**Грамматика языка:**

**Аксиома:**

program

**Правила:**

**program** ::=statement EOL EOF | statement EOL program

**statement** ::= declaration | assigning | function-call | function-construction | if-construction| while-construction | return-statement | switch-statement | for-statement | array-statement

**declaration** ::= simple-declaration | simple-declaration EQUALS expression | simple-declaration EQUALS term

**simple-declaration** ::= VAL ID | VAR ID

**value** ::= STR | CHAR | NUM | TRUE | FALSE | MINUS NUM

**expression** ::= expression-inside | LBR expression-inside RBR | LBR logical-expression RBR | logical-expression| LBR expression-inside RBR expression-inside’ | LBR logical-expression RBR logical-expression’ | logical-expression logical-expression’

**expression-inside** ::= LBR expression-inside RBR | term sign expression-inside | term sign term | LBR expression-inside RBR expression-inside’ | term sign expression-inside expression-inside’ | term sign term expression-inside’

**expression-inside’** ::= sign expression | sign term | sign expression expression-inside’ | sign term expression-inside’

**logical-expression** ::= LBR logical-expression RBR | NOT term | NOT logical-expression | term logical-sign logical-expression | term logical-sign term | ID IS type LBR logical-expression RBR logical-expression’ | NOT term logical-expression’ | NOT logical-expression logical-expression’ | expression-inside logical-expression’

**logical-expression’** ::= logical-sign term | logical-sign logical-expression | logical-sign term logical-expression’ | logical-sign logical-expression logical-expression’

**term** ::= value | ID | function-call | ID LSQBR NUM RSQBR| MINUS ID | MINUS ID LSQBR NUM RSQBR | MINUS function-call

**logical-sign** ::= MORE | LESS | MOREOREQUAL | LESSOREQUAL | AND | OR | ISEQUAL | ISNOTEQUAL

**sign** ::= PLUS | MINUS | MUL | MOD | DIV

**function-call** ::= ID LBR actual-params RBR | ID LBR RBR

**actual-params** ::= actual-param COMMA actual-params | actual-param

**actual-param** ::= expression | term

**assigning** ::= ID EQUALS expression | ID EQUALS term

**function-construction** ::= FUNCTION ID LBR params RBR LCBR function-block RCBR | FUNCTION ID LBR RBR LCBR function-block RCBR | FUNCTION ID LBR params RBR LCBR RCBR | FUNCTION ID LBR RBR LCBR RCBR | FUNCTION ID LBR RBR COLON type LCBR RCBR | FUNCTION ID LBR params RBR COLON type LCBR function-block RCBR | FUNCTION ID LBR RBR COLON type LCBR function-block RCBR

**params** ::= param COMMA params | param

**param** ::= ID COLON type

**type** ::= BYTE | SHORT | INT | LONG | FLOAT | DOUBLE | BOOL | CHARTYPE | STRING

**function-block** ::= EOL | EOL function-block | function-command function-block | function-command

**function-command** ::= declaration | assigning | function-call | if-construction| while-construction | switch-statement | return-statement| for-statement| array-statement

**if-construction** ::= IF LBR if-condition RBR LCBR function-block RCBR | IF LBR if-condition RBR LCBR RCBR | IF LBR if-condition RBR LCBR function-block RCBR ELSE LCBR function-block RCBR | IF LBR if-condition RBR LCBR RCBR ELSE LCBR function-block RCBR | IF LBR if-condition RBR LCBR function-block RCBR ELSE LCBR RCBR | IF LBR if-condition RBR LCBR RCBR ELSE LCBR RCBR

**if-condition** ::= logical-expression | term

**while-construction** ::= WHILE LBR if-condition RBR LCBR function-block RCBR | WHILE LBR if-condition RBR EOL LCBR function-block RCBR | WHILE LBR if-condition RBR LCBR EOL function-block RCBR

**return-statement** ::= RETURN term | RETURN expression

**switch-statement** ::= WHEN LBR ID RBR LCBR switch-block RCBR

**switch-block** ::= switch-command | switch-command EOL | switch-command EOL switch-block |EOL switch-command

**switch-command** ::= value CASE LCBR function-block RCBR | value CASE LCBR RCBR | ELSE CASE LCBR function-block RCBR | ELSE CASE LCBR RCBR | value CASE function-command | ELSE CASE function-command

**for-statement** ::= FOR LBR ID IN ID RBR LCBR for-block RCBR | FOR LBR ID IN ID RBR LCBR RCBR | FOR LBR ID IN NUM TO NUM RBR LCBR for-block RCBR | FOR LBR ID IN NUM TO NUM RBR LCBR RCBR | FOR LBR ID IN ID RBR for-statement | FOR LBR ID IN NUM TO NUM RBR for-statement

**for-block** ::= EOL | EOL for-block | for-command for-block | for-command

**for-command** ::= declaration | assigning | function-call | if-construction| while-construction | switch-statement | return-statement | CONTINUE | BREAK

**array-statement** ::= ID POINT ADD LBR value RBR | ID POINT REMOVE LBR NUM RBR