Revisão

Lógica de Programaçao

Maurício Manoel
mauriciomanoel@gmail.com
https://github.com/mauriciomanoel/ete-webdesign
26 de março de 2018

Introdução

Amanda é aluna da **ETE Limoeiro**.

Para ser aprovado, um aluno da **ETE Limoeiro** precisa obter nota maior ou igual a

6 e comparecer a mais de 75% das aulas.

Amanda compareceu a todas as aulas e obteve nota igual a 8.

Então, o que podemos concluir?

Introdução

Sejam os seguintes fatos:

Todos os filhos de João são mais altos do que Maria.

Antônio é filho de João.

Então, o que podemos concluir logicamente?

Antônio é mais alto que Maria

Maria é mais baixa que João

Maria é a mais baixa de todos os filhos

Lógica

- Utilizamos a lógica de forma natural em nosso dia a dia, então...

O que é lógica?

Lógica

- Pode ser vista como a arte de **pensar corretamente**. A lógica visa a colocar **ordem no pensamento** (FARRER, 1999);
- Ciência que estuda as **leis do raciocínio**;
- Correção/validação do pensamento;
- Encadeamento/ordem de ideias;
- Arte de pensar bem.

Raciocínio Lógica

- Precisamos mais do que fórmulas, precisamos **APRENDER A PENSAR**!
- É preciso aprender a pensar **sobre o problema**, **extraindo** o máximo de **informações** sobre ele;
- **Lógica ensina a colocar ordem no pensamento**.

Lógica de Programação

- Permite o **APERFEIÇOAMENTO** de nossa forma de pensar e raciocinar sobre um **PROBLEMA computacional**, afim de obter uma solução eficaz e/ou eficiente;
- A lógica de programação é **NECESSÁRIA** para pessoas que desejam **trabalhar com desenvolvimento de sistemas e programas**, ela permite definir a **SEQUENCIA LÓGICA** para o desenvolvimento.

Sequencia Lógica

- São passos executados até atingir um **OBJETIVO** ou **SOLUÇÃO** de um problema, Por exemplo: Preparar um Café:
- Sequencia de passos para a solução:

1. Encher a chaleira com água
2. Colocar a chaleira para ferver
3. Colocar duas colheres de sopa de pó de café no coador
4. Aguarde a água ferver
5. Acrescente a água ao pó aos poucos
6. Aguarde coar
7. Adoce a gosto

