Software-Praktikum

SS 2021, Blatt T5

Praktische Übung: 06.09.2021

Prof. Dr. Sven Apel

Aufgabe 1 Grundlagen

Im Folgenden finden Sie nützliche Tutorials, um das generelle Verständnis für das Programmieren in Java und das Zusammenwirken der benötigten Tools zu verstehen, sowie unsere Spezifikationen.

a) **Entwicklungsumgebung**

Wir empfehlen dringend die **IntelliJ IDEA Community Edition** von JetBrains als Entwicklungsumgebung zu nutzen, da sie alle für das SoPra benötigten Funktionalitäten einfach bereitstellt.

Aus diesem Grund bieten wir ausschließlich Support für IntelliJ an.

b) **Java Version**

- Sie brauchen für das diesjährige SoPra zwingend Java Version `16.0.2`.

  Tipp:
  Über die aktuellste Version von IntelliJ kann man sich die benötigte JDK[1] direkt herunterladen:

  `Tutorial: Define/Setup a Project SDK in IntelliJ`.

- Generell kann man über den Befehl `java -version` seine Java Version über die Kommandozeile abfragen.

  `Hier` findet ihr außerdem eine ausführliche Anleitung für Windows.

- Wenn man Java manuell installiert, muss man darauf achten die Umgebungsvariablen `JAVA_HOME` und `PATH` gegebenfalls geupdated oder gesetzt werden, damit der `java`-Befehl gefunden werden kann.

c) **Versionskontrolle mit Git**

Wir empfehlen die leicht verständlichen Tutorials von atlassian.com, um sich mit Git vertraut zu machen. Die meisten von Ihnen sollten in der Vorlesung Programmierung 2 schon mit Git gearbeitet haben.

Da die Gruppenarbeit im SoPra dieses Jahr teilweise online stattfinden muss, ist es wichtiger denn je, dass jeder nachvollziehbar gleichermaßen viel zur Entwicklung der Software beiträgt. Um dies zu überprüfen, werden die Git-Statistiken für jedes Gruppen-Repository während der Gruppenphase kontinuierlich und nach der Codeabnahme endgültig ausgewertet.

Hinweise:

- Achten Sie darauf, dass Sie Git mit Ihrer Uni-Mailadresse (...@stud.uni-saarland.de) konfiguriert haben, und auch nur über diesen Account in das Gruppen-Repo pushen, sodass Ihre Code Beiträge ersichtlich sind.
- Über die Git BASH kann man auch mit Windows komfortabel über die Kommandozeile arbeiten.
- Git ist als Plugin in IntelliJ integriert. Sie können entweder über die Kommandozeile oder direkt in IntelliJ mit Git arbeiten.
- Vermeiden Sie in jedem Fall das Austauschen von Dateien über E-Mail, Dropbox. Skype, Messenger Dienste etc. Es gibt nichts, was sich nicht mit Git lösen lässt. Verwalten Sie Ihr Repo **nicht** über Dropbox o.ä.!

  In der Vorlesung am Donnerstag (09.09.21) wird unter anderem das Arbeiten mit Branches und das Lösen von Merge-Konflikten (bzw. das Vermeiden des Entstehens solcher) behandelt.
- Um eine sichere Remote-Verbindung von Ihrem Computer zu dem SoPra-Gitlab herstellen zu können, müssen Sie einen SSH-Key in Ihrem Gitlab-Account hinterlegen.

  Sie finden diese Anleitung zum Generieren von SSH-Keys ebenso in Gitlab unter `Profile` → `Settings` → `SSH-Keys`.

  Lassen Sie die Felder für Dateinamen und Passwort leer, damit die erzeugten Dateien an den Default-Ort (`[user]/.ssh/`) auf Ihrem Rechner gelegt werden!

[1] `https://www.codejava.net/java-core/what-are-jvm-jre-and-jdk`

Aufgabe 2 Testprojekt

Sie werden vor Beginn der Gruppenphase ein eigenes Repository erhalten, in dem Sie ein kleines Testprojekt finden, dass sie klonen und importieren können. Dieses können Sie nutzen, um sich einerseits mit der IDE Ihrer Wahl und andererseits mit dem Interface von GitLab vertraut zu machen, da Sie beides in den nächsten Wochen durchgängig nutzen werden.

a) Installieren Sie IntelliJ, die richtige Java Version und hinterlegen Sie einen SSH-Key in ihrem SoPra-Gitlab Account wie in Aufgabe 1 beschrieben.

b) Klonen Sie das Testprojekt über SSH. Der Zugriff über HTTPS ist nicht erlaubt.

   Navigieren Sie dazu über die Kommandozeile in den Ordner, wo Sie das Testprojekt ablegen möchten und geben Sie den folgenden Befehl zum Klonen des Projektes ein:

   `git clone [link]`

   wobei Sie den Link auf der Projekt-Seite Ihres Testprojekts im GitLab bei Klick auf den blauen Button "Clone"finden. Achten Sie darauf, den SSH-Link zu benutzen.

c) Sie haben zwei Möglichkeiten das Projekt in IntelliJ zu öffnen:

   - 1. `Open` → Ordner auswählen, in den Sie das Projekt geklont haben.
   - 2. `Import` → Sie wählen nur die `build.gradle` Datei des Testprojektes aus.

