

MicroJava – Grammatik

| | |
|------------|--|
| Program | = "class" 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 } BlockStat . |
| FormPars | = Type ident { "," Type ident } . |
| Type | = ident ["[" "]"] . |
| Statement | = Designator ("=" Expr "(" [ActPars] ")" "++" "--") ";" "if" "(" Condition ")" Statement ["else" Statement] "while" "(" Condition ")" Statement "do" Statement "while" "(" Condition ")" ";" "continue" ";" "return" [Expr] ";" "read" "(" Designator ")" ";" "print" "(" Expr ["," number] ")" ";" BlockStat ";" . |
| BlockStat | = "{" { Statement } "}" . |
| ActPars | = Expr { "," Expr } . |
| Condition | = CondTerm { " " CondTerm } . |
| CondTerm | = CondFact { "&&" CondFact } . |
| CondFact | = Expr Relop Expr . |
| Expr | = ["-"] Term { Addop Term } . |
| Term | = Factor { Mulop Factor } . |
| Factor | = Designator ["(" [ActPars] ")"] number charConst "new" ident ["[" Expr "]"] "(" Expr ")" . |
| Designator | = ident { "." ident "[" Expr "]" } . |
| Relop | = "==" "!=" ">" ">=" "<" "<=" . |
| Addop | = "+" "-" . |
| Mulop | = "*" "/" "%" . |