Język Pascalis

a) Gramatyka języka:

Comments ::= //.* (to new line)

Type t ::= **numeri integri | logica booleana | titulus | litera**

Num n ::= . . . −1 | 0 | 1 | . . .

Char c ::= . . . 'a' | 'b' | 'c' | . . .

Var x ::= x1 | x2 | . . .

VarP p ::= p1 | p2 | . . .

Param pa ::= pa1, pa2 | ae | x |{epsilon}

Expr e ::= n | x | x[i] | e1 + e2 | e1 ∗ e2 | e1 − e2 | (e1) | p(pa) | **lege**(x) | **ordo**(c) | **longitudo**(x)

C Ecpr ce := c | x | x[i]

BExpr b ::= x | x[i] | **verum** {true} | **falsum** {false} | e1 < e2 | e1 <= e2 | e1 > e2 | e1 >= e2 | e1 = e2 | e1 <> e2 | b1 = b2 | b1 <> b2 | b1 **et** b2 {and} | b1 **uel** b2 {or} | **non** b {not} | (b1) | f(x)| s1 > s2 | s1 < s2 | s1 = s2 | s1 >= s2 | s1 <= s2 | s1 <> s2 | ce1 < ce2 | ce1 <= ce2 | ce1 > ce2 | ce1 >= ce2 | ce1 = ce2 | ce1 <> ce2

S Expr s ::= s1 | s1 + s2 | s[e:e] | s[:e] | s[e:] | ”przykładowy napis” | ”a” | ...

Index i ::= e | ce

AnyExpr ae = e | b | s | ce

Block block ::= **incipe** I **fini | incipe** I **fini**

Dparam dpa ::= v**ariabilis** x: t | x: t | dpa1; dpa2 | {epsilon} | **variabilis** `x:matrix{t} autem t | x:matrix{t} autem t | variabilis x: dictionarium{t} autem t| x: dictionarium{t} autem t`

Dec D ::= v**ariabilis** x : t | **variabilis** `x:matrix[ce1..ce2] autem t` | **variabilis** `x:matrix[e1..e2] |variabilis x:dictionarium{t} autem t` | D1; D2 | **functio** p(dpa): **t**; D block; | **processus** p(dpa); D block;

Instr I ::= | x := ae | x[i] := ae | x := **lege**() | I1; I2 | **persulta** | **si** b **tunc** I1 **alter** I2 | **si** b **tunc** I1 | p(pa) | block | **incribo** (x) | **donec** b **fac** I | **pro** x:=e **ut** e **fac** I | **refer** ae |

Program pr ::= **program** x; D **incipe** I **fini**.

b) Przykładowy program

```
program examplum_program;

functio examplum_functio(variabilis x1: numeri integri; x2: numeri integri): numeri integri;

incipe
        x1 := 3;
        x2 := 4;
        refer  x2;
fini;


processus examplum_processus();
// Ta procedura wczytuje liczbę i ją wypisuje
incipe
        variabilis x2: numeri integri;
        x2 := lege();
        incribo(x2);
fini;


incipe
        variabilis x: dictionarum{litera} autem {litera};
        variabilis y1: numeri integri;
        variabilis y2: numeri integri;
        processus examplum_processus2();
        incipe
                variabilis x: numeri integri;
                // slownik x jest nadpisywane przez numeri integri x
                variabilis x2: numeri integri;
                functio exaplum_functio2(): numeri integri;
                incipe
                        // funkcja examplum_funtio widzi zmienną x
                        refer x;
                fini;
```

```
        x := 2;
        x2 := examplum_functio2()
        // x2 = 2
    fini;
    x['a'] := 'b';
    x['b'] := 'c'
fini.
```

c) Tesktowy opis języka

Język jest podzbiorem języka pascal, w którym zamieniono słowa angielskie na ich łacińskie odpowiedniki. Dodatkowo zawiera wycinanki srtingów podobne do pythona.

Typy:

Język zawiera następujące typy:

Nemeri Integri – Integer w pascalu

Logica Booleana – Boolean

Titulus - String

Litera – Char

Oznaczające to co ich odpowiedniki w języku Pascal.

Wyrażenia Logiczne:

verum – true

falsum – false

et -and

uel – or

non – not

Wyrażenia numeryczne:

lege – read

ordo – ord

longitudo – length

Bloki:

inicipe – begin

fini – end

(Bloki działają trochę inaczej niż w pascalu co widać w przykładowym programie)

Deklaracje:

variabilis – var

matrix – array, tab

autem – of

dictionarium – dict (struktura której w pascalu nie ma)

Funkcje i procedury:

functio – function

processus - procedure

Instrukcje:

persulta – skip

incribo - wypisz

si tunc aliter (if then else)

Pętle:

donec fac (while do)

pro ut fac (for to do)

Funkcje i procedury mają działać tak jak w pascalu. Można przekazywać do nich wartości oraz zmienne. Możliwe jest deklarowanie procedury w funkcji i używanie zmiennych (z tej funkcji).

Zaimplementowana również będzie rekurencja.

Każda zmienna będzie miała też statyczne typowanie.

Będą obsługiwane dynamiczne błędy wykonania, np. dzielenie przez zero.

Język będzie zawierać wbudowany typ **titulus** wraz z wycinankami listowymi takimi jak w języku python.

Ponad to język będzie zawierać słowniki parametryzowane typami oraz tablice takie jak w pascalu, indeksowane **numeri integri** lub **litera**.

W przypadkach nieopisanych język będzie się zachowywał jak język pascal z przetłumaczoną składnią.