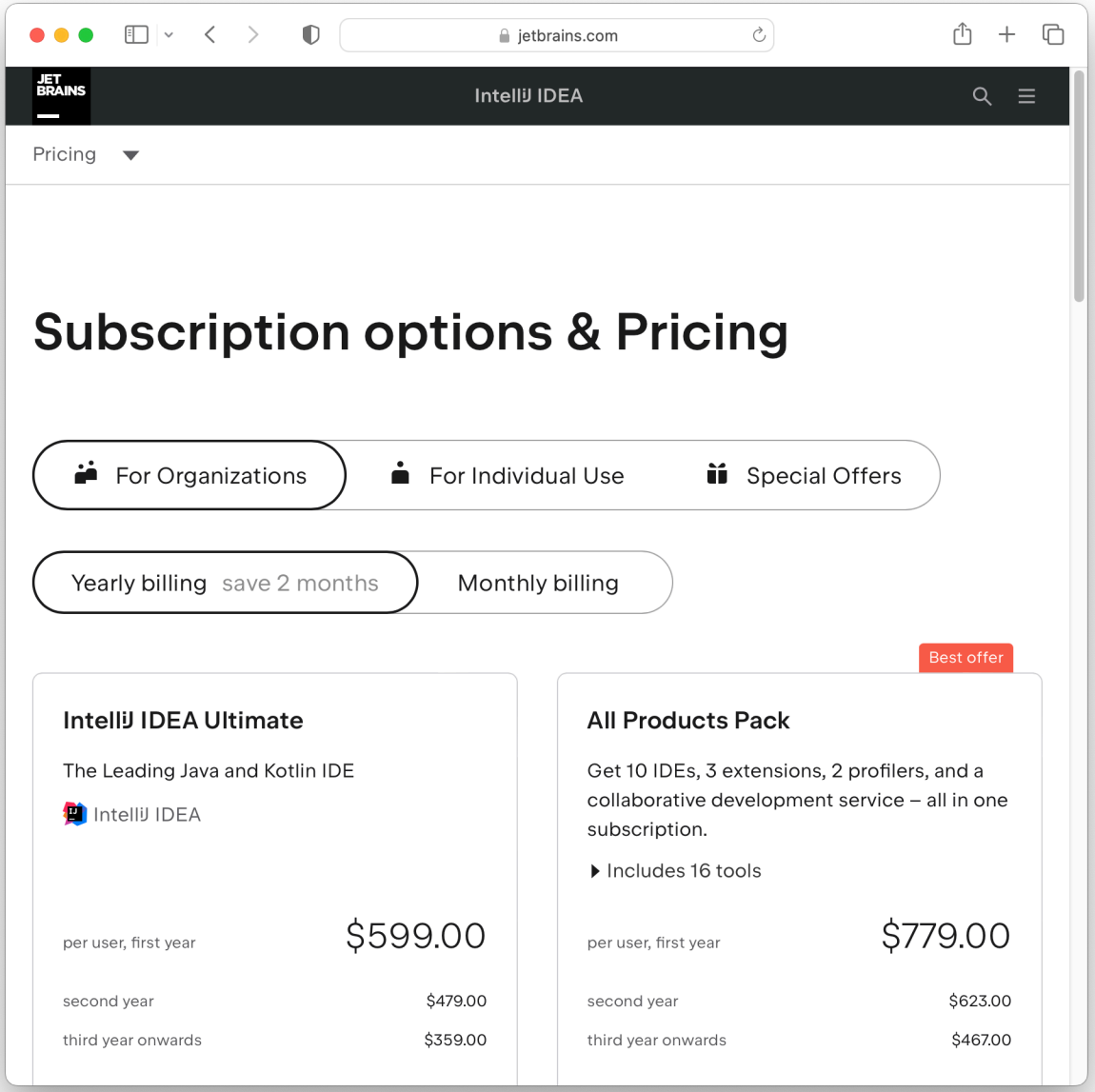
More details for this section in the course material. You can find other resources and alternatives as well.

