– Program –

<Program> ::= <PackageClause> <ImportDecl> <Declaration>

– Package Clause –

<PackageClause> ::= "package" <PackageName>

<PackageName> ::= <Identifier>

– Import Declaration –

<ImportDecl> ::= "import" <ImportSpec>

<ImportSpec> ::= <PackageName> <ImportPath>

<ImportPath> ::= <String>

– Declarations –

<Declaration> ::= <VarDec>l | <FunctionDecl>

<VarDecl> ::= "var" <VarSpec>

<VarSpec> ::= <IdentifierList> <Type> | <IdentifierList> "=" ExpressionList

– Function Declaration –

<FunctionDecl> ::= "func" <FunctionName> "(" <ParameterList> ")" <Signature> <FunctionBody>

<FunctionName> ::= <Identifier>

<ParameterList> ::= <ParameterDecl> | <ParameterDecl>”,”<ParameterList>

<ParameterDecl> ::= <IdentifierList> <Type>

<FunctionBody> ::= <Block>

– Statements –

<Statement> ::= <AssignStmt> | <IOStmt> | <IfStmt> | <ForStmt>

<AssignStmt> ::= <Identifier> ":=" <Expression>

<IOStmt> ::= "fmt.Scan" "(" <Identifier> ")"

| "fmt.Print" "(" <Expression> ")"

| "fmt.Printf" "(" <ExpressionList> ")"

| "fmt.Println" "(" <ExpressionList> ")"

<IfStmt> ::= "if" <SimpleStmt> <Block>

<ForStmt> ::= "for" <Condition> <Block>

<Condition> ::=<Expression>

– Blocks –

<Block> ::= "{" <StatementList> "}"

<StatementList> ::= <Statement> | <Statement><StatementList>

– Expressions –

<ExpressionList> ::= <Expression> | <Expression> "," <ExpressionList>

<Expression> ::= <UnaryExpr> | <Expression> <BinaryOp> <Expression>

<UnaryExpr> ::= <PrimaryExpr> | <UnaryOp> <UnaryExpr>

<PrimaryExpr> ::= <Identifier> | <Constant> | "(" <Expression> ")"

<BinaryOp> ::= "||" | "&&" | <RelOp> | <AddOp> | <MulOp>

<RelOp> ::= "==" | "!=" | "<" | "<=" | ">" | ">="

<AddOp> ::= "+" | "-"

<MulOp> ::= "*" | "/"

<UnaryOp> ::= "+" | "-" | "!" | "&"

– Types –

<Type> ::= "bool" | "int" | "float64" | "char" | <ArrayType> | <StructType>

<ArrayType> ::= "[" <ArrayLength> "]" <Type>

<ArrayLength> ::= <Digit>|<Digit><ArrayLength>

<StructType> ::= "struct" "{" FieldDecl "}"

<FieldDecl> ::= <IdentifierList> <Type>

– Identifiers and Constants –

<IdentifierList> ::= <Identifier> | <Identifier> "," <IdentifierList>

<Identifier> ::= <Letter>| <Letter><Digit>|<Identifier>

<Constant> ::= <IntConst> | <CharConst> | <BoolConst> | <String>

<IntConst> ::= "+" <Integer> | "-" <Integer> | <Integer>

<Integer> ::= <Digit> | <Digit><Integer>

<CharConst> ::= "'" <Character> "'"

<BoolConst> ::= "true" | "false"

<String> ::= "”" <Character>| <Character><String> "”"

– Base Symbols –

<Digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<Letter> ::= "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"

<Character> ::= <Letter> | <Digit> | <Symbol>

<Symbol> ::= "+" | "-" | "*" | "/" | ... | ";"

__Set of Non-terminals (N)__

Character, String, CharConst, FloatConst, IntConst, Constant, Identifier, IdentifierList, FieldDecl, StructType, ArrayLength, ExpressionList, Expression, UnaryExpr, PrimaryExpr, BinaryOp, Declaration, VarDecl, VarSpec, FunctionDecl, FunctionName, ParameterList, ParameterDecl, FunctionBody, Statement, AssignStmt, IOStmt, IfStmt, ForStmt, Block, StatementList, Program, PackageClause, PackageName, ImportDecl, ImportSpec, ImportPath, Symbol, Letter, Digit, BoolConst, TypeRelOp, AddOp, MulOp, UnaryOp

__Set of Terminals (Σ)__

"+" | "-" | "*" | "/" | ... | ";"| "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"| "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"| "true" | "false"| "struct"| “(” | “)” | “[” | “]”| “{” | “}” | "bool" | "int" | "float64" | "char" | "==" | "!=" | "<" | "<=" | ">" | ">=" | "||" | "&&" | fmt.Scan" | "fmt.Print" | "fmt.Printf" | "fmt.Println" | "if" |"for"| “:=” | "func" | “var” | “import” | “package”

__**Start (S)**__

Program