

Software Entwicklung & Programmierung

Build Management Tools

Bilder und Texte der Veranstaltungsfolien und -unterlagen sowie das gesprochene Wort innerhalb der Veranstaltung und Lehr-Lern-Videos dienen allein dem Selbst- bzw. Gruppenstudium. Jede weiterführende Nutzung ist den Teilnehmenden der Moodle-Kurse untersagt, z.B. Verbreitung an andere Studierende, in sozialen Netzwerken, dem Internet!

Darüber hinaus ist ein studentischer Mitschnitt von Webkonferenzen im Rahmen der Lehre nicht erlaubt.

Am Ende dieser Präsenzstunde könnt Ihr:

- Erläutern was ein **Build Management Tool** ist und welche Komponenten es umfasst
- die zwei verschiedenen Build Management Tools **Maven** und **Gradle** voneinander unterscheiden und ihre Anwendung erläutern
- Mittels der vorgestellten Build Management Tools ein **eigenes Projekt** erstellen

Agenda

1. Grundlagen
2. Maven
3. Gradle



Maven und Gradle

- Build Management Tools:
 - helfen bei der Verwaltung des Projekts im kompletten Lebenszyklus (Kompilieren, Testen, Verpacken)
 - Ermöglichen die automatische Einbindung von benötigten Bibliotheken
- Support durch Entwicklungsumgebung:
 - Projekte können direkt als Maven- oder Gradle-Projekt erstellt werden (spezifische Beispiele folgen auf den nächsten Folien)
 - Bietet Projektstruktur gemäß des Build Management Tools inklusive zugehöriger Dateien (z.B. pom.xml)

