A Linguagem **PASCAL**

= = = = = = = = = = = = = = = = =

1) ALFABETO: são os símbolos (ié, "caracteres") permitidos na linguagem. São as **letras** (maiúsculas e minúsculas), os **algarismos** (ou *dígitos*) decimais e alguns **caracteres especiais**.

Os caracteres especiais, que servem como operadores ou delimitadores, são os seguintes:
+ - * / = < > ( ) [ ] { } . , : ; ^

Os seguintes símbolos aparecem aos pares:
<> <= >= := .. (* *)

2) ITENS LÉXICOS: são os elementos básicos de um programa. São *palavras reservadas*, os *dados*, os *identificadores* e os *comentários*.

2.1) As **palavras reservadas** têm um significado pré-determinado. São elas: **and, array, begin, case, const, div, do, downto, else, end, file, for, function, goto, if, in, label, mod, not, of, or, procedure, program, record, repeat, set, then, to, type, until, var, while, with, xor.**

Obs: Existem ainda "identificadores" que também têm um significado pré-definido: **false, true, nil, integer, real, boolean, char, string, abs, sqr, sqrt, pred, ord, chr, eof, eoln, reset, rewrite, read, readln, write, writeln, new, forward, external, implementation, unit, uses.**

2.2) Os **dados** são valores constantes que aparecem no texto do programa. São os **números** (inteiros e reais), as **cadeias de caracteres** (ou *strings*, ié, sequências de caracteres escritas entre apóstrofos) e os **valores lógicos** (*true* e *false*).
Os números são sequências de dígitos que podem ser precedidos por um sinal (+,-). Os números reais podem ter um ponto decimal seguido de outros dígitos, e podem também estar na representação de ponto flutuante (ié, conter um expoente da forma: a letra E+sinal+dígitos, correspondentes à potência de 10).

2.3) Os **identificadores** são nomes, escolhidos pelo programador, para

representar *constantes*, *variáveis*, *programas*, *procedimentos*, *tipos*, *campos* (de registros) e *unidades*. São sequências de letras e dígitos (podendo ser utilizado ainda o símbolo _), começando obrigatoriamente por uma letra.

2.4) Os **comentários** são sequências de caracteres escritas entre **{** e **}**, ou entre **(*** e ***)**.

3) VARIÁVEIS:

Cada *variável* corresponde a uma **posição** de memória do computador, utilizada para armazenar dados (de entrada ou gerados durante a execução do programa). Podemos imaginá-la como uma "caixinha"!

A cada variável estão associados:
(1) um **nome** (ié, um identificador);
(2) um **conteúdo** (ié, um valor), que pode ser modificado por um comando de leitura (READ ou READLN) ou por um comando de atribuição;
(3) um **tipo** (de dados), que é o conjunto de valores que podem servir como conteúdo da variável.

Obs: Toda variável usada por um programa deve ser previamente declarada.

4) TIPOS DE DADOS:

Um *tipo de dados* é um conjunto de valores (constantes) que uma variável pode assumir. De outra maneira, é o *intervalo* a que uma variável pertence.

São divididos em: (1) *Pré-definidos*, ié, já existentes nas linguagem e (2) *Definidos pelo programador* (num programa).

4.1) Tipos **PRÉ-DEFINIDOS**:

(a) **Numéricos**:

- INTEIROS
  - *integer*: -32768 a 32767
  - *shortint*: -128 a 127
  - *byte*: 0 a 255
  - *word*: 0 a 65535
  - *longint*: -2147483648 a 2147483647

- REAIS
  - *real*: 2.9E-39 a 1.7E+38_
  - *single*: 1.5E-45 a 3.4E+38
  - *double*: 5.0E-324 a 1.7E+308
  - *extended*: 3.4E-4932 a 1.1E+4932
  - *comp*: -9.2E+18 a 9.2E+18

Obs: Estes tipos são representados em *ponto flutuante* cujas *mantissas* têm, respectivamente, os seguintes números de dígitos: 11(*real*), 7 a 8(*single*), 15 a 16(*double*) e 19 a 20 (*extended* e *comp*).

(b) **Literal** - *char*: refere-se aos caracteres do teclado (código ASCII)

(c) **Lógico** - *boolean*: utiliza dois valores constantes: TRUE e FALSE. São valores admitidos como resultado de uma expressão lógica ou assumida por uma variável lógica.