IntelliJ IDEA

- IDE for (Java) software development
- Developed by JetBrains
- Works on Windows, macOS, Linux
- Quite a standard in the industry

Community Edition and Ultimate Edition

- Community (free) and Ultimate (paid)
- Free student license available - you can get it!
- The Community Edition is sufficient for this course

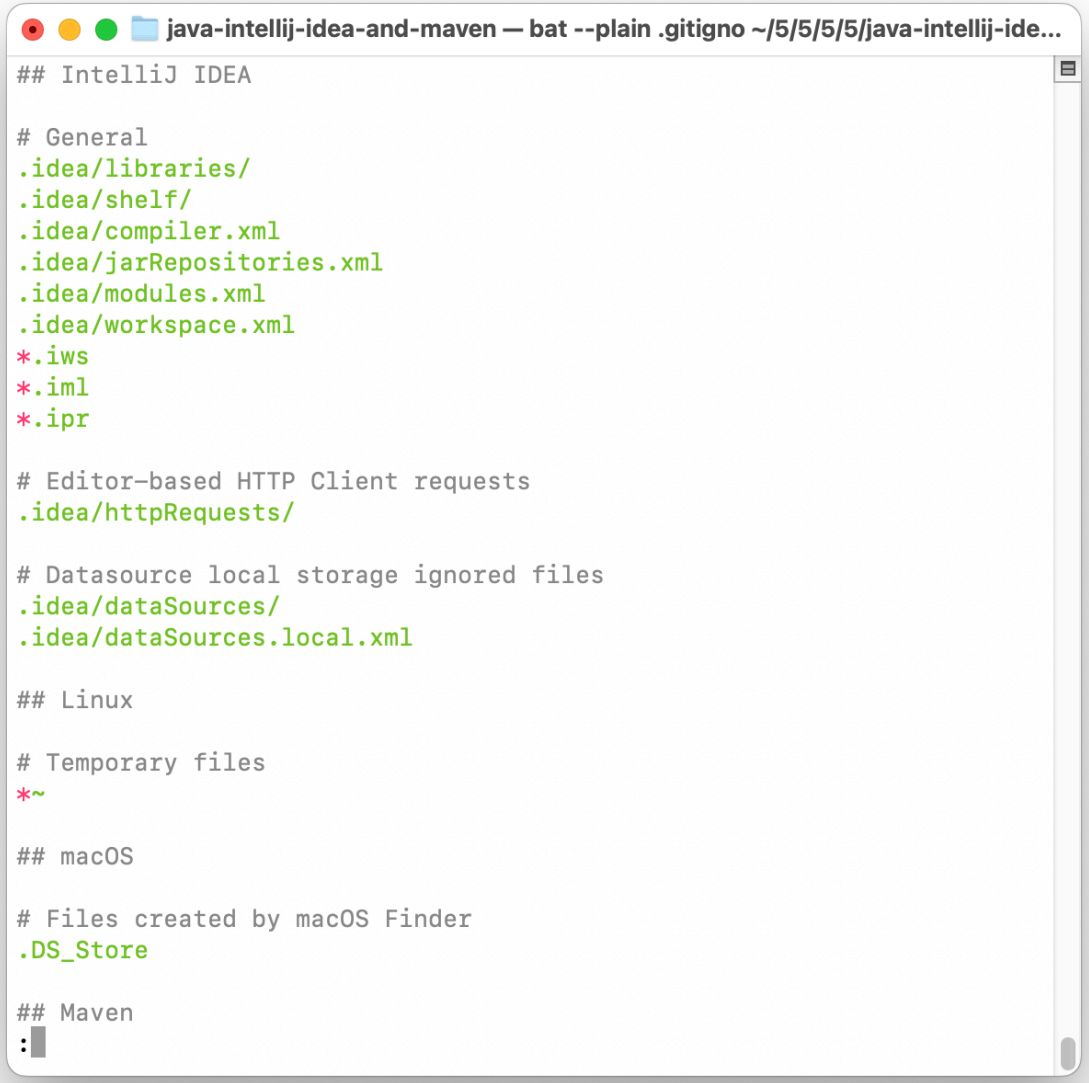


JetBrains Toolbox App

- Manage multiple JetBrains IDEs
- Install and update in one place
- Optional but very useful

Configuration files and Git

- `.idea` directory for project config
- Allow to share project settings between developers
- Some files must be ignored to avoid issue (local config) in Git



Summary

- IntelliJ IDEA is an integrated development environment (IDE) written in Java for developing computer software.
- IntelliJ IDEA is available in two editions: the Community Edition (free and open-source) and the Ultimate Edition (proprietary).
- You are eligible for a free student license for the Ultimate Edition.
- When creating a new project, IntelliJ IDEA will create a `.idea` directory containing the project configuration files.
- Some of these files must be ignored by Git, as they contain local configuration that is specific to your computer.