Links: <https://maven.apache.org/>, <https://gradle.org/>

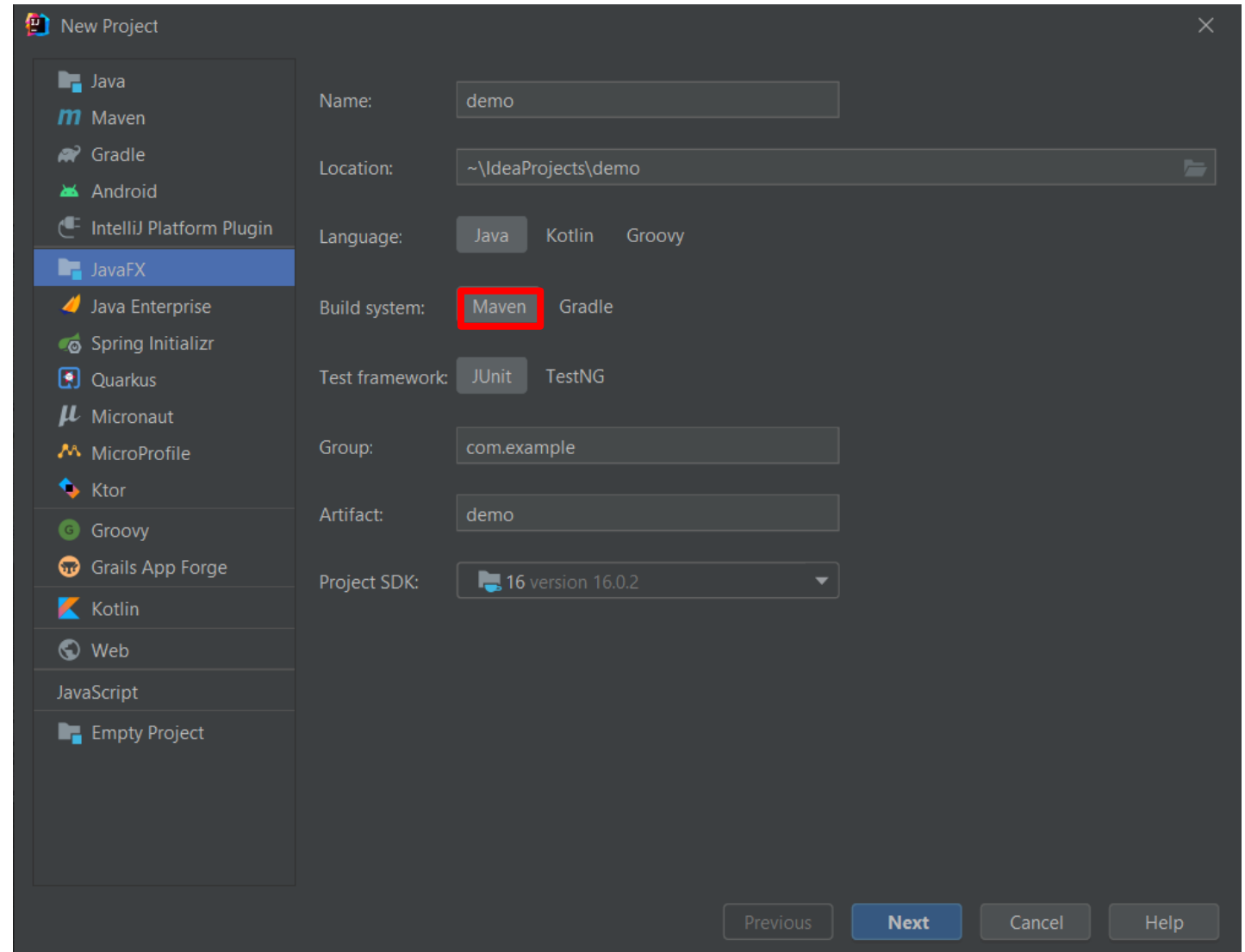
Agenda

1. Grundlagen
2. **Maven**
3. Gradle

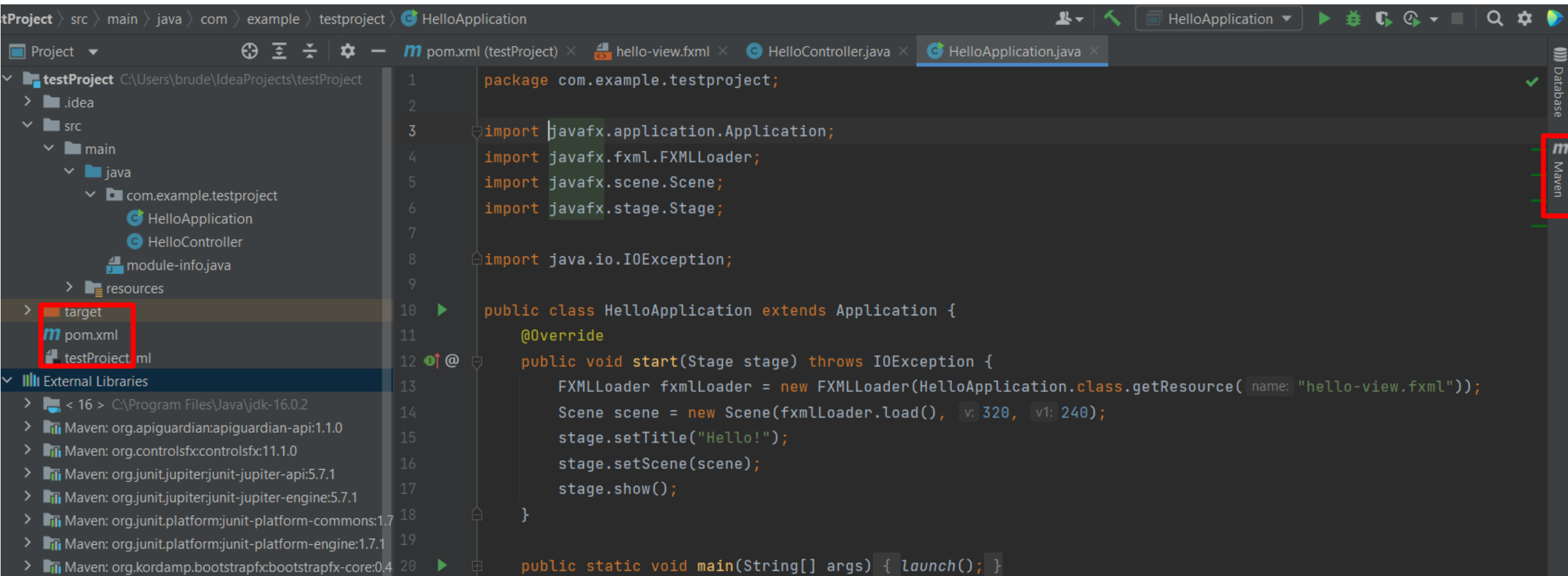


Mavenprojekt - Am Beispiel der IntelliJ IDE (2021.2)

- Durch die neue IntelliJ-Version, kann direkt ein Javafx-Projekt mit Maven oder Gradle erstellt werden (s. Bild)
- Alternativ kann auch ein alleiniges Maven- oder Gradle-Gerüst ausgewählt werden
 - z.B. zum Arbeiten mit React, Spring etc.



Auto-generated JavaFX Maven Projekt IntelliJ