   Danach öffnet sich ein Fenster mit Einstellungen 1.

   Gradle ist ein Build-Management-Automatisierungs-Tool wie Apache Ant oder Apache Maven. Wir benutzen einen Gradle Wrapper[2] für das Projekt. Daher wählen Sie bitte `Use gradle 'wrapper' task configuration`, damit das Projekt mit unserer vorgegebenen Gradle Konfiguration eingerichtet werden kann. Außerdem müssen Sie als `Gradle JVM` Java Version 16.0.2 auswählen.

   Der Gradle Wrapper sollte automatisch die richtige Gradle Version herunterladen, falls diese noch nicht installiert ist.

d) Überprüfen Sie die Einstellungen der `Project SDK` in den `Project Settings`, wie schon in Aufgabe 1 b) beschrieben.

e) In den allgemeinen Einstellungen von IntelliJ (`Preferences` → `Editor` → `Code Style` → `Java`) können Sie die Style Einstellungen anpassen. Beispielhaft können Sie die folgenden Einstellungen ändern, um den Auto-Formatter kompatibel zum Style-Check-Tool zu machen.

   (i) `Editor -> Code Style -> Java -> Code Generation` → bei "Final Modifier" alle Haken markieren

   (ii) `Editor -> Inspections -> Java -> Serialization issues`
   `-> Serializable class without 'serialVersionUID'` → Häkchen setzen und 'OK' drücken.

f) In der Übung werden wir Ihnen das Bauen, Ausführen, Schreiben von JUnit-Tests und Debuggen im Debug-Modus zeigen.

[2] `https://docs.gradle.org/current/userguide/gradle_wrapper.html`

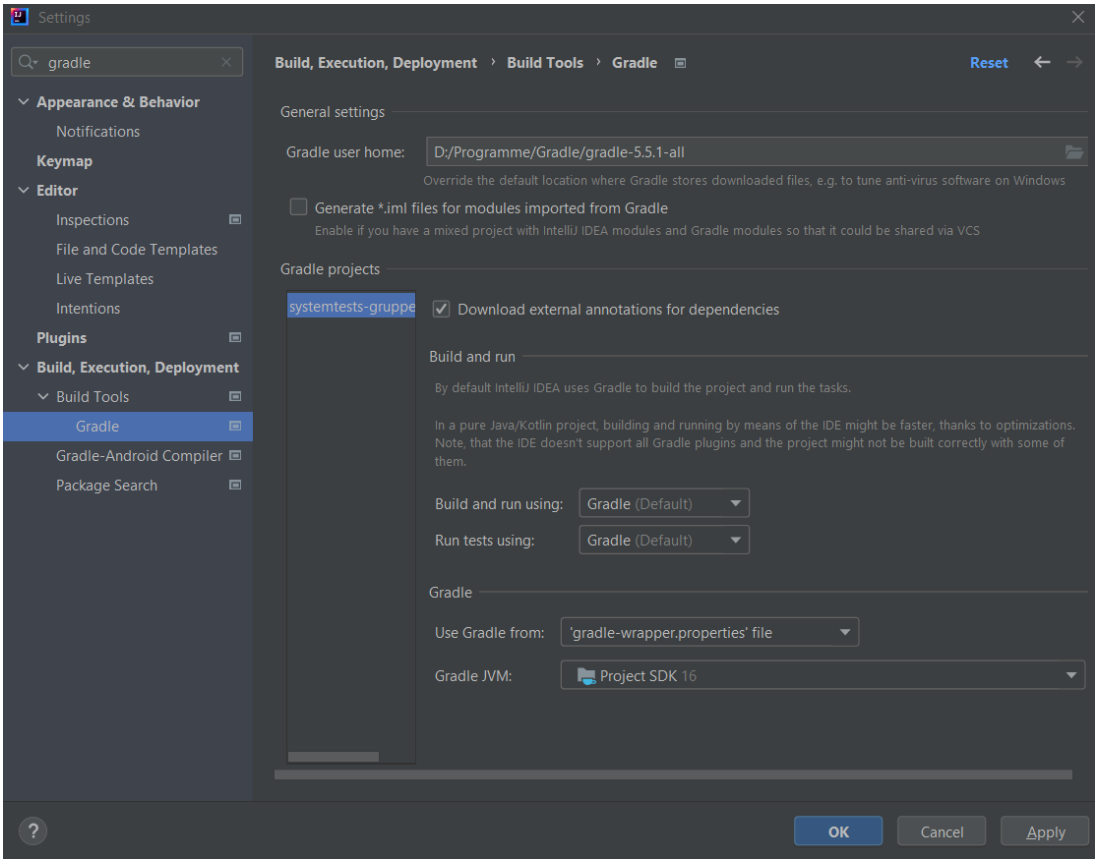

Abbildung 1: Import Options für ein SoPra-Gradle-Projekt