

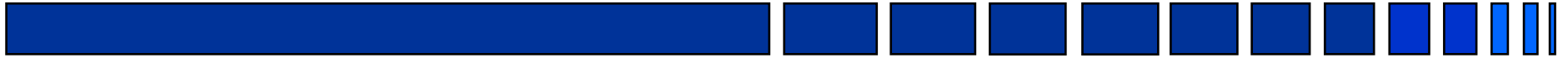


# Algoritmo

Prof. Anderson Almeida Ferreira

Agradeço ao prof. Guilherme Tavares de Assis  
por fornecer slides que fazem parte desta  
apresentação

# Desenvolvimento de programas



- Análise do problema
- Desenvolvimento do algoritmo
- Codificação do programa
- Compilação e execução / Interpretação
- Teste e depuração

# Análise do problema



- Conhecer exatamente o que o problema requer é vital para a solução do mesmo.
- Metodologia:
  - Especificar de forma clara e precisa os dados de entrada e os dados de saída (resultados) do problema.
  - A especificação dos dados de entrada e saída deve responder às seguintes questões:
    - » Quais são os dados de entrada?
    - » Quais são os seus valores válidos e inválidos?
    - » Quais valores serão produzidos?
    - » Qual o formato dos resultados?

# Análise do problema



**Problema:** cálculo da área de um triângulo

- Quais são os dados de entrada?

Base e altura do triângulo.

- Quais são os seus valores válidos e inválidos?

Os dados de entrada devem ser valores numéricos positivos.

- Quais valores serão produzidos?

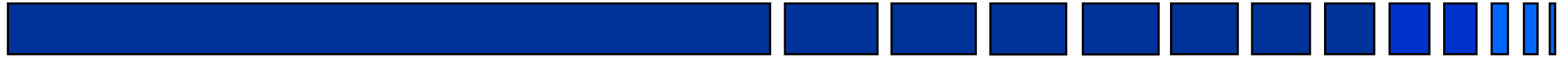
Apenas a área do triângulo através da fórmula:

$$(base \times altura) / 2.$$

- Qual o formato dos resultados?

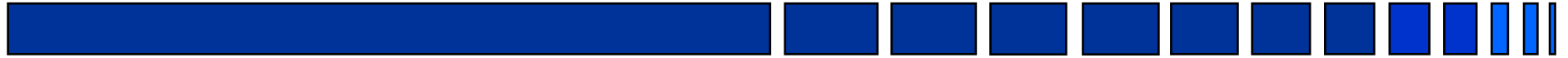
Um valor numérico positivo.

# Desenvolvimento do algoritmo



