

Vorlesung

Einführung in die objektorientierte Programmierung

Benötigte Hard- und Software

Benötigte Hardware

Notebook

- keine blockierenden Company Proxies
- Admin-Zugriff (um stressfrei installieren zu können)

Wenn nicht: Lösungen für dieses Probleme innerhalb des Unternehmens heraussuchen und parat halten.

Benötigte Software

1. Java Development Kit (JDK), Version 21+
2. IDE (IntelliJ IDEA Ultimate/Community)
3. Maven
4. Git

IntelliJ IDEA

About - IntelliJ

- Integrated Development Environment (IDE) von JetBrains
- Entwicklungsumgebung, die wir in dieser Vorlesung verwenden
- Alternativen: Visual Studio Code, Eclipse, ...
- Support in der Vorlesung: Nur für IntelliJ IDEA

IntelliJ IDEA (Check)

- Öffnen von IntelliJ IDEA führt zu Projekt-Dialog:
 - ermöglicht ein neues Projekt zu anzulegen
 - ermöglicht ein bestehendes Projekt zu öffnen

Installation von IntelliJ IDEA Ultimate

Schritt 1: Download

- JetBrains Webseite: <https://www.jetbrains.com/de-de/idea/>
- **Jetbrains Toolbox** runterladen
 - Wrapper um alle möglichen JetBrains IDEs
 - z. B. IntelliJ, PyCharm, RubyMine, etc.
- in Toolbox: IntelliJ Ultimate / Community installieren
- je nach Betriebssystem andere Anweisungen nach Download

Schritt 2: JetBrains Student Abo abschließen

- <https://www.jetbrains.com/shop/eform/students>
- Dialog ausfüllen
- JetBrains Konto anlegen (Campus-Email)
- Auf diesem Konto die Student Benefits verfügbar bekommen

Schritt 3: Installation starten

- im Dateibrowser zum Download navigieren
- Executable ausführen
- Installationsassistenten folgen und gewünschten Einstellungen auswählen
- Falls nach Lizenz gefragt wird: Mit erstelltem Konto einloggen

Schritt 4: Initiale Konfiguration

- IntelliJ IDEA Ultimate-Symbol auf Ihrem Desktop anklicken
- ein paar letzte Einstellungen festlegen

Java Development Kit (JDK)

About - JDKs

- Java Distributionen, die alle Tools beinhalten, die zur Software-Entwicklung mit Java benötigt werden
- Bestandteile:
 1. Java Virtual Machine (Konsole: `java`)
 2. Java Compiler (Konsole: `javac`)
 3. Debugger, Signer etc.

Beispiele

- Oracle JDK
- OpenJDK
- ...

Installation von JDKs

1. Durch die IDE (IntelliJ)
2. Mit `sdkman` (Linux, MacOS)
3. Manuell (Windows)

Java Virtual Machine (Check)

```
$ java --version
```

...führt zu bspw. folgender Konsolenausgabe:

```
openjdk 21.0.2 2022-01-18 LTS  
OpenJDK Runtime Environment XY (build 21.0.2+8-LTS)  
OpenJDK 64-Bit Server VM XY (build 21.0.2+8-LTS, mixed mode)
```

