

3.1 Abstrakte Syntax für Java-Statements

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

statement	→	ifthenstatement ifthenelsestatement whilestatement block emptystatement returnstatement
ifthenstatement	→	IF LBACE bexpression RBACE statement
ifthenelsestatement	→	IF LBACE bexpression RBACE statement ELSE statement
whilestatement	→	WHILE LBACE bexpression RBACE statement
block	→	LBACKET RBACKET LBACKET statements RBACKET
statements	→	statement statements statement
emptystatement	→	SEMICOLON
returnstatement	→	RETURN SEMICOLON RETURN expression SEMICOLON
bexpression	→	TRUE
expression	→	1

Definieren Sie eine abstrakte Syntax für die Menge von Java-Statements

3.2 Java-Statement-Kombinatorparser (Prüfungsleistung, 10 Punkte)

Schreiben Sie einen Kombinatorparser für die Menge Java-Statements, der die Eingabe in abstrakte Syntax umwandelt. Nutzen Sie den Scanner aus Aufgabe 2.1 oder 2.2.

3.3 Happy-Parser (Prüfungsleistung, 10 Punkte)

Schreiben Sie einen Happy-Parser für die Menge Java-Statements, der die Eingabe in abstrakte Syntax umwandelt. Nutzen Sie den Scanner aus Aufgabe 2.1 oder 2.2.