ELO Flows Development

Grundlagen

Inhaltsverzeichnis

Grundlagen Flows Entwicklung

Technische Systemvoraussetzungen

In diesem Kapitel finden wir einen ersten Überblick über die Implementierung der notwendigen Werkzeuge.

Visual Studio Code (VS Code)

Visual Studio Code ist ein Quelltexteditor, der in verschiedenen Betriebssystemen (Windows, Linux und macOS) eingesetzt werden kann. Bei dem Editor handelt es sich um keine echte integrierte Entwicklungsumgebung (IDE). Dennoch stellt VS Code alle gängigen Programmierwerkzeuge wie Versionsverwaltung, Debugging, Code-Vervollständigung oder Syntaxhervorhebung zur Verfügung. Der Quelltexteditor arbeitet nicht auf Projektbasis. Die Implementierungsarbeiten werden in Arbeitsumgebungen und Ordnern organisiert. Durch den Einsatz von Plug-ins kann VS Code nahezu für jede Programmiersprache eingesetzt, individuell angepasst und erweitert werden.

Java Runtime Environment (Version Java 17)

Als Basis für die Implementierung von ELO Flows-Komponenten kommt die Programmiersprache Java zum Einsatz. Damit Java-Anwendungen erstellt und ausgeführt werden können, bedarf es einer Java-Laufzeitumgebung (JRE). Für umfangreichere Implementierung ist ein Java Development Kit (JDK) zu empfehlen, das weitere Programmierwerkzeuge zur Verfügung stellt. Die Java-Laufzeitumgebung ist bereits im JDK enthalten. Für unsere Schulung werden wird die Version Java 17 verwenden.

Gradle 7.x

Gradle ist ein Build-Management-Projekt-Tool, das in der Softwareentwicklung für verschiedene Programmiersprachen eingesetzt werden kann. Das Werkzeug automatisiert den Entwicklungsprozess vom Kompilieren des Quellcodes in Binärcode über das Erstellen von Paketen bis hin zu Ausführung automatisierter Tests.

VS Code Plug-in – Framework zur Komponentenentwicklung

Das von ELO zur Verfügung gestellte Plug-in für VS Code ist ein Framework für die Implementierung von ELO Flows-Komponenten. Das Framework soll die Entwicklung vereinfachen und einheitliche Richtlinien für die Implementierung schaffen. Es beinhaltet eine Beispiel-Komponente, die den Einstieg in die Umsetzung erheblich beschleunigt. Der Entwickler kann nach der Installation ein Projekt starten und mit der Programmierarbeit beginnen. Es sind keine weiteren Konfigurationen bzw. Erweiterungen mehr nötig.

Framework (VS Code Plug-in)

In diesem Kapitel werfen wir einen Blick auf die Konfiguration der im Kapitel Technische Systemvoraussetzungen erwähnten Entwicklungswerkzeuge.

Visual Studio Code (VS Code)

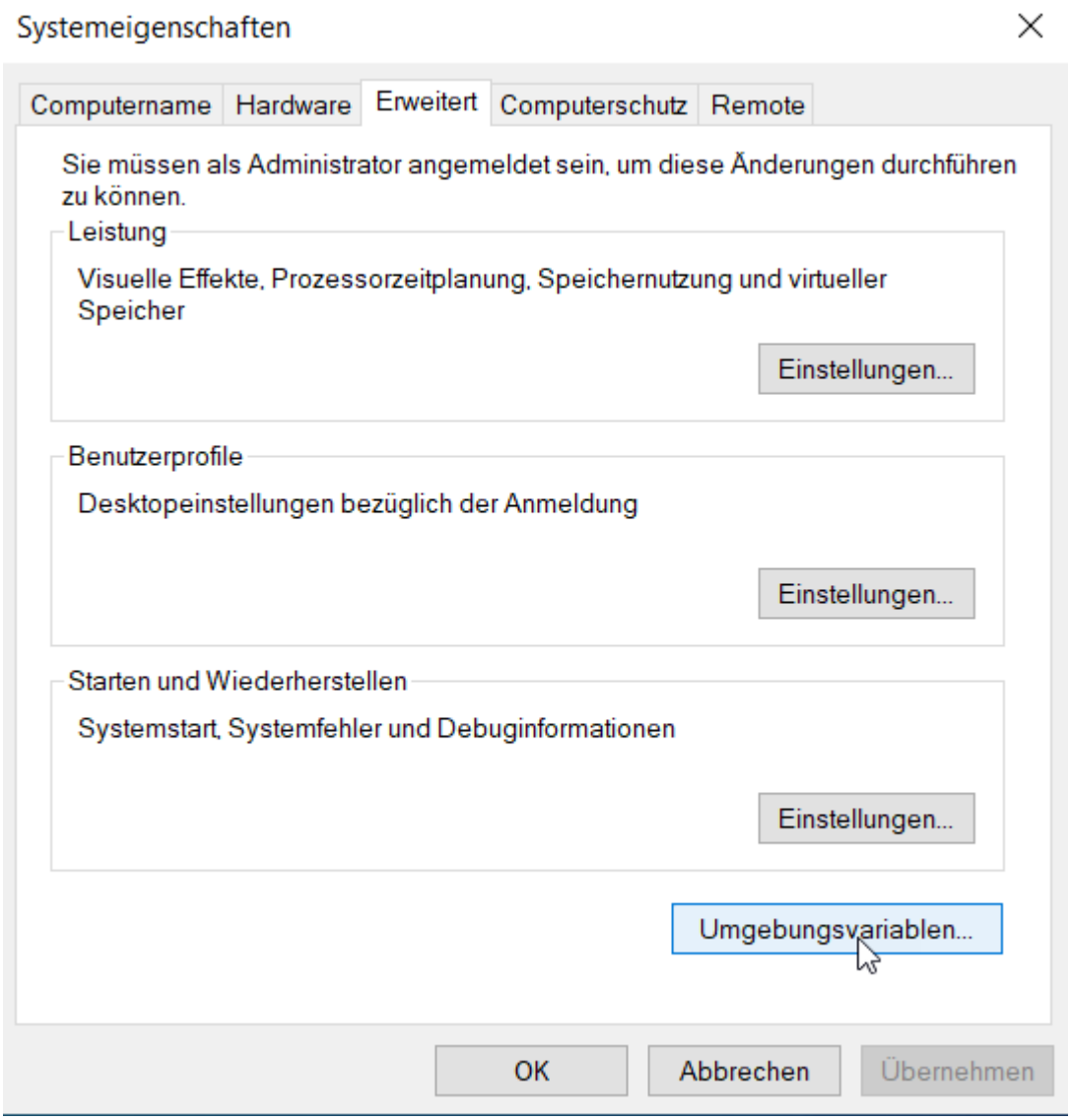
Hier sind keine Konfigurationsschritte notwendig.