Java Compiler (Check)

```
$ javac --version
```

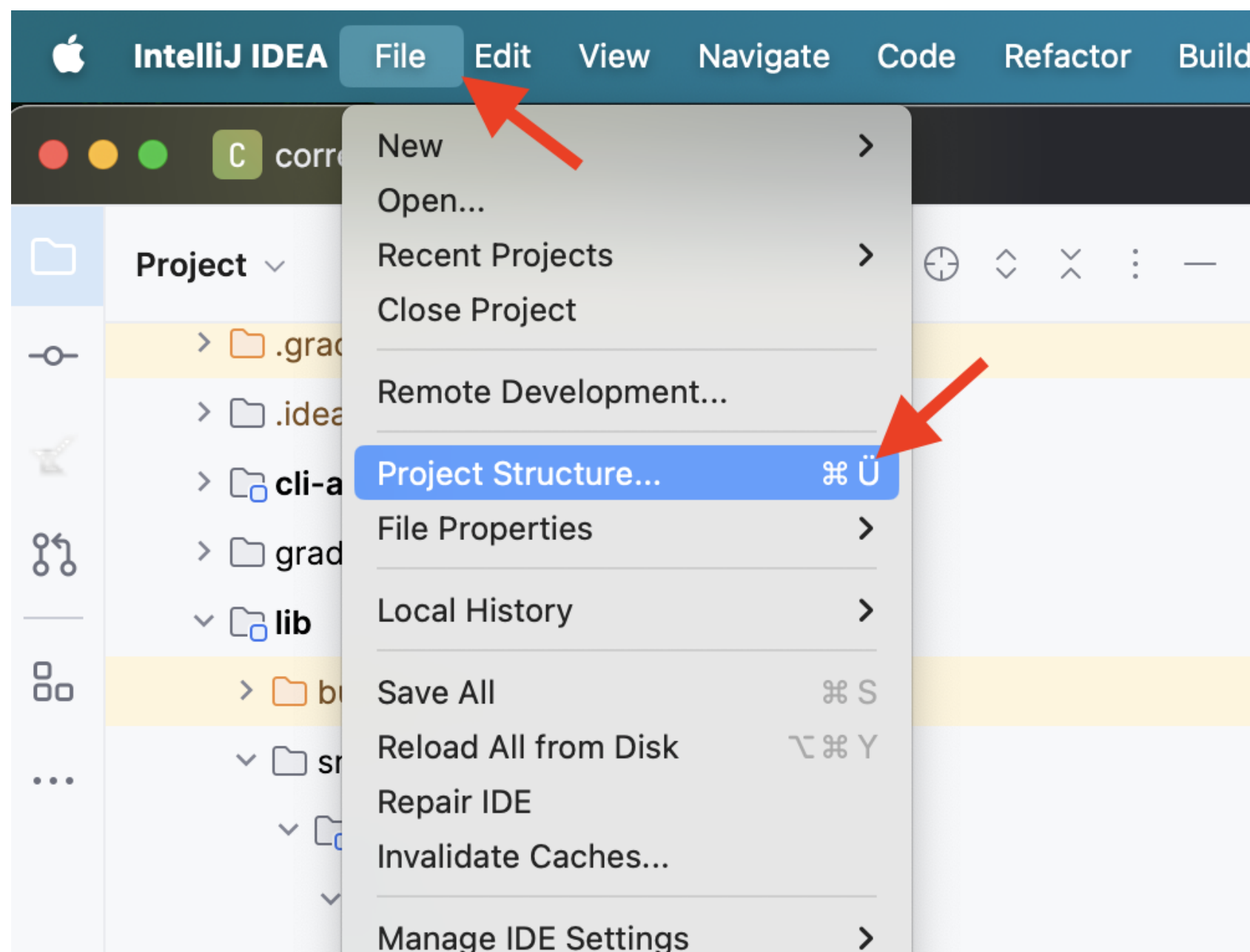
...führt zu folgender Konsolenausgabe:

```
javac 21.0.2
```