(d) **Intervalo de Tipo** (ou *subrange*) - é um subconjunto de um conjunto ordenado (portanto, o tipo real não é admitido).

Obs: Os tipos a seguir são classificados como "**estruturados**":

(e) **Cadeia de caracteres** – *string*: corresponde a sequências de caracteres, isto é, de elementos do tipo *char*.

(f) ***array*;** (g) ***record*;** (h) ***file*;** (i) ***text***;
(j) ***set***; (k) ***pointer***; (l) ***object***

4.2) Tipos **DEFINIDOS PELO PROGRAMADOR**:

ex-1) *type* cor = (azul, verde, amarelo, alaranjado,vermelho);

ex-2) *type* mes =
(jan, fev, mar, abr, mai, jun, jul, ago, set, out, nov, dez);

• • •

Usamos ainda a definição de tipos quando se trata da descrição de parâmetros de procedimentos...

• • •

5) EXPRESSÕES: São sequências de *operandos* e *operadores* devidamente arrumadas (ié, obedecendo a notação *infixa*), que quando avaliadas retornam como resultado um valor constante. As expressões podem ser **aritméticas** ou **lógicas**.

Os **operandos** são *variáve*is, *constantes*, (resultados das aplicações de) *funções* e *expressões entre (e)*.

Os **operadores** são:

- ***aritméticos***: + - * / *div* *mod*
- ***lógicos***: *not* *and* *or* *xor*
- ***relacionais***: = <> > < >= <= *in*

Observações:

(1) { / - corresponde á divisão cujo resultado é *real.*
div - é a divisão de inteiros; dá o quociente.

(2) *mod* - para operandos inteiros; dá o resto da divisão.

(3) variáveis e constantes são "casos simples" de expressões.

(4) **Precedência** dos operadores:

not + - ("unários")
* / *div* *mod* *and*
+ - *or* *xor*
= <> < <= > >= in

E, da esquerda para a direita, para operadores de mesma prioridade.

(5) A prioridade pode ser modificada pelo uso de parênteses.

(6) Operadores relacionais podem ser usados tanto entre operandos numéricos como entre operandos de tipo *char* e *string*.

(7) Expressões lógicas envolvem variáveis, constantes e funções do tipo *boolean*, relações (ié, comparações) e operadores lógicos. A sua avaliação resulta em um valor *false* ou *true*.

(8) No caso de ocorrer "mistura" de tipos numa expressão aritmética,

ocorre a "submissão" de valores inteiros em real. Assim, o resultado será real.

(9) FUNÇÕES (e PROCEDIMENTOS) já disponíveis na linguagem:

(func/proc)	***(tipo do x)***	***(tipo do resultado)***	***(operação)***
sin(x)	real (radianos)	real	seno
cos(x)	real (radianos)	real	cosseno
arctan(x)	real	real	arco-tangente
trunc(x)	real	inteiro	parte inteira
round(x)	real	inteiro	parte inteira "arredondada"
frac(x)	real	real	parte fracionária
abs(x)	real/inteiro	mesmo tipo	valor absoluto
sqr(x)	real/inteiro	mesmo tipo	quadrado
sqrt(x)	real/inteiro	real	raiz quadrada
ln(x)	real	real	logaritmo neperiano
exp(x)	real	real	número *e* elevado a x
randomize	-	-	chama rotina de nºs aleatórios
random(x)	word	word	gera *n* tal que $0<n<x$
random	-	real	gera *k* tal que $0<k<1$
dec(x)	inteiro	inteiro	decrementa x de 1
dec(x,n)	inteiros	inteiro	decrementa x de n
inc(x)	inteiro	inteiro	incrementa x de 1
pi	-	real	3.14...
odd(x)	inteiro	*boolean*	se x é ímpar
ord(x)	qq tipo "ordenado"	inteiro	ordem no tipo de dados
chr(x)	*byte*	*char*	caractere cujo código ASCII é x
pred(x)	qq tipo	mesmo	predecessor

	"ordenado"	tipo	de x
succ(x)	qq tipo "ordenado"	mesmo tipo	sucessor de x
upcase(x)	*char*	*char*	maiúsculo
eof(x)	*file*	*boolean*	se é fim do arquivo
eoln(x)	*text*	*boolean*	se é fim de linha

Obs: outras funções (e procedimentos) devem ser definidas pelo programador.