



SCC0221 – Introdução à Ciência de Computação I

Prof.: Dr. Rudinei Goularte
(rudinei@icmc.usp.br)

Aulas 5 e 6 – Introdução a Algoritmos

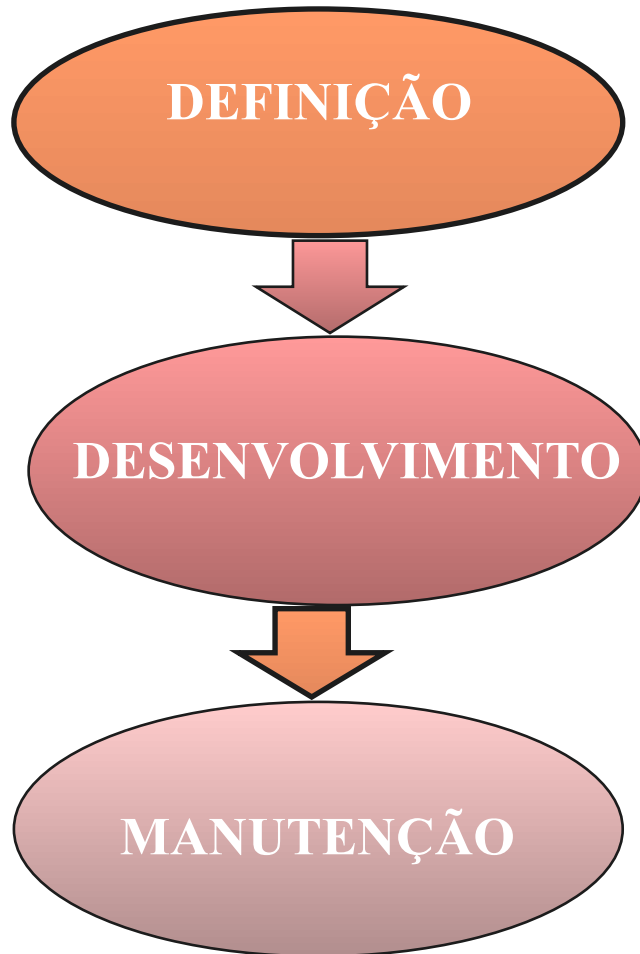
Instituto de Ciências Matemáticas e de Computação - ICMC
Sala 4-229



Sumário

- 1. Etapas da construção de programas.
- 2. Definição de Algoritmo.
- 3. Introdução às estruturas algorítmicas e refinamentos sucessivos.
- 4. Constantes e variáveis.
- 5. Operadores e expressões.
- 6. Entrada e Saída.
- 7. Comandos de seleção.