Installation über die IDE

...mit IntelliJ

Schritt 1: Project Structure



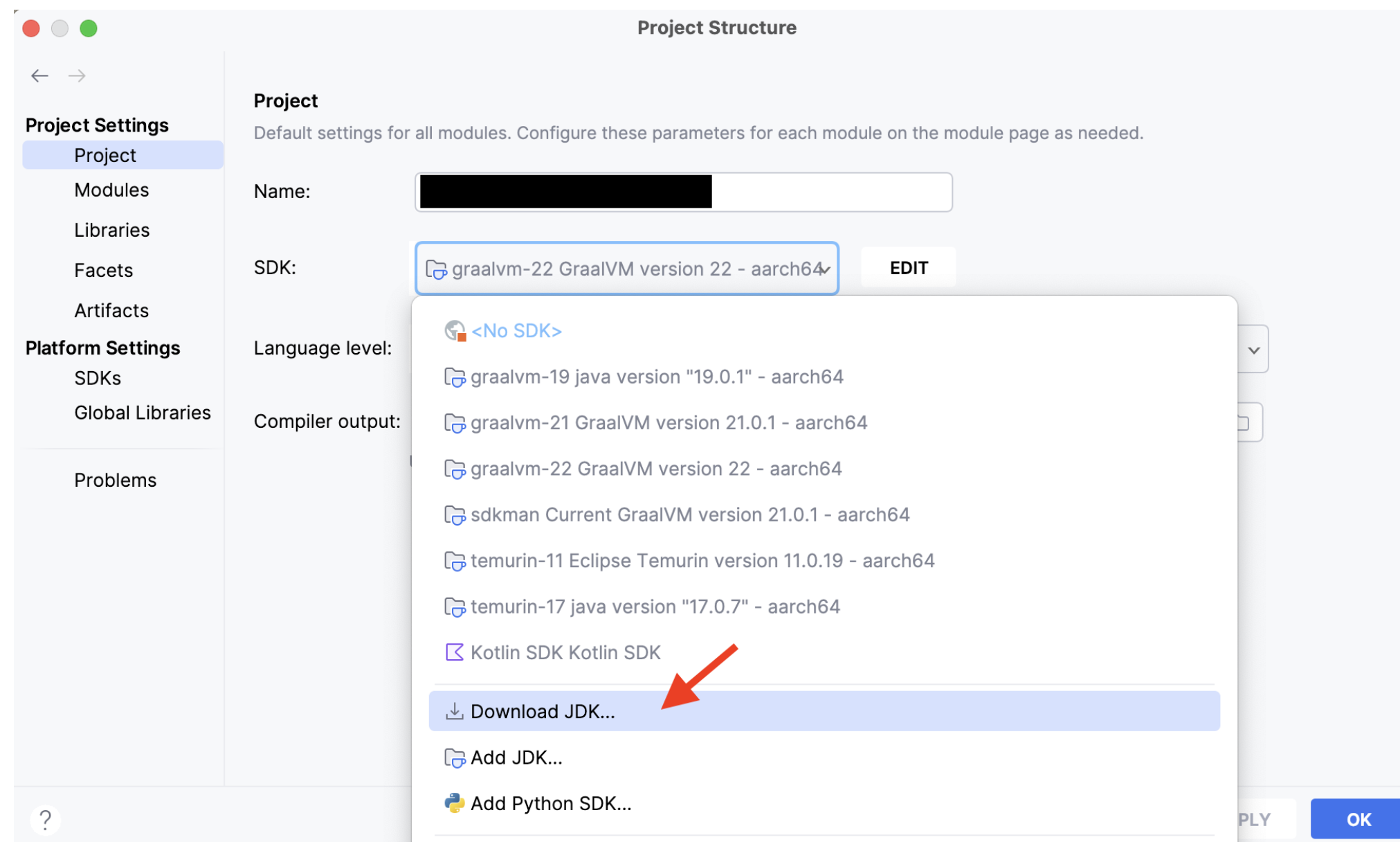
- File / Project Structure
- öffnet Project Settings, wo die passenden SDKs konfiguriert werden können