Pom.xml

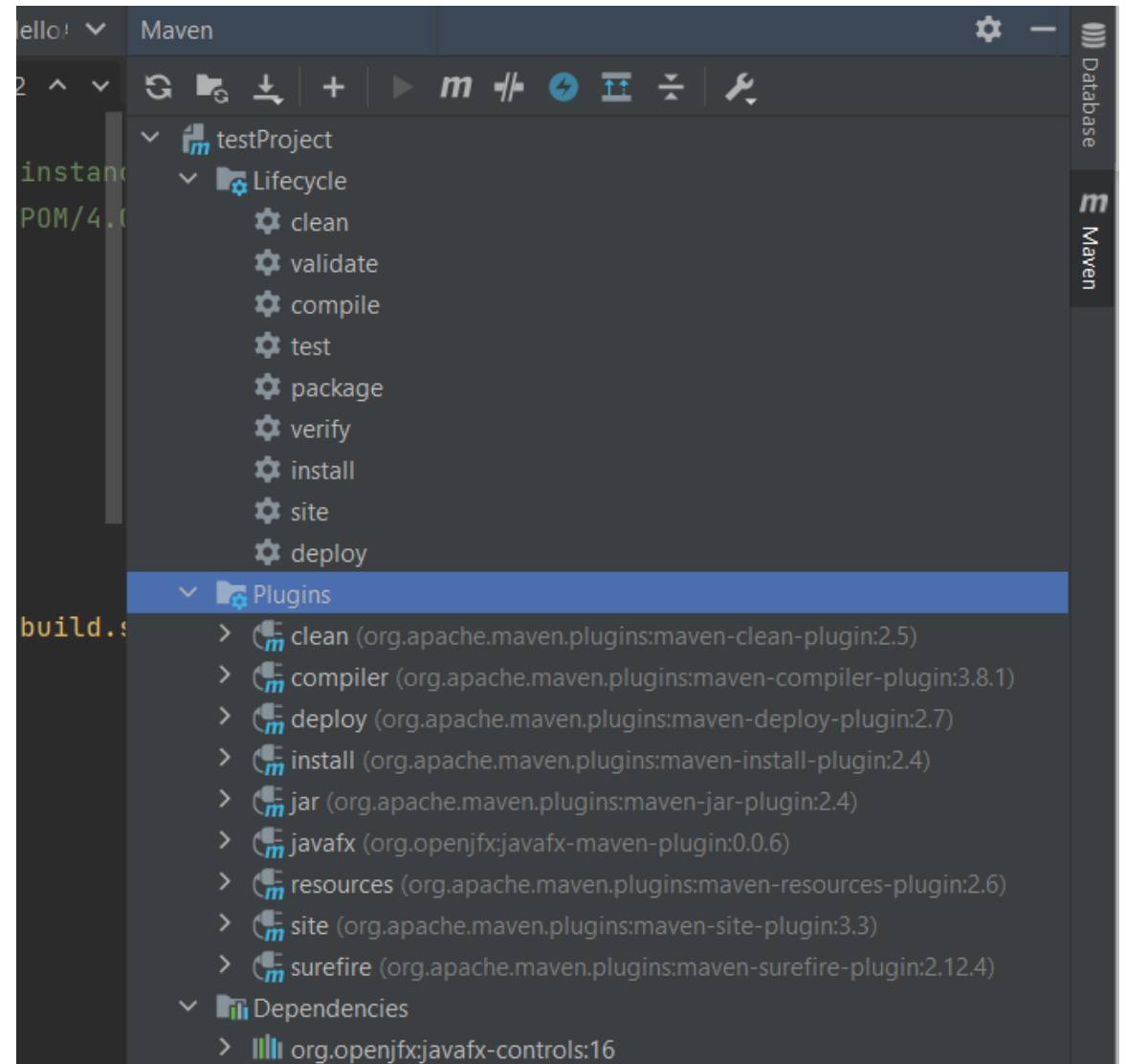
- Pom steht für **Project Object Model**
- Bezeichnet die zentrale Konfigurationsdatei im .xml Format
- Enthält alle Informationen des zugehörigen Projekts in standartisierter Form
 - Verwaltet beispielsweise zentral die im Projekt genutzten Libraries

```
m pom.xml (testProject) x hello-view.fxml x HelloController.java x HelloApplication.java x
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.example</groupId>
8   <artifactId>testProject</artifactId>
9   <version>1.0-SNAPSHOT</version>
10  <name>testProject</name>
11
12  <properties>
13    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
14    <junit.version>5.7.1</junit.version>
15  </properties>
16
17  <dependencies>
18    <dependency>
19      <groupId>org.openjfx</groupId>
20      <artifactId>javafx-controls</artifactId>
21      <version>16</version>
22    </dependency>
23    <dependency>
24      <groupId>org.openjfx</groupId>
25      <artifactId>javafx-fxml</artifactId>
26      <version>16</version>
27    </dependency>
28  </dependencies>
```

