Michał Niedźwiecki
Tomasz Karczmarczyk
Sławomir Napiórkowski

Notacja BNF

1. Złożenia projektu.

Projekt będzie obejmował stworzenie interpretera języka Pascal. Zakres jaki będzie on obejmował ograniczony zostanie do podstawowych instrukcji języka oraz pętli for. Zastosowane będą instrukcje logiczne If – Then oraz If – Then – Else.

2. Notacja Bachusa Naura – BNF.

Notacja Backhusa-Naura dopuszcza stosowanie następujących symboli:
<> – w nawiasach ostrych podaje się definiowanej składni,
::= – oznacza przypisanie do definiowanej nazwy jej składni,
| – lub
[ ] – nawiasy kwadratowe oznaczają opcjonalne wystąpienie,
{} - wybierz jeden z elementów występujących (dokładnie 1 raz),
{}+ - element musi wystąpić co najmniej 1 raz (1 lub więcej razy),
{}* - element może wystąpić 0 lub więcej razy (0 lub więcej razy),

3. Składnia w notacji BNF.

<1..9> ::= 1|2|3|4|5|6|7|8|9
<zero> ::= {0}
<mala_litera>::={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}
<duza_litera>::={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}
<typ_zm> ::= real | integer
<operatory_aryt.> :: = + | - | * | / | div
<operatory_por> ::= < | = | > | >= | <=
<operator_por=>::= =
<n_p> :: = (
<n_k> ::=)
<od>::= for
<do>::= to | downto
<wykonuj>::= do
<jezeli> ::= if
<to> ::= then
<w_przeciwnym_wyp> ::= else
<deklaracja> ::= :
<poczatek_b> ::= begin
<koniec_b> ::= end
<wczytaj> ::= readln
<wypisz> ::= writeln
<separator> ::= ;
<dek_zm> ::= var

<op_przypisania> ::= :=
<prog> ::= program

<cyfra> ::= <zero> | <1..9>
<nazwa> ::= litera { litera | cyfra }*

<liczba_nat_bez_0> ::= <1..9>{<cyfra>}*
<liczba_nat> ::= <zero> | {"-"<liczba_nat_bez_0>} |<liczba_nat_bez_0>
<liczba_zmiennoprzecinkowa> ::= <liczba_nat>"."{<cyfra>}*
<liczba> ::= {<liczba_nat> | <liczba_zmiennoprzecinkowa>}

<deklaracja_zm_globalna> ::= <nazwa> <deklaracja> <typ_zm> <separator>
<zm_globalna> ::= <nazwa>
<wartosc> ::= <zm_ globalna >| <liczba>
<operacja_arytm> ::= wartosc{<operator_aryt> <wartosc>}*
operacja_arytm_w_nawias::={ <operacja_arytm> |"(" <operacja_arytm> ")"}
operacja_arytm_z_nawias::= < operacja_arytm_w_nawias > { <operator_aryt>
<operacja_arytm_w_nawias >}*

wyrażenie::=

<porownanie>:: = <zm_ globalna> <operator_por><zm_globalna> | <zm_ globalna>
<operator_por> <liczba>
<przypisanie> ::= <zm_ globalna> <op_przypisania> {< wyrazenie_mat > | <wartosc>}
<separator>
<blok> ::= <początek_b> {operacje}* <koniec_b> <separator>
<op_if> ::= {<jeżeli> <operator_por> <to> <blok>|<operacja>} | {<jeżeli> <porownanie>
<to> <blok> < w_przeciwnym_wyp > <blok>}
<petla> ::= <od><zm_ globalna> < operator_por=>
<liczba_nat><do><liczba_nat><wykonuj><blok>|<operacja>
<czyt_pis> ::= {{<wczytaj> <n_p> <zm_globalna> <n_k>} | {<wypisz> <n_p> <zm_
globalna> <n_k>} } <separator>
<operacje> ::= czyt_pis | petla | op_if | przypisanie
<def_prog> ::= <prog> <nazwa> <separator> <dek_zm> {<deklaracja_zm_globalna> }*
<blok>