Maven

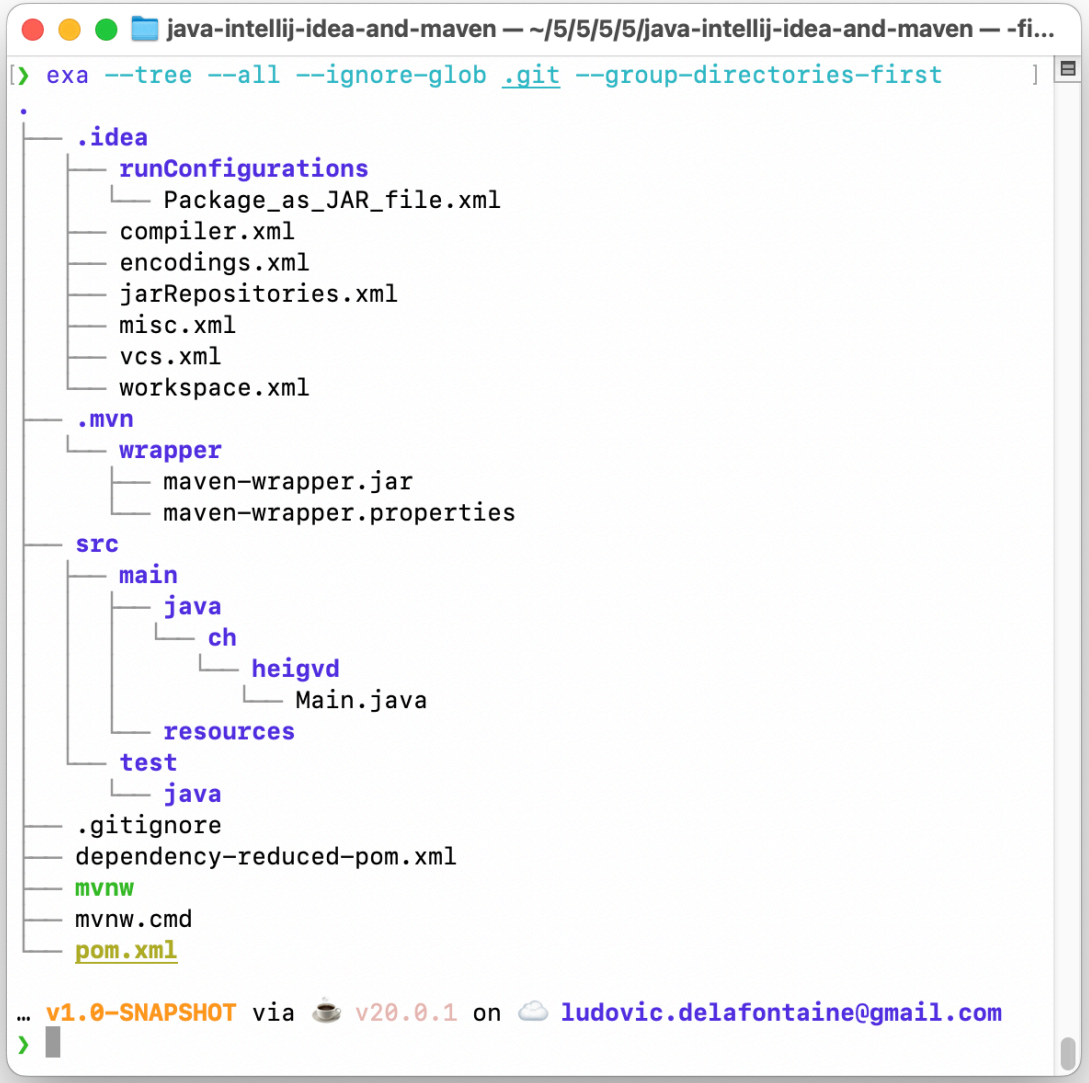
More details for this section in the course material. You can find other resources and alternatives as well.