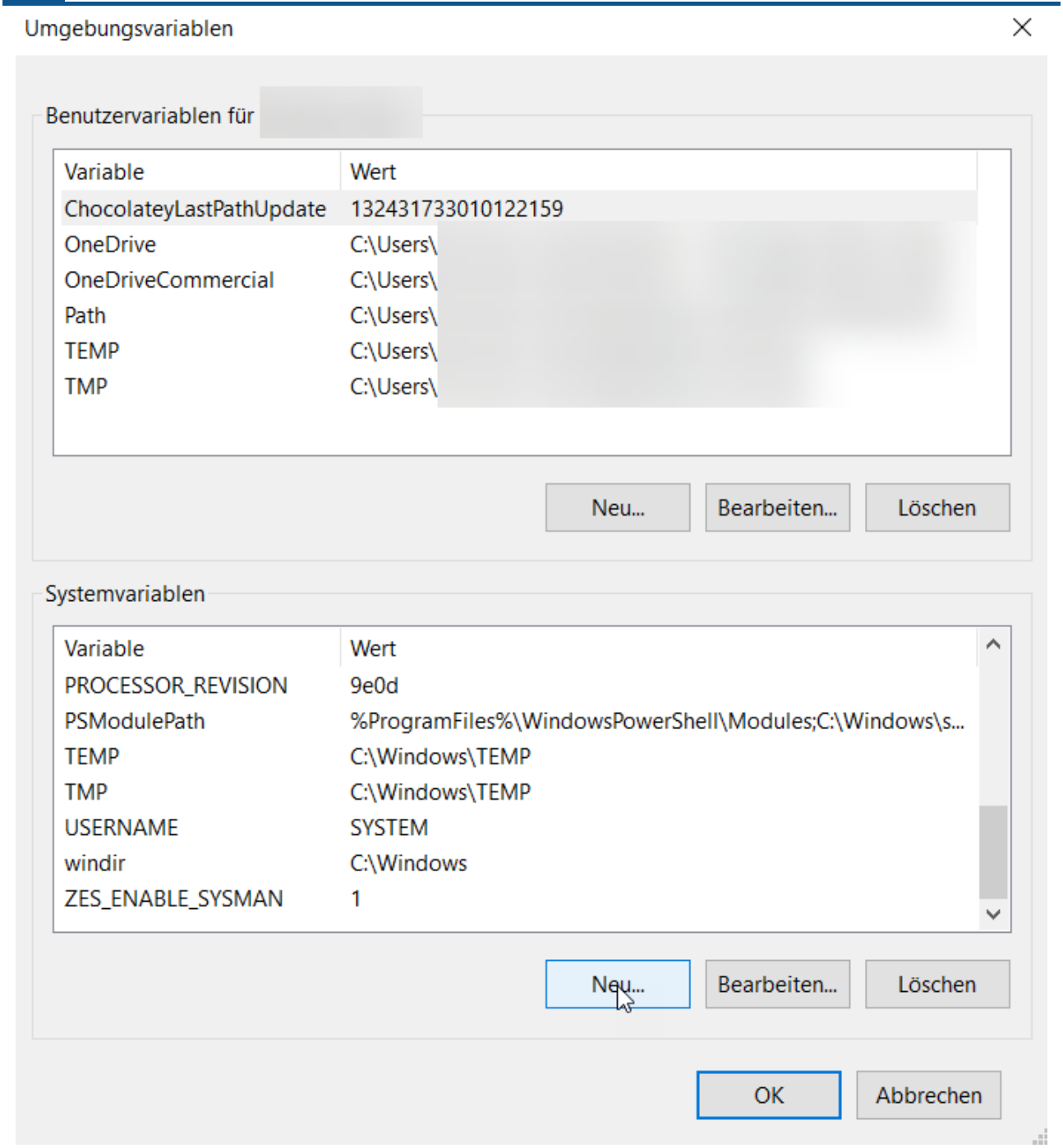
Java Runtime Environment (Open JDK 17)