Schritt 2: Project SDK



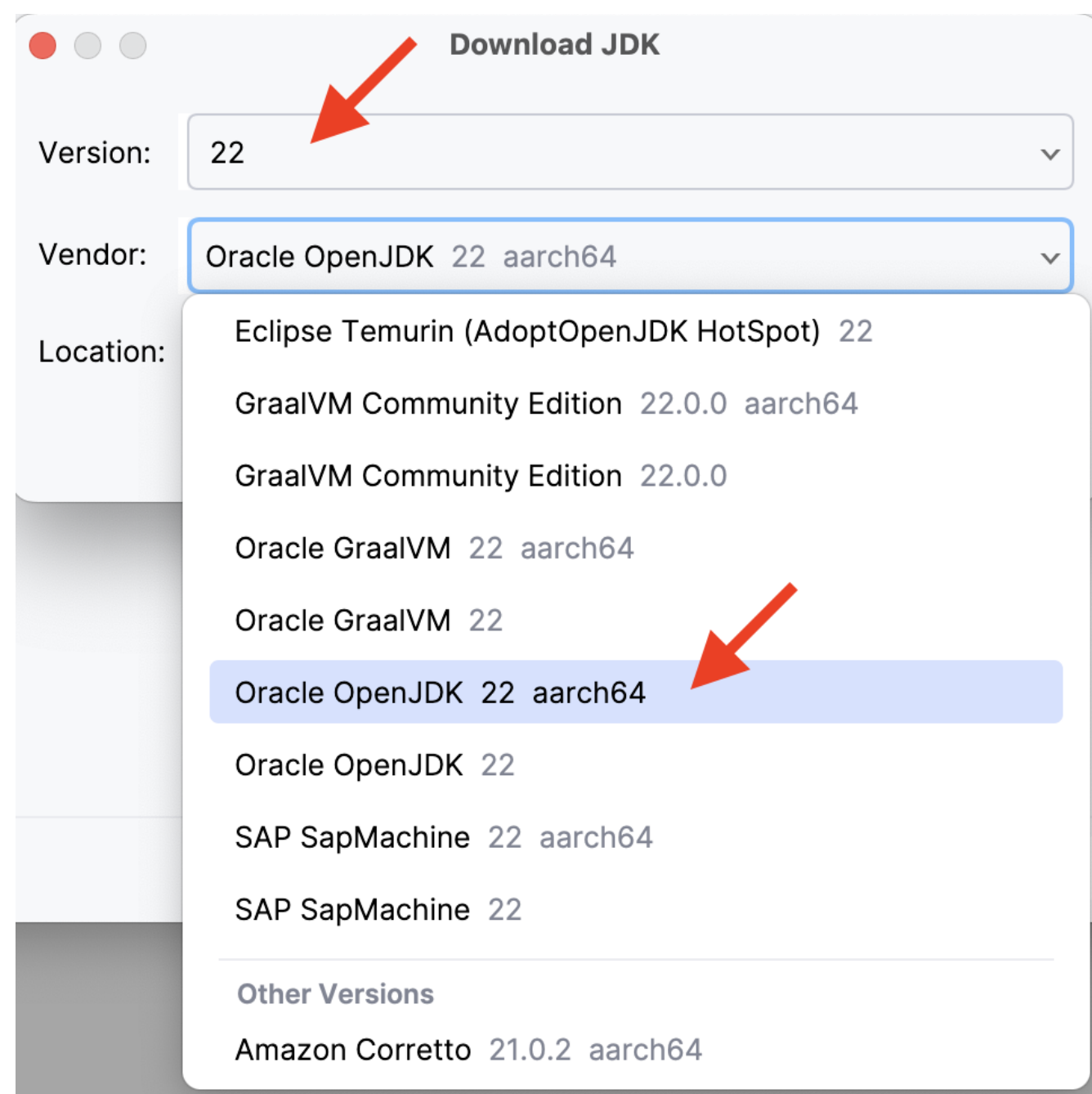
- unter Project / SDK muss das JDK für das Project ausgewählt werden
- initial leer, falls bereits JDK installiert, ggf. voreingestellt