Maven

- Maintained by Apache Software Foundation
- Software project management tool
- Manages dependencies
- Build automation tool
- Allows to define a standard project structure

Maven project structure

- Standardized directory structure
  - `src/main/java`
  - `src/main/resources`
  - `src/test/java`
- Simplifies build process with conventions

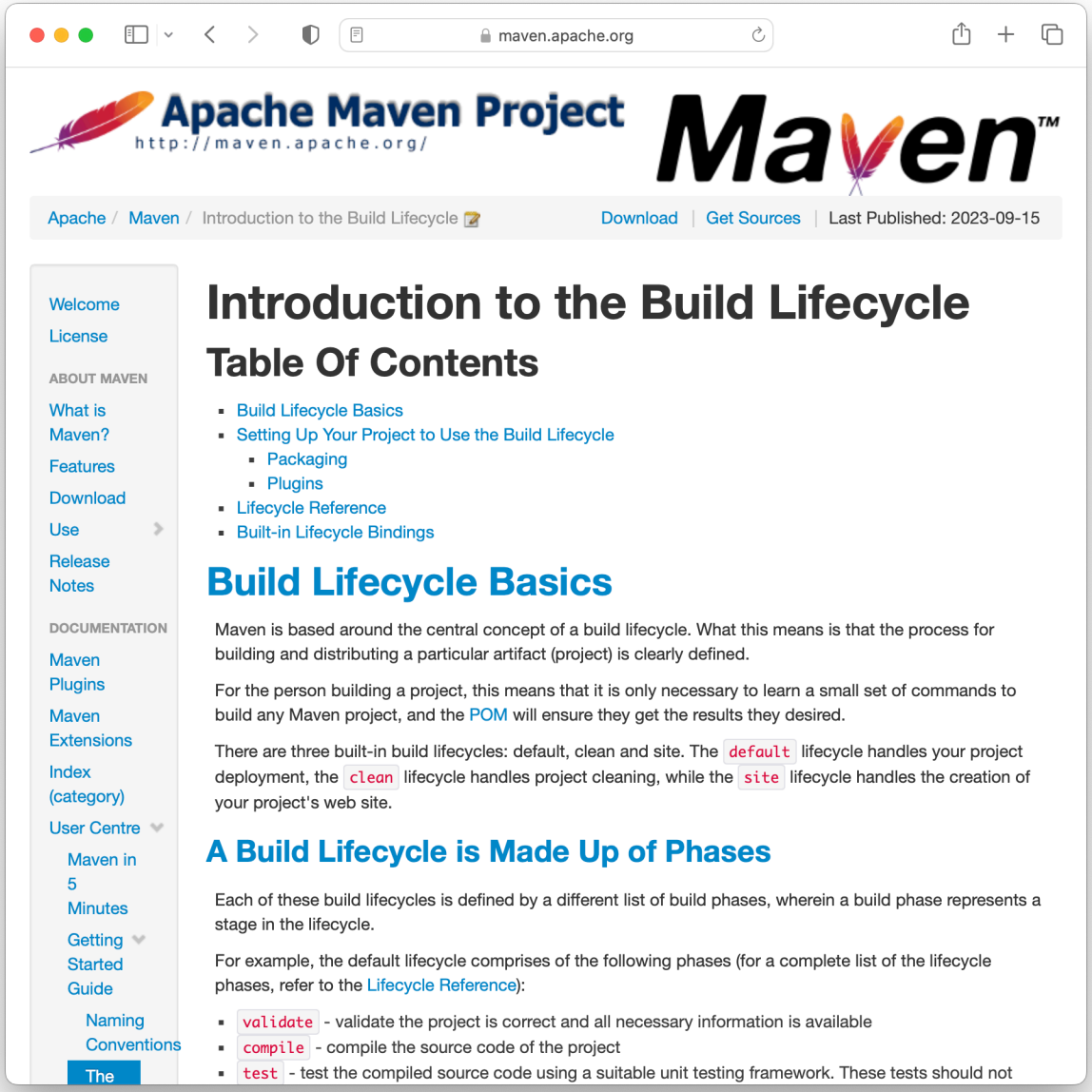


`pom.xml` file

- Configuration and build settings
- Shared among developers
- Defines dependencies and plugins:
  - Plugins extend Maven functionality
  - Dependencies are external libraries

