Compiler Design Mini Project

Grammar for Java Compiler

CompilationUnit:

ImportDec CompilationUnit|ClassDec|**ε**

ImportDec:

import Identifier.Identifier.\*;

ClassDec:

Modifier class Identifier {ClassBody}

ClassBody:

GlobVarDec ClassBody|MethodDec ClassBody|**ε**

GlobVarDec:

Modifier Type DecIdenList;

MethodDec:

Modifier Type Identifier(Params)Block

Modifier:

public Modifier1|private Modifier1|protected Modifier1|Modifier1

Modifier1: static Modifier2| Modifier2

Modifier2: final | **ε**

Params:

ParamList|**ε**

ParamList:

Type Identifier| Type Identifier, ParamList

Block:

{Statement}

Statement: Assign Statement| break; Statement| continue; Statement| while(Expression)Block| while(Expression)Statement |return Expression; Statement| switch(Expression) {SwitchBlock} Statement| VarDec Statement| ArrDec Statement| ArrInit Statement **| ε**

SwitchBlock:

SwitchLabel: Statement SwitchBlock| **ε**

SwitchLabel:

case (Expression)| default

VarDec:

Type Identifier = Expression DecIdenList;| Type Identifier DecIdenList;

DecIdenList:

,Identifier = Expression DecIdenList| ,Identifier DecIdenList| **ε**

ArrDec:

int Identifier[] ArrList;| Identifier[][] ArrList; | Identifier ArrList;

ArrList;

,Identifier[]|,Identifier[][]|,Identifier| **ε**

ArrInit:

int Indentifier [] = new int[Expression];|int Indentifier [][] = new int[Expression][Expression];|

int[] Indentifier = new int[Expression];|int[][] Indentifier = new int[Expression][Expression];|

Identifier = new int[Expression];| Identifier = new int[Expression][Expression];

ArrExp:

Identifier| Identifier[Expression]| Identifier[Expression][Expression]

Type:

int| double| char

Assign:

ArrExp AssignOp Expression;

AssignOp:

=| +=| -=| \*=| /=| &=| |=| ^=| %= | **ε**

InfixOp:

|| | && | | | ^ | & | == |!= | < | >| <=| >=| <<| >>| +| -| \*| /| % | **ε**

PrefixOp:

++|--|!|~|+|-| **ε**

PostfixOp:

++|--| **ε**

Expression:

PrefixOp Expression PostfixOp InfixOp PrefixOp Expression PostfixOp; | (Expression) | ArrExp| NUM| **ε**