

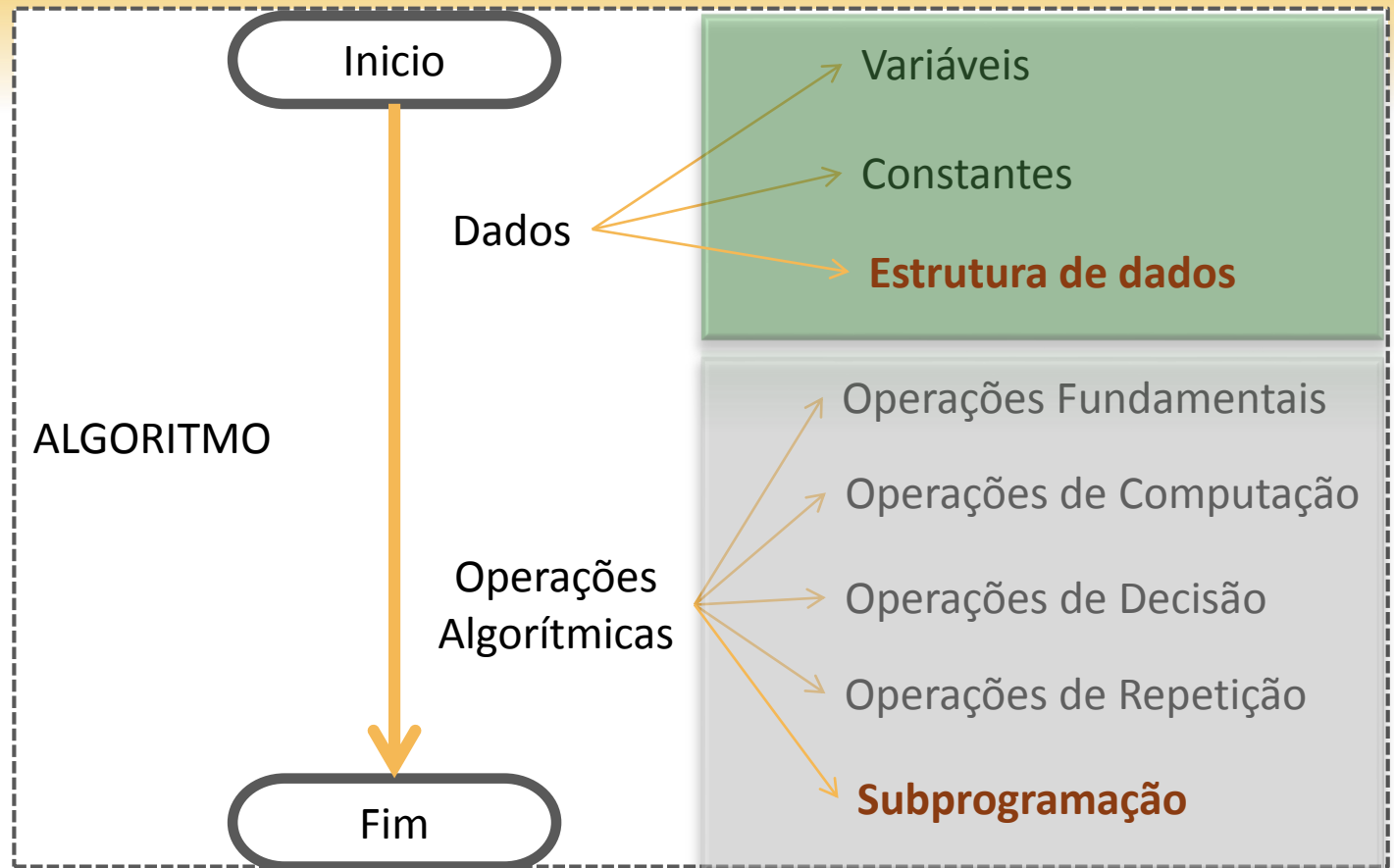
SUBPROGRAMAÇÃO

Prof. Josildo Silva

AGENDA

- ⊙ Técnicas de algoritmo computacional
- ⊙ Paradigma: divisão e conquista
- ⊙ Subprogramas e seus tipos

