

Pascal (Introdução)

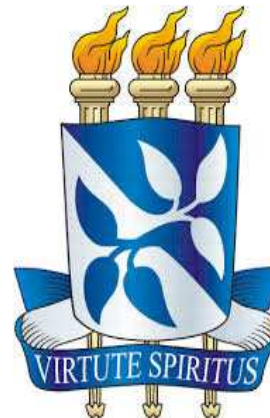
MATA37: Introdução à Lógica de Programação

Prof.: Rafael A. Melo (melo@dcc.ufba.br)

Departamento de Ciência da Computação

Instituto de Matemática

Universidade Federal da Bahia



Linguagens de Alto Nível

- Linguagem com um nível de abstração relativamente elevado, longe do código de máquina e mais próximo à linguagem humana.
- Ex.: Pascal, C, C++, C#, Delphi, Visual Basic, Java, etc.

Linguagens de Alto Nível

- Um algoritmo escrito em uma linguagem de alto nível é chamado *programa fonte* ou simplesmente *programa*.
- Um *programa fonte* deve ser traduzido para a linguagem de máquina
 - Interpretadores;
 - Compiladores.

Linguagens de Alto Nível

- Sintaxe
 - Conjunto de regras que define a forma de uma linguagem, estabelecendo como são compostas as suas estruturas básicas (palavras).
 - Como cada instrução deve ser escrita.
- Semântica
 - A semântica descreve o significado de construções sintáticas válidas.
 - Ação resultante da execução de uma instrução.

Sobre Pascal

- Linguagem de programação desenvolvida em 1970 por Niklaus Wirth.
- Simples para estimular boas práticas de programação.
- Linguagem usada para ensino de programação.
- Linguagem imperativa.
- Fortemente tipada.
- Compilada.

Cronograma do restante da aula

- Estrutura básica
- Tipos primitivos de dados
- Constantes e variáveis
- Expressões lógicas e aritméticas, precedência entre operadores
- Identificadores, símbolos especiais, números, cadeias de caracteres
- Comandos de atribuição
- Comandos de entrada e saída
- Estruturas de decisão

Primeiro programa: Hello World

```
program Hello;  
begin  
    writeln ( 'Hello, world.' );  
end.
```

Estrutura Básica

program NomeDoPrograma;

uses

(* declaração de bibliotecas usadas *)

const

(* declaração de constantes *)

type

(* declaração de tipos *)

var

(* declaração de variáveis *)

(* definições de procedimentos e funções *)

begin

(* programa principal *)

end.

Variáveis

var Lista de identificadores: tipo de dado;

var

Quant: **integer**;

Num, Soma, Media: **real**;

Falhou: **boolean**;

Resposta: **char**;

Contantes

- Uma *constante* armazena um valor fixado pelo programa que não pode ser alterado durante sua execução.

const

```
Identificador1 = valor1;  
Identificador2 = valor2;  
Identificador3 = valor3;
```

const

```
nome = 'claudio';  
primeiraLetra = 'c';  
ano = 2008;  
pi = 3.1415926535897932;  
achou = TRUE;
```

```
a : real = 12;
```

Identificadores

- Seqüência de letras, dígitos e caractere para sublinhamento.
- O primeiro caractere do identificador não pode ser um dígito.
- Não diferencia minúsculas e maiúsculas
- Palavras reservadas
 - Uso -> Erro sintático

Identificadores

- Palavras reservadas.

and	downto	In	or	then
asm	else	Inline	packed	to
array	end	Interface	procedure	type
begin	exports	Label	program	unit
case	file	Library	record	until
const	for	Mod	repeat	uses
constructor	function	Nil	set	var
destructor	goto	Not	shl	while
div	if	Object	shr	with
do	implementation	Of	string	xor

Tipos de Dados

- **char**: letras, dígitos e caracteres especiais como ':', '?', e '*'.
- **shortint**: números inteiros do intervalo [-128, 127].
- **integer**: números inteiros do intervalo [-32768, 32767].
- **longint**: números inteiros do intervalo [-2147483648, 2147483647].
- **byte**: números inteiros do intervalo [0, 255].
- **word**: números inteiros do intervalo [0, 65535].
- **real**: números reais do conjunto $[-3,4 \times 10^{38}, 3,4 \times 10^{-38}]$.
- **boolean**: conjunto dos valores **false** (falso) e **true** (verdadeiro).

