

## Aufgabe 7: Skriptsprache für boolesche Ausdrücke

Folgende Grammatik beschreibt eine kleine Sprache, mit der boolesche Ausdrücke berechnet, in Variablen gespeichert und ausgegeben werden können:

```
Program  = { Statement }.
Statement = ident "=" Expr ";"
          | "print" Expr ";"
Expr     = Term { "||" Term }.
Term     = Factor { "&&" Factor }.
Factor   = "true" | "false" | ident | "(" Expr ")" | "!" Factor.
```

Attributieren Sie diese Grammatik, sodass der Wert der booleschen Ausdrücke berechnet und in Variablen abgespeichert oder ausgegeben werden kann. Verwalten Sie die verwendeten Variablen und ihren Wert in einer Hashtabelle (HashMap). Ein Beispielprogramm könnte wie folgt aussehen:

```
big = true;
small = ! big;
ready = false;
print (big || small) && ready || big && ! ready;
```

### Lösung

Zur Verwaltung der Namen und ihrer Werte verwenden wir eine globale HashTabelle:

```
HashMap<String, Boolean> values = new HashMap<String, Boolean>();
```

Die attributierte Grammatik sieht dann wie folgt aus:

```
Program = { Statement }.

Statement          (. boolean val; .)
= ident              (. String name = t.val; .)
  "="
  Expr <↑val>        (. values.put(name, val); .)
  ";"
| "print"
  Expr <↑val>        (. System.out.println(val); .)
  ";"

Expr <↑val>        (. boolean val, val2; .)
= Term <↑val>
  { "||" Term <↑val2> (. val = val || val2; .)
  }.

Term <↑val>        (. boolean val, val2; .)
= Factor <↑val>
  { "&&" Factor <↑val2> (. val = val && val2; .)
  }.

Factor <↑val>      (. boolean val; .)
= "true"             (. val = true; .)
| "false"            (. val = false; .)
| ident              (. Boolean c = values.get(t.val);
                     if (c != null) val = c;
                     else { error("undeclared name"); val = false; } .)
| "(" Expr <↑val> ")"
| "!" Factor <↑val> (. val = ! val; .)
```