Charakteristika von Maven

Maven Konsole

- Stellt ohne Konsoleneingaben den Maven-Lebenszyklus dar
- Kann dazu genutzt werden, um beispielsweise eine .jar-Datei eures Projekts zu erstellen
- *Clean* – löscht Ergebnisse vorheriger Builds und kann bei Bugs als erster Anhaltspunkt gewählt werden



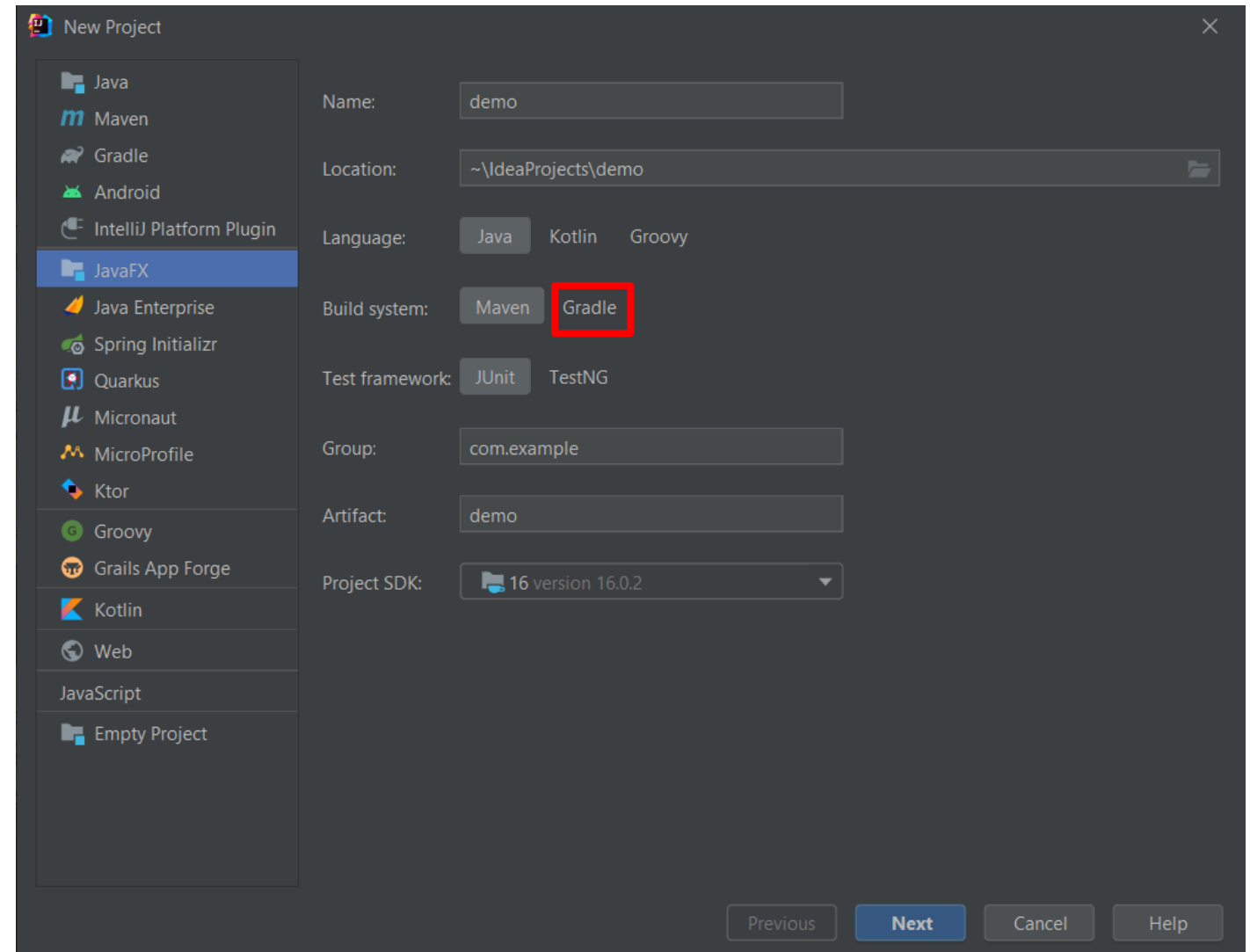
Agenda

1. Grundlagen