Comentários

- (* isso é um comentário *)
- { isso é um comentário }

Expressões Aritméticas

- $+$: adição
- $-$: subtração
- $*$: multiplicação
- $/$: divisão
- div : quociente
- mod : resto
- $()$: Parênteses
 - Ex: $2 + (3 * 4)$
- Precedência
 1. $()$
 2. operador unário
 3. div , mod , $*$, $/$
 4. $+$, $-$

Expressões Relacionais

- > maior do que
- >= maior do que ou igual a
- < menor do que
- <= menor do que ou igual a
- = igual
- <> diferente

Expressões Lógicas

- and
 - or
 - not
-
- operadores relacionais têm prioridade mais baixa que os operadores lógicos.

Comando de Atribuição

- Identificador de variável **:=** expressão;

$a := 2.8;$

$b := a * a / 2;$

Entrada e Saída de Dados

- **readln**(lista de identificadores);
- **read**(lista de identificadores);
- **writeln**(lista de identificadores/Expressão/Mensagem);
- **write**(lista de identificadores/Expressão/Mensagem);

```
program TesteEntradaSaida  
var c: char;  
begin  
    writeln('Digite um caractere');  
    readln(c);  
    writeln('Você digitou' , c, '.');  
end.
```

Você digitou 5.

Entrada e Saída de Dados

```
program MediaDe3;  
var a, b, c : real;  
begin  
    write('Digite três números reais: ');  
    readln(a, b, c);  
    writeln('A média dos números ', a, ', ', b, ' e ', c, ' é ', (a + b + c) / 3);  
end.
```

Digite três números reais: 8.43 6.1 13.145

A média dos números 8.4300000000E+00, 6.1000000000E+00 e
13.1450000000 é 9.2250000000E+00

Entrada e Saída de Dados

- Como padrão, as expressões ou mensagens num comando `write` ou `writeln` são exibidas na primeira coluna à esquerda do monitor.
- É possível alterar esse padrão.

writeln(a:n);	exibe o valor da variável a , reservando n colunas para isso, alinhando pela direita.
----------------------	---

por ex., se **a = 125** (inteiro), o comando **writeln(a:20);** exibirá:

[illegible]

Entrada e Saída de Dados

```
writeln(x:n:d);
```

para uma variável real x , o parâmetro d define o número de casas decimais a serem exibidos .

por ex., se $a = 125$ e $b = 48.125$, ambas reais, os comandos

```
writeln(a:20:8);
```

```
writeln(b:10:2);
```

exibirão:

[illegible]

Exemplo

```
program converteC2F;  
var  
    celsius, fahrenheit: real;  
begin  
    readln(celsius);  
    fahrenheit := (9*celsius)/5 + 32;  
    writeln (fahrenheit);  
end.
```

Exercício 1

- Escreva um programa em Pascal que determine o algarismo da casa das unidades de um inteiro fornecido pelo usuário.

Exercício 2

- Escreva um programa em Pascal que inverta um número com dois algoritmos.

Exercício 3

- Escreva um programa em Pascal que permuta os conteúdos de duas variáveis.

Algumas Funções Predefinidas

- Funções da unidade System

Função	Argumento	Valor	Valor calculado
Abs(x)	real/integer	mesmo	valor absoluto de x
ArcTan(x)	real	real	número cuja tangente é x
Chr(x)	byte	char	caractere cujo código ASCII é x
Cos(x)	real	real	coseno de x
Exp(x)	real	real	exponencial de x (e^x)
Frac(x)	real	real	parte fracionária de x
Ln(x)	real	real	logaritmo natural de x
Odd(x)	integer	boolean	verifica se x é ímpar
Ord(x)	tipo ordenado	integer	ordem de x no tipo de dado
Pred(x)	tipo ordenado	mesmo	antecessor de x
Round(x)	real	integer	arredondamento de x
Sin(x)	real	real	seno de x
Sqr(x)	real/integer	mesmo	quadrado de x
SqrT(x)	real/integer	real	raiz quadrada x
Succ(x)	tipo ordenado	mesmo	sucessor de x
Trunc(x)	real	integer	parte inteira de x
UpCase(x)	char	char	X maiúsculo

Funções Predefinidas

Outras unidades -> depende do compilador.

```
program LimpaTela;  
    uses Crt;  
begin  
    ClrScr;  
end.
```