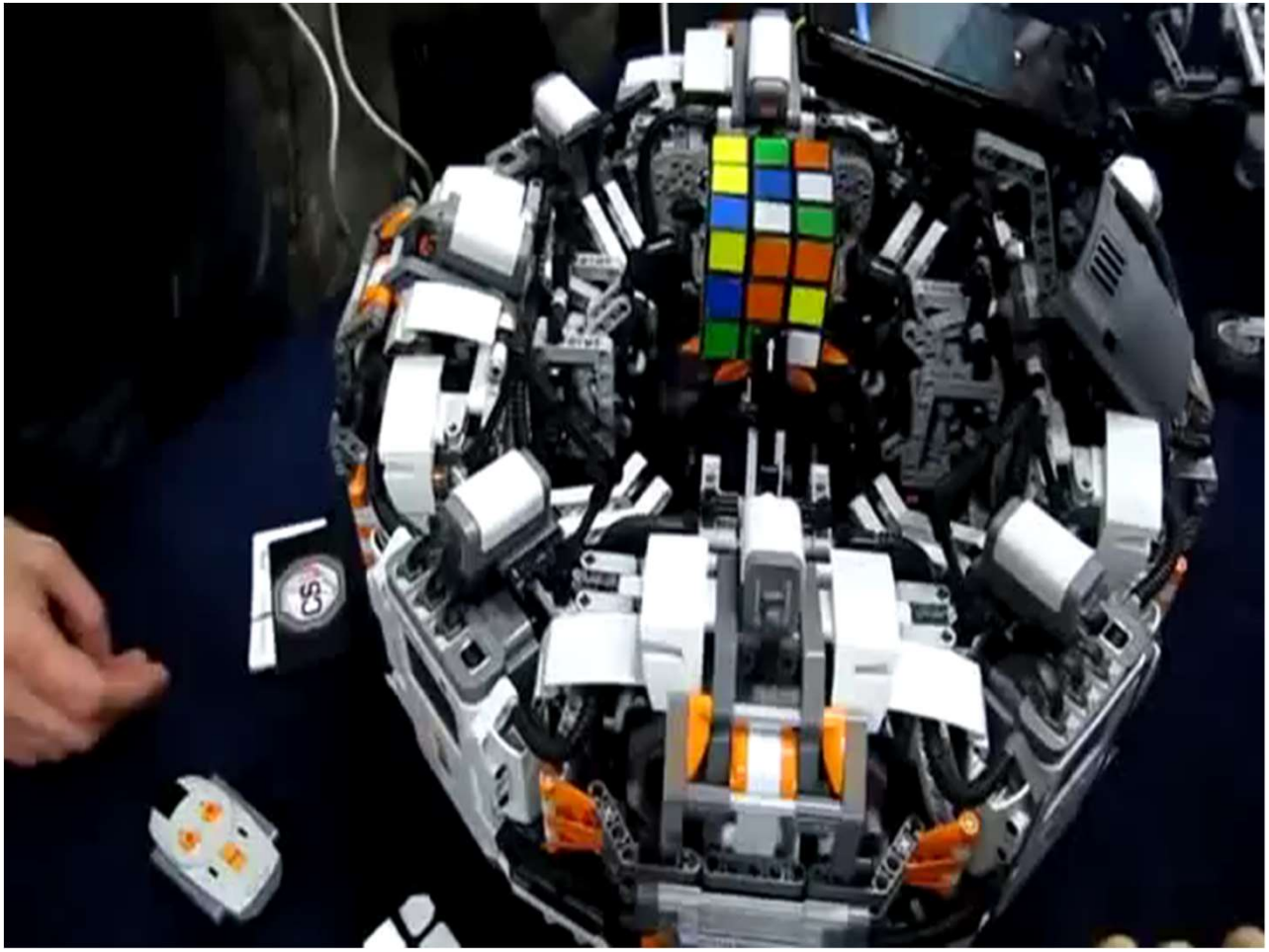
Dúvidas

Definição

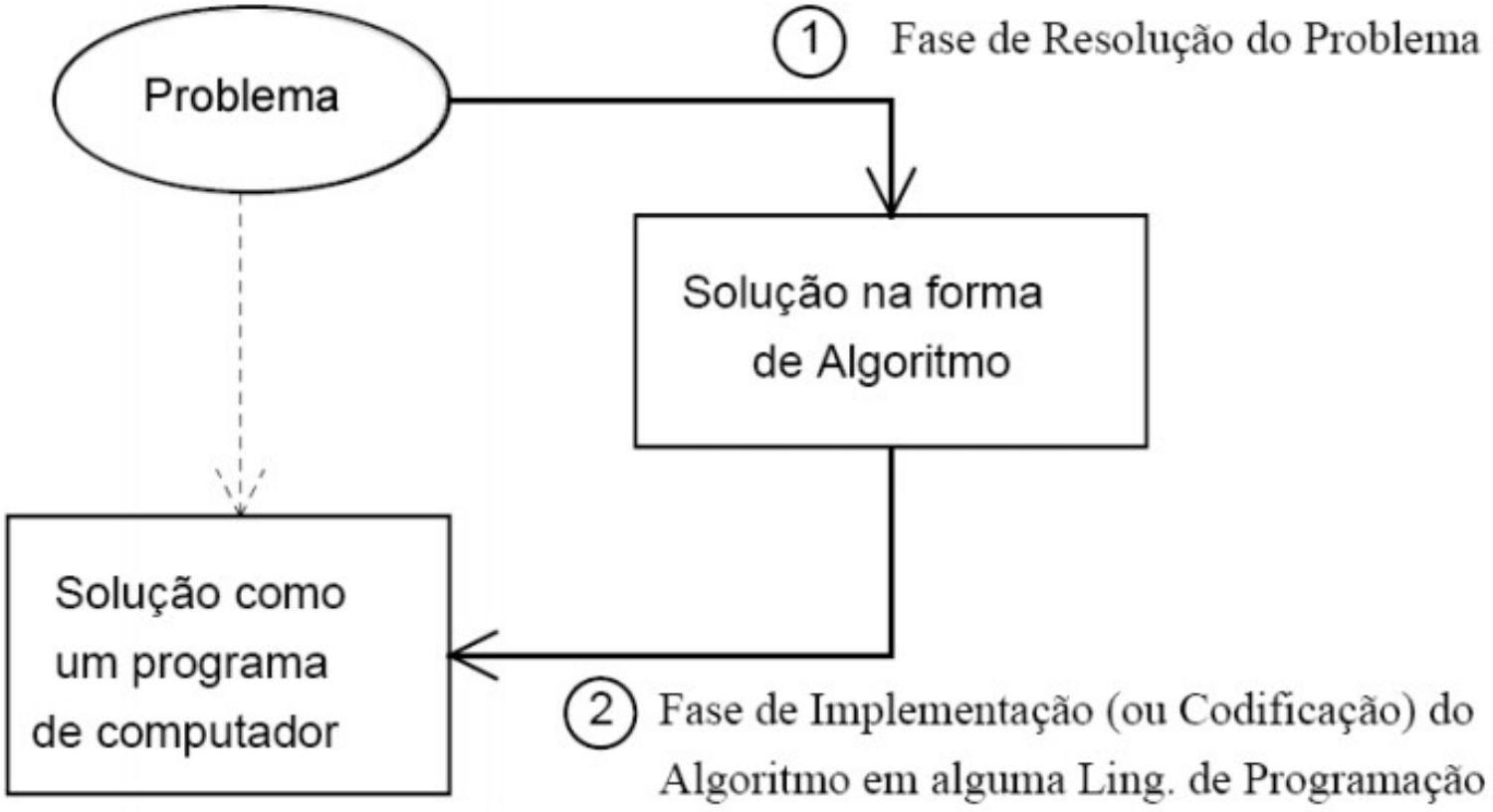
- Formalmente é uma sequência **FINITA DE PASSOS** que levam à **EXECUÇÃO DE UMA TAREFA**. Pensar em algoritmo como uma receita, uma **SEQUÊNCIA DE INSTRUÇÕES** que busca atingir um **OBJETIVO BEM DEFINIDO** (FORBELLONE et al., 2005);
- Ao definir uma sequência de passos é necessário **PENSAR ORDENADAMENTE**, utilizar lógica;
- O objetivo fundamental de toda **PROGRAMAÇÃO** é construir **ALGORITMOS**;
- Ao criar um algoritmo, apenas **APONTAMOS** uma **SEQUÊNCIA DE ATIVIDADES** que levam à **SOLUÇÃO DE UM PROBLEMA**.

