**EBNF**

**<программа>** ::= <список\_инструкций>

**<список\_инструкций>** ::= <условие> |

  <цикл> |

  <объявление> |

  <присвоение>

**<условие>** ::= '(' <выражение> ')' <блок\_кода> [ 'else' <блок\_кода> ]

**<цикл>** ::= '(' <выражение> ')' <блок\_кода>

**<объявление>** ::= 'var' <идентификатор> '=' <выражение> [ ',' <объявление>] ';'

**<присвоение>** ::= <идентификатор> “=” <выражение> [ ',' <присвоение>] ';'

**<выражение>** ::= <конъюнкт> 'or' <выражение> | <конъюнкт>

**<конъюнкт>** ::= <дизъюнкт> “and” <конъюнкт> | <дизъюнкт>

**<дизъюнкт>** ::= <отношение> '<' <дизъюнкт> |

  <отношение> '>' <дизъюнкт> |

  <отношение> '<=' <дизъюнкт> |

  <отношение> '>=' <дизъюнкт> |

  <отношение> '==' <дизъюнкт> |

  <отношение> '!=' <дизъюнкт> |

  <отношение>

**<отношение>** ::= <фактор> '+' <отношение> |

    <фактор> '-' <отношение> |

    <фактор>

**<фактор>** ::= <терм> '\*' <фактор> |

       <терм> '/' <фактор> |

       <терм>

**<терм>** ::= <терм1> '.' <терм> |

  <терм1>

**<терм1>** ::= <число> |

                      <идентификатор> |

                      <вызов\_метода> |

                      '(' <выражение> ')' |

                      <объект>

**<вызов\_метода>** ::= '.' <идентификатор>  '(' <объект> ')'

**<объект>** ::= '{' <список\_полей> '}'

**<список\_полей>** ::= <строковый\_идентификатор> ':'<выражение> [ ',' <список\_полей> ]

**<число>** ::= '-'? <десятичная\_цифра> + ('.' <десятичная\_цифра> + ) ?

**<десятичная\_цифра>** ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

**<идентификатор>** ::= <буква> |

                                       <идентификатор> <буква> |

                                        <идентификатор> <цифра>

**<буква>** ::= 'A' | 'B' … 'Z' | 'a' | 'b' … 'z'

**<строковый\_идентификатор>** ::= '”' <идентификатор> '”'

**Reduced LL(1)-Grammar**

**GOAL** -> INSTRUCTION\_LIST

**INSTRUCTION\_LIST** ->

INSTRUCTION INSTRUCTION\_LIST |

E

**INSTRUCTION** ->

var VAR

if IF |

while WHILE |

return RETURN |

ID ASSIGNMENT

**RETURN** -> EXPR ;

**IF** -> ( EXPR ) BODY ELSE

**ELSE** -> else BODY |

E

**WHILE** -> ( EXPR ) BODY

**BODY->** { INSTRUCTION\_LIST } |

INSTRUCTION

**ASSIGNMENT ->** = EXPR ;

**VAR ->** ID = EXPR VAR1 ;

**VAR1 ->** , ID = EXPR VAR1 |

        E

**EXPR**  ->  T1 or EXPR |

          T1

**T1**   ->   T2 and T1 |

    T2

**T2**   ->   T3 < T2 |

    T3 > T2 |

    T3 <= T2 |

    T3 >= T2 |

    T3 == T2 |

    T3 != T2 |

    T3

**T3**   ->   T4 + T3 |

   T4 - T3 |

   T4

**T4**   ->   T5 \* T4 |

    T5 / T4 |

    T5

**T5**   ->   T6 . T5 |

    T6

**T6** ->   NUM |

  ID |

  API CALL |

  ( CONDITION ) |

  OBJECT

**CALL** -> . ID ( OBJECT)

**OBJECT** -> { FIELDS }

**FIELDS** -> STRING\_ID : EXPR FIELDS\_1 |

E

**FIELDS\_1** -> , FIELDS |

   E