2. Maven
3. **Gradle**

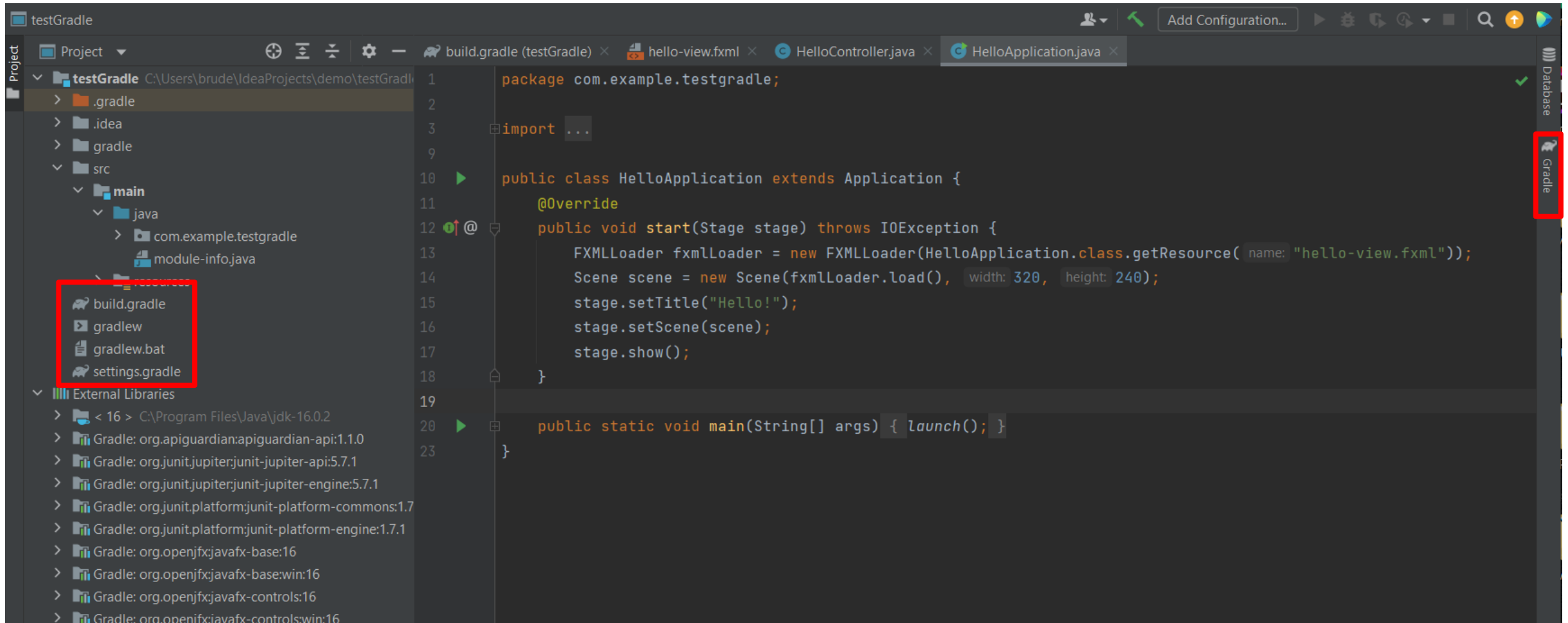


Gradleprojekt - Am Beispiel der IntelliJ IDE (2021.2)

- Durch die neue IntelliJ-Version, kann direkt ein Javafx-Projekt mit Maven oder Gradle erstellt werden (s. Bild)
- Alternativ kann auch ein alleiniges Maven- oder Gradle-Gerüst ausgewählt werden
 - z.B. beim Arbeiten mit React, Spring etc.