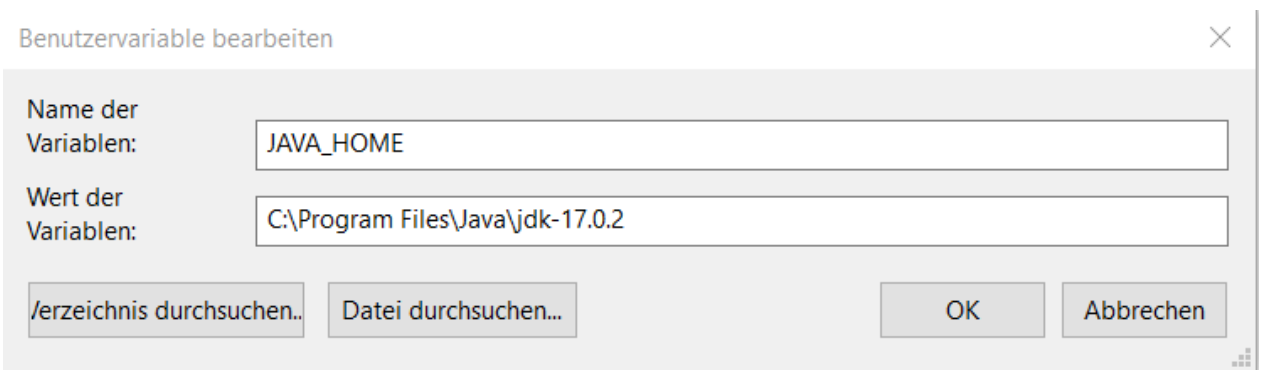
Nach der Installation des JRE wird die Umgebungsvariable *JAVA_HOME* angepasst.

1. Setzen Sie für Java die Umgebungsvariable `JAVA_HOME`. Tragen Sie den jeweils passenden Pfad ein (z. B. *JAVA_HOME=C:\Program Files\Java\jdk-17.0.2*)

In Systemeingenschaften wird die neue *JAVA_HOME* Umgebungsvariable definiert.





