

Parser Grammar for Java while and if-else constructs

Team Members :

Mehul Garg 01FB16ECS206

Naman Bansal 01FB16ECS220

P Abhishikta Sai 01FB16ECS242

T_ABC : This indicates the symbol is non-terminal.

T_ABC? : This indicates the non-terminal can be present zero or more number of times

Names

T_Name > -> T_SimpleName | T_QualifiedName

T_SimpleName -> Id

T_QualifiedName -> T_Name . Id

T_Literal -> IntLit | FloatLit | BoolLit | CharLit | StringLit | NullLit

Packages

S -> T_PackageDecl? T_ImportDecl? T_TypeDecl

T_PackageDecl -> package T_Name ;

T_ImportDecl -> import T_Name; | import T_name . *

T_TypeDecl -> ; | T_Class

Types

T_Type -> T_PrimType

T_PrimType -> T_NumType > | boolean

T_NumType > -> byte | short | int | long | char | float | double

T_ArrayType -> T_PrimType [] | T_Name [] | T_ArrayType []

Modifiers

T_Modifiers -> T_Modifier | T_Modifiers T_Modifier

T_Modifier > -> public | private | protected | static | abstract | final | native

Classes & Methods

T_ClassDecl -> T_Modifiers class Id T_ClassBody

T_ClassBody -> {T_Main T_ClassBodyDeclList?}

T_Main -> T_Modifiers? T_Type main (T_FormalParmList?) T_Block | T_Modifiers? void main (T_FormalParmList?) T_Block

T_ClassBodyDeclList -> T_MethodDecl | static T_Block

T_MethodDecl -> T_MethodHdr T_Block

T_MethodHdr -> T_Modifiers? T_Type Id (T_FormalParmList?) | T_Modifiers? void Id (T_FormalParmList?) T_Block

T_FormalParmList > -> T_FormalParam | T_FormalParmList , T_FormalParam

T_FormalParam -> T_Type T_VarDeclId

T_VarDeclId -> Id | T_VarDeclId []

Blocks

T_Block -> {T_BlockStmt | T_BlockStmt T_BlockStmt?}

T_BlockStmt -> T_LocalVar | T_Stmt
 T_LocalVar -> T_Type T_VarDeclId ; | T_Type T_VarDeclId , T_VarDeclId? ;
 T_Stmt -> T_StmtOther | T_If | T_IfElse | T_While
 T_StmtOther -> T_Block | T_ExprStmt | T_BreakStmt | T_RetStmt | ;
 T_ExprStmt -> T_Assign | T_DecExpr | T_IncExpr
T_If -> if (T_Expr) T_Stmt
T_IfElse -> if (T_Expr) T_Stmt else T_Stmt
T_While -> while (T_Expr) T_Stmt
 T_BreakStmt -> break Id? ;
 T_RetStmt -> return T_Expr? ;

Expressions

T_Expr -> T_Assign | T_CondExpr | T_DecExpr | T_IncExpr | T_Primary
 T_Assign -> T_Name T_AssignOp T_CondExpr | T_Name T_AssignOp T_Primary
 T_CondExpr -> T_RelatExpr | T_ArithExpr | T_OtherExpr
 T_Relat -> T_Expr T_Relop T_Expr
 T_ArithExpr -> T_Expr T_Arithop T_Expr
 T_OtherExpr -> T_Expr T_Otherop T_Expr
 T_DecExpr -> --T_Expr | T_Expr--
 T_IncExpr -> ++T_Expr | T_Expr++
 T_Primary -> T_Literal | Id | (T_Expr)

Operators

T_AssignOp -> = | *= | /= | %= | += | -=
 T_Relop -> == | < | > | != | <= | >=
 T_Arithop -> + | - | * | / | %
 T_Otherop -> && | || | & | ^