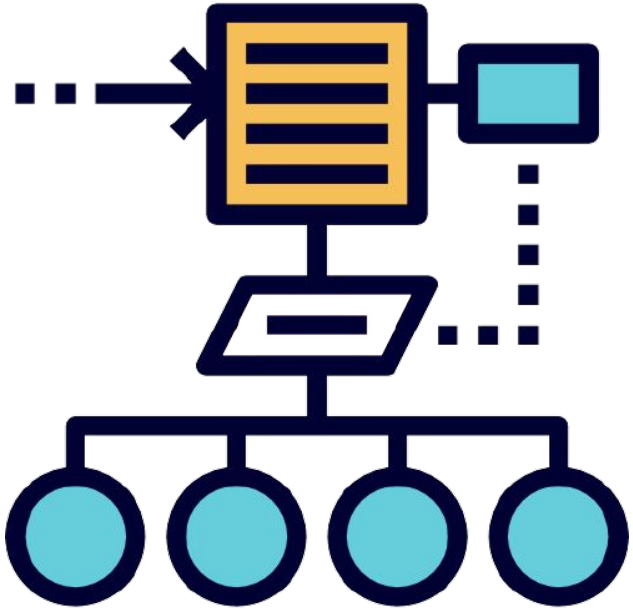
Linguagem de Programação Pascal

Estrutura Sequencial

Introdução: Algoritmo

Algoritmo *é a descrição, de forma lógica, dos passos a serem executados no cumprimento de determinada tarefa.*

Um ***programa*** *é a formalização de um algoritmo em uma determinada linguagem de programação, segundo suas regras de sintaxe e semântica, de forma a permitir que o computador possa entender a seqüência de ações.*



Introdução: Pascal

LINGUAGEM DE PROGRAMAÇÃO PASCAL

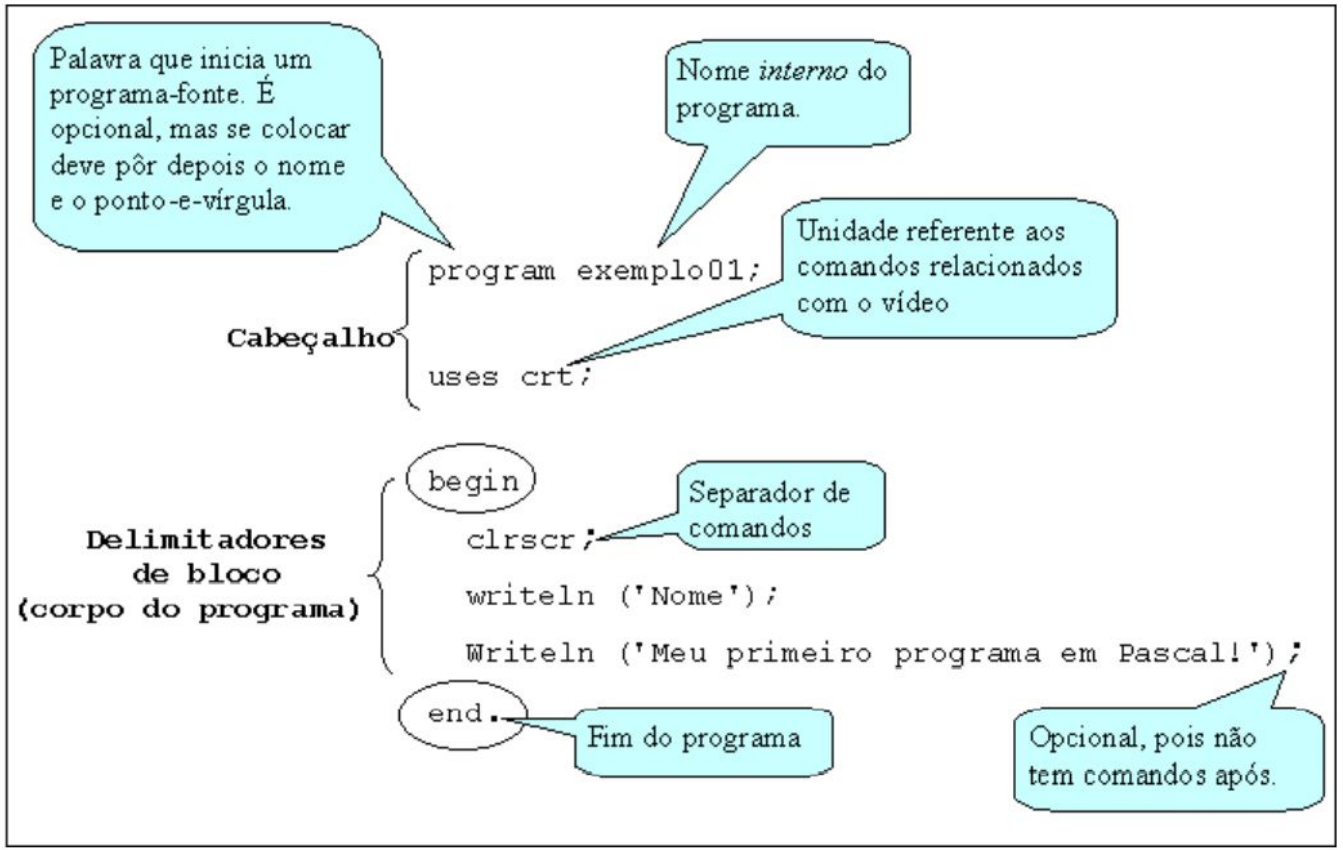
Origem: *desenvolvida nos anos entre 1968 e 1970 por Nicklaus Wirth na Universidade Técnica de Zurique, Suíça. Em 1970 é disponibilizado o primeiro compilador para a linguagem.*

Objetivo: *desenvolver uma linguagem de programação disciplinada de alto nível para ensinar programação estruturada. Esta linguagem foi batizada com o nome de Pascal, em homenagem a Blaise Pascal, filósofo e matemático francês que viveu entre 1623 e 1662.*

Padronização: *ANSI (American National Standars Institute) e IEEE (Institute of Eletrical and Eletronics Engineers)*

Borland International cria em 1983 o Turbo Pascal.

Estrutura de um programa em Pascal



INSTITUTO FEDERAL
Sergipe

Variáveis e Constantes

Os identificadores são os nomes das variáveis, dos programas, das constantes, das rotinas e unidades, procedimentos e funções, entre outras.

As regras básicas para a formação dos identificadores são:

- *Podem ter qualquer tamanho. Entretanto, apenas os 63 primeiros caracteres são utilizados pelo compilador;*
- *Os caracteres que podem ser utilizados na formação dos identificadores são: os números, as letras maiúsculas, as letras minúsculas e o caractere sublinhado;*
- *O compilador não faz distinção entre letras maiúsculas e minúsculas;*
- *O primeiro caractere deve ser sempre uma letra ou o caractere sublinhado;*
- *Não são permitidos espaços em branco e caracteres especiais (@, $, +, , %, !);*
- *Não é permitido usar palavras reservadas.*

Palavras reservadas do Pascal

São nomes utilizados pelo compilador para representar comandos, operadores e nomes de seções de programas.

and	downto	In	packed	to
array	else	inline	procedure	type
asm	End	interface	program	unit
begin	File	Label	record	until
case	For	mod	repeat	until
const	Foward	nil	set	uses
constructor	Function	not	shl	var
destructor	Goto	object	shr	while
div	If	of	string	with
do	implementation	or	then	xor

Link – http://pascalzimbr.blogspot.com/p/linguagem.html

INSTITUTO FEDERAL
Sergipe

Tipos de Dados mais Utilizados

TIPO	FAIXA DE VALORES	TAMANHO (aproximado)
shortint	–128 a 127	8 bits
integer	–32.768 a 32.767	16 bits
longint	–2.147.483.648 a 2.147.483.647	32 bits
byte	0 a 255	8 bits
word	0 a 65.535	16 bits
real	$2{,}9 \times 10^{-39}$ a $1{,}7 \times 10^{38}$ (11 a 12 dígitos com sinal)	6 bytes
single	$1{,}5 \times 10^{-45}$ a $3{,}4 \times 10^{38}$ (7 a 8 dígitos com sinal)	4 bytes
double	$5{,}0 \times 10^{-324}$ a $1{,}7 \times 10^{308}$ (15 a 16 dígitos com sinal)	8 bytes
extended	$3{,}4 \times 10^{-4932}$ a $1{,}1 \times 10^{4932}$ (19 a 20 dígitos com sinal)	10 bytes
comp	$-9{,}2 \times 10^{18}$ a $9{,}2 \times 10^{18}$ (19 a 20 dígitos com sinal)	8 bytes
boolean	true ou false	8 bits
wordbool	true ou false	16 bits
longbool	true ou false	32 bits
bytebool	true ou false	8 bits
char	1 caractere qualquer	1 byte
string	cadeia de caracteres (no máximo 255)	tantos bytes quantos forem os caracteres

Tipo de dados Inteiro

São caracterizados tipos inteiros, os dados numéricos positivos ou negativos, excluindo-se qualquer número fracionário.

Em Pascal, este tipo de dado pode ser referenciado por um dos seguintes identificadores:

Tipo de dado inteiro	Faixa de abrangência	Tamanho(bytes)
shortint	de –128 até 127	1 byte
integer	de –32.768 a 32.767	2 bytes
longint	de –2.147.483.648 a 2.147.483.647	4 bytes
byte	de 0 até 255	1 byte
word	de 0 até 65535	2 bytes

Tipo de Dados Reais

O tipo de dado real permite trabalhar com números fracionários, tanto positivos como negativos, sendo sua capacidade de armazenamento maior que dos números inteiros.

Vejamos os tipos:

Tipo de dado real	Faixa de abrangência	Tamanho(bytes)
real	de 2.9 e-39 até 1.7 e38	6 bytes
single	de 1.5 e-45 até 3.4 e38	4 bytes
double	de 5.0 e-324 até 1.7e308	8 bytes
extended	de 3.4 e-4.932 até 1.1 e4.932	10 bytes
comp	de –9.2 e18 até 9.2 e18	8 bytes

Tipos de Dados Caracteres: String

São considerados tipos caracteres, as seqüências contendo letras, números e símbolos especiais.

Uma seqüência de caracteres, em Pascal, deve ser representada entre apóstrofos (‘’).

Este tipo de dado é referenciado pelo identificador string, podendo armazenar de 1 até 255 caracteres.

Tipos de Dados Caracteres

Podemos ainda especificar um tamanho menor do que os 255 caracteres permitidos.

Vejamos a sintaxe para criarmos uma variável do tipo string com tamanho limitado.

```
1 Program variavel ;
2 var
3 palavra:string[4];
4 Begin
5 writeln ('Digite uma palavra com 4 letras: ');
6 readln(palavra);
7 writeln (palavra);
8 End.
```

Pascalzim Console

```
Digite uma palavra com 4 letras:
Abacaxi - apesar de ter digitado 7 letras
Abac
1 2 3 4 - exibe apenas as 4 primeiras letras digitadas
```

Tipo de Dados Caracter: Char

Existe ainda o tipo char, utilizado da mesma forma que o tipo string, porém com uma pequena diferença: é usado para strings de apenas um caracter.

Vejamos um exemplo do tipo de dado char:

```
var
   Sexo : char;
```

Observações:

a) Em Pascal, os caracteres literais são representados entre apóstrofos.

Ex.: 'maria dos anjos'

b) Os números reais utilizamos ponto como separador decimal.

Ex.: 3.00 ao invés de 3,00

c) Cada comando é finalizado com o sinal de ponto e vírgula.

Declaração de Variáveis

As variáveis são declaradas após a palavra VAR

Os tipos mais utilizados são:

- *INTEGER (para números inteiros),*
- *REAL (para números reais),*
- *CHAR (para um caractere),*
- *STRING (para vários caracteres) e*
- *BOOLEAN (para verdadeiro ou falso).*

```
VAR X: INTEGER;
    Y,Z: REAL;
    NOME: STRING;
    SEXO: CHAR;
    TESTE: BOOLEAN;
```

Declaração de Constante

As constantes são declaradas após a palavra CONST e seus valores não podem ser alterados durante a execução do programa.

```
CONST X = 8;
      Y = 2.8;
      NOME = 'MARIA';
      SEXO = 'm';
      TESTE = TRUE;
```

```
program Area_Circulo;
{ Programa para calcular a área de um círculo. }

const
  PI = 3.141519265;
var
  Area, Comprimento, Raio   : real;

begin
  writeln( 'Digite o Raio : ' );
  readln( Raio );
  Area := PI * Raio * Raio;
  Comprimento := 2 * PI * Raio;
  writeln( 'Área = ', Area );
  writeln( 'Comprimento da Circunferencia = ',    Comprimento );
end.
```

Operador e Expressão

Operador

O símbolo + é um operador que representa a operação aritmética de adição.

Expressão

Uma expressão é um arranjo de operadores e operandos.

A cada expressão válida é atribuído um valor numérico.

4 + 6 é uma expressão cujo valor é 10.

4 e 6 são operandos.

Os operandos podem ser variáveis, constantes ou valores gerados por funções.

Operador de Atribuição

O operador de atribuição é utilizado para atribuir o valor de uma expressão a uma variável.

Sintaxe:

identificador_variável := expressão;

Ex;

A := 10;

Nome := ´José´;

Operador Aritmético

Os operadores aritméticos são utilizados para efetuar operações aritméticas com número inteiros e reais.

Operador	Símbolo
Subtração	-
Adição	+
Multiplicação	*
Divisão Real	/
Divisão Inteira (truncada)	div
Resto da Divisão Inteira	mod
Inverte Sinal	-
Mantém Sinal	+

Operador de Concatenação

O operador de concatenação efetua a junção de duas variáveis ou constantes do tipo string.

```
var
  PreNome, SobreNome, NomeCompleto : string[ 30 ];

begin
{ Suponhamos o nome Josias Lima Santos }
   PreNome      := ´Josias´;
   SobreNome    := ´Santos´;

   NomeCompleto := PreNome + SobreNome;
   writeln( NomeCompleto );

   NomeCompleto := ´Jose´ + ´Maria´;
   writeln( NomeCompleto );
   ...
end.
```

Operadores Relacionais

Os operadores relacionais são utilizados para efetuar a comparação entre dados de mesmo tipo.

Operador	Símbolo
Maior que	>
Menor que	<
Maior ou igual	>=
Menor ou igual	<=
Igual	=
Diferente	<>

Operadores Relacionais

Os operadores relacionais são utilizados para efetuar a comparação entre dados de mesmo tipo.

Exemplo:

```
var
  Nota1, Nota2              : real;
  NomeAluno1, NomeAluno2 : string[ 30 ];
  A, B, C                   : integer;

begin
  A := 2;
  B := 3;
  C := 1;
  if B = A + C then
    writeln( B );
  Nota1 :=  5.0;
  Nota2 := 10.0;
  if Nota1 < Nota2 then
    writeln( Nota1 );
  NomeAluno1 := ´Maria Jose´;
  NomeAluno2 := ´MariaJose´;
  if NomeAluno1 < NomeAluno2 then
    writeln( NomeAluno1 );
end.
```

Operador Lógico

Os operadores lógicos são utilizados para se analisar duas ou mais expressões interrelacionadas.

Operador	Símbolo
E	and
OU	or
NÃO	not

```
var
  Nota1, Nota2 : real;
  NomeAluno1, NomeAluno2 : string[ 30 ];
  A, B, C : integer;

begin
  A := 2;
  B := 3;
  C := 1;
  NomeAluno1 := ´Maria Jose´;
  NomeAluno2 := ´MariaJose´;

  if ( B = A + C ) and ( NomeAluno1 <> NomeAluno2 ) then
     writeln( NomeAluno1, B );
  if ( A = C ) or ( NomeAluno1 = NomeAluno2 ) then
     writeln( NomeAluno1 );
  if not( A = C ) then
     writeln( NomeAluno1 );
end.
```

INSTITUTO FEDERAL
Sergipe

Funções Predefinidas

Nome Função	Objetivo	Tipo Parâmetro	Tipo do Retorno
abs(x)	Calcula o valor absoluto de x.(Positivo)	inteiro ou real	o mesmo que x
cos(x)	Calcula o coseno de x em radianos	inteiro ou real	real
exp(x)	Calcula e^x, em que e=2.7182818 é sistema natural de logaritmos neperianos.	inteiro ou real	real
ln(x)	Calcula o logaritmo natural de x (x>0)	inteiro ou real	real
exp(ln(x)*y)	Retorna x elevado a y {utilizando regras de logaritmos}.	Inteiro ou real	real
sin(x)	Calcula o seno de x em radianos	inteiro ou real	real
sqr(x)	Calcula o quadrado de x	inteiro ou real	o mesmo que x
sqrt(x)	Calcula a raiz quadrada de x (x>=0)	inteiro ou real	real
odd(x)	Determina se x é par ou impar TRUE, X é impar FALSE, X é par	inteiro	boolean
random(x)	Retorna um número pseudo-aleatório entre 0 e x. Se x não for especificado retorna um valor entre 0.0 e 1.0	inteiro	real
pi	Retorna o valor de PI (3.1415...)	Nenhum	real

INSTITUTO FEDERAL
Sergipe

Funções Predefinidas

Funções Literais:

Nome Função	Objetivo	Tipo Parâmetro	Tipo Retorno
length(x)	Determina o número de caracteres de x	string	inteiro
concat(x1, x2, x3,...)	Concatena duas ou mais strings (máx 255 caracteres)	string	string
copy(x, y, z)	Retorna uma subcadeia da cadeia x, com z caracteres, começando no caracter y.	string, inteiro, inteiro	string
UpCase(x)	Retorna x convertido para maiúscula	char	char

Funções Predefinidas

Funções para Conversão:

Nome Função	Objetivo	Tipo Parâmetro	Tipo Retorno
trunc(x)	Trunca x para um número inteiro	real	inteiro
int(x)	Retorna a parte inteira de x	real	real
frac(x)	Retorna a parte fracionária de x	real	real
round(x)	Arredonda x para um inteiro	real	inteiro
chr(x)	Determina o caracter ASCII representado por x	inteiro	char

Funções e Procedimentos de Uso Geral:

Nome Função	Objetivo	Tipo Parâmetro	Tipo do Retorno
sizeof(x)	Retorna o número de byte de x	qualquer tipo	inteiro
gotoxy(x,y)	Move o curso para a coluna x e linha y	inteiro	

Funções Predefinidas

Por não existir o operador de potenciação, temos:

$$AB = EXP(B*LN(A))$$

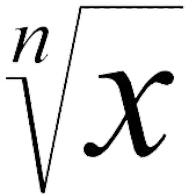
Exemplo:

$3^4 = exp(4*ln(3))$

$5^{10} = exp(10*ln(5))$

Por não existir o operador de raiz, temos:

raíz cúbica: exp(ln(X)/3).



Comando de Entrada em Pascal

O comando de entrada é utilizado para receber dados digitados pelo usuário.

Esses dados são armazenados em variáveis.

Esse comando é representado pela palavra READ ou READLN .

Sua sintaxe

sintaxe:

```
READLN(nome_da_variável);
READLN(nome_da_variável1,nome_da_variável2);
```

Comando de Saída em Pascal

O comando de saída é utilizado para mostrar dados na tela ou na impressora.

Esse comando é representado pelas palavras WRITE ou WRITELN e os dados podem ser conteúdos de variáveis ou mensagens.

A diferença entre esses comandos é que o comando WRITELN mostra seu conteúdo e passa o cursor para a linha de baixo, enquanto o comando WRITE mantém o cursor na mesma linha após mostrar a mensagem.

```
WRITELN('Conteúdo de Y = ',Y);
WRITE('Conteúdo de Y = ',Y);
```

INSTITUTO FEDERAL
Sergipe

Comando de Saída em Pascal

sintaxe:

```
WRITE(nome_da_variável);
WRITELN(nome_da_variável);
WRITE('mensagem');
WRITELN('mensagem');
WRITE('mensagem', nome_da_variável);
WRITELN('mensagem', nome_da_variável);
```

Casas decimais - comando de saída

Nos comandos de saída, é possível ainda fazer a formatação de variáveis do tipo real, single, double, extended e comp.

Após o nome da variável, coloca-se ***:m:n****, onde*

m *significa a quantidade de espaços da tela e*

n *o número de caracteres gastos com a parte fracionária do número.*

O ponto, que é o separador decimal, ocupará um caractere do total de caracteres.

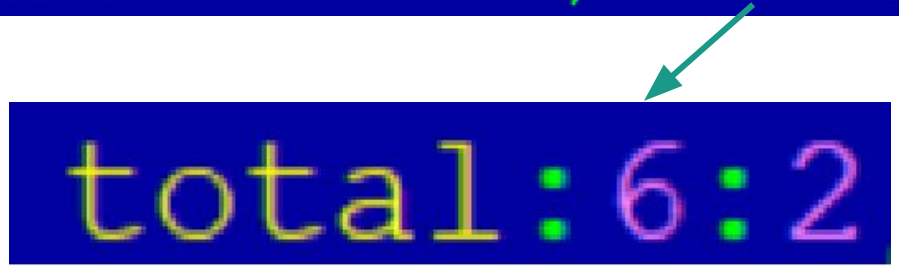
Os espaços que sobram à esquerda serão preenchidos com branco, e, quando faltam espaços, o compilador completa com a necessidade para mostrar o resultado.

Casas decimais - comando de saída

Exemplo:

O conteúdo da variável total ocupará 6 espaços na tela. Dois espaços para a parte fracionária, um espaço para o ponto e três espaços para a parte inteira.

```
writeln ('O valor total foi de: ', total:6:2, ' reais.');
```




```
O valor total foi de: 130.00 reais.
```

INSTITUTO FEDERAL
Sergipe

Exercício

```
1  Program funcao ;
2  uses crt;
3
4  Const
5  A = -10;
6  B = 5.8;
7  C = 9;
8  D = 3;
9
10 Begin
11
12 writeln ('O valor da constante A é: ',A);
13 writeln ('O valor da constante A positivo é: ',abs(A));
14 writeln ('A parte inteira da constante B = ', B:5:2, ' é: ',trunc(B));
15 writeln ('A parte fracionária da constante B = ', B:5:2, ' é: ',frac(B):5:2);
16 writeln ('A raiz quadrada da constante c = ', C, ' é: ',sqrt(C):5:2);
17 writeln ('A constante c = ', C, ' elevado ao quadrado é: ',sqr(C));
18 writeln ('A constante c = ', C, ' elevado a D = ',D,' é: ',exp(d*ln(c)):0:0);
19 readkey;
20 End.
```

Exercício:

Dado as seguintes informações de um funcionário: Nome, idade cargo e o seu salário bruto considere:

a) **O salário bruto teve um reajuste de 38%.**
b) **O funcionário receberá uma gratificação de 20% do salário bruto.**
c) **O Salário total é descontado em 15%**

Faça um algoritmo e um algoritmo para:

- **Imprimir Nome, idade e cargo.**
- **Imprimir o salário bruto.**
- **Imprimir o salário líquido.**

Exercício

Uma empresa tem para um determinado funcionário uma ficha contendo o nome, número de horas trabalhadas e o n^0 de dependentes de um funcionário.

Considerando que:

a) A empresa paga 12 reais por hora e 40 reais por dependentes.

b) Sobre o salário são feito descontos de 8,5% para o INSS e 5% para IR.

Faça um algoritmo para ler o Nome, número de horas trabalhadas e número de dependentes de um funcionário. Após a leitura, escreva qual o Nome, salário bruto, os valores descontados para cada tipo de imposto e finalmente qual o salário líquido do funcionário.

Faça um programa que receba o salário base de um funcionário, calcule e mostre o salário a receber, sabendo-se que o funcionário tem gratificação de 5% sobre o salário base e paga imposto de 7% também sobre o salário base.

O custo ao consumidor de um carro novo é a soma do preço de fábrica com o percentual de lucro do distribuidor e dos impostos aplicados ao preço de fábrica. Faça um programa que receba o preço de fábrica de um veículo, o percentual de lucro do distribuidor e o percentual de impostos, calcule e mostre:

a) o valor correspondente ao lucro do distribuidor;

b) o valor correspondente aos impostos;

c) o preço final do veículo.