

## Blatt 03 – A3.1: ANTLR-Grammatik

Modul: Compilerbau — Gruppe: Adrian Kramkowski, Abdelhadi Fares, Yousef Al Sahli, Abdelraoof Sahli

### A3.1 – Grammatik (Lexer + Parser)

```
grammar Mini;
```

```
start
: stmt+ EOF
;
```

```
stmt
: assign NL
| ifStmt
| whileStmt
;
```

```
assign
: ID ':=' expr
;
```

```
ifStmt
: 'if' expr 'do' NL stmt+ 'end' NL
| 'if' expr 'do' NL stmt+ 'else' 'do' NL stmt+ 'end' NL
;
```

```
whileStmt
: 'while' expr 'do' NL stmt+ 'end' NL
;
```

```
expr
: expr ("*" | "/") expr
| expr ("+" | "-") expr
| expr ("==" | "!=" | "<" | ">") expr
| '(' expr ')'
| ID
| INT
| STRING
```

```
;  
  
ID    : [a-zA-Z_][a-zA-Z_0-9]* ;  
INT   : [0-9]+ ;  
STRING : "" ( ~["\r\n])* "" ;  
NL    : ('\r'? '\n')+ ;  
WS    : [ \t]+ -> skip ;  
COMMENT : '#' ~[\r\n]* -> skip ;
```

Lexer- und Parser-Struktur nach *sheet03.md*, *antlr-lexing.md*, *antlr-parsing.md*.

Startregel `start : stmt+ EOF`, Vorrang über Alternativenreihenfolge.