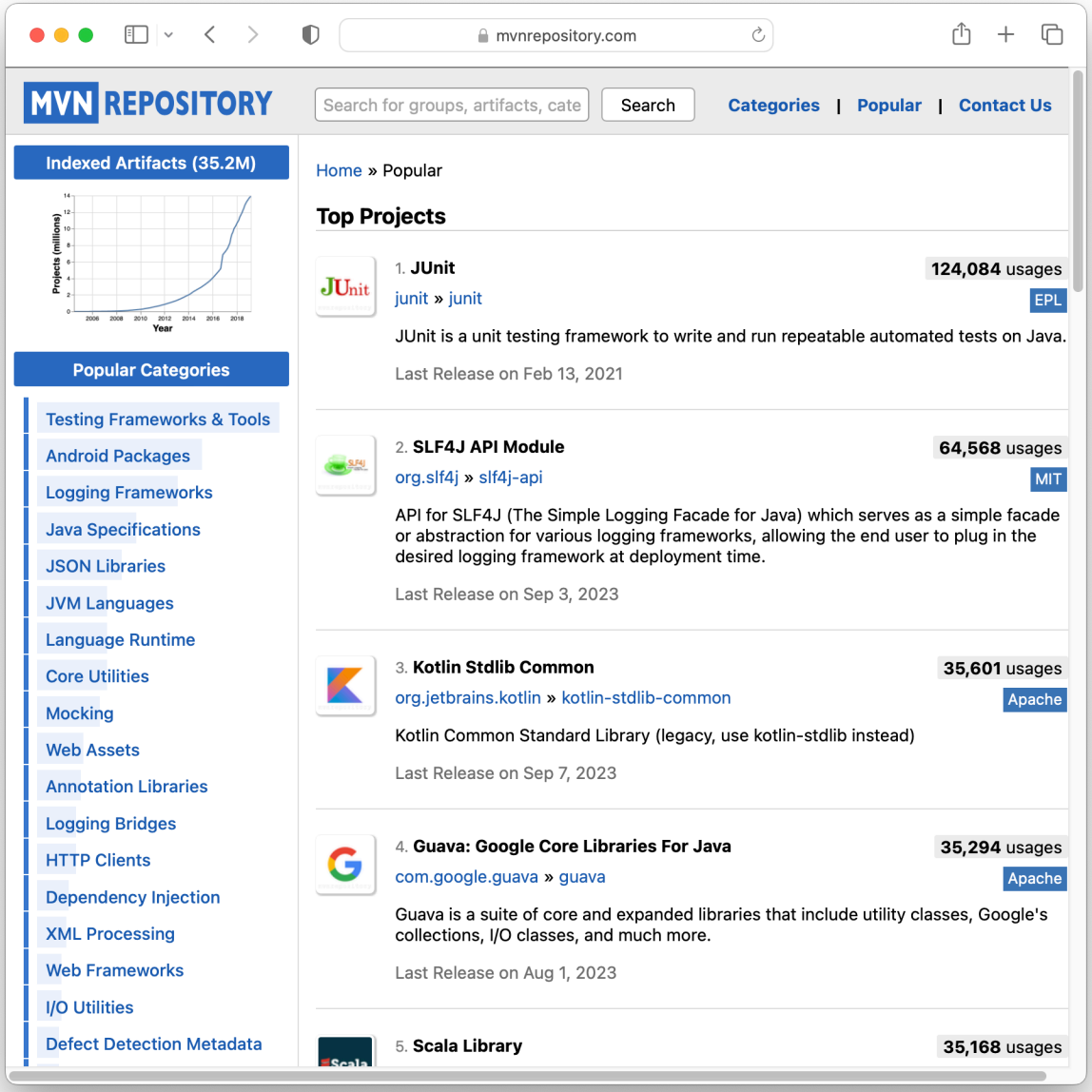
Maven lifecycle

- Maven defines build process
- Composed of phases and goals
- Phases load plugin goals
- Goals execute tasks on your project (e.g., compile, test, package)



Maven Repository

- Public repository of Java libraries
- Maven can download dependencies automatically
- You can publish your own libraries
- Many libraries available such as picocli, a library for building CLI applications



Maven wrapper

- Allows to use Maven without installing it
- Wrapper script downloads Maven
- Ensures consistent Maven version
- Use `mvnw` or `mvnw.cmd` instead of `mvn`

Summary

- Maven is a software project management and comprehension tool.
- Maven is a dependency manager for Java projects.
- Maven is a build automation tool for Java projects.
- Maven defines a standard directory structure for Java projects.
- Maven defines a standard build process for Java projects.
- The `pom.xml` file contains the configuration of your Maven project.