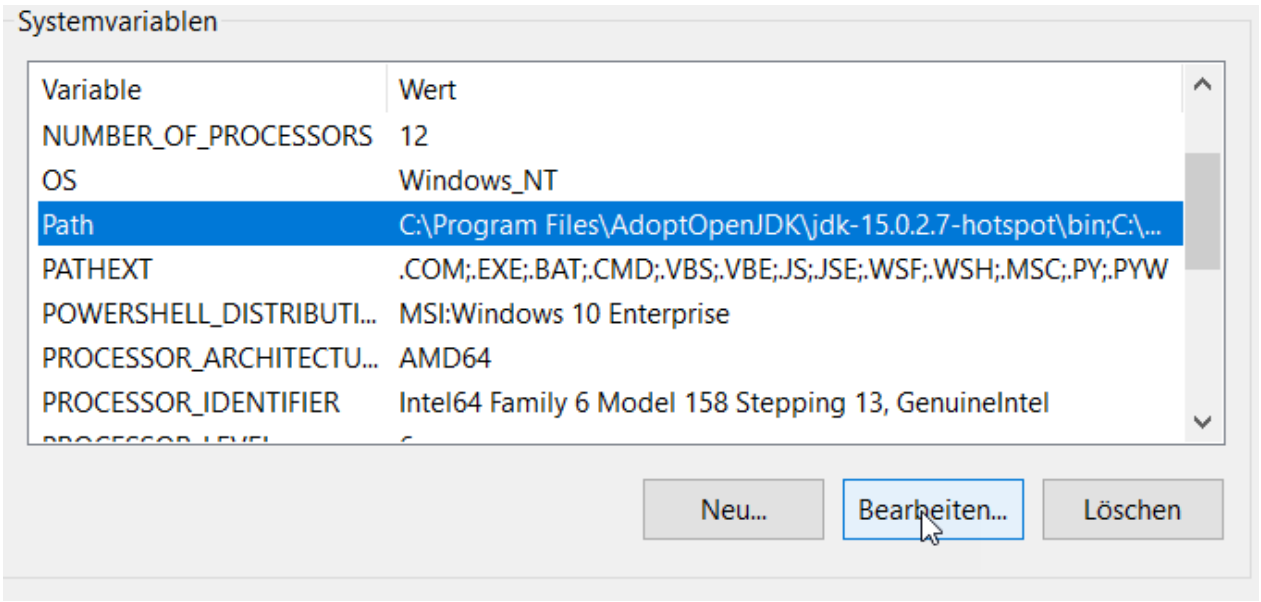


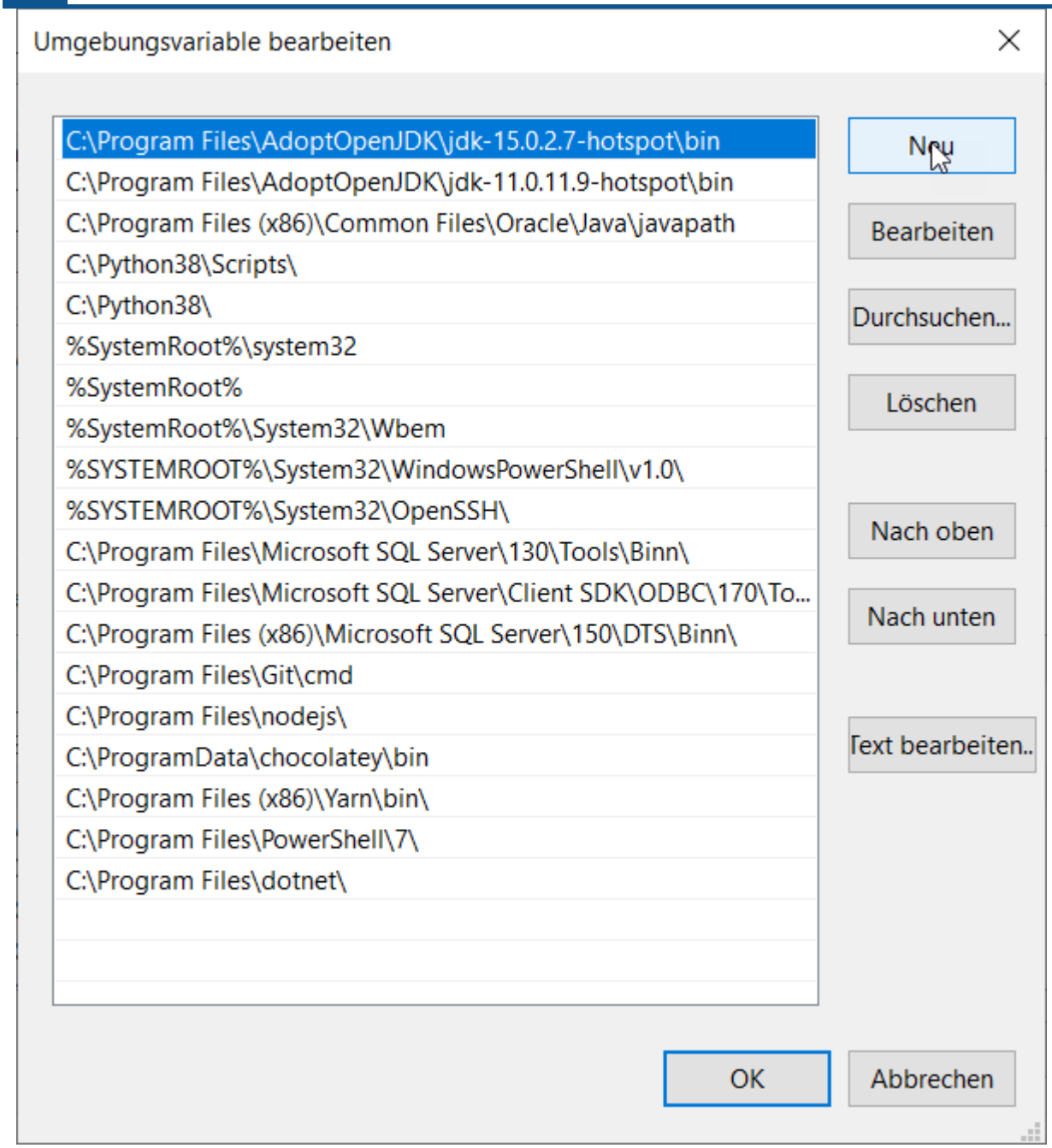
Gradle 7.3

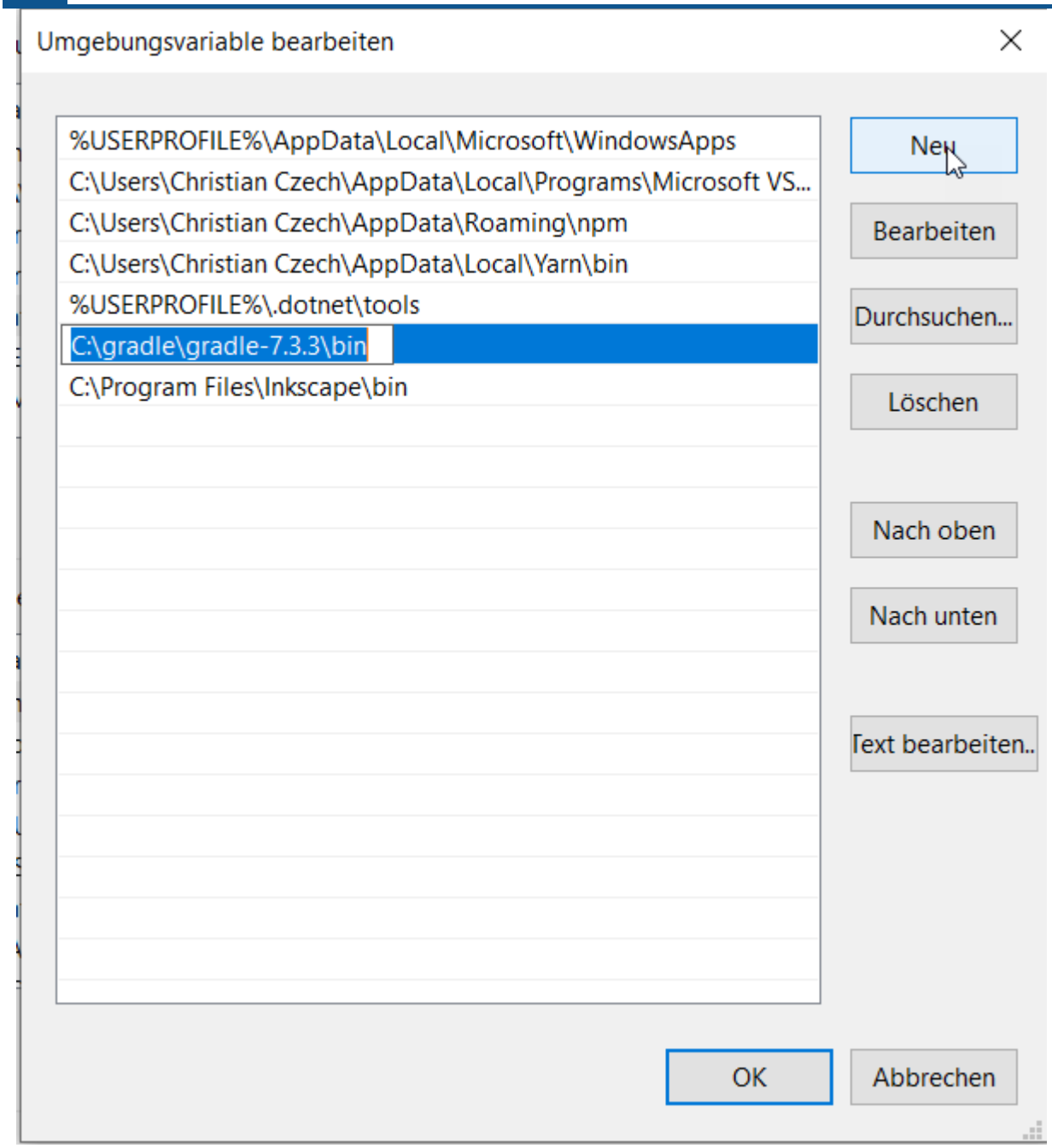
Nach der Installation der Gradle 7.3 Version wird die Umgebungsvariable `Path` angepasst.

