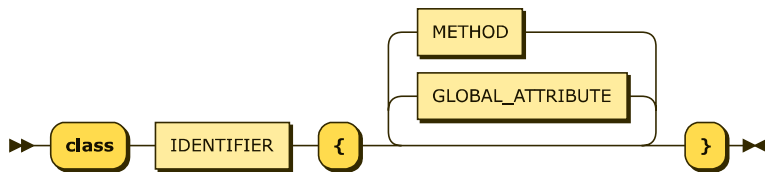
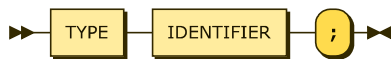


PROGRAM:

PROGRAM ::= 'class' IDENTIFIER '{' (GLOBAL_ATTRIBUTE | METHOD)* '}'

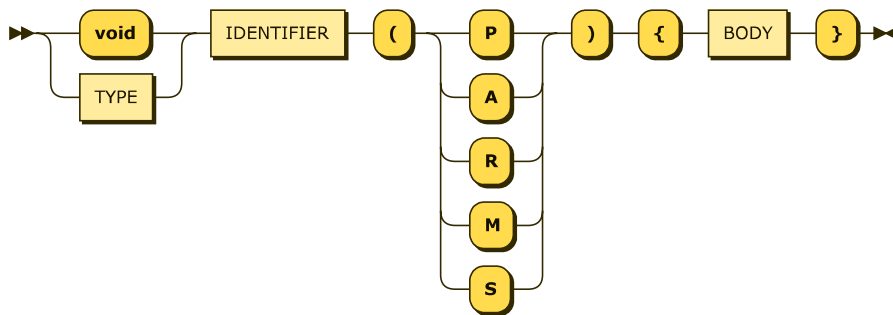
no references

GLOBAL_ATTRIBUTE:

GLOBAL_ATTRIBUTE
::= TYPE IDENTIFIER ';'

referenced by:

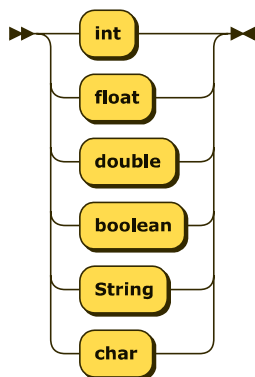
- [PROGRAM](#)

METHOD:

METHOD ::= ('void' | TYPE) IDENTIFIER '(' [PARMS] ')' '{' BODY '}'

referenced by:

- [PROGRAM](#)

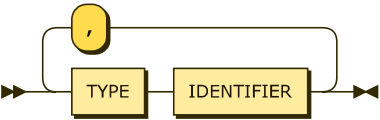
TYPE:

TYPE
::= 'int'
| 'float'
| 'double'
| 'boolean'
| 'String'
| 'char'

referenced by:

- GLOBAL_ATTRIBUTE
- METHOD
- PARAMS
- VARIABLE

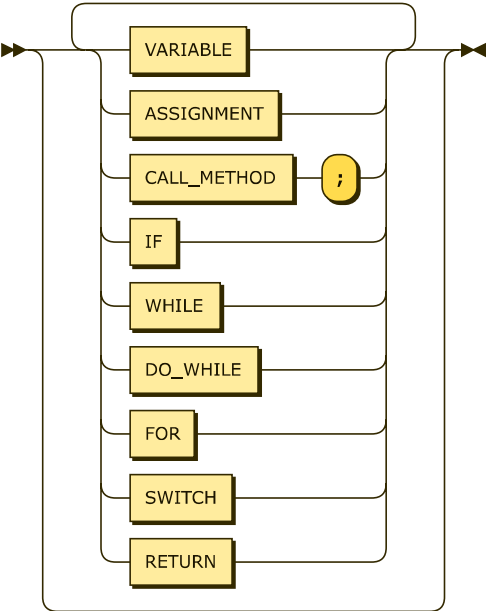
PARAMS:



PARAMS ::= TYPE IDENTIFIER (',' TYPE IDENTIFIER)*

no references

BODY:

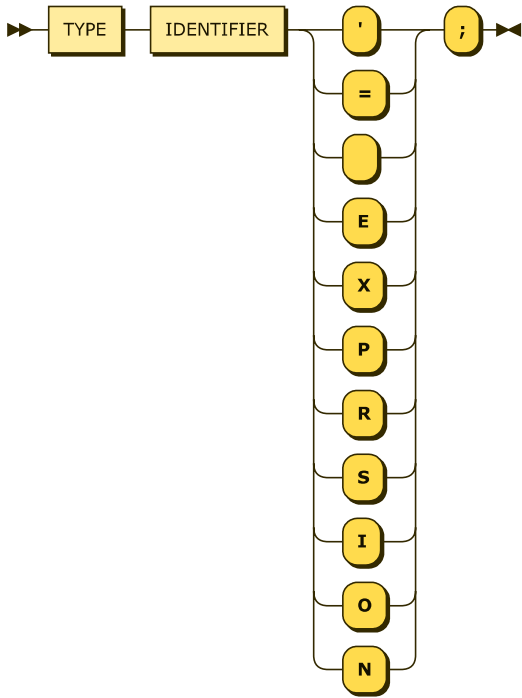


BODY ::= (VARIABLE | ASSIGNMENT | CALL_METHOD ';' | IF | WHILE | DO_WHILE | FOR | SWITCH | RETURN)*

referenced by:

- DO_WHILE
- FOR
- IF
- METHOD
- WHILE

VARIABLE:

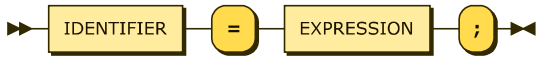


VARIABLE ::= TYPE IDENTIFIER ['=' EXPRESSION] ';' ;

referenced by:

- BODY
- STATEMENT

ASSIGNMENT:

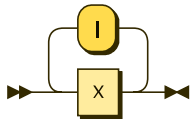


ASSIGNMENT ::= IDENTIFIER '=' EXPRESSION ';' ;

referenced by:

- BODY
- STATEMENT

EXPRESSION:

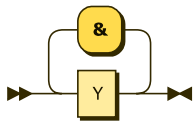


EXPRESSION ::= X ('I' X)*

referenced by:

- ASSIGNMENT
- C
- DO WHILE
- IF
- PARAM VALUES
- SWITCH
- WHILE

X:

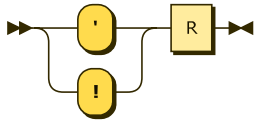


X ::= Y ('&' Y)*

referenced by:

- EXPRESSION

Y:

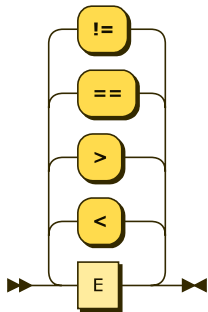


Y ::= ['!'] R

referenced by:

- X

R:

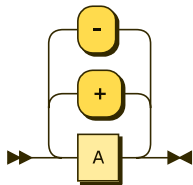


R ::= E (('<' | '>' | '==' | '!=') E)*

referenced by:

- Y

E:

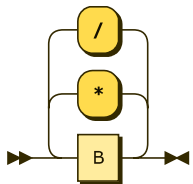


E ::= A (('+' | '-') A)*

referenced by:

- R

A:



A ::= B (('*' | '/') B)*

referenced by:

- E

B:

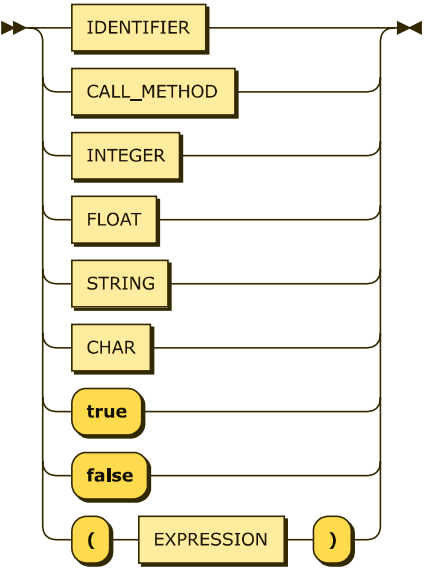


B ::= ['-'] C

referenced by:

- A

C:

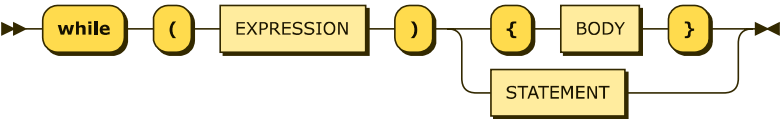


C ::= IDENTIFIER
| CALL_METHOD
| INTEGER
| FLOAT
| STRING
| CHAR
| 'true'
| 'false'
| '(' EXPRESSION ')'

referenced by:

- B

WHILE:

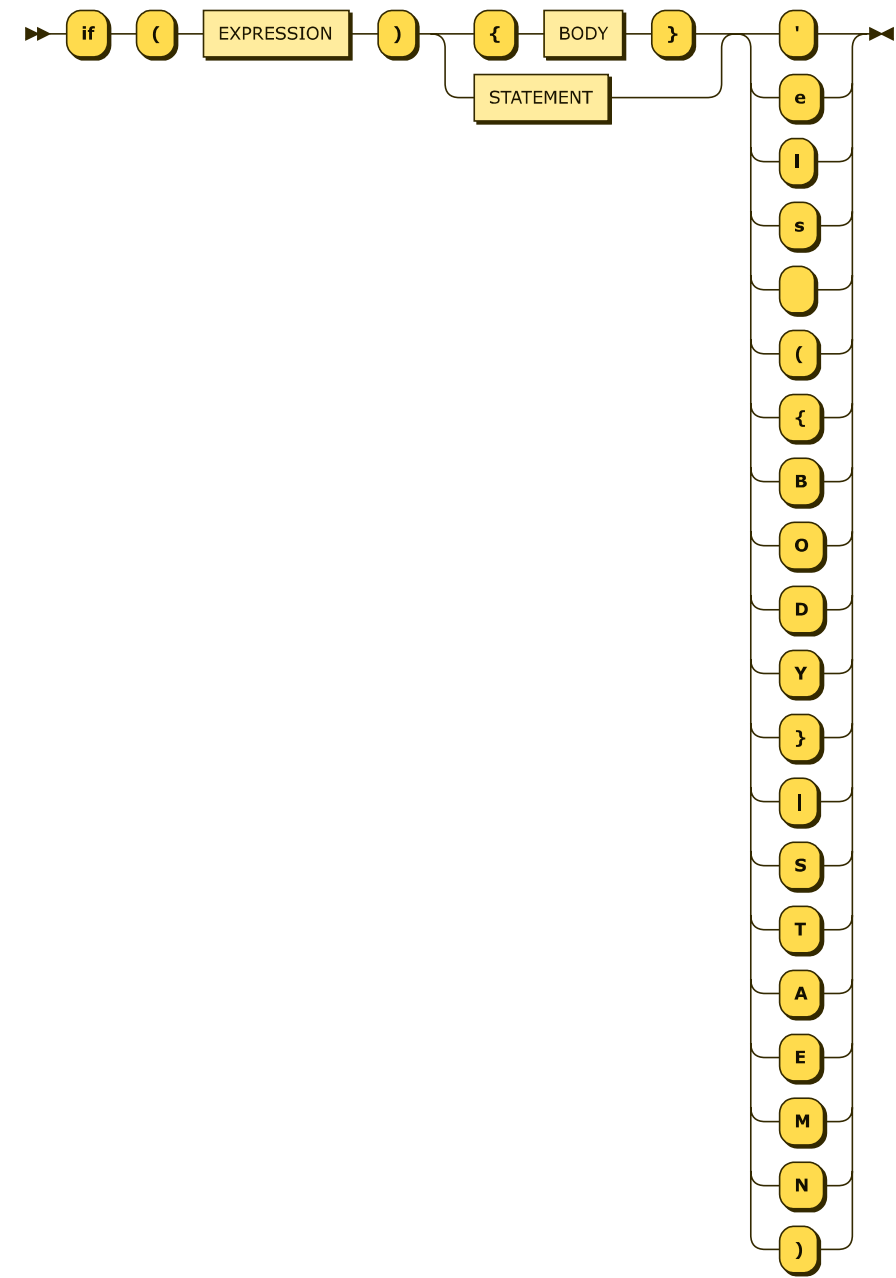


WHILE ::= 'while' '(' EXPRESSION ')' ('{' BODY '}' | STATEMENT)

referenced by:

- BODY
- STATEMENT

IF:

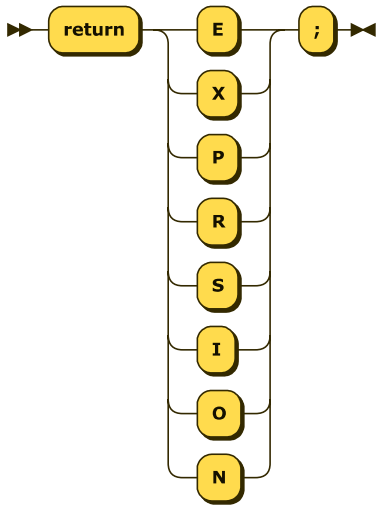


IF ::= 'if' '(' EXPRESSION ')' ('{' BODY '}' | STATEMENT) ['els' ({BODY}|STAEMN)]

referenced by:

- BODY
- STATEMENT

RETURN:

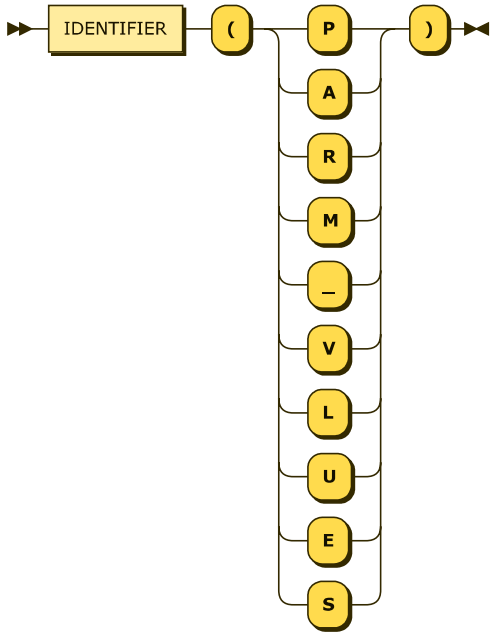


RETURN ::= 'return' [EXPRSION] ';' ;

referenced by:

- BODY
- STATEMENT

CALL_METHOD:

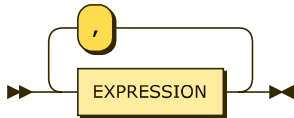


CALL_METHOD
::= IDENTIFIER '(' [PARAM_VLUES] ')' ;

referenced by:

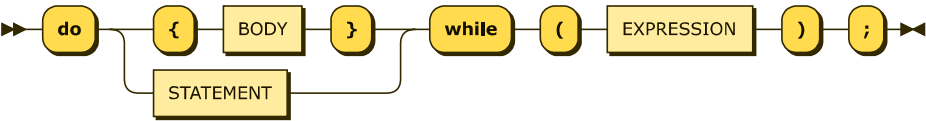
- BODY
- C
- STATEMENT

PARAM_VALUES:



PARAM_VALUES
::= EXPRESSION (',' EXPRESSION)* ;

DO_WHILE:

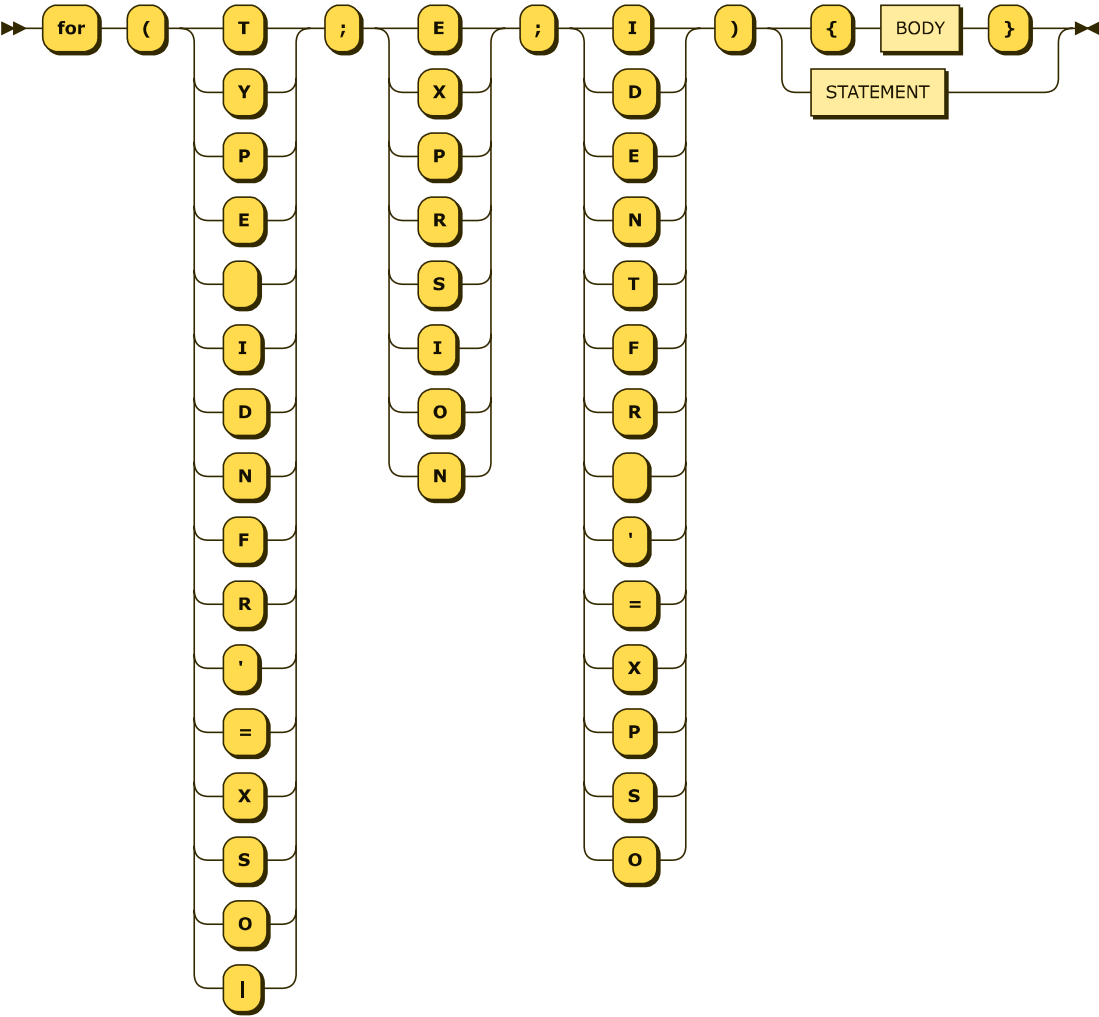


DO_WHILE ::= 'do' ('{' BODY '}' | STATEMENT) 'while' '(' EXPRESSION ')' ';' ;

referenced by:

- BODY
- STATEMENT

FOR:

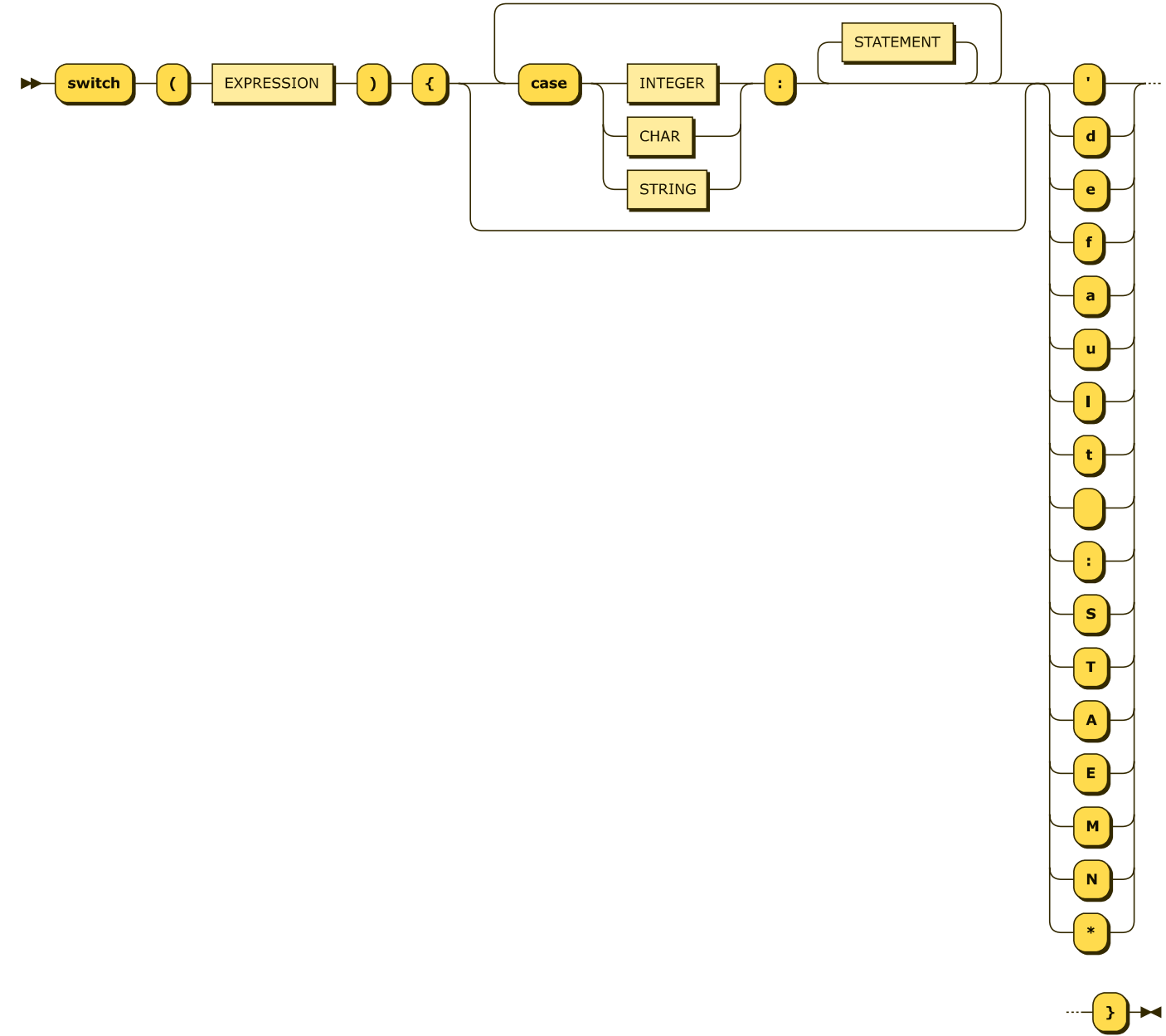


FOR ::= 'for' '(' [TYPE IDENTFR'=XS0|] ';' [EXPRSION] ';' [IDENTFR '=XPS0|] ')' ('{' BODY '}' | STATEMENT)

referenced by:

- BODY
- STATEMENT

SWITCH:

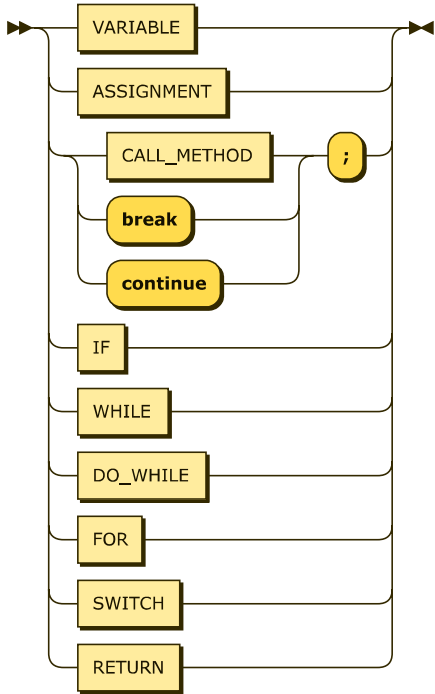


SWITCH ::= 'switch' '(' EXPRESSION ')' '{' ('case' (INTEGER | CHAR | STRING) ':' STATEMENT*)* ['default :STAEMN*'] '}'

referenced by:

- BODY
- STATEMENT

STATEMENT:



```
STATEMENT ::= VARIABLE
           | ASSIGNMENT
           | ( CALL_METHOD | 'break' | 'continue' ) ';'
           | IF
           | WHILE
           | DO_WHILE
           | FOR
           | SWITCH
           | RETURN
```

referenced by:

- DO_WHILE
- FOR
- IF
- SWITCH
- WHILE