ALGORITMO COMPUTACIONAL

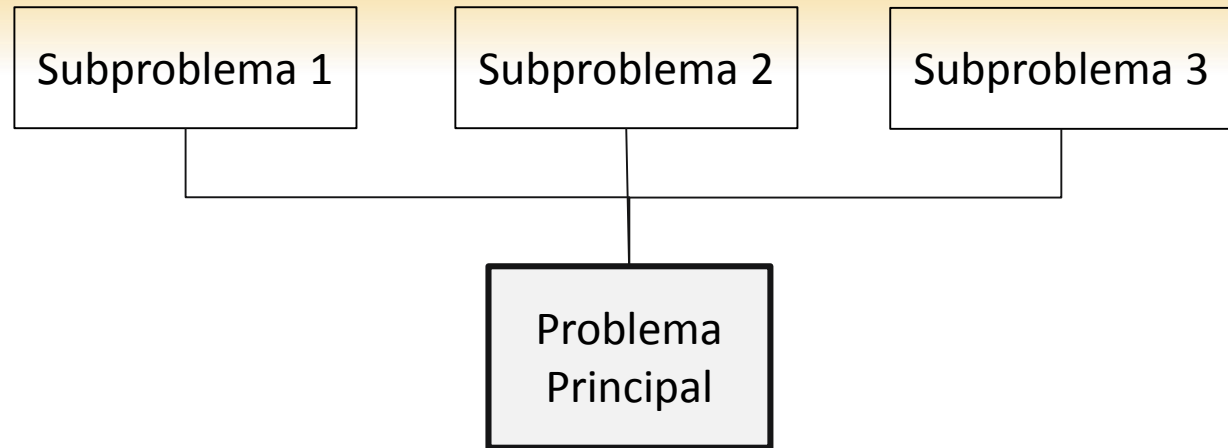


Conjunto ordenado de passos automatizáveis atuando sobre dados que definem um processo finito.

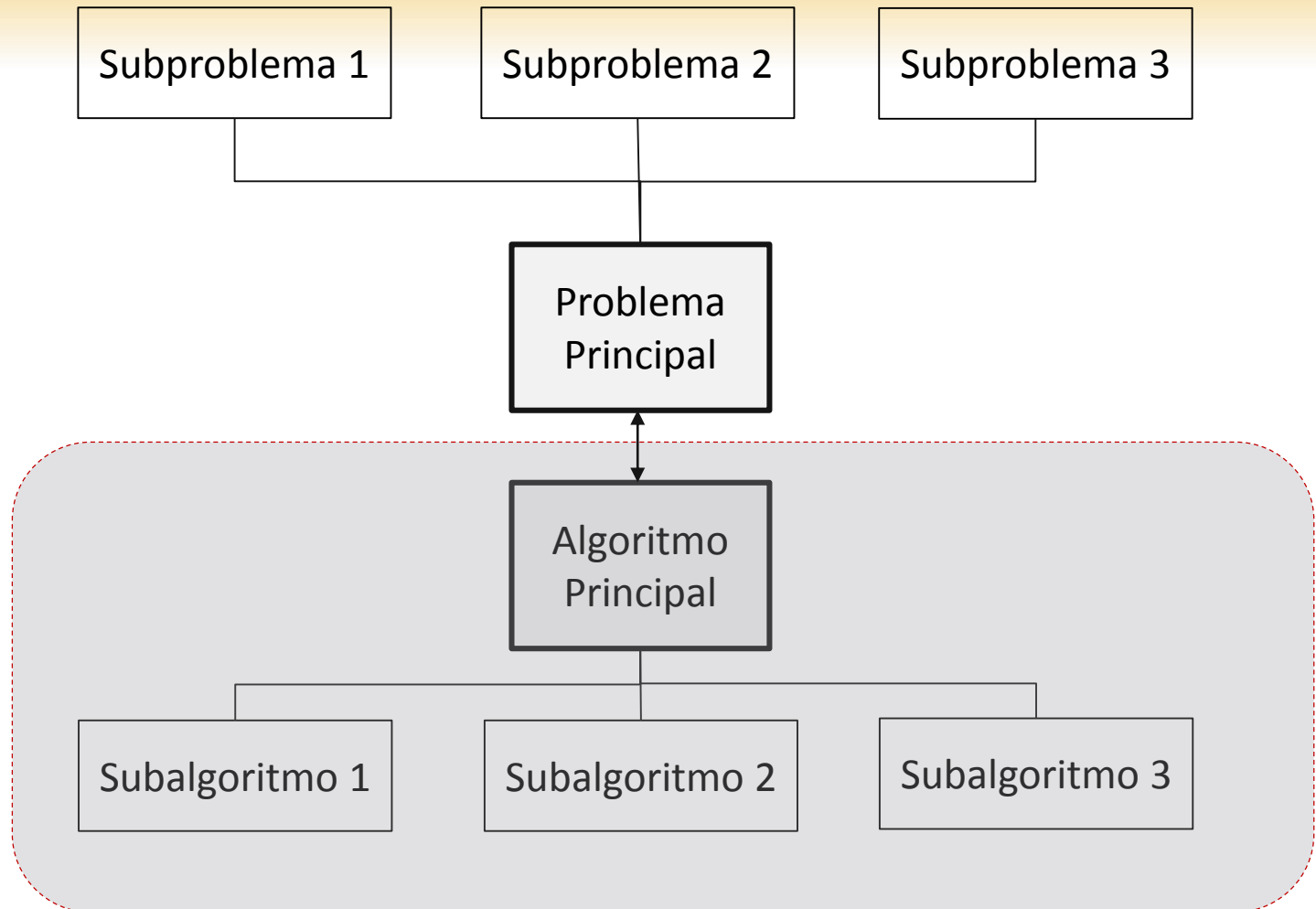
PARADIGMA DIVISÃO E CONQUISTA

- ⊙ Um paradigma é a representação de um padrão a ser seguido.
- ⊙ Em paradigmas de projetos de algoritmos estuda-se: *“dividir para conquistar”*.
- ⊙ Consiste em dividir o problema em partes menores, encontrar soluções para as partes, e combiná-las em uma solução global.