1.

Ergänzen Sie für Gradle die Path-Variable. Tragen Sie den jeweils passenden Pfad ein.



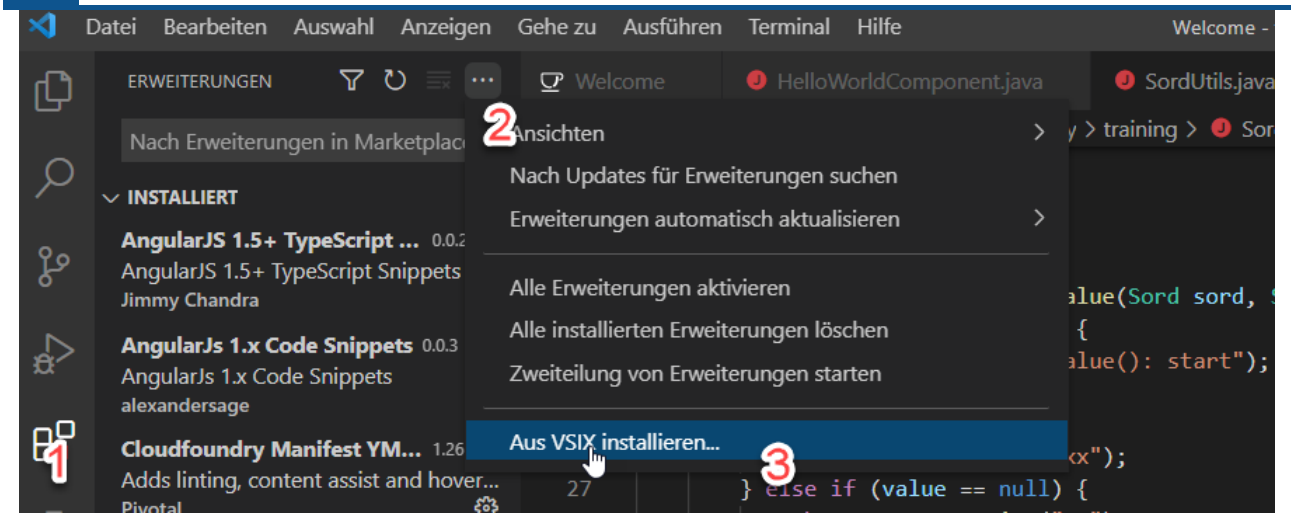

Umgebungsvariable bearbeiten
C:\Program Files\AdoptOpenJDK\jdk-15.0.2.7-hotspot\bin
C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\Python38\Scripts\
C:\Python38\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Microsoft SQL Server\130\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\To...
C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\
C:\Program Files\Git\cmd
C:\Program Files\nodejs\
C:\ProgramData\chocolatey\bin
C:\Program Files (x86)\Yarn\bin\
C:\Program Files\PowerShell\7\
C:\Program Files\dotnet\
Neu
Bearbeiten
Durchsuchen...
Löschen
Nach oben
Nach unten
Text bearbeiten...
OK
Abbrechen



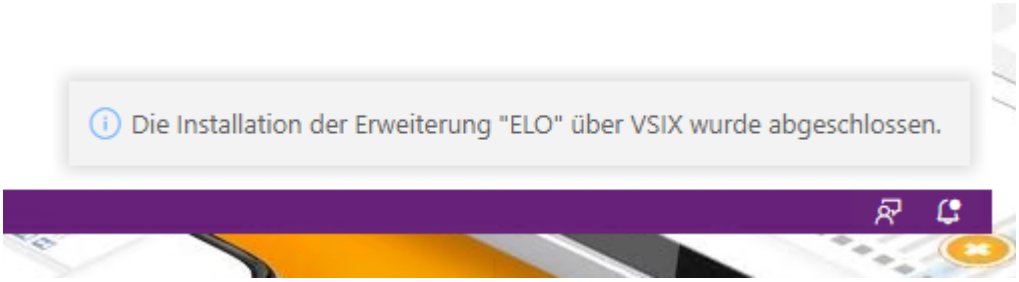
ELO Plug-in Installation

Nachdem alle Werkzeuge konfiguriert sind, können wir das Plug-in zur Flows-Komponentenentwicklung installieren.