Ajuda a resolver problemas...

Ensinar a maquina como resolver problemas...



Fases da Programação



Regras para Criação do Algoritmo

- **Ter início e fim;**
- Usar somente um verbo por frase;
- Imaginar que você está desenvolvendo um algoritmo para **pessoas** que **não trabalham** com **informática**;
- Usar frases curtas e simples;

Regras para Criação do Algoritmo

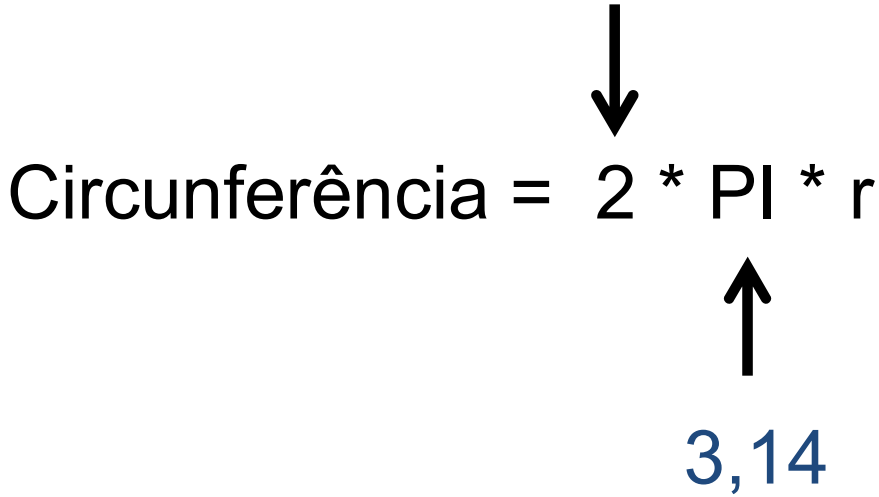
- Problema: Trocar uma lâmpada.
- Sequencia de passos para a solução:
    1. Pegue uma escada;
    2. Posicione a escada embaixo da lâmpada;
    3. Pegue uma lâmpada nova;
    4. Suba na escada;
    5. Retire a lâmpada velha;
    6. Coloque a lâmpada nova.

Revisão

Constantes e Variáveis

Constantes

- É um valor fixo, não muda durante a execução de um programa;



- “São valores que não se alteram.”

