

Für die folgenden Aufgaben gelten die Vorgaben: Mini-Java ist eine Untermenge von Java und umfasst (mindestens) *Klassen*, *Attribute*, *for*-, *while*-Schleifen, *if/else*-Anweisungen, Zuweisungen, Methodenaufrufe und *Ausdrücke* mit verschiedenen Operationen.

2.1 Mini-Java-Scanner I (Prüfungsaufgabe)

Schreiben Sie für eine sinnvolle Menge Tokens der Programmiersprache Java einen Scanner/Lexer direkt in Haskell.

2.2 Mini-Java-Scanner II (Prüfungsaufgabe)

Schreiben Sie mit Alex für einen Ausschnitt aus Java einen Scanner.

Dazu müssen Sie eine Lex-Spezifikation der Form erstellen:

```
{
...
}

%wrapper ...

tokens :-
  abstract {}
  boolean {}
  break {}
  case {}
  ...
  [1-9][0-9]*[L]?      { }
  0[xX][0-9a-fA-F]+[L]? { }

{
data Token
  = ABSTRACT
  | BOOLEAN
  | BREAK
  | CASE
  ...
  | INTLITERAL Integer
  | BOOLLITERAL Bool
  | JNULL
  | CHARLITERAL Char
  | STRINGLITERAL String
  | IDENTIFIER String
  ...
}
```

Sei folgende Grammatik für einen Ausschnitt der Java-Statements gegeben:

statement	→	ifthenstatement ifthenelsestatement whilestatement block emptystatement returnstatement
ifthenstatement	→	IF LBRACE bexpression RBRACE statement
ifthenelsestatement	→	IF LBRACE bexpression RBRACE statement ELSE statement
whilestatement	→	WHILE LBRACE bexpression RBRACE statement
block	→	LBRACKET RBRACKET LBRACKET statements RBRACKET
statements	→	statement statements statement
emptystatement	→	SEMICOLON
returnstatement	→	RETURN SEMICOLON RETURN expression SEMICOLON
bexpression	→	TRUE
expression	→	1

2.3 Java-Statement-LL-Parser (Prüfungsaufgabe)

Schreiben Sie einen LL-Parser für die gegebene Menge von Java-Statements. Nutzen Sie den Scanner aus Aufgabe 2.1 oder 2.2.