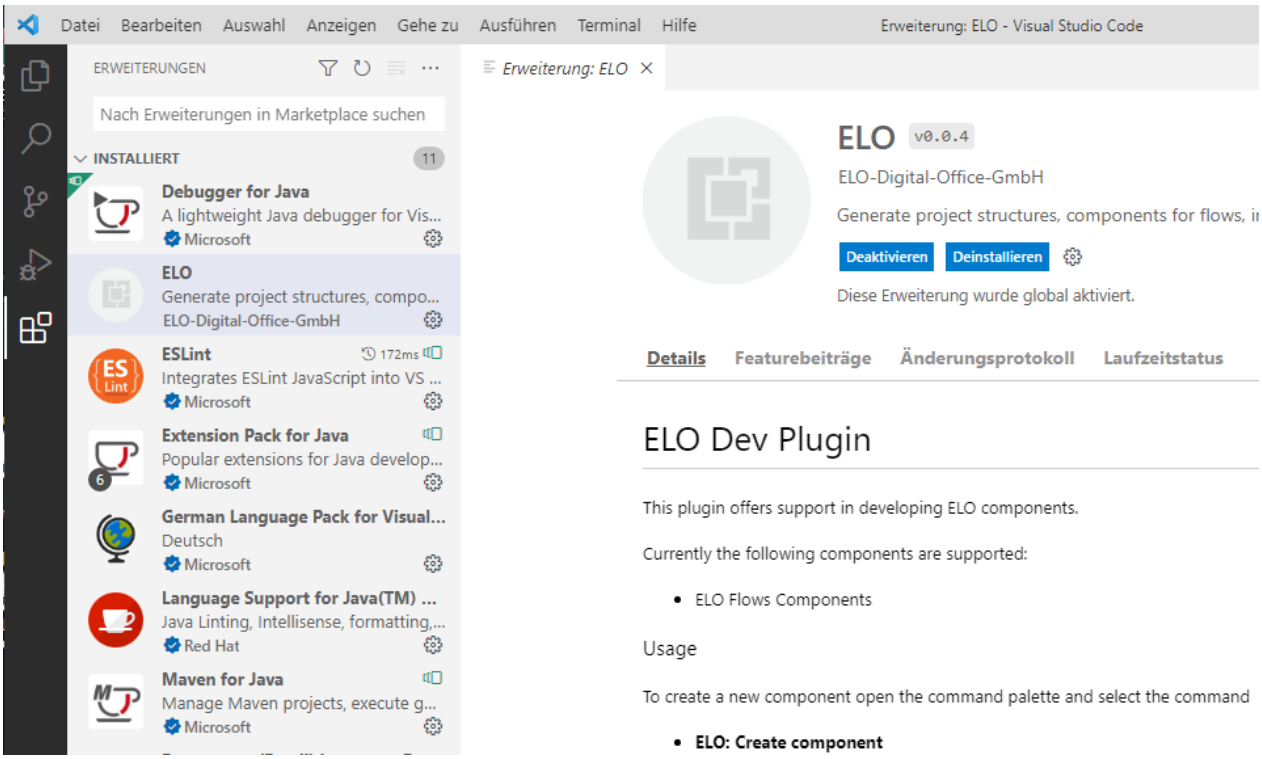
1. Öffnen Sie VS Code.
2. Navigieren Sie in den Bereich *Erweiterungen* (1).
3. Öffnen Sie das Menü *Views and more Actions* (2).
4. Klicken Sie auf *Aus VSIX installieren...* (3).



Die Fertigstellung der Installation wird im VS Code (rechts unten) angezeigt.