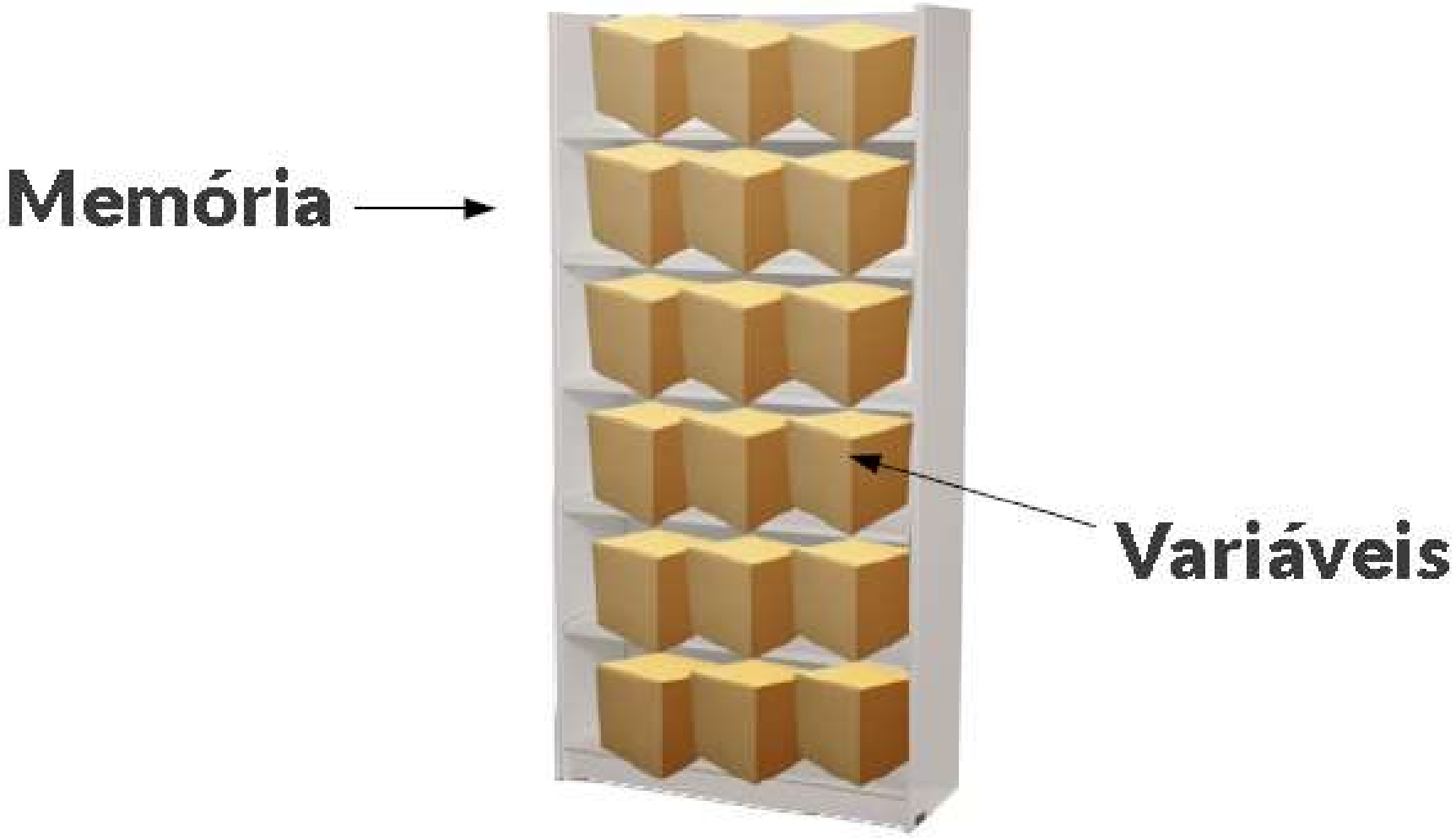
Características

- São imutáveis;
- Não podem ser alteradas durante uma execução;
- Geralmente são representadas em **CAIXA ALTA**.
- PI = 3.14159265
- VALOR_MAX = 100

Variáveis

- Um espaço reservado na memória do computador para armazenar algum tipo de dado;
- Valores modificáveis;
- Armazena um valor a cada instante;
- Devem ser declaradas antes de serem utilizadas: *Geralmente no início do programa*.
- Programadores geralmente escolhem **nomes** para **variáveis** que são **significativos**.
- Eles **documentam** para o que a **variável** é **usada**.

Variáveis



Variáveis

Dados Cadastrais

Nome: João Guilherme
Idade: 30
Endereço: Rua João Pinho, 123
Peso: 85,5
Altura: 1,90
IMC: 23,7

Dúvidas

Declaração

- Declarar uma variável significa definir o seu tipo e seu nome;
- Devem ser declaradas antes de seu uso no programa, geralmente no topo;
- Duas variáveis não devem ter o mesmo nome;

Regras para Variáveis

- Nunca comece com números;
- Não é permitido o uso de espaços em branco, acentos ou caracteres especiais;
- Não é permitido utilizar palavras reservadas;
- Geralmente começam com um caractere alfabético;
- Geralmente são escritas em minúsculo;
- Só são permitidos caracteres alfanuméricos ou sublinhado (_).

Regras para Constantes

- Segue todas as regras para variáveis, exceto que são escritas em **CAIXA ALTA**.

Tipos de Dados

- Podem ser basicamente de três tipos:
  - Numéricas;
  - Caracteres;
  - Lógicas.

Numéricas

- Podem ser divididos em dois conjuntos:
  - **Inteiro** → Podem ser positivos, negativos ou nulos, mas não possuem componente decimal:
    - 5; -9; 0; 189; -800.
  - **Ponto Flutuante** → Podem ser positivos, negativos ou nulos, e possuem componente decimal:
    - 5,89; -6,8978; 0; 7,986; -1458,252.