Exercício 4

- Escreva um programa que determine a área de um triângulo de lados de comprimentos dados
 - a área do triângulo cujos lados têm comprimentos x , y e z é dada por

$$S = \sqrt{p \cdot (p - x) \cdot (p - y) \cdot (p - z)}$$

- onde $p = \frac{x + y + z}{2}$ é o semiperímetro do triângulo

Estruturas de Seleção

- O comando *if - then*

```
if expressão lógica then  
  begin  
    seqüência de comandos;  
  end;
```

```
if expressão lógica then  
  único comando;
```

Estruturas de Seleção

{Programa para determinar o maior de dois números dados}

program MaiorDe2;

var a, b, Maior : **real**;

begin

writeln('Digite dois números');

readln(a, b);

 Maior := a;

if (b > a) **then**

 Maior := b;

writeln('O maior dos números ', a, ' e ', b, ' é ', Maior);

end.

Estruturas de Seleção

{Programa para ordenar os conteúdos de duas variáveis}

program Ordena2;

var x, y, Aux : **real**;

begin

writeln ('**Digite os dois números** ');

readln(x, y);

writeln('**Números digitados: x =** ', x, ' **y =** ', y);

if x > y **then**

begin

 Aux := x;

 x := y;

 y := Aux;

end;

writeln('**Números ordenados: x =** ', x, ' **y =** ', y);

end.

Estruturas de Seleção

- O comando *if-then-else*

```
if Expressão lógica
  then
    begin
      seqüência de comandos;
    end
  else
    begin
      seqüência de comandos;
    end;
end;
```

Exercício 5

- Altere o programa que calcula a área do triângulo de forma que:
 - Antes fazer o cálculo verifique se os valores fornecidos são comprimentos válidos de lados de um triângulo
 - Cada lado é menor que a soma dos outros dois
 - Se não, o programa deve apresentar uma mensagem informando que aqueles valores não podem ser os lados de um triângulo

Exercício 5: algoritmo anterior

{Programa que determina a área de um triângulo a partir de} {seus lados}

program AreaTriangulo;

var a, b, c, SemiPer, Area : **real**;

begin

writeln('Digite os lados do triângulo')

readln(a, b, c);

 SemiPer := (a + b + c)/2;

 Area := **SqrT**(SemiPer*(SemiPer - a)*(SemiPer - b)*(SemiPer - c));

writeln('Área do triângulo = ', Area);

end.

Exercício 6

Suponhamos que uma empresa decidiu dar um aumento escalonado a seus funcionários de acordo com a seguinte regra: 13% para os salários inferiores ou iguais a R\$ 200,00; 11% para os salários situados entre R\$ 200,0 e R\$ 400,00 (inclusive); 9 % para os salários entre R\$ 400,00 e R\$ 800,00 (inclusive) e 7% para os demais salários. Escreva um programa que receba o salário atual de um funcionário e forneça o valor do seu novo salário.

Estruturas de Seleção

- O comando **case**

case *Expressao* **of**

Lista de valores 1 : seqüência de comandos 1;

. . .

Lista de valores n : seqüência de comandos n;

else seqüência de comandos x;

end;

Estruturas de Seleção

```
program GerenciaCaixaEletronico;  
var Opcao :char;  
...  
begin  
    writeln('1. Extrato');  
    writeln('2. Saldo');  
    writeln('3. Saque');  
    writeln('4. Depósito');  
    write(' Digite sua opção > ');  
    readln(Opcao);  
    case Opcao of  
        '1' : writeln('Digitou a opcao 1');  
        '2' : writeln('Digitou a opcao 2');  
        '3' : writeln('Digitou a opcao 3');  
        '4' : writeln('Digitou a opcao 4');  
    else  
        writeln('Opcao invalida');  
    end;  
end.
```

Estruturas de Seleção

```
program ConvertePontuacaoEmConceito;  
var NPontos: integer;  
Conc : char;  
begin  
    writeln('Digite a pontuação obtida');  
    readln(NPontos);  
    case NPontos of  
        0 .. 49:   Conc := 'D';  
        50 .. 69 : Conc := 'C';  
        70 .. 89 : Conc := 'B';  
        90 .. 100: Conc := 'A';  
    end;  
    writeln('O numero de pontos igual a ', NPontos, ' corresponde a um  
conceito ', Conc);  
end.
```

Pascal (Introdução)

MATA37: Introdução à Lógica de Programação

Prof.: Rafael A. Melo (melo@dcc.ufba.br)

Departamento de Ciência da Computação

Instituto de Matemática

Universidade Federal da Bahia

