

# Syntaxerkenner mit ANTLR

## Vorbereitung

Studieren Sie Kapitel 4.1 des ANTLR4-Buches: „Matching an Arithmetic Expression Language“. Das Kapitel 5.4: „Dealing with Precedence, Left Recursion, and Associativity“ erläutert, wie in ANTLR4 die Priorität und Assoziativität eines Operators spezifiziert wird.

Allgemeine Erläuterungen zum Entwurf von Grammatiken mit ANTLR finden Sie im Kapitel 5.3: „Recognizing Common Language Patterns with ANTLR Grammars“ des ANTLR4-Buches.

## 1. Aufgabenstellung

Implementieren Sie einen Parser für die folgende Grammatik mit Hilfe von ANTLR. Der Parser soll die arithmetischen Ausdrücke nicht berechnen, sondern nur deren Syntax auf Korrektheit prüfen.

$$\begin{aligned} N &= \{\text{expr}\} \\ T &= \{\text{int}, +, -, *, /, (, )\} \\ P &= \left\{ \begin{array}{l} \text{expr} \rightarrow \text{int} \\ \text{expr} \rightarrow \text{expr} + \text{expr} \\ \text{expr} \rightarrow \text{expr} - \text{expr} \\ \text{expr} \rightarrow \text{expr} * \text{expr} \\ \text{expr} \rightarrow \text{expr} / \text{expr} \\ \text{expr} \rightarrow (\text{expr}) \end{array} \right\} \end{aligned}$$

## 2. Erweiterungen des Syntaxerkenners

### 2.1. Potenzoperator

Erweitern Sie den Syntaxerkenner um einen **rechtsassoziativen** Potenzoperator. Der Potenzoperator hat eine höhere Priorität als Addition/Subtraktion und Multiplikation/Division.

### 2.2. Vergleichsoperatoren

Erweitern sie den Syntaxerkenner um die nichtassoziativen Vergleichsoperatoren "<" und ">". Die Vergleichsoperatoren haben eine geringere Priorität als Addition/Subtraktion.

### 2.3. Zuweisungsoperator

Erweitern Sie den Syntaxerkenner um einen Zuweisungsoperator "=", so dass einfache Variablen verwendet werden können. Variablennamen sollen aus Groß- und Kleinbuchstaben bestehen können.