Caracteres ou Literais

- Composto por letras e números (alfanuméricos);
- Escrito entre aspas: A = "Texto"
- Não confundir A com "A"

A é a variável A

"A" é o literal A

- A = "Minha Casa a Lua 37“

Lógicas

- Dados que assumem apenas dois valores:
  - Verdadeiro (**True**);
  - Falso (**False**).

Estes valores também são chamados de booleanos.

Tipos de Dados

Dados Cadastrais

Nome: João Guilherme
Idade: 30
Endereço: Rua João Pinho, 123
Peso: 85,5
Altura: 1,90
IMC: 23,7
Peso Ideal: ( ) Sim ( ) Não

IMC - Índice de Massa Corporal	HOMEM	MULHER
Obesidade Mórbida	+ de 43	+ de 39
Obesidade Moderada	30 a 39,9	29 a 38,9
Obesidade Leve	25 a 29,9	24 a 28,9
Normal	20 a 24,9	19 a 23,9
Abaixo do Normal	- de 20	- de 19

Em Java?

Atribuição

- É o processo de “colocar os dados nas caixas”.
- É utilizado o operador de atribuição (=);
- O que está à esquerda do operador é o identificador;
- O que está à direita do operador é o valor;
- VALOR_MAX = 100
- nome = "Luiz Augusto"

Atribuição

- Variável = **expressão**
- Exemplo:
  - A = verdadeiro
  - B = 5 * 3
  - C = A e B
  - D = B * A – 2 > 4
- A atribuição é a operação que modifica o valor de uma variável.
- É importante notar que se atribui o resultado da expressão a variável e não a expressão em si.

Lógicos

Não (NOT)

E (AND)

Ou (OR)

- Operadores lógicos são utilizados para modificar valores como **verdadeiro** e **falso**, criando **expressões lógicas**;
- O resultado das operações é definido pelas chamadas **tabelas-verdade** de cada operador;
- Expressão avaliada da esquerda para a direita

Lógicos

Operação	Resultado
a E b	VERDADEIRO se ambas as partes (**a** e **b**) forem verdadeiras
a OU b	VERDADEIRO se apenas uma das partes (**a** ou **b**) é verdadeira.
NAO a	Nega uma afirmação, invertendo o seu valor lógico: se **a** for VERDADEIRO retorna FALSO, se **a** for FALSO retorna VERDADEIRO.

Tabela Verdade

a	b	a E b	a OU b	NAO a	NAO b
V	V	V	V	F	F
V	F	F	V	F	V
F	V	F	V	V	F
F	F	F	F	V	V

Tabela Verdade: Exercício

- Sendo A verdadeiro e B falso, resolva:
  - A) A E B
  - B) B E NÃO A
  - C) A OU B
  - D) (A E B) ou NÃO B
  - E) NÃO A
  - F) NÃO B

Em Java?

Tipos de Dados

```
public class ExemploOperador {
    public static void main(String[] args) {
        boolean a = true;
        boolean b = false;

        System.out.println(a && b);
        System.out.println(a && !b);
        System.out.println(a || b);
        System.out.println((a && b) || !b);
        System.out.println(!a);
        System.out.println(!b);
    }
}
```

Material para Complementação

- JAVA - 006 - Variáveis e Sintaxe: https://youtu.be/jLbnH968gtk
- JAVA - 009 - Constantes e Modificador final: https://youtu.be/IwLR8kDLXT0
- JAVA - 007 - Tipos Primitivos: https://youtu.be/1v0heZtQnlI
- JAVA - 013 - Operadores: https://youtu.be/1chFYRNRl7I
- JAVA - 016 - Operadores Lógicos: https://youtu.be/CgtskFPE-RA
- JAVA - 017 - Operadores de Atribuição: https://youtu.be/EiWgZeVGrl0

Revisão

Instruções de Decisão (Condicionais)

Introdução

- Problemas reais, em sua maioria, exigem uma **TOMADA DE DECISÃO** no algoritmo;
- Geralmente, é possível **seguir mais de um caminho**.

