BNF:
{} - 0 lub wiecej razy
[] - opcja: 0 lub 1
| - lub
* * - komentarz

w rozszerzonej notacji **EBNF** stosuje sie takze nawiasy okrągłe () do grupowania, jednak w tym wypadku uzywamy prostej notacji **BNF.**

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

<cyfra> ::= "0" | .. | "9"

<znak> ::= "_"

<znak_linii> ::= * znak konca linii <LF><CR> *

<znaki_specjalne> ::= ";" | "(" | ")"

<inne_znaki> ::= * wszystkie inne znaki (np. dialektyczne) rozne od <litera>, <cyfra>, <znak>, <znaki_specjalne>, <znak_linii> *

<ciag_znakow> ::= ' { <litera> | <cyfra> | <znak> | <znaki_specjalne> | <znak_linii> | <inne_znaki> } '

<slowo_kluczowe> ::= program | var | real | integer | begin | end | if | then | while | do | read | write | not | true | false

<typ_zmiennej> ::= real | integer

<identyfikator> ::= <litera> { <litera> | <cyfra> | <znak> } * <identyfikator> nie może być taki sam jak <slowo_kluczowe> *

<liczba_calkowita> ::= <cyfra> { <cyfra> }

<liczba_rzeczywista> ::= <liczba_calkowita> . <liczba_calkowita>

<liczba> ::= <liczba_calkowita> | <liczba_rzeczywista>

<operator_arytmetyczny> ::= "+" | "-" | "/" | "*"

<operator_logiczny> ::= ">" | "<" | "<>" | "="

<zmienna_logiczna> ::= <identyfikator> | <liczba>

<wyrażenie_logiczne> ::= <zmienna_logiczna> <operator_logiczny> <zmienna_logiczna>

<warunek_logiczny> ::= [not] <wyrażenie_logiczne>

<instrukcja_przypisania> ::= <identyfikator> := <liczba> ;

<instrukcja_podstawienia> ::= <identyfikator> := <identyfikator> ;

<instrukcja_warunkowa> ::= if (<warunek_logiczny>) then [<instrukcja>] ;

<instrukcja_petli> ::= while (<warunek_logiczny>) do [<instrukcja>] ;

<instrukcja_wejscia> ::= read (<identyfikator>) ;

```
<instrukcja_wyjscia> ::=  write ( <liczba> ) ; |
                          write ( <identyfikator> ) ; |
                          write ( <ciag_znakow> ) ;

<instrukcja> ::=  <instrukcja_przypisania> | <instrukcja_podstawienia> |
                  <instrukcja_warunkowa> | <instrukcja_petli> |
                  <instrukcja_wejscia> | <instrukcja_wyjscia>

<nazwa_programu> ::= <identyfikator>

<deklaracja_zmiennych> ::= var
                               <identyfikator> : <typ_zmiennej> ;
                               { <identyfikator> : <typ_zmiennej> ; }

<cialo_programu> := { <instrukcja> }

<program> ::=  program <nazwa_programu> ;
                  [ <deklaracja_zmiennych> ]
               begin
                  [ <cialo_programu> ]
               end.
```