

Programação de Computadores I

Aula 03

Linguagem algorítmica: Portugol

José Romildo Malaquias

Departamento de Computação
Universidade Federal de Ouro Preto

2011-1

Introdução I

Lógica

A **lógica** é usada no dia a dia das **pessoas** que trabalham com **computação** para **solucionar problemas** de forma **eficiente**.

Algoritmo

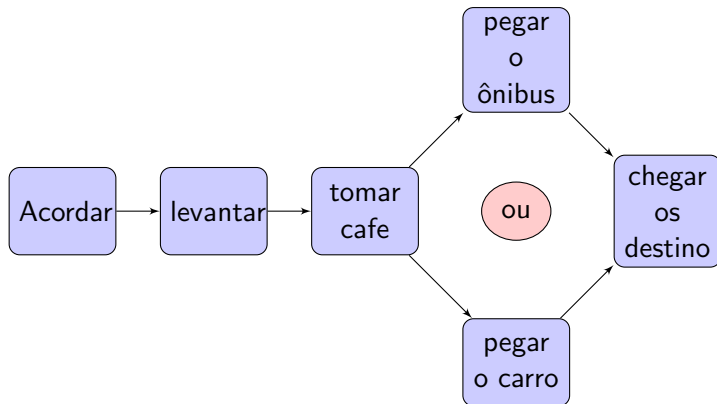
Um algoritmo representa de forma estruturada, uma seqüência de ações, que levam a um resultado esperado.

Resumindo:

- ▶ algoritmo: exercício de raciocínio (definir o problema);
- ▶ técnicas de programação: exercício da implementação

Introdução II

- ▶ Exemplo: Seqüência de ações para chegar ao trabalho/universidade



Introdução III

- ▶ Para cada ação acontecer, é necessário que a ação anterior tenha sido executada
- ▶ Cada ação pode conter outros eventos associados (outros algoritmos)

Portugol I

Portugol

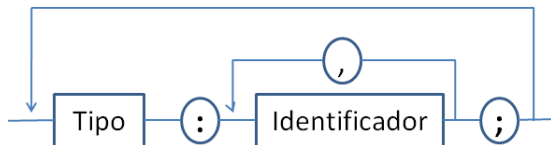
É uma pseudolinguagem que permite ao programador pensar no problema em si e não no equipamento que irá executar o algoritmo.

Estrutura de um algoritmo

```
Inicio
    <declarações de variáveis>
    <comandos>
Fim
```


► **Variável**

- ▶ Variável é um local que armazena um tipo específico de conteúdo
 - ▶ Contém um valor que se modifica durante a execução de um programa.
 - ▶ É identificada por um nome (identificador), que pode ser representando:
- ▶ Declaração de variáveis:



Portugol VI

- ▶ **Inteiro:** qualquer número inteiro (negativo, nulo ou positivo)
Exemplo: 100, 0, 1, 2, 1250
- ▶ **Real:** qualquer número real (negativo, nulo ou positivo)
Exemplo: -10, -1.5, 11.2, 0, 1, 2, 50
- ▶ **Caracter:** caracteres alfanuméricos
Exemplo: casa, Win31, 123
- ▶ **Lógico:** valor lógico *verdadeiro* ou *falso*
Exemplo: $x > y$?

Portugol VII

► Exemplo:

```
inteiro: idade;  
real: nota1, nota2, media;  
character: nome_aluno;  
logico: maior;
```

Portugol VIII

- ▶ É importante não esquecer

1. Não é possível definir variáveis de diferentes tipos com o mesmo identificador.

Exemplo: `real A; inteiro A;` causaria erro na programação.

2. Tomar cuidado em relação à sintaxe da linguagem. Não é possível ter identificador como:

```
caracter nome—empregado;  
real valor*;  
inteiro 8x;  
logico positivo?;
```

3. Letras maiúsculas e minúsculas são tratadas de forma diferente.

Exemplo: *Media* é diferente de *media*, como também de *MEDIA*;

► Constantes

- Uma constante é um valor fixo que não se modifica ao longo do tempo
- Em algoritmo representaremos constantes pelo tipo **const** ou **#define** (eventualmente alguns elementos da linguagem C podem ser escritos no algoritmo)

```
const M 10;
```

► Comandos básicos:

- O comando de **atribuição** é utilizado para atribuir um valor a uma variável.
- Para isso usamos o símbolo “←”



- ▶ A notação usada para expressões é basicamente uma forma **linear** comumente usada na matemática, que pode conter operadores:
 - ▶ Aritméticos: $+$, $-$, $*$, $/$, $\text{raiz}()$, \wedge , $\text{sen}()$, $\text{cos}()$, mod , div , ...
 - ▶ Lógicos: e , ou , não
 - ▶ Relacionais: $=$, \neq , $>$, \geq (ou \geq), $<$, \leq (ou \leq)

Portugol XII

Exemplos:

- ▶ Atribuição de um valor constante

```
inteiro valor;  
valor <- 10;
```

- ▶ Atribuição entre variáveis

```
inteiro valor;  
inteiro x;  
x <- 10;  
valor <- x;
```

- ▶ Resultado de expressões

```
inteiro valor;  
inteiro x, y;  
x <- 10;  
y <- 5;  
valor <- x + y * 2;
```


Exercício 1 I

Desenvolva um algoritmo em portugol para somar dois valores inteiros (10+5)

Inicio

```
inteiro x, y, z; //declara três variáveis inteiras  
x ← 10; // atribui 10 para x  
y ← 5; // atribui 5 para y  
z ← x + y; // soma x e y, o resultado é atribuído a z
```

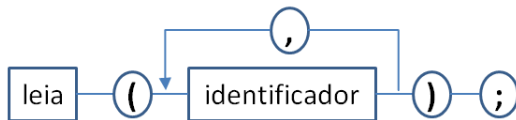
Fim

Entrada e Saída de dados I

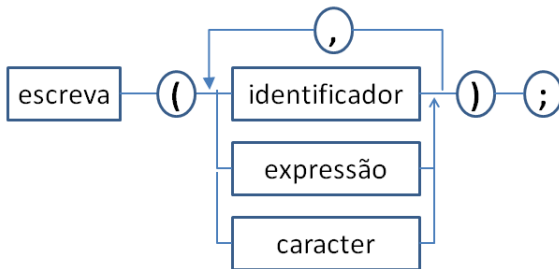
- ▶ Um algoritmo pode receber um dado informado através de um comando de **leitura**
- ▶ Também pode ser necessário conhecer o resultado de uma determinada operação, nesse caso usaremos um comando de **escrita**

Entrada e Saída de dados II

O comando de entrada é **leia**



O comando de saída é **escreva**



Exercício 2 I

Inserir dois números inteiros e encontrar a soma

Início

```
inteiro: x, y, soma;
```

```
leia(x, y);
```

```
soma ← x + y;
```

```
escreva("A soma entre", x, " e ", y, " é ", soma);
```

Fim

Regras para escrever algoritmos I

- ▶ Incluir comentários
- ▶ Usar nomes significativos para as variáveis que possam identificar o conteúdo
- ▶ Identar os comandos **facilita a legibilidade** do algoritmo e **reduz** a possibilidade de **erros**

Exercício 3 I

Desenvolva um algoritmo que calcule o volume de uma esfera de raio R , fornecido pelo usuário. ($V = 4/3\pi R^3$)

Exercício 3 II

Início

```
const pi 3.14159;  
real: R, volume;  
escreva("Inserir raio da esfera");  
leia(R);  
volume ← 4/3 * pi * (R^3);  
escreva("O volume da esfera: ", volume);
```

Fim

Exercício 3 III

Desenvolva um algoritmo para encontrar a média entre 4 valores fornecidos pelo usuário

Exercício 3 IV

Início

```
real: nota1, nota2, nota3, nota4, media;  
escreva("Inserir quatro notas");  
leia(nota1, nota2, nota3, nota4);  
media <- (nota1 + nota2 + nota3 + nota4) / 4;  
escreva("Valor da media: ", media);
```

Fim

Comandos de Controle I

- ▶ Permite
 - ▶ alterar a direção tomada por um programa, ou (desvio)
 - ▶ fazer com que partes específicas de um algoritmo sejam executadas mais de uma vez (*loop*)

Comandos de Controle II

- ▶ **Desvio condicional:** muitas vezes será necessário desviar a execução do programa segundo uma condição.
 - ▶ Por exemplo, ir a universidade de carro ou de ônibus?
 - ▶ Para se testar condições pode ser necessário utilizar operadores lógicos e relacionais

Comandos de Controle III

► Desvio condicional simples

```
se (condição) então  
    lista de comandos  
fim_se
```

Inserir um número real, se ele for positivo imprimir o número

Comandos de Controle IV

```
Inicio
    inteiro A;
    escreva("Inserir valor ");
    leia(A);
    se A > 0 então
        escreva(A);
    fim_se
Fim
```

Comandos de Controle V

► Desvio condicional composto

- As condições, verdadeiras ou falsas, geram ações através de um único comando de desvio condicional

```
se (condição) entao  
    lista de comandos  
senão  
    lista de comandos  
fim_se
```

Inserir dois valores numéricos e encontrar o maior deles

Comandos de Controle VI

Inicio

```
    real num1, num2;  
    escreva("Inserir dois valores");  
    leia(num1, num2);  
    se num1 > num2  
        escreva("o maior é ", num1);  
    senão  
        escreva("o maior é ", num2);  
    fim_se
```

Fim

Exercícios em aula I

1. Desenvolva um algoritmo que lê o nome de um aluno, as notas a , b e c de suas três provas, calcula e exibe a média harmônica das provas, dada por

$$\frac{3}{\frac{1}{a} + \frac{1}{b} + \frac{1}{c}}$$

2. Faça um algoritmo para calcular a área de um círculo, sendo fornecido o valor do raio, que deve ser positivo.
3. Escreva um algoritmo que leia um número inteiro. Se o número lido for positivo, escreva uma mensagem indicando se ele é par ou ímpar. Se o número for negativo, escreva a seguinte mensagem “Este número não é positivo”.

Exercícios propostos I

1. Crie um algoritmo que faça a soma de dois valores inteiros obtidos através de leitura e imprima o resultado dessa soma. Em seguida leia outro número inteiro e calcule e exiba a quociente da soma calculada anteriormente por este número.
2. Faça um algoritmo que leia o nome de um piloto, uma distância percorrida em km e o tempo que o piloto levou para percorrê-la (em horas). O programa deve calcular a velocidade média

$$\text{Velocidade} = \frac{\text{Distância}}{\text{Tempo}}$$

em km/h, e exibir a seguinte frase:

A velocidade média do *<nome do piloto>* foi *<velocidade media calculada>* km/h.

FIM

Créditos:

Baseado no material preparado pelo
Prof. Guillermo Cámara-Chávez.