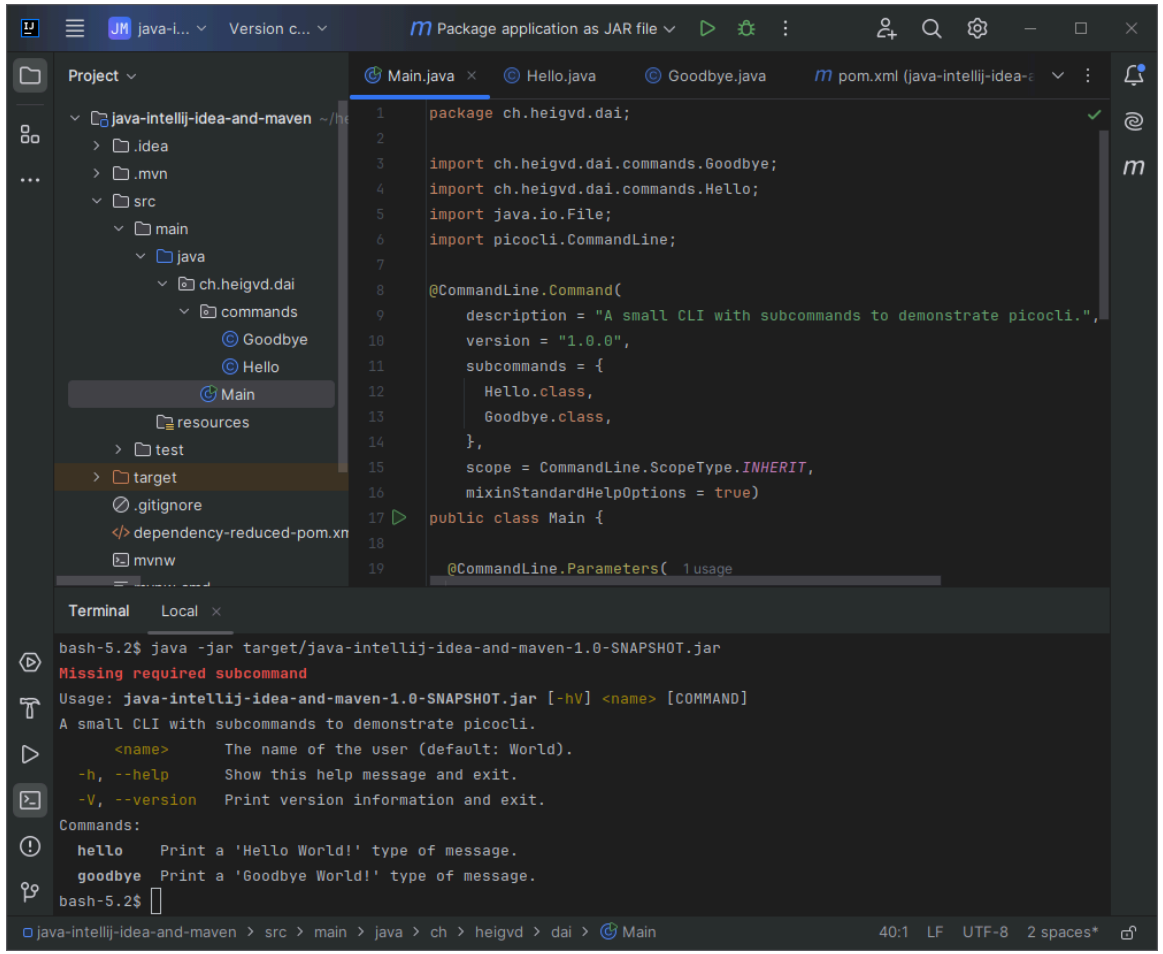
Questions

Do you have any questions?

Practical content

What will you do?

- Install and configure SDKMAN!, Java, Maven and IntelliJ IDEA
- Create and run a small CLI application with picocli, an external Maven dependency
- Publish your project on GitHub



Find the practical content

You can find the practical content for this chapter on GitHub.

Finished? Was it easy? Was it hard?

Can you let us know what was easy and what was difficult for you during this chapter?

This will help us to improve the course and adapt the content to your needs. If we notice some difficulties, we will come back to you to help you.