Programação e Níveis de Linguagem

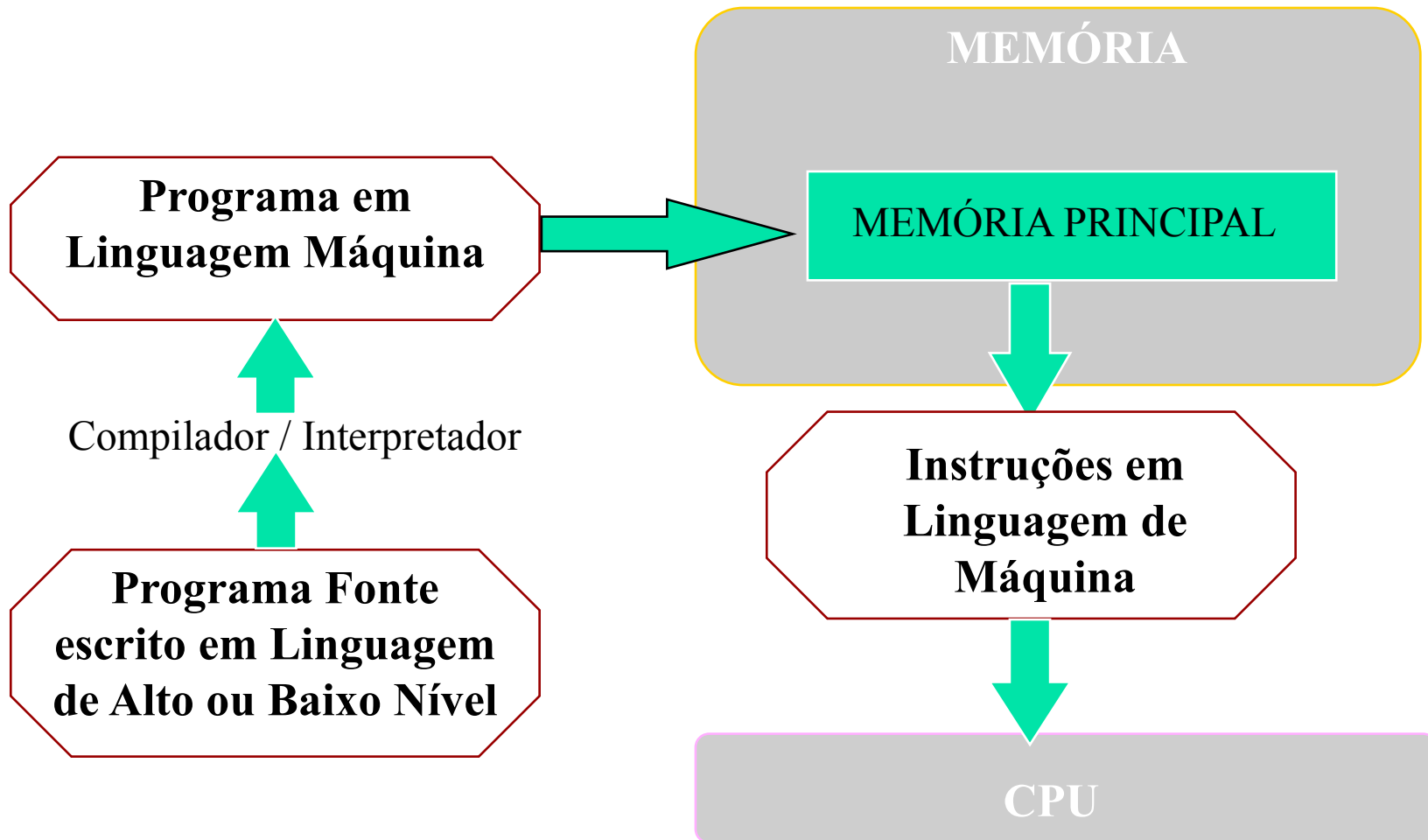


O QUE

COMO

ALTERAÇÕES

Programação e Níveis de Linguagem



Programação e Níveis de Linguagem



■ Linguagem de máquina

- É o conjunto das instruções primitivas projetadas para um computador. Uma CPU somente pode compreender instruções que sejam expressas em termos de sua LINGUAGEM DE MÁQUINA.
- Um programa escrito em linguagem de máquina consiste de uma série de números binários e é muito difícil de ser entendido pelas pessoas.
- Exemplo: Cada instrução é constituída de 2 partes:

o código da operação e o operando

0010 (Load)

011000 (endereço)

0110 (multiplica)

000101 (valor)

...

0010011000

0110000101

...



Programação e Níveis de Linguagem

- Linguagem de baixo nível
 - Os programas são escritos em uma notação que está próxima da linguagem de máquina

Exemplo:

código da operação	operando	significado
LD	A	load 10
MPI	5	multiplica 5



Programação e Níveis de Linguagem

♦ Linguagem de alto nível

- Se pode escrever programas em uma notação próxima à maneira natural de expressar o problema que se deseja resolver.

- Exemplo:

`RESULT = D-((A+B)/C)`

- Aplicações Científicas : FORTRAN, ALGOL, BASIC, APL, LISP, PASCAL, ADA, C, PROLOG, PLI, Java, Python, ...
- Aplicações Comerciais: COBOL, RPG, PLI, C, Java, Ruby, ...



Programação e Níveis de Linguagem

- **COMPILADOR**

- Traduz os comandos simbólicos de uma linguagem de alto nível, para linguagem de máquina.

- **MONTADOR**

- Traduz os comandos simbólicos de uma linguagem de baixo nível , para linguagem de máquina.

