

# Specyfikacja języka

## Tokeny

### Słowa kluczowe

- fun
- var
- for
- while
- do
- if
- then
- else
- and
- or
- is
- not
- return
- true
- false
- none

### Znaki zarezerwowane

Znak	Znaczenie
+	Dodawanie
-	Odejmowanie
*	Mnożenie
/	Dzielenie
{	Początek bloku
}	Koniec bloku
,	Separator w definicjach list i funkcji oraz w wywołaniach funkcji
;	Znak końca instrukcji
(	Priorytet wyrażeń, definicje i wywołania funkcji
)	Jak wyżej
[	Inicjalizacja list
}	Jak wyżej
+	Przypisanie

### Inne tokeny

Nazwa	Wyrażenie regularne
COMPARISON	>=   <=   >   <
ID	[a-zA-Z_][a-zA-Z_0-9]*
FLOAT	\d\.\(d+\)
INT	\d+
STRING	".*?"
COMMENT (ignorowane)	\#(\.)*
WHITESPACE (ignorowane)	\s

## Typy

### Dostępne typy

- Liczba całkowita – INT
- Liczba zmiennoprzecinkowa – FLOAT
- Wartość logiczna – BOOL
- Napis - STRING
- Lista – LIST
- Brak wartości – NONE

### Zasady konwersji pomiędzy typami

Typ początkowy	Typ docelowy	Zasady konwersji
INT	FLOAT	Dodajemy część dziesiętną z wartością 0.0
INT	BOOL	0 – false Pozostałe wartości – true
INT	STRING	Liczba zamieniana jest na jej reprezentację w formie napisu
INT	LIST	Niedozwolone
INT	NONE	Niedozwolone
FLOAT	INT	Z liczby obcinana jest część dziesiętna
FLOAT	BOOL	0.0 – false Pozostałe wartości – true
FLOAT	STRING	Liczba zamieniana jest na jej reprezentację w formie napisu
FLOAT	LIST	Niedozwolone
FLOAT	NONE	Niedozwolone
BOOL	INT	true – 1 false – 0
BOOL	FLOAT	true – 1.0 false – 0.0
BOOL	STRING	Niedozwolone
BOOL	LIST	Niedozwolone
BOOL	NONE	Niedozwolone
STRING	INT	Jeżeli napis reprezentuje poprawną liczbę to zamieniany jest na tą liczbę. W przeciwnym przypadku występuje błąd konwersji
STRING	FLOAT	Jeżeli napis reprezentuje poprawną liczbę zmiennoprzecinkową (z separatorem '.') to zamieniany jest na tą liczbę. W przeciwnym przypadku występuje błąd konwersji
STRING	BOOL	Pusty napis – false Pozostałe wartości – true
STRING	LIST	Zwracana jest lista, której elementami są znaki tworzące napis
STRING	NONE	Niedozwolone
LIST	INT	Niedozwolone
LIST	FLOAT	Niedozwolone
LIST	BOOL	[] – false Pozostałe wartości – True
LIST	STRING	Zwracana jest reprezentacja listy w postaci napisu
LIST	NONE	Niedozwolone
NONE	INT	Niedozwolone

NONE	FLOAT	Niedozwolone
NONE	BOOL	Zawsze false
NONE	STRING	Napis 'none'
NONE	LIST	Niedozwolone

## Opis gramatyki

### Opis notacji

W zapisie produkcji stosowana jest notacja.

lewa\_strona : prawa\_strona

Jeżeli dla danej lewej strony produkcji możliwych jest kilka prawych stron, to oddzielane są one znakiem „\”.

lewa\_strona : prawa\_strona | inna\_prawa\_strona

Tokeny składające się z jednego znaku są zapisywane w produkcji jako 'znak'. Symbole terminale są zapisywane dużymi literami.

### Lista produkcji

program : declarations

statement : expression ';' |

if\_statement

| while\_statement

| for\_statement

| block

| return\_statement

return\_statement : RETURN expression ';' |

RETURN ';' |

for\_statement : FOR variable\_declaration expression ';' expression DO statement

| FOR statement expression ';' expression DO statement

| FOR variable\_declaration ';' expression DO statement

| FOR statement ';' expression DO statement

| FOR variable\_declaration ';' DO statement

| FOR statement expression ';' DO statement

| FOR ';' expression ';' expression DO statement

| FOR ';' expression ';' DO statement

| FOR ';' ';' expression DO statement

| FOR ';' ';' DO statement

while\_statement : WHILE expression DO statement  
if\_statement : IF expression THEN statement  
                  | IF expression THEN statement ELSE statement  
block : '{' declarations '}'  
declarations : declarations declaration  
                  | declaration  
declaration : variable\_declaration  
                  | function\_declaration  
                  | statement  
function\_declaration : FUN function  
function : ID '(' parameters\_list ')' block  
                  | ID '(' ')' block  
parameters\_list : parameters\_list ',' ID  
                  | ID  
variable\_declaration : VAR ID '=' expression ';' ;  
                  | VAR ID ';' ;  
expression : assignment  
assignment : l\_value\_expression '=' assignment  
                  | binary\_logic\_operator  
l\_value\_expression : equality  
binary\_logic\_operator : and\_operator  
                  | or\_operator  
                  | equality  
and\_operator : and\_operator AND equality  
                  | equality  
or\_operator : or\_operator OR equality  
                  | equality  
equality : expression IS comparison  
                  | expression IS NOT comparison  
                  | comparison

comparison : comparison COMPARISON term

| term

term : term '-' factor

| term '+' factor

| factor

factor : factor '/' unary

| factor '\*' unary

| unary

unary : NOT unary

| '-' unary

| call\_or\_subscript

call\_or\_subscript : function\_call

| ID '[' expression ']'

| primary

function\_call : ID '(' arguments ')'

arguments : arguments ',' expression

| expression

primary : number\_literal

| boolean\_literal

| STRING

| NONE

| '(' expression ')'

| ID

| list\_literal

list\_literal : '[' arguments ']'

| '[' ']'

number\_literal : INT

| FLOAT

boolean\_literal : TRUE

| FALSE