

Vereinfachte Java-Grammatik

1 Ausdrücke

PrimExp: Lit|Var|IncDecExp|(Exp)|MethodInvoc
Var: Id {[Exp]}
Id: [Id.] Name
IncDecExp: Var IncDec | IncDec Var
IncDec: ++|--
Exp: UnaryExp | BinExp | CondExp | AssignExp | CreationExp
UnaryExp: PrimExp | UnaryOp UnaryExp
UnaryOp: + | - | ! | ... | Cast
Cast: (PrimType)
BinExp: Exp BinOp Exp
BinOp: + | - | * | / | && | || | ... | < | > | != | ==
CondExp: Exp ? Exp : Exp
AssignExp: Var AssignOp Exp
AssignOp: = | += | -= | *= | /= | ...
MethodInvoc: Method ([ActualArgs])
Method: Id
ActualArgs: Exp {,Exp}
CreationExp: new ClassId([ActualArgs]) | ArrayCreationExp
ArrayCreationExp: new Id [Exp] {[Exp]} {[]}
PrimType: Name

2 Anweisungen

Stmt: SimpleStmt | CompStmt | Label: Stmt
SimpleStmt: ;| StmtExp; | **break** [Label]; | **continue** [Label]; |
 return [Exp]; |**throw** Exp;
StmtExp: AssignExp | IncDecExp | MethodInvoc | CreationExp

Label: Name
 CompStmt: Block | CaseStmt | RepStmt | TryBlock
 Block: {{VarDec | Stmt}}
 VarDec: [final] Type VDec;
 VDec: Name [= Exp]{, Name [= Exp]}
 Type: PrimType | ClassType | ArrayType
 PrimType: Name
 ClassType: Id
 ArrayType: Id [] {[]}
 CaseStmt: CondStmt | SwitchStmt
 RepStmt: WhileStmt | DoStmt | ForStmt
 CondStmt: if (Exp) Stmt [else Stmt]
 SwitchStmt: switch (Exp) SwitchBlock
 SwitchBlock: {{CaseGroup}}
 CaseGroup: SwitchLabel {SwitchLabel} Stmt {Stmt}
 SwitchLabel: case Exp: | default:
 WhileStmt: while (Exp) Stmt
 DoStmt: do Stmt while (Exp);
 ForStmt: for ([Init]; [Exp]; [Update]) Stmt
 Init: StmtExp {, StmtExp } | VarDec
 Update: StmtExp {, StmtExp }
 TryBlock: try Block CatchClause {CatchClause} [finally Block]
 try Block finally Block
 CatchClause: catch (Exception Name) Block
 Exception: Id

3 Methoden

MethodDec: MethodHeader MethodBody
 MethodHeader: {Modifier} ResultType MDec [ThrowsClause]
 Modifier: public | static | final | ...
 ResultType: Type | void

MDec: ID ([FormalArg {,Formal Arg}])
FormalArg: Type Name
ThrowsClause: Throws Exception {,Exception}
MethodBody: ; | Block

4 Klassen

ClassDec: [public] class Name ClassBody
ClassBody: {{{Modifier} Dec}}
Dec: VarDec | MethodDec | ClassDec | ConstructorDec | Initializer
ConstructorDec: [public] Name ([FormalArg] {,FormalArg}) [ThrowsClause] Block
Initializer: [static] Block

5 Paket

CompUnit: [PackageHeader] {Import} PackageBody
PackageHeader: package Identifier;
Import: import ClassType; | import Package . * ;
Package: Id
PackageBody: {ClassDec}