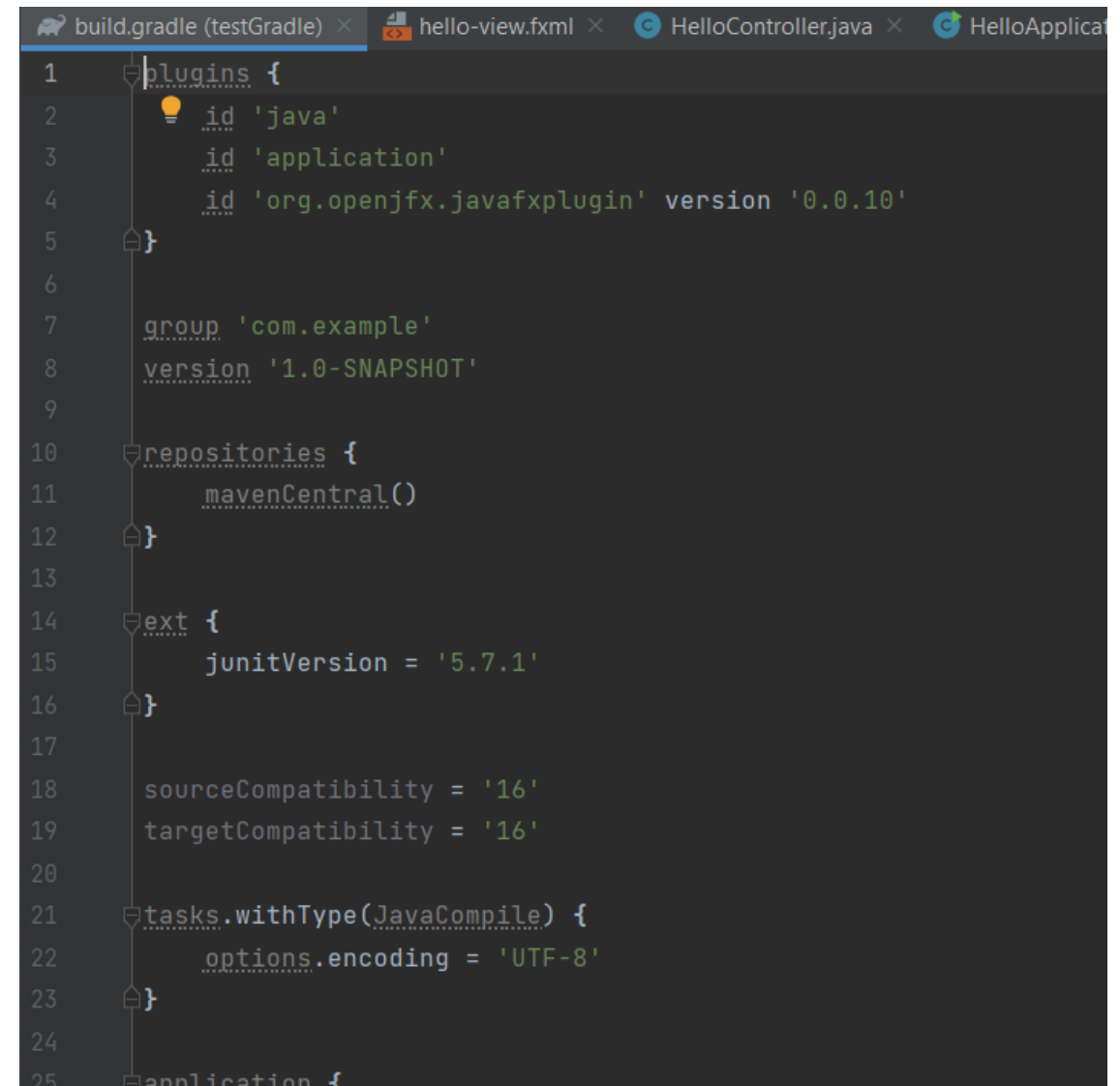
Auto-generated JavaFX Gradle Projekt



Charakteristika von Gradle

DSL und build.gradle

- Im Gegensatz zu Maven gibt es keine XML Deklarationen, sondern Domain Specific Language (DSL) von Gradle
- Eigene Buildskripte erfordern das Lernen von Groovy
- *Build.gradle* – beinhaltet ähnlich wie die pom.xml, Abhängigkeiten und Tasks des Projekts



```
1 plugins {  
2     id 'java'  
3     id 'application'  
4     id 'org.openjfx.javafxplugin' version '0.0.10'  
5 }  
6  
7 group 'com.example'  
8 version '1.0-SNAPSHOT'  
9  
10 repositories {  
11     mavenCentral()  
12 }  
13  
14 ext {  
15     junitVersion = '5.7.1'  
16 }  
17  
18 sourceCompatibility = '16'  
19 targetCompatibility = '16'  
20  
21 tasks.withType(JavaCompile) {  
22     options.encoding = 'UTF-8'  
23 }  
24  
25 application {
```

Quellen

- <https://maven.apache.org/>
- <https://gradle.org/>

Vielen Dank für Eure Aufmerksamkeit