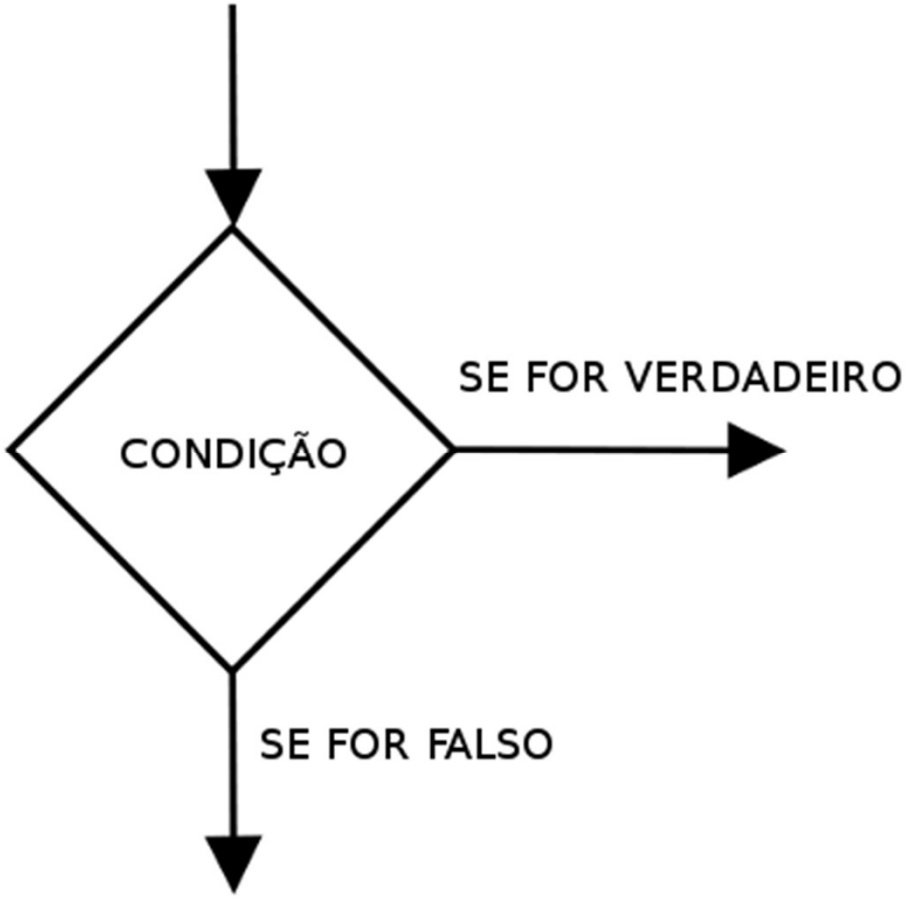
Algoritmo: Narrativa - Estrutura de Decisão

- Trocar uma Lâmpada
  - 1. pegar uma escada
  - 2. posicionar a escada embaixo da lâmpada
  - 3. buscar uma lâmpada nova
  - 4. acionar o interruptor
  - 5. **SE** a lâmpada não acender, **ENTÃO**
    - 5.1 subir na escada
    - 5.2 retirar lâmpada queimada
    - 5.3 colocar lâmpada nova

Quando Usar as Instruções de Decisão?

- Quando queremos que uma **CONDIÇÃO SEJA ANALISADA**;
- Caso esta **CONDIÇÃO SEJA VERDADEIRA**, **UM COMANDO** será executado;
- Caso esta **CONDIÇÃO SEJA FALSA**, **OUTRO COMANDO** será executado.
- Os principais comandos das estruturas condicionais (ou de seleção) são:
  - if / If ... else
  - Switch/Case

Quando Usar as Instruções de Decisão?



Em Java?

Quando Usar as Instruções de Decisão?

Estrutura switch-case

Quando Usar as Instruções de Decisão?

Estrutura switch-case

break é opcional: seu uso faz com que a execução pare no caso escolhido.

```
switch( opção )
{
    case opção1:
            comandos para opção 1 seja escolhida
            break;
    case opção2:
            comandos caso a opção 2 seja escolhida
            break;
    case opção3:
            comandos caso a opção 3 seja escolhida
            break;
    default:
            comandos caso nenhuma opção seja
escolhida
}
```

Quando Usar as Instruções de Decisão?

Estrutura switch-case

```
public class ExemploSwitch {

    public static void main(String[] args) {
        int diaDaSemana = 1;
        switch (diaDaSemana) {
            case 1:
                System.out.println("Domingo");
                break;
            case 2:
                System.out.println("Segunda-feira");
                break;
            case 3:
                System.out.println("Terça-feira");
                break;
```

Material para Complementação

- JAVA - 023 - if else: https://youtu.be/E4JZ8leWhkA
- JAVA - 024 - switch: https://youtu.be/NnsGEY2cnQ8

Revisão

Estrutura de Repetição

Introdução

- Como fazer a leitura de 2 números e informar a média entre eles?
- Como fazer a leitura de 4 números e informar a média entre eles?
- Como fazer a leitura de 8 números e informar a média entre eles?
- Como fazer a leitura de **n** números e informar a média entre eles?

