

BNF – Pascal

Kinga Kosicka.

Krzysztof Mazurek

Tomasz Michałowski

`<identyfikator> ::= <litera> { <litera_cyfra> }`

`<litera_cyfra> ::= <litera> | <cyfra>`

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

`<cyfra> ::= "0" | "1" | "2" | ... | "9"`

`<typ> ::= "integer" | "real" | "^integer" | "^real"`

`<liczba> ::= <całkowita> | <rzeczywista>`

`<niezero> ::= "1" | "2" | ... | "9"`

`<całkowita> ::= <niezero> { <cyfra> } | -<niezero> { <cyfra> } | "0"`

`<rzeczywista> ::=`

`-<cyfra> { <cyfra> } "." <cyfra> { <cyfra> } | <cyfra> { <cyfra> } "." <cyfra> { <cyfra> }`

`<nawias_lewy> ::= "("`

`<nawias_prawy> ::= ")"`

`<program> ::= <blok_deklaracji> < blok_implementationi>`

`<blok_deklaracji> ::= "var" <deklaracja> { <deklaracja> }`

`<deklaracja> ::= <identyfikator> { ',' <identyfikator> } ':' <typ> <srednik>`

`<blok_implementationi> ::= "begin" { <instrukcja_zlozona> } "end"`

`<instrukcja_prosta> ::= <instrukcja_warunkowa> | <instrukcja_przypisania> | <instrukcja_petli>`

`<instrukcja_dispose> | <instrukcja_new>`

`<instrukcja_zlozona> ::= { <instrukcja_prosta> }`

`<instrukcja_warunkowa> ::= "if" <instrukcja_porownania> "then" "begin" <instrukcja_zlozona> "end" <srednik> | "else" "begin" <instrukcja_zlozona> "end" <srednik>`

`<instrukcja_dispose> ::= "dispose (" <identyfikator> ")" <srednik>`

`<instrukcja_new> ::= "new (" <identyfikator> ")" <srednik>`

<instrukcja_porównania>::=<identyfikator>|<liczba><operator_porównania><identyfikator>|<liczba>

<operator_porównania>::=">"|">="|"="|"<"|"<="|"<"<"

<instrukcja_przypisania>::=<identyfikator>":="<identyfikator>|<liczba>|

<operacja_arytmetyczna><srednik>

<operacja_arytmetyczna>::=<identyfikator>|<liczba><operator_arytmetyczny><identyfikator>|<liczba>|

|<nawias_lewy>{<nawias_lewy>}<identyfikator>|

<liczba><operator_arytmetyczny>{<nawias_lewy>}<identyfikator>|

<liczba>{<nawias_prawy>}<nawias_prawy> <srednik>

<operator_arytmetyczny>::="+|"-'|'|'*'|'div'|'mod'

<instrukcja_pętli>::='while' '(' <instrukcja_porównania> ')' 'do begin' <instrukcja_złożona> 'end' <srednik>

<srednik>::=';'

<operacja_wejscia>::='read('<identyfikator>')'|'readln('<identyfikator>')'<srednik>

<operacja_wyjscia>::='write('<string>"|<identyfikator>')'|'writeln('<string>|<identyfikator>')'<srednik>

<string>::="''" {<litera_cyfra>}'