labelpoint = label , ";";

gotolabel = "goto", ident, ";";

program\_name = ident;

value\_type = "int16\_t", ident, { ",", ident };

declaration\_ident = ident;

other\_declaration = ",", ident;

declaration = value\_type, declaration\_ident, { other\_declaration };

operation\_not = "!" , inseparable\_expression;

and\_action = "and" , inseparable\_expression;

or\_action = "or" , high\_priority\_expression;

equal\_action = "eg" , middle\_priority\_expression;

not\_equal\_action = "ne" , middle\_priority\_expression;

less\_or\_equal\_action = "<=" , middle\_priority\_expression;

greater\_or\_equal\_action = ">=" , middle\_priority\_expression;

add\_action = "+" , high\_priority\_expression;

sub\_action = "-" , high\_priority\_expression;

mul\_action = "\*" , inseparable\_expression;

div\_action = "/" , inseparable\_expression;

mod\_action = "%" , inseparable\_expression;

unary\_operation = operation\_not;

inseparable\_expression = group\_expression | unary\_operation | ident\_read | value\_read;

high\_priority\_left\_expression = group\_expression | unary\_operation | ident\_read | value\_read;

high\_priority\_action = mul\_action | div\_action | mod\_action | and\_action;

high\_priority\_expression = high\_priority\_left\_expression , { high\_priority\_action };

middle\_priority\_left\_expression = high\_priority\_expression | group\_expression | unary\_operation | ident\_read | value\_read;

middle\_priority\_action = add\_action | sub\_action | or\_action;

middle\_priority\_expression = middle\_priority\_left\_expression , { middle\_priority\_action };

low\_priority\_left\_expression = middle\_priority\_expression | high\_priority\_expression | group\_expression | unary\_operation | ident\_read | value\_read;

low\_priority\_action = less\_or\_equal\_action | greater\_or\_equal\_action | equal\_action | not\_equal\_action;

low\_priority\_expression = low\_priority\_left\_expression , { low\_priority\_action };

group\_expression = "(", low\_priority\_expression , ")";

bind = ident\_write, "==>", low\_priority\_expression;

if\_expression = expression

body\_for\_true = {statement}, tokenSEMICOLON ";"

body\_for\_false = tokenELSE, "else"; {statement}, "statement"; tokenSEMICOLON, ";"

cond\_block = tokenIF, "if"; if\_expression; body\_for\_true; [body\_for\_false]

cycle\_begin\_expression = low\_priority\_expression;

cycle\_counter = ident;

cycle\_counter\_last\_value = value;

cycle\_body = "do" , statement , { statement };

forto\_cycle = "for" , cycle\_begin\_expression , "==>", cycle\_counter , "to", cycle\_counter\_last\_value , cycle\_body , ";";

while\_cycle\_head\_expression = low\_priority\_expression;

while\_cycle = "while" , while\_cycle\_head\_expression , { statement } , ";";

repeat\_until\_cycle\_cond = group\_expression

do\_while\_cycle\_cond = low\_priority\_expression;

do\_while\_cycle = "do" , { statement } , "while" , do\_while\_cycle\_cond;

tokenCONTINUE = "continue";

tokenWHILE = "while";

tokenEXIT = "exit";

continue\_while = tokenCONTINUE , tokenWHILE;

exit\_while = tokenEXIT , tokenWHILE;

statement\_in\_while\_body = statement | continue\_while | exit\_while;

input = "get", "(", ident\_write , ")";

output = "put", "(", low\_priority\_expression , ")";

statement = recursive\_descent\_end\_point | bind | cond\_block | forto\_cycle | while\_cycle | do\_while\_cycle | labeled\_point | gotolabel | input | output;

program = program\_name , ";", "start", "finish", "data", [ declaration ], ";", { statement }, "end";

digit = digit\_0 | digit\_1 | digit\_2 | digit\_3 | digit\_4 | digit\_5 | digit\_6 | digit\_7 | digit\_8 | digit\_9;

non\_zero\_digit = digit\_1 | digit\_2 | digit\_3 | digit\_4 | digit\_5 | digit\_6 | digit\_7 | digit\_8 | digit\_9;

unsigned\_value = (non\_zero\_digit , { digit } | "0");

value = [ sign ] , unsigned\_value;

letter\_in\_lower\_case = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z";

letter\_in\_upper\_case = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z";

ident = letter\_in\_upper\_case, letter\_in\_upper\_case, letter\_in\_upper\_case, letter\_in\_upper\_case,

letter\_in\_upper\_case, letter\_in\_upper\_case, letter\_in\_upper\_case, letter\_in\_upper\_case;

label = letter\_in\_lower\_case , { letter\_in\_lower\_case } ;

sign = sign\_plus | sign\_minus;

sign\_plus = "+";

sign\_minus = "-";

digit\_0 = '0';

digit\_1 = '1';

digit\_2 = '2';

digit\_3 = '3';

digit\_4 = '4';

digit\_5 = '5';

digit\_6 = '6';

digit\_7 = '7';

digit\_8 = '8';

digit\_9 = '9';