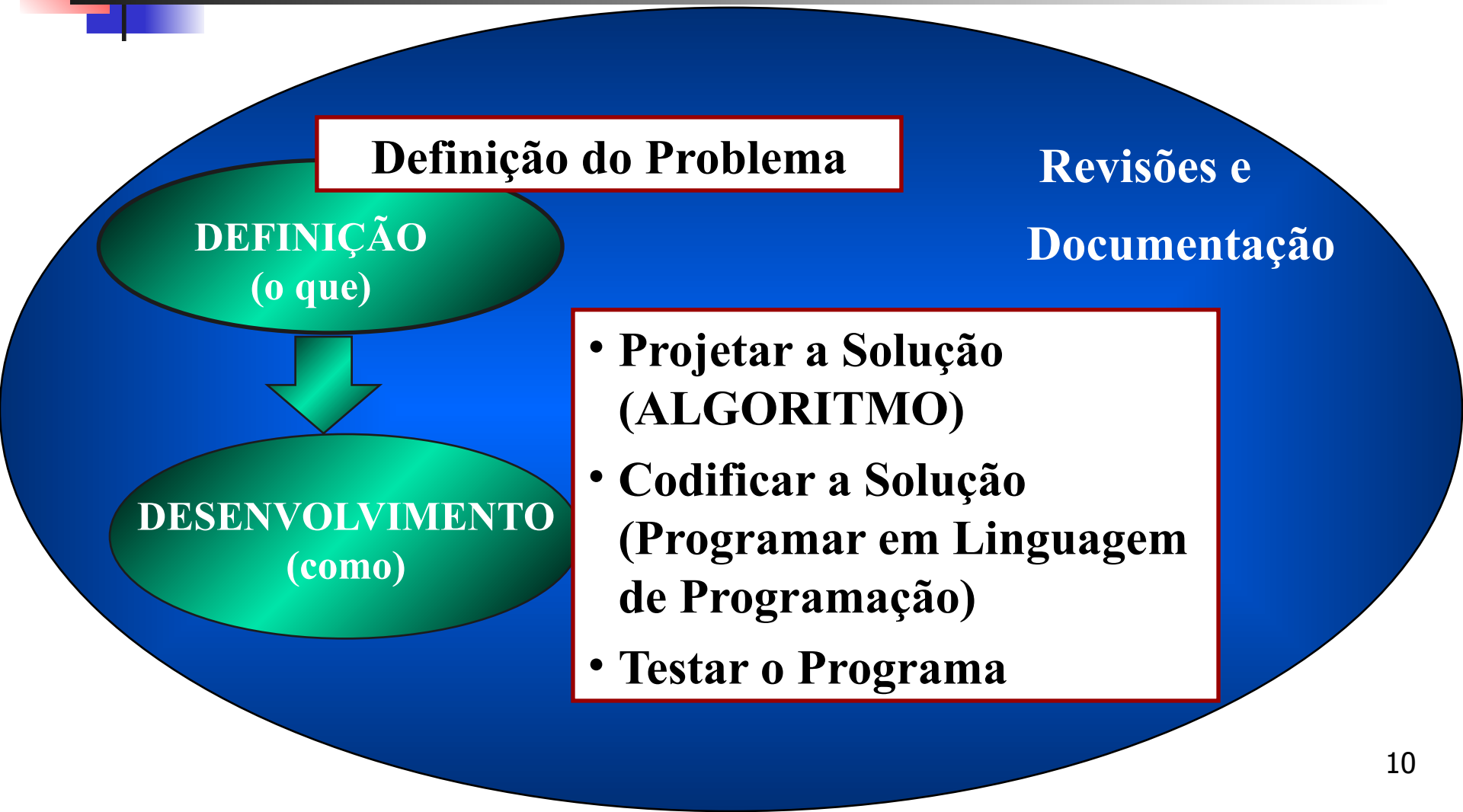
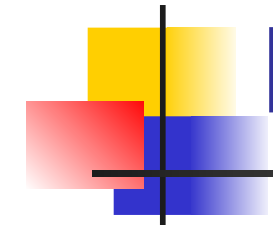


Programação e Níveis de Linguagem

- INTERPRETADOR

- Lê e executa uma declaração do programa por vez.
- Nenhuma fase intermediária de compilação é necessária.
- A execução do programa interpretado requer que o interpretador da linguagem esteja sendo executado no computador.

1. Etapas da Construção de Programas





2. Definição de Algoritmo

- ♦ Um algoritmo é uma seqüência de passos que visam atingir um objetivo bem definido.
 - ♦ Pessoas tem inteligência e habilidade racional => fazem perguntas para se esclarecer.
 - ♦ Computador não tem senso próprio => deve receber instruções explícitas e precisas (algoritmos).



2. Definição de Algoritmo

- ♦ Um algoritmo correto deve possuir 3 qualidades:
 - 1- Cada passo do algoritmo deve ser uma instrução que possa ser realizada.
 - 2- A ordem dos passos deve ser precisamente determinada.
 - 3- O algoritmo deve ter fim.

3. Introdução às estruturas algorítmicas e refinamentos sucessivos

- Todo algoritmo possui início e fim.