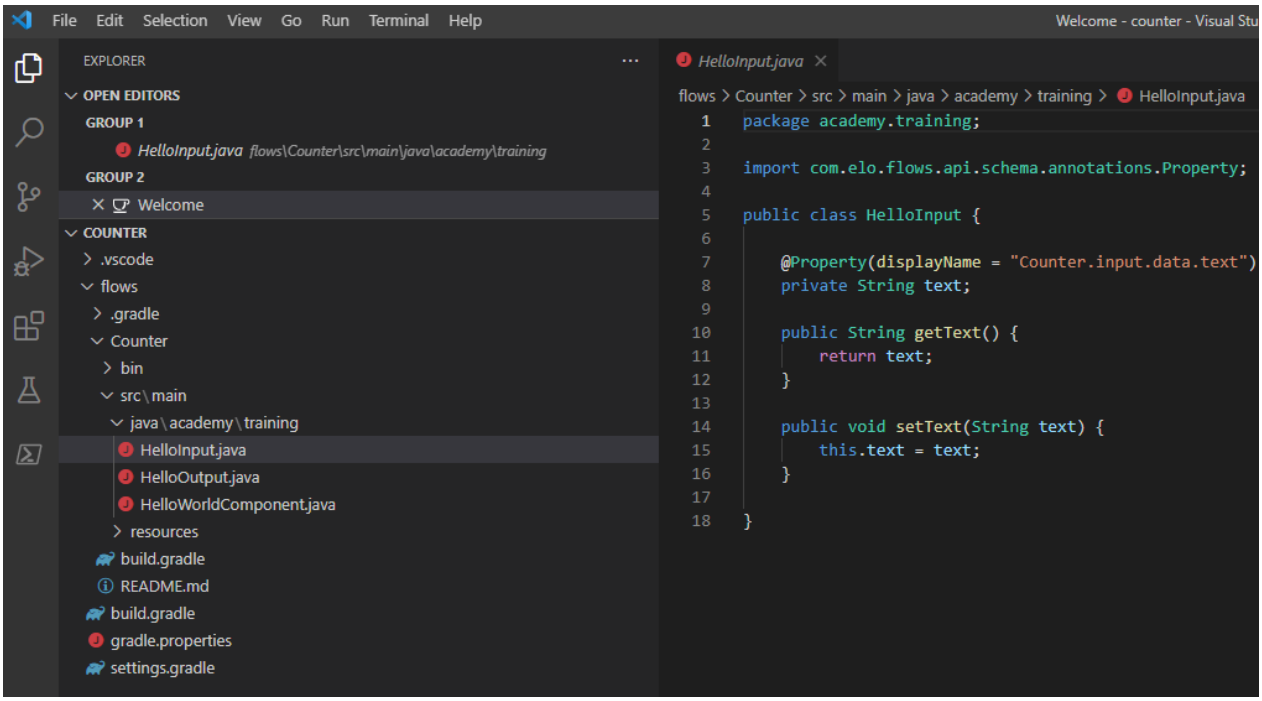


Weitere Informationen über das Framework findet man in der Erweiterungsübersicht.



Projektstruktur

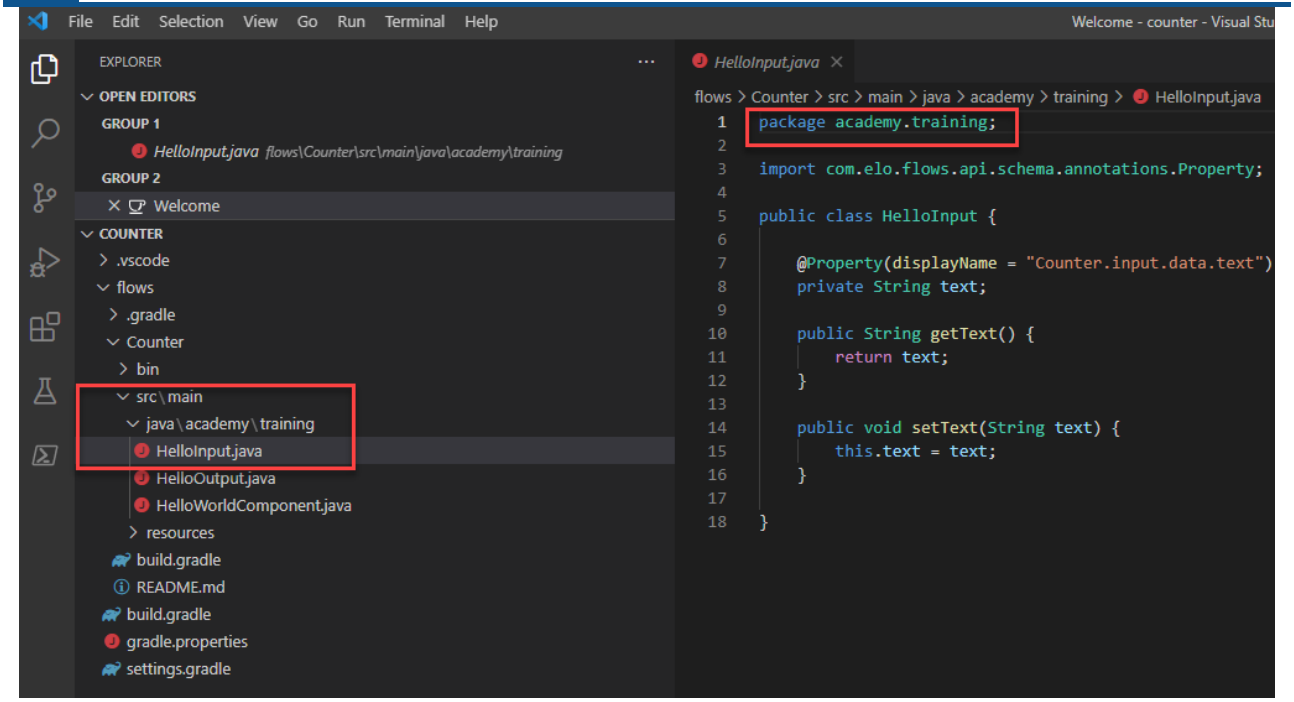
Das initiale Projekt im Framework besteht aus mehreren unterschiedlichen Bereichen (Verzeichnissen), die in jedem typischen Java-Projekt auch so zu finden wären. Bitte beachten Sie, dass in dem Entwicklungsprozess weiter Strukturbereiche nach und nach ergänzt werden. Dazu zählen Ihre Tests, zusätzliche Klassen oder die erstellte Bibliothek Ihres Projekts.



Source

Der Quellcode (Source) des Projekts befinden sich im Ordner *src\main*.

> **Information**
>
> Die Projekt-Ordner-Struktur entspricht nicht 1:1 der Paketierung im Quellcode. Das hängt von den Projekteinstellungen ab und kann später individuell konfiguriert werden.

