

## Expression-Lexer mit ANTLR4

### Vorbereitung

Studieren Sie das Kapitel 5.5 „Recognizing Common Lexical Structures“ des ANTLR4-Buches.

### Aufgabenstellung

Ersetzen Sie im Postfix-Interpreter aus dem Praktikum der Lehrveranstaltung Informatik II den dort vorgegebenen Expression-Lexer durch einen mit ANTLR4 generierten Lexer. Testen Sie den Postfix-Interpreter mit dem neuen Expression-Lexer!

### Verwendung des Lexers

Das folgende Programm demonstriert die Verwendung eines mit ANTLR4 entwickelten Expression-Lexers. Das Programm zerlegt die Eingabe in einen Tokenstrom. Für jedes Token werden Zeilennummer, Spaltennummer, Tokentyp-Code und Lexem ausgegeben. Die Tokentyp-Codes sind in der Klasse `ExprLexer` als ganzzahlige Konstanten deklariert.

```
import org.antlr.v4.runtime.*;
public class Test {
    public static void main(String[] args) throws Exception {
        CharStream input = null;
        // Pick an input stream (filename from commandline or stdin)
        if (args.length > 0) input = new ANTLRFileStream(args[0]);
        else input = new ANTLRInputStream(System.in);
        ExprLexer lex = new ExprLexer(input);
        Token t = lex.nextToken();
        while (t.getType() != Token.EOF) { //ExprLexer.EOF works as well
            System.out.printf("%2d:%2d Typ-Code: %2d Lexem: %s\n",
                t.getLine(),
                t.getCharPositionInLine(),
                t.getType(),
                t.getText());
            t = lex.nextToken();
        }
    }
}
```

Sie müssen die ANTLR-Bibliothek **antlr-4.5.2-complete.jar** in Ihr Projekt einbinden!

**Bewertung:** Ohne Erweiterungen sind mit einem Vortag max. 3 Punkte erreichbar.