

**AKADEMIA PODLASKA
S.S.W.**

Projekt: *Kompilator języka PASCAL w C++*

**Temat pracy: „Notacja BNF”
v 1.4.1**

**Autorzy: Krackowski Adam
Wawryniuk Izabela
Zacharczuk Dariusz
Zalewski Michał**

BNF:

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

w rozszerzonej notacji **EBNF** stosuje się także nawiasy okrągłe () do grupowania, jednak w tym wypadku używamy prostej notacji **BNF**.

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

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

<znak> ::= "_"

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

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

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

<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> | true | false

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

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

<DANA_PODSTAWIANA> ::= <IDENTYFIKATOR> | <LICZBA>

<INSTRUKCJA_PODSTAWIENIA> ::= <IDENTYFIKATOR> " :="
 [-] <DANA_PODSTAWIANA>
 { <OPERATOR_ARYTMETYCZNY> <DANA_PODSTAWIANA> } ;

```

<instrukcja_warunkowa> ::= if <warunek_logiczny> then <blok_instrukcji>
<instrukcja_petli>    ::= while <warunek_logiczny> do <blok_instrukcji>
<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>

<blok_instrukcji> ::= <instrukcja>
                    | begin <instrukcja> { <instrukcja> } end;

<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.

```