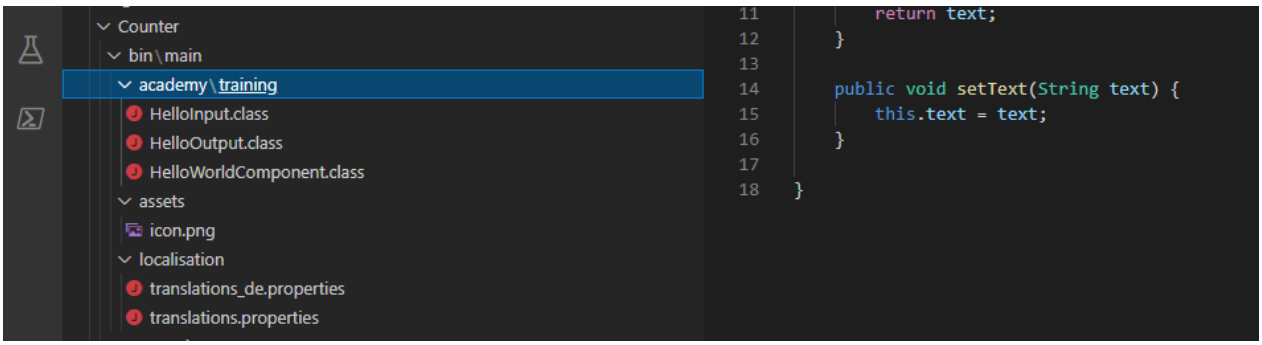


Resourcen

In dem Projektordner *resourcen* befinden sich z. B. die Lokalisierungsdateien.

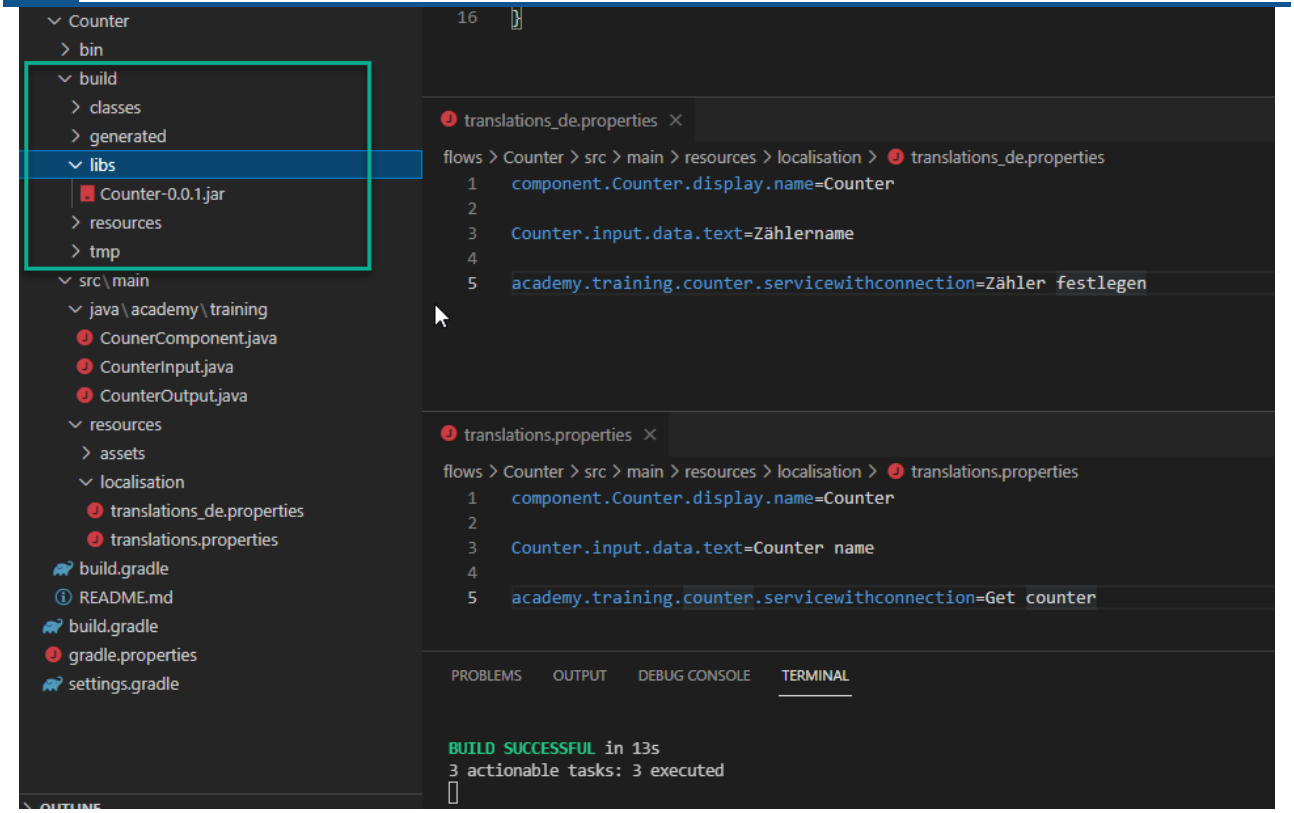
Binäries

Im *bin*-Projektordner werden die vorübersetzten Java-Klassen als auch Ressourcen abgelegt. Hieraus werden über den Build-Prozess die JAR-Dateien später gebaut.



Libs (während der Projektentwicklung)

Nach dem ersten Build-Prozess erscheint ein neuer Ordner mit der JAR-Bibliothek für den Deploy-Prozess.



Test (während der Projektentwicklung)

Den Testordner werden wir im Laufe des Projekts erstellen. Dies kann wie folgt aussehen.