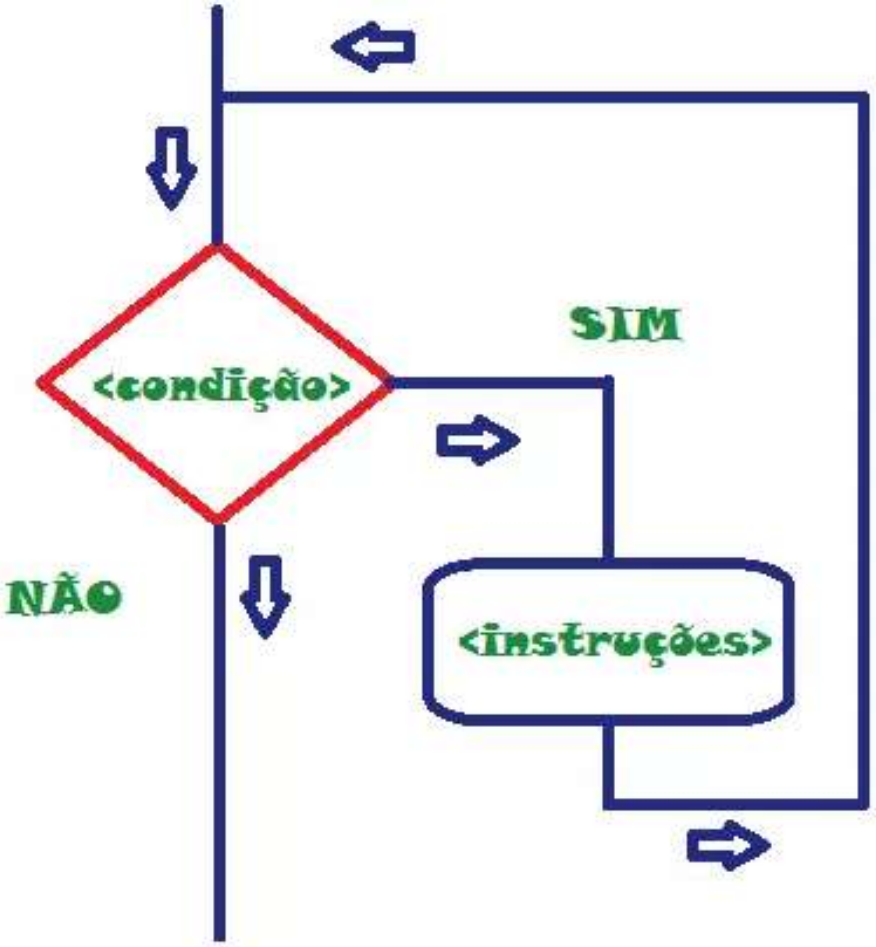
Introdução

- Há situações em que temos que trabalhar com a mesma informação várias vezes, no mesmo algoritmo;
- Se uma ação se repete várias vezes, podemos, na maioria dos casos, utilizar uma **estrutura de repetição**, e poupar linhas e tempo.

Há três Principais

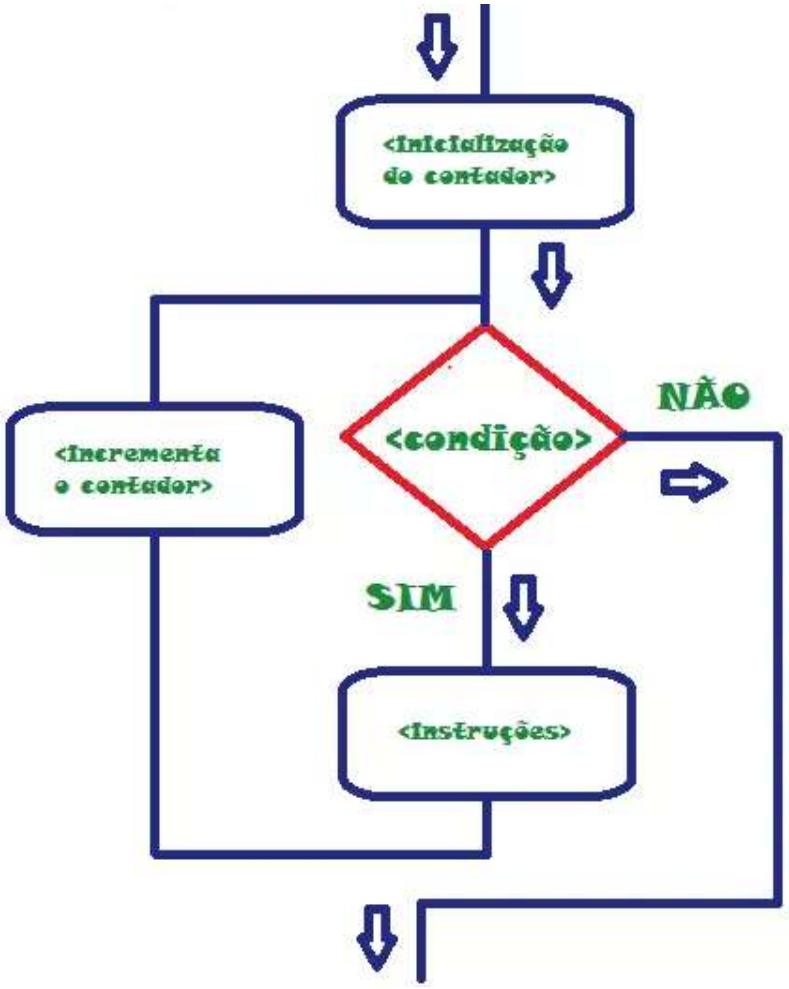
- Repetição com teste inicial: **enquanto... faça**
- Repetição com teste final: **repita... ate**
- Repetição controlada: **para... ate... faça**

