start -> KPACKAGE KMAIN statement import\_statement statement main\_statement statement

import\_statement -> KIMPORT STRING | empty

main\_statement -> KFUNC KMAIN ‘(‘ ‘)’ ‘{‘ statement ‘}’

statement -> global\_statement statement

| if\_statement statement

| switch\_statement statement

| for\_statement statement

| func\_statement statement

| assign\_statement statement

| print\_statement statement

| empty

global\_statement -> func\_statement global\_statement

| global\_assign\_statement global\_statement

| empty

global\_assign\_statement -> var\_assign\_statement

| const\_assign\_statement

if\_statement -> KIF condition ‘{‘ statement ‘}’ else\_statement

else\_statement -> KELSE KIF condition ‘{‘ statement ‘}’ else\_statement

| KELSE ‘{‘ statement ‘}’

switch\_statement -> KSWITCH ‘{‘ case\_statement ‘}’

| KSWITCH ID ‘{‘ case\_var\_statement ‘}’

case\_statement -> KCASE condition ‘:’ statement case\_statement

| KDEFAULT ‘:’ statement

| empty

case\_var\_statement -> KCASE expressions ‘:’ statement case\_var\_statement

| KDEFAULT ‘:’ statement

| empty

expressions -> expression comma\_expr

comma\_expr -> ‘,’ expression comma\_expr

| empty

for\_statement -> KFOR for\_condition ‘{‘ statement ‘}’

| KFOR condition ‘{‘ statement ‘}’

| KFOR range\_statement ‘{‘ statement ‘}’

| KFOR ‘{‘ statement ‘}’

for\_condition -> put on hold

range\_statement -> put on hold

func\_statement -> KFUNC ID ‘(‘ parameters ‘)’ ‘{‘ statement ‘}’

parameters -> ID type comma\_para

comma\_para -> ‘,’ ID type comma\_para

| empty

assign\_statement : var\_assign\_statement

| const\_assign\_statement

| def\_statement

| ID ‘=’ expr\_cond

| ID assign\_oper expression

assign\_oper : PE | ME | TE | DE | MOE

var\_assign\_statement : KVAR ID type assign\_expr

assign\_expr : ‘=’ expr\_cond

| empty

type : KINT

| KBOOL

| KSTRING

| empty

expr\_cond : expression | condition

const\_assign\_statement : KCONST ID type assign\_expr

def\_statement : ID DEF expr\_cond

expression : expression oper expression

| ‘(‘ expression ‘)’

| ‘-’ expression

| INT | STRING | ID

oper : ‘+’ | ‘-’ | ‘\*’ | ‘/’ | ‘%’

condition : condition LAND condition

| condition LOR condition

| expression rel\_op expression

| condition EQ condition

| condition NE condition

| ‘!’ condition

| ‘(‘ condition ‘)’

| BOOL

rel\_op : LT | LE | GT | GE | EQ | NE