**1. Specificarea mini limbajului de programare (EBNF)**

program = header , declarations , ‘*begin’* , instruction\_list , ‘*end*’ ;

header = ‘*program*’ , program\_name , ‘*;*’ ;

program\_name = ID;

declarations = [‘*type*’ , user\_defined\_type , ‘*:*’ , other\_type , ‘*;*’] ‘*var*’ , variable\_list , ‘*:*’ , data\_type , ‘*;*’ ;

variable\_list = ID , { ‘*,*’ , ID } ;

data\_type = ‘*integer*’ | ‘*real*’ | user\_defined\_type ;

user\_defined\_type = ID ;

other\_type = ‘*array[1..5] of integer*’ ;

letter = ‘*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*’ ;

digit = ‘*0*’ | ‘*1*’ | ‘*2*’ | ‘*3*’ | ‘*4*’ | ‘*5*’ | ‘*6*’ | ‘*7*’ | ‘*8*’ | ‘*9*’ ;

instruction\_list = instruction , { ‘*;*’ , instruction } ;

instruction = assign\_instruction | if\_instruction | io\_instruction | while\_loop ;

assign\_instruction = ID , ‘*:=*’ , expression , ‘*;*’ ;

expression = (ID | CONST) , { operator , (ID | CONST) } ;

if\_instruction = ‘*if*’ , boolean\_expression , ‘*then* *begin’* , instruction\_list , ‘*end;*’ , [‘*else* *begin’* , instruction\_list , ‘*end;*’ ] ;

boolean\_expression = ID , bool\_operator , expression ;

io\_instruction = ‘*readln(*‘ , ID , ‘*);*’ | ‘*writeln(*‘ , ID , ‘*);*’ ;

while\_loop = ‘*while*‘ , boolean\_expression , ’ *do begin* ‘ , instruction\_list , ‘*end;*’ ;

bool\_operator = ‘*>*’ | ‘*<*’ | ‘*<>*’; operator = ‘*+*’ | ‘*-*‘ | ‘*\**’ ;

ID = letter , { letter | digit } ;

CONST = digit , [‘*.*’] , { digit } ;

# Textele sursă a 3 mini-programre

1. Perimetrul și aria unui cerc cu rază dată:

program cerc;

var a, p, r: real;

begin

readln(r);

p := 2\*3.14\*r;

a := 3.14 \* r \*r;

writeln(p);

writeln(a);

end

1. CMMDC a două numere:

program cmmdc;

var a, b: integer;

begin

readln(a);

readln(b);

while a<>b do

begin

if a>b then

begin

a:=a-b;

end;

else begin

b:=b-a;

end;

end;

writeln(a);

end

1. Suma a n numere citite de la tastatură:

program suma;

var n, sum, nr: integer;

begin

readln(n);

sum := 0;

while n>0 do

begin

readln(nr);

sum := sum + nr;

n := n - 1;

end;

writeln(sum);

end

**3. Programe ce conțin erori:**

1. Două erori care sunt în același timp erori în limbajul original (pentru care MLP definește un subset)

program err1;

var a, **2b**: integer;

begin

**readln(a)**

writeln(a);

end

1. Două erori conform MLP, dar care nu sunt erori în limbajul original.

program ex1;

var a, b: **word**;

begin

**readln(a, b);**

a := a + b;

writeln(a);

end