

# MicroJava – Grammatik

```
Program      = "program" ident { ConstDecl | VarDecl | ClassDecl }
              "{" {MethodDecl} "}".

ConstDecl   = "final" Type ident "=" ( number | charConst ) ";".

VarDecl     = Type ident { "," ident } ";".

ClassDecl   = "class" ident "{" { VarDecl } }".

MethodDecl  = ( Type | "void" ) ident "(" [ FormPars ] ")"
              { VarDecl } Block.

FormPars    = Type ident { "," Type ident }.

Type        = ident [ "[" "]" ].

Block       = "{" { Statement } }".

Statement   = Designator ( Assignop Expr | ActPars | "++" | "--" ) ";"
              | "if" "(" Condition ")" Statement [ "else" Statement ]
              | "while" "(" Condition ")" Statement
              | "break" ";"
              | "return" [ Expr ] ";"
              | "read" "(" Designator ")" ";"
              | "print" "(" Expr [ "," number ] ")" ";"
              | Block
              | ";".

Assignop    = "=" | "+=" | "-=" | "*=" | "/=" | "%=".

ActPars     = "(" [ Expr { "," Expr } ] ")".

Condition   = CondTerm { "||" CondTerm }.

CondTerm    = CondFact { "&&" CondFact }.

CondFact    = Expr Relop Expr.

Relop       = "==" | "!=" | ">" | ">=" | "<" | "<=".

Expr        = [ "-" ] Term { Addop Term }.

Term        = Factor { Mulop Factor }.

Factor      = Designator [ ActPars ]
              | number
              | charConst
              | "new" ident [ "[" Expr "]" ]
              | "(" Expr ")".

Designator  = ident { "." ident | "[" Expr "]" }.

Addop       = "+" | "-".

Mulop       = "*" | "/" | "%".
```