Repetição com Teste Inicial



Repetição com Teste Final

Repetição Controlada



Principais Estruturas de Repetição

```java
public class EstruturaRepeticao {
    public static void main(String[] args) {
        int i = 0;
        // Exemplo: Estrutura de Repetição - Teste Incial
        System.out.println("Estrutura de Repetição - Teste Inicial");
        while(i<10) {
            System.out.println(i);
            i++;
        }

        System.out.println("Estrutura de Repetição - Teste Final");
        i = 0;
        do {
            System.out.println(i);
            i++;
        }while(i<10);

        System.out.println("Estrutura de Repetição - Repetição Controlada");
        for(i=0;i<10;i++) {
            System.out.println(i);
        }
    }
}
```

Material para Complementação

- JAVA - 027 - foreach: https://youtu.be/jMV0yzy83-I
- JAVA - 028 - while, do while: https://youtu.be/XfIXOm-5BXg
- JAVA - 026 - for: https://youtu.be/fGXNBqP3RnM

Revisão

Vetores

Introdução

- Faça um programa para auxiliar a escrever “Parabéns!” nas melhores provas de uma disciplina com 3 alunos
  - Ler os nomes e as notas de 3 alunos
  - Calcular a média da turma
  - Listar os alunos que tiveram nota acima da média

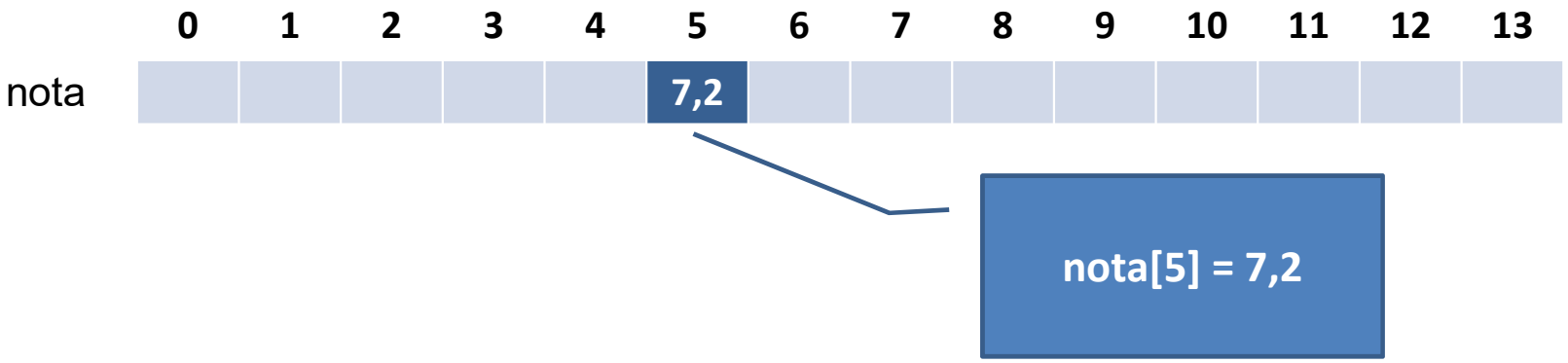
Introdução

```
aluno1 = input("Entre com o nome do Aluno 1: ")
aluno2 = input("Entre com o nome do Aluno 2: ")
aluno3 = input("Entre com o nome do Aluno 3: ")

nota1 = float(input("Informe a nota de " + aluno1 + ": "))
nota2 = float(input("Informe a nota de " + aluno2 + ": "))
nota3 = float(input("Informe a nota de " + aluno3 + ": "))

media = (nota1+nota2+nota3)/3
print("Media da turma foi ", media)

if (nota1>media):
    print("Parabéns ", aluno1)
if (nota2>media):
    print("Parabéns ", aluno2)
if (nota3>media):
    print("Parabéns ", aluno3)
```

Calcule agora para **40** pessoas

Vetores

- Variável composta unidimensional
  - Contém espaço para armazenar diversos valores
  - É acessada via um índice
- Ou seja, é um conjunto ordenado de informações, onde o índice do array começa com zero.



Vetores

```
public class ExemploVetor {
    public static void main(String[] args) {
        double[] notas = {7, 5, 6, 7, 8, 8, 6, 5};
        System.out.println(notas[7]);

    }
}
```

Material para Complementação

- JAVA - 020 - Array: https://youtu.be/XRMQR0H8NJU

Bibliografia

- Universidade XTI: https://www.youtube.com/playlist?list=PLxQNfKs8YwvGhXHbHtxtoB-tRRv6r3Rlr