PARADIGMA DIVISÃO E CONQUISTA



PARADIGMA DIVISÃO E CONQUISTA



PARADIGMA DIVISÃO E CONQUISTA

- ⊙ O problema principal é solucionado pelo correspondente algoritmo principal - também denominado controlador ou condutor (*driver*).
- ⊙ A divisão do problema principal em subproblemas por consequência leva à soluções menores (localizadas) e por sua vez aos subalgoritmos.

SUBPROGRAMAÇÃO

DIVISÃO E CONQUISTA

- ⊙ Implementar algoritmo principal ou resolver o problema com divisão e conquista diretamente em alguma linguagem de programação se vale de uma característica das linguagens conhecida como SUBPROGRAMAÇÃO.
- ⊙ Assim, o problema principal é solucionado pelo correspondente programa principal contendo subprogramas que por sua vez solucionam os subproblemas do problema principal.

SUBPROGRAMA

- ☉ Um subprograma pode realizar as mesmas ações que um programa típico:
 - ☉ 1) aceitar dados
 - ☉ 2) realizar alguns cálculos e
 - ☉ 3) devolver resultados
 - ☉ 4) Acionar outros subprogramas
- ☉ Um subprograma possui um propósito específico. Ele recebe dados do programa principal, realiza processamentos e devolve resultados(função) ou não(procedimentos) ao programa principal.

PROGRAMA PRINCIPAL

SUBPROGRAMAS

Programa *TODO_PROGRAMA*

Subprograma *Entrada_Dados*

Subprograma *Processamento_Dados*

Subprograma *Exibicao_Dados*

Programa Principal

CALL Subprograma *Entrada_Dados*

CALL Subprograma *Processamento_Dados*

CALL Subprograma *Saida_Dados*

SUBPROGRAMAS

CLASSIFICAÇÃO

- ◎ Subprogramas ou subalgoritmos podem ser “pensados” ou desenvolvidos considerando duas abordagens:
 - I. DEVOLVER RESULTADOS (ao chamador).
 - II. NÃO DEVOLVER RESULTADOS (ao chamador).

SUBPROGRAMAS

CLASSIFICAÇÃO

- ◎ Subprogramas ou subalgoritmos podem ser “pensados” ou desenvolvidos considerando duas abordagens:
 - I. **Funções**: retornam um valor;
 - II. **Procedimentos** ou **subrotinas**: não retornam valor;

SUBPROGRAMA: FUNÇÃO

- ⊙ Matematicamente, pode-se pensar uma **função** como uma operação que recebe um ou mais valores chamados **argumentos** e produz um valor denominado **resultado**.

$$f(x) = \frac{x}{1 + x^2}$$

$$f(x, y) = \frac{x - y}{\sqrt{x} - \sqrt{y}}$$

- ✓ f é nome da função.
- ✓ (x) e (x, y) são argumentos ou parâmetros.

SUBPROGRAMA: FUNÇÃO

$$f(x) = \frac{x}{1 + x^2}$$

Argumento formal (definição)

$$f(2) = \frac{2}{1 + 2^2} = \frac{2}{5}$$

Argumento real (execução)

SUBPROGRAMA: FUNÇÃO

$$f(x) = \frac{x}{1 + x^2}$$

$$f(x, y) = \frac{x - y}{\sqrt{x} - \sqrt{y}}$$

- ⊙ As linguagens de programação possuem funções incorporadas ou intrínsecas e funções definidas pelo usuário.
- ⊙ Cada função é executada (ativada, chamada) utilizando-se seu nome em uma expressão com os argumentos atuais ou reais entre parênteses.

SUBPROGRAMA: FUNÇÃO

- Em geral as linguagens permitem utilização de dois tipos de funções:
- Funções internas:** incorporadas ou intrínseca à linguagem.
- Funções externas:** definidas pelo programador.

SUBPROGRAMA: SUBROTINA

- ⊙ Com frequência são necessários subprogramas que calculem/retornem vários resultados em vez de apenas um (como é no subprograma tipo função).
- ⊙ Necessidade de aplicar operações em um conjunto de valores sem ter que retornar valor (ordenação).
- ⊙ Necessidade execução de atividades muito específicas como construção de objetos gráficos em dispositivos de saída.

FUNÇÃO VS SUBROTINA

Definição de função

inteiro **elevaQuadro**(x)

Definição de subrotina

limpaTela()

Execução:

inteiro **result**

1) **result** = **elevaQuadro**(3)

2) **limpaTela**()

3) **escreva**(**result**)

REFERÊNCIAS

1. AGUILAR, L.J. Fundamentos de Programação: Algoritmos, estruturas de dados e objetos. McGraw-Hill, São Paulo, 3.ed, 2008..
2. ASCENCIO, Ana F.G. e CAMPOS, Edilene A. V. Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++ e Java. São Paulo: Prentice Hall.