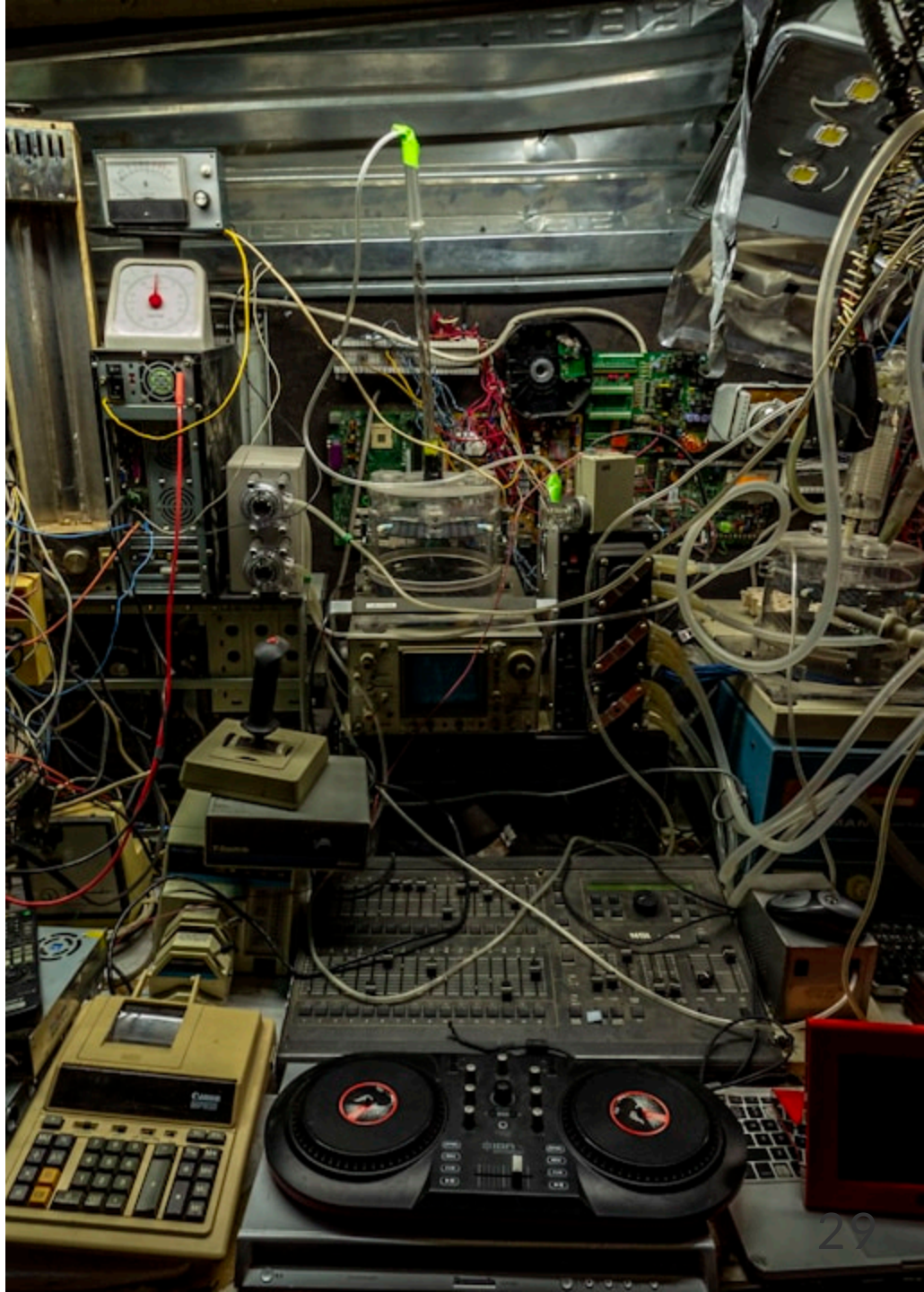
➡️ GitHub Discussions

You can use reactions to express your opinion on a comment!

What will you do next?

In the next chapter, you will learn the following topics:

- Java IOs: input/output processing
  - How to read and write files?
  - Why is encoding important?
  - How to deal with exceptions?



Sources

- Main illustration by Nathan Dumlao on Unsplash
- Illustration by Aline de Nadai on Unsplash
- Java logo by Java
- SDKMAN! logo by SDKMAN!
- IntelliJ IDEA and Intellij Toolbox logos by JetBrains
- Maven logo by Apache Software Foundation
- Illustration by Nathan Dumlao on Unsplash