Schritt 3: Download JDK



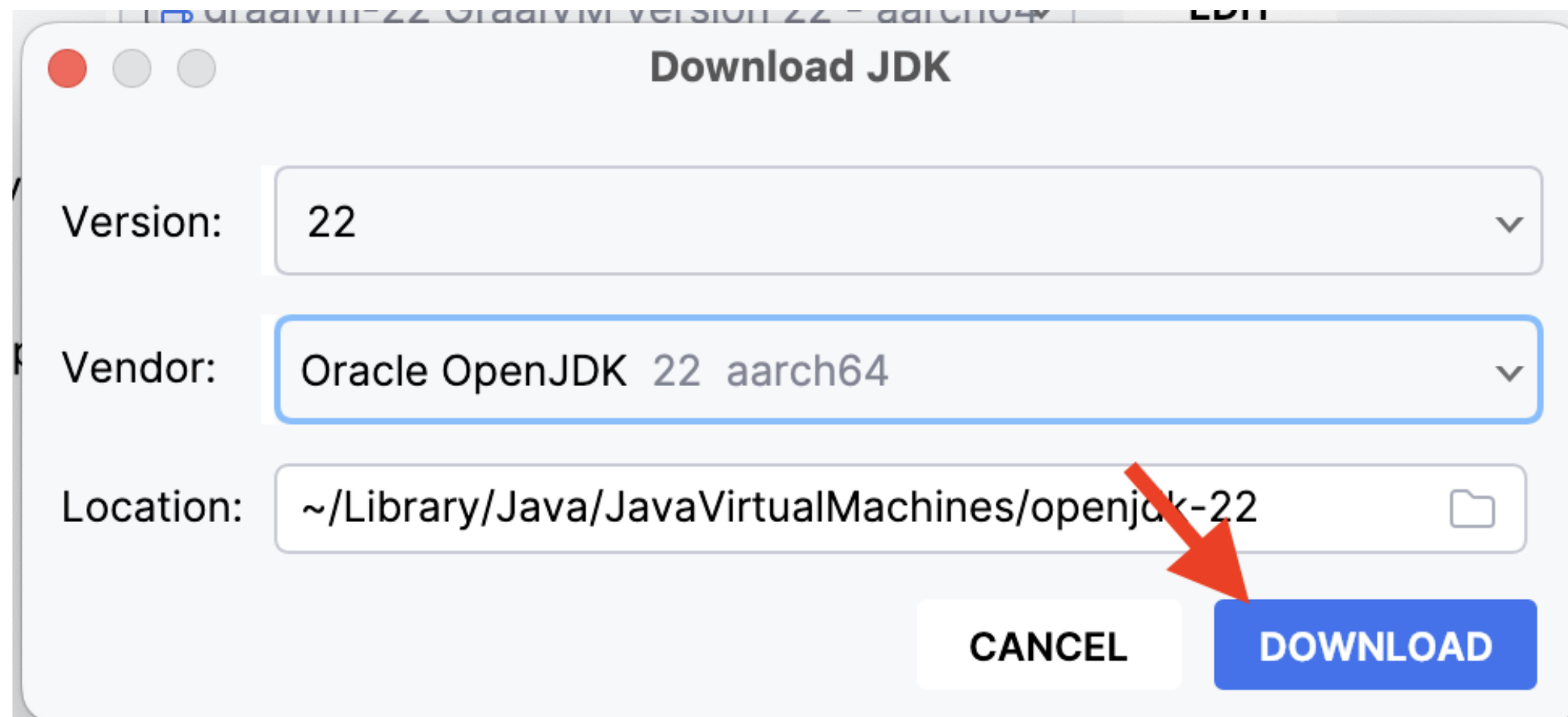
- unten im Dropdown gibt es die Möglichkeit ein neues JDK runterzuladen
- IntelliJ kümmert sich um das passende Setzen von Umgebungsvariablen
- **WICHTIG** JAVA_HOME und PATH

Schritt 4: JDK auswählen



- Aktuelle LTS Version: **JDK 21**
 - *virtual threads* und *pattern matching for switch*
 - vorheriger Lecture-Durchlauf noch Java 17
 - dieses Mal schon Java 21
- JSK 22 (verfügbar seit Ende März)

Schritt 5: JDK herunterladen



WICHTIG Download nicht vergessen!

Installation über **sdkman**

...für die Praxis am einfachsten/sinnvollsten!

Installation sdkman

- Online-Anleitung: <https://sdkman.io/install>
- Ergebnis: `sdk` CLI Tool verfügbar

Installation von JDKs

```
$ sdk list java
```

- listet alle für `sdkman` verfügbaren JDKs auf
- alle mit Version 21 oder höher passen

```
$ sdk install java 21.0.2-oracle
```

- installiert passendes LTS OpenJDK

Manuelle Installation (Windows)

Schritt 1: Download

- z. B. Adoptium JDK: <https://adoptium.net/de/temurin/releases/>
- Gängige Archiv-Formate: `.zip`, `.tar.gz`
- Fürs passende Betriebssystem eine Datei auswählen

Schritt 2: OpenJDK-Archiv extrahieren

- im Dateibrowser zum Speicherort der heruntergeladenen Datei gehen
- Archiv in einen Ordner der Wahl extrahieren
- Empfehlung:
 - Windows: C:/User/**Benutzername**/dev/runtimes/jdk/**konkretesJDK**
 - Linux/MacOS: ~/dev/runtimes/jdk/**konkretesJDK**

Schritt 3: Umgebungsvariable JAVA_HOME

- Start, "Umgebungsvariablen" in der Suche eingeben und öffnen
- dann "Neu" unter "Systemvariablen"
- Name der Variable: "JAVA_HOME", dann Pfad zum Ordner indem das Archiv extrahiert wurde

Schritt 4: Java-Binärpfad zu PATH hinzufügen

- Bereithalten: Pfad vom "bin"-Ordner im JDK-Ordner
- Umgebungsvariablen-Dialog (wie vorher) öffnen
- **Path** unter "Systemvariablen" auswählen, dann "bearbeiten"
- in der geöffneten Tabelle auf "Neu"
- in der neuen Zeile den Pfad zum "bin"-Ordner hinzufügen

Schritt 5: Installation überprüfen

1. Java Virtual Machine ✓
2. Java Compiler ✓

Maven

About - Maven

- populäres Buildtool für Java Projekte
- ermöglicht einfaches "Bauen" der Software
- Vorteil: Wir müssen nicht jedes Mal den Compiler bedienen
- Konfiguration über `pom.xml` Datei in Projekten

Installation von Apache Maven

1. Überprüfen ob nicht bereits installiert

- Müsste durch IntelliJ bereits installiert sein
- wenn nicht: Manuell installieren

2. Manuell (Anleitung für Windows)

Maven (Check)

```
$ mvn --version
```

...führt zu folgender Konsolenausgabe:

```
Maven home: /opt/homebrew/Cellar/maven/3.8.7/libexec  
Java version: 21.0.2, vendor: XY, runtime: YOUR_JAVA_DIRECTORY  
Default locale: de_DE, platform encoding: UTF-8  
OS name: "HOPEFULLY_!W6", version: "42", arch: "ARCH", family: "FAM"
```

Installation auf Windows

Schritt 1: Download

- Apache Maven Webseite: <https://maven.apache.org/download.cgi>
- Archiv-Format auswählen: `bin.zip` oder `bin.tar.gz`

Schritt 2: Maven-Archiv extrahieren

- im Dateibrowser zum Speicherort der heruntergeladenen Datei gehen
- Archiv in einen Ordner der Wahl extrahieren
- Empfehlung:
 - Windows:
C:/User/**Benutzername**/dev/tools/maven/**konkreteMavenVersion**
 - Linux/MacOS: ~/dev/tools/**konkreteMavenVersions**

Schritt 3: Umgebungsvariable MAVEN_HOME

- Start, "Umgebungsvariablen" in der Suche eingeben und öffnen
- dann "Neu" unter "Systemvariablen"
- Name der Variable: "MAVEN_HOME", dann Pfad zum Ordner indem das Archiv extrahiert wurde

Schritt 4: Java-Binärpfad zu PATH hinzufügen

- Bereithalten: Pfad vom "bin"-Ordner im Maven-Ordner
- Umgebungsvariablen-Dialog (wie vorher) öffnen
- **Path** unter "Systemvariablen" auswählen, dann "bearbeiten"
- in der geöffneten Tabelle auf "Neu"
- in der neuen Zeile den Pfad zum "bin"-Ordner hinzufügen

Git

About - Git

- Software zur verteilten Versionsverwaltung von Dateien
- Version Control System (VCS)
- durch Linus Torvalds et. al. initiiert
- en. *git*: britische Umgangssprache für "Blödmann"

Installation von Git

Git (Check)

```
$ git --version
```

...führt zur Ausgabe der installierten Git Version und zu keiner Fehlermeldung.

Manuelle Installation

Schritt 1: Download

- Git-Website: <https://git-scm.com/downloads>
- Installer herunterladen

Schritt 2: Installationsprozess starten

- Navigieren zum Speicherort der heruntergeladenene Datei
- Öffnen und Installationsprozess starten

Schritt 3: Installationskomponenten auswählen

- Standardkomponenten reichen
- dasselbe gilt für alle folgenden Fragen, die der Installer stellt
- Ausnahme: **Editor!**

Schritt 5: Editor auswählen

- Editor auswählen, den man mit Git verwenden möchte
- Standardeditor passt, ansonsten einen, den man gut kennt
- **nicht Vim, wenn man nicht Vim kann**