Início

----;

----;

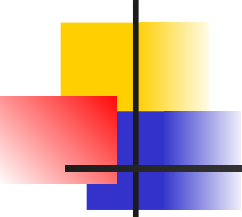
----;

Fim.

- Cada instrução do algoritmo termina com ;
- O algoritmo termina com ponto final.

3. Introdução às estruturas algorítmicas e refinamentos sucessivos

- Primeiro algoritmo: trocar uma lâmpada no teto.
 - Quais serão as instruções?
 - Usaremos o português coloquial.



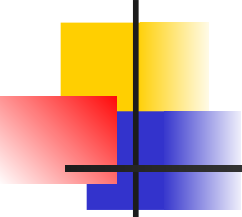
3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

Início

Remova a lâmpada queimada;
Coloque a nova lâmpada;

Fim.



3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

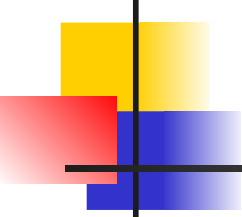
Início

Remova a lâmpada queimada;

Coloque a nova lâmpada;

Fim.

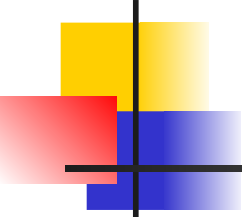
O que é necessário para remover a lâmpada queimada?



3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

- Posicione a escada debaixo da lâmpada queimada;
- Suba na escada até que a lâmpada possa ser alcançada;
- Gire a lâmpada queimada no sentido anti-horário até que se solte;



3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

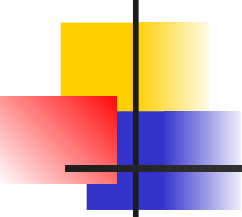
Início

Remova a lâmpada queimada;

Coloque a nova lâmpada;

Fim.

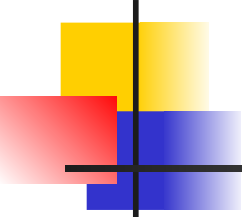
O que é necessário para colocar a lâmpada nova?



3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

- Escolha uma lâmpada da mesma potência da queimada;
- Posicione a nova lâmpada no soquete;
- Gire a lâmpada no sentido horário até que ela se firme;
- Desça da escada;



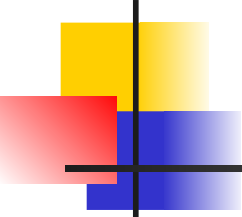
3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

Início

Posicione a escada debaixo da lâmpada queimada;
Suba na escada até que a lâmpada possa ser alcançada;
Gire a lâmpada queimada no sentido anti-horário até que se solte;
Remova a lâmpada queimada;
Escolha uma lâmpada da mesma potência da queimada;
Posicione a nova lâmpada no soquete;
Gire a lâmpada no sentido horário até que ela se firme;
Desça da escada;

Fim.



3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

Início

Posicione a escada debaixo da lâmpada queimada;

Suba a escada até que a lâmpada possa ser alcançada;

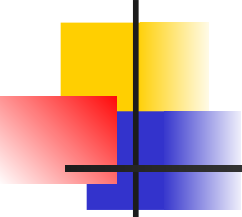
Sequenciamento: estabelece um padrão de comportamento. As ações devem ser executadas linearmente, em seqüência, uma após a outra.

Posicione a nova lâmpada no soquete;

Gire a lâmpada no sentido horário até que ela se firme;

Desça da escada;

Fim.



3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

Início

Posicione a escada debaixo da lâmpada queimada;

Suba na escada até que a lâmpada possa ser alcançada;

G

R

E

E se a lâmpada não estiver
queimada?

Posicione a nova lâmpada no soquete;

Gire a lâmpada no sentido horário até que ela se firme;

Desça da escada;

Fim.

3. Introdução às estruturas algorítmicas e refinamentos sucessivos

ALGORITMO PARA TROCAR UMA LÂMPADA NO TETO

Início

Posicione a escada debaixo da lâmpada queimada;

Suba na escada até que a lâmpada possa ser alcançada;

G

E se a lâmpada não estiver

R

E

**Precisamos de um
teste seletivo!**

Posicione a nova lâmpada;

Gire a lâmpada no sentido horário até que ela se fixe;

Desça da escada;

Fim.

3. Introdução às estruturas algorítmicas e refinamentos sucessivos

Início

Acionar o interruptor;
se a lâmpada não ascender, então
início

Posicione a escada debaixo da lâmpada queimada;
Suba na escada até que a lâmpada possa ser alcançada;
Gire a lâmpada queimada no sentido anti-horário até que se solte;
Remova a lâmpada queimada;
Escolha uma lâmpada da mesma potência da queimada;
Posicione a nova lâmpada no soquete;
Gire a lâmpada no sentido horário até que ela se firme;
Desça da escada;

fim;

Fim.

3. Introdução às estruturas algorítmicas e refinamentos sucessivos

Início

Acionar o interruptor;
se a lâmpada não ascender, então
início

Posicione a escada debaixo da lâmpada queimada;

Teste seletivo: determina qual conjunto de ações deve ser seguido, dependendo do resultado da **condição** resultar em **verdadeiro** ou **falso**.

Posicione a nova lâmpada no soquete;

Gire a lâmpada no sentido horário até que ela se firme;

Desça da escada;

fim;

Fim.

3. Introdução às estruturas algorítmicas e refinamentos sucessivos

Início

Acionar o interruptor;
se a lâmpada não ascender, então
início

Decida-se a escada debaixo da lâmpada queimada;

**E se a nova lâmpada estiver
queimada?**

Remova a lâmpada queimada;

Escolha uma lâmpada da mesma potência da queimada;

Posicione a nova lâmpada no soquete;

Gire a lâmpada no sentido horário até que ela se firme;

Desça da escada;

fim;

Fim.



Início

Acionar o interruptor;
se a lâmpada não ascender, então

início

Posicione a escada debaixo da lâmpada queimada;
Suba na escada até que a lâmpada possa ser alcançada;
Gire a lâmpada queimada no sentido anti-horário até que se solte;
Remova a lâmpada queimada;
Escolha uma lâmpada da mesma potência da queimada;
Posicione a nova lâmpada no soquete;
Gire a lâmpada no sentido horário até que ela se firme;

fim;

enquanto a lâmpada não ascender, faça

início

Gire a lâmpada queimada no sentido anti-horário até que se solte;
Remova a lâmpada queimada;
Escolha uma lâmpada da mesma potência da queimada;
Posicione a nova lâmpada no soquete;
Gire a lâmpada no sentido horário até que ela se firme;

fim;

Desça da escada;

Fim.

Repetição: mesmo trecho é repetido várias vezes, até que a condição de parada seja alcançada!

Qual a **condição de parada?**

O número de repetições é **indefinido**, porém, **finito**.

fim;

enquanto a lâmpada não ascender, faça

início

Gire a lâmpada queimada no sentido anti-horário até que se solte;

Remova a lâmpada queimada;

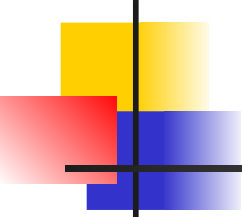
Escolha uma lâmpada da mesma potência da queimada;

Posicione a nova lâmpada no soquete;

Gire a lâmpada no sentido horário até que ela se firme;

fim;

Desça da escada;



3. Introdução às estruturas algorítmicas e refinamentos sucessivos

**E se tivermos que testar 10
soquetes de lâmpadas?**

Note-se: a quantidade é conhecida.

Início

Ir até o interruptor do prim. soquete;
enquanto a qtd de soquetes testados for menor que 10 faça
início

Acionar o interruptor;
se a lâmpada não ascender, então
início

Posicione a escada debaixo da lâmpada queimada;
Suba na escada até que a lâmpada possa ser alcançada;
Gire a lâmpada queimada no sentido anti-horário até que se solte;
Remova a lâmpada queimada;
Escolha uma lâmpada da mesma potência da queimada;
Posicione a nova lâmpada no soquete;
Gire a lâmpada no sentido horário até que ela se firme;

fim;

enquanto a lâmpada não ascender, faça

início

Gire a lâmpada queimada no sentido anti-horário até que se solte;
Remova a lâmpada queimada;
Escolha uma lâmpada da mesma potência da queimada;
Posicione a nova lâmpada no soquete;
Gire a lâmpada no sentido horário até que ela se firme;

fim;

Desça da escada;
Ir para o próximo soquete;

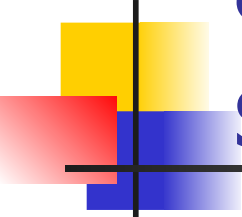
fim;

Fim

3. Introdução às estruturas algorítmicas e refinamentos sucessivos

- **Até quando devemos refinar o algoritmo?**
 - Até que as instruções cheguem o mais próximo possível das instruções de uma linguagem de programação. **Instruções Primitivas!**
- **Instrução Primitiva**
 - Comandos que definem integralmente uma ação a ser executada.

3. Introdução às estruturas algorítmicas e refinamentos sucessivos



- O algoritmo deve ser independente de linguagem de programação.
- As linguagens de programação, assim como os computadores, têm um **conjunto restrito de instruções**.
 - Muitas dessas instruções são comuns: **testes seletivos, repetições, entrada e saída**, etc.
- Pseudocódigo
 - Sintaxe

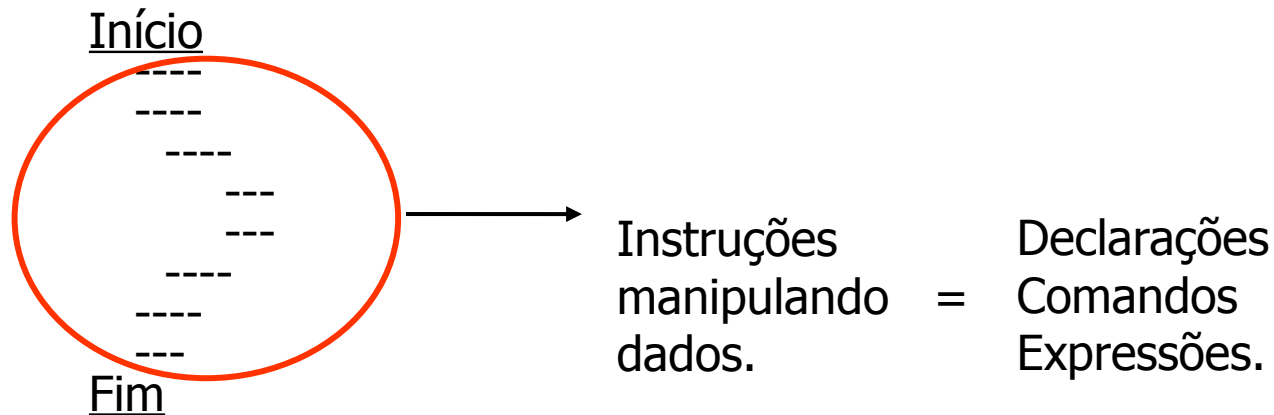


4. Constantes e Variáveis

- Constantes
 - Caracteres – aspas duplas.
- Identificadores
- Tipos primitivos
 - Inteiro, real, caracter, lógico
- Declarações
- Atribuições



4. Constantes e Variáveis



- Como representar os dados computacionalmente?



4. Constantes e Variáveis

- Os dados são representados através de **constantes e variáveis**, as quais possuem um tipo de dado associado.
- Dizemos que uma determinada constante ou variável é de um determinado tipo.



4. Constantes e Variáveis

- Relembrando...:
- Tipos de dados podem ser vistos como métodos para interpretar o conteúdo da memória do computador.
- Um tipo de dado especifica:
 - A quantidade de bytes que deve ser reservada para uma constante ou variável.
 - Como o dado representado por esses bytes deve ser interpretado (o que significa a cadeia de bits).



4. Constantes e Variáveis

- As linguagens de programação conseguem manipular um conjunto de tipos de dados.
- Dentre eles, os tipos primitivos de dados (básicos) são classificados em dados numéricos, literais e lógicos.



4. Constantes e Variáveis

- Em nossos algoritmos usaremos quatro tipos de dados básicos:
 - inteiro: -5; 218; etc.
 - real: 4,5; -3,659; 0,82.
 - caracter: "a"; "123"; "ABcdE".
 - Observe as aspas duplas.
 - Não distinguiremos, por ora, caracter de string.
 - lógico: verdadeiro e falso.



4. Constantes e Variáveis

- Declaração de variáveis
 - Modo geral: ***tipo: identificador;***
 - Tipo: um tipo de dado primitivo
 - Identificador: um nome válido para a variável
 - Exemplo: inteiro: temperatura;



4. Constantes e Variáveis

- Declaração de constantes
 - Modo geral: ***constante tipo: identificador = valor;***
 - *tipo*: um tipo de dado primitivo
 - *identificador*: um nome válido para a variável/constante
 - *valor*: um valor válido para a constante
 - Exemplo:

constante real: $\pi = 3,14159265359;$



4. Constantes e Variáveis

- Exercício 1:
 - Faça um algoritmo que declare variáveis para os seguintes dados sobre uma pessoa: o nome, a altura, a idade, o peso, se a pessoa é fumante ou não.



4. Constantes e Variáveis

- Exercício 2:
 - Faça um algoritmo que declare variáveis e constantes necessárias para converter distância em quilômetros para anos-luz.

4. Constantes e Variáveis



- Atribuição
 - Operador: <-
 - Atribui o valor à direita à variável à esquerda.
 - Exemplo:
 - inteiro: x;
 - x <- 10;



5. Operadores e Expressões

- Aritméticos
- Pot, Rad, Div, Mod
- Relacionais
- Lógicos
 - E, OU, NÃO
 - Tabelas verdade
- Precedência

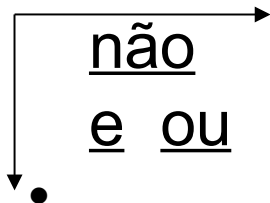


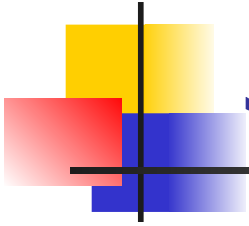
5. Operadores e Expressões

- Prioridades entre todos os operadores:
parênteses mais internos
operadores aritméticos
operadores relacionais
operadores lógicos



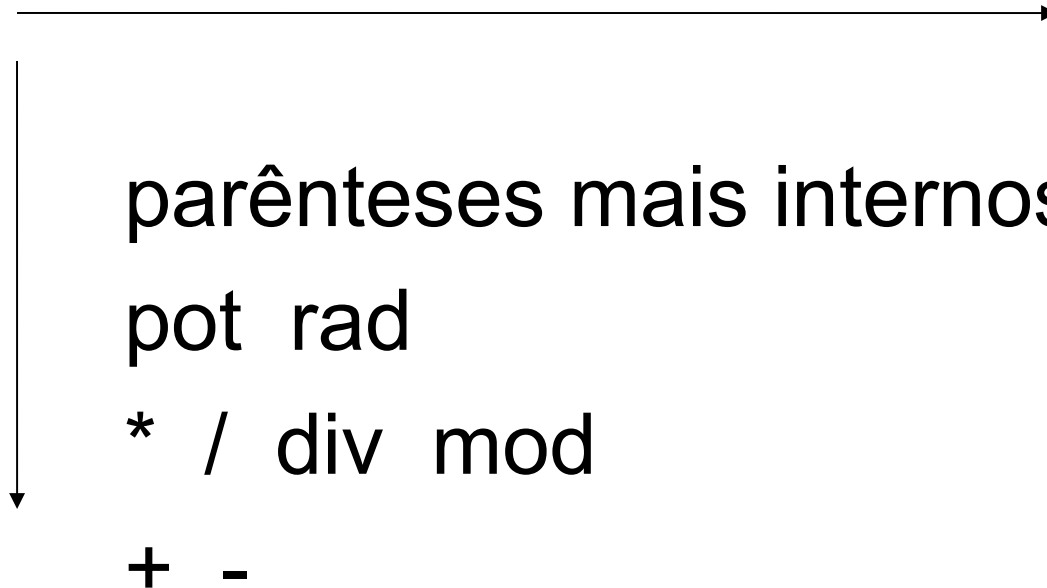
Prioridades entre operadores lógicos:





5. Operadores e Expressões

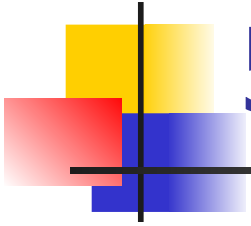
- Prioridades entre operadores aritméticos





5. Operadores e Expressões

- Exercícios:
 - $(2 + 3) + 5 \bmod 2 * 3$
 - R: 10
 - $3 + 2 * \text{rad}(9) \text{ div } 22 / 2$
 - R: 3
 - $2 * 4 = 24 / 3$
 - R: verdadeiro
 - $15 \bmod 4 < 19 \bmod 6$
 - R: falso
 - $3 * 5 \text{ div } 4 \leq \text{pot}(3, 2) / 0,5$
 - R: verdadeiro



5. Operadores e Expressões

Operadores lógicos:

- Tabelas-verdade: conjunto de todas as possibilidades combinatórias entre valores de variáveis lógicas e operadores lógicos.



5. Operadores e Expressões

- Tabela-verdade para negação:

A	<u>NÃO(A)</u>
0	1
1	0

Se $A = 0$, $\text{NÃO}(A) = 1$, e vice-versa.

Corresponde à porta lógica NOT: 

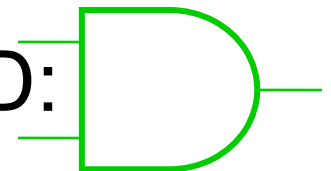


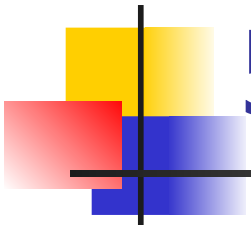
5. Operadores e Expressões

- Tabela-verdade para conjunção:

A	B	$A \text{ E } B$
0	0	0
0	1	0
1	0	0
1	1	1

Corresponde à porta lógica AND:



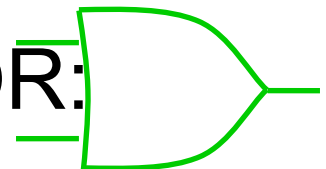


5. Operadores e Expressões

Tabela-verdade para disjunção:

A	B	A <u>OU</u> B
0	0	0
0	1	1
1	0	1
1	1	1

Corresponde à porta lógica OR:





5. Operadores e Expressões

- Expressões
 - Poder ser atribuídas a variáveis:
 - Sintaxe: `identificador <- expressão;`
 - Identificador deve ser do tipo da expressão
 - Exemplos:
 - Seja $A = 4$, $B = 10$, $C = -8$, $D = 1,5$
 - `Media <- (A + B) / 2;`
 - `X1 <- 2 * A mod 5 - C;`
 - `E <- 2 * 4 < 3 - pot(3, 2);`



6. Entrada e Saída

- Leia
 - Sintaxe: `leia (identificador);`
 - *identificador* pode ser uma lista de identificadores, separados por vírgula.
 - Exemplo:
 - inteiro: A,B,C;
 - leia (A);
 - leia (B,C,A);



6. Entrada e Saída

- Escreva
 - Sintaxe: escreva (*identificador* ou *expressão*);
 - Exemplos:
 - escreva (5);
 - escreva ("ola!");
 - caracter: nome;
leia (nome);
 - escreva ("Bom dia", nome, "!");



7. Comandos de seleção

- Também chamados de condicionais.
- Seleção simples

- Sintaxe:

- se (*condição*) então
 comando;
fimse;

- se (*condição*) então
 início
 comando 1;
 ...
 comando n;
 fim;
fimse;

- *condição* é uma expressão lógica!

- Pode-se usar *bloco* de comandos!



7. Comandos de seleção

- Seleção composta

- Sintaxe:

- se (*condição*) então
 comando 1;
senão
 comando 2;
fimse;

Pode-se usar blocos de comandos para o *se*, ou para o *senão*, ou para ambos.



7. Comandos de seleção

- Seleção encadeada

```
se (condição 1) então
  início
    se (condição 2) então
      comando 1;
    fimse;
    se (condição 3) então
      início
        comando A;
        ...
        comando Z;
      fim;
    fimse;
  fim;
senão
  comando A1;
fimse;
```



Exercício

- Dados três valores de entrada, A , B e C , construir um algoritmo para verificar se os mesmos podem ser os comprimentos dos lados de um triângulo. Se forem, verificar e imprimir se o triângulo é equilátero, isósceles ou escaleno. Informar se não formarem nenhum triângulo.



7. Comandos de seleção

- Casos especiais
 - se então se (aninhados, longos)
 - Operadores lógicos
 - se senão se (aninhados, longos)
 - Seleção de múltipla escolha



7. Comandos de seleção

- Seleção de múltipla escolha

- Sintaxe:

- escolha X

- caso v1: c1;
 - caso v2: c2;
 - ...
 - caso vn: cn;
 - fimescolha;

- escolha X

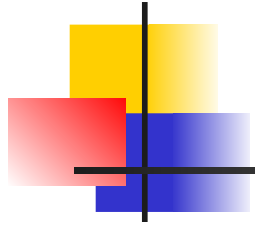
- caso v1: c1;
 - caso v2, v3, v4: c2;
 - caso v5: c3;
 - caso contrário: c4;
 - fimescolha;



Exercício

- Escreva um algoritmo que leia o código de um determinado produto e mostre a sua classificação. Utilize a seguinte tabela como referência:

Código:	Classificação:
1	Alimento não perecível
2, 3 ou 4	Alimento perecível
5 ou 6	Vestuário
7	Higiene pessoal
8 a 15	Limpeza e utensílios domésticos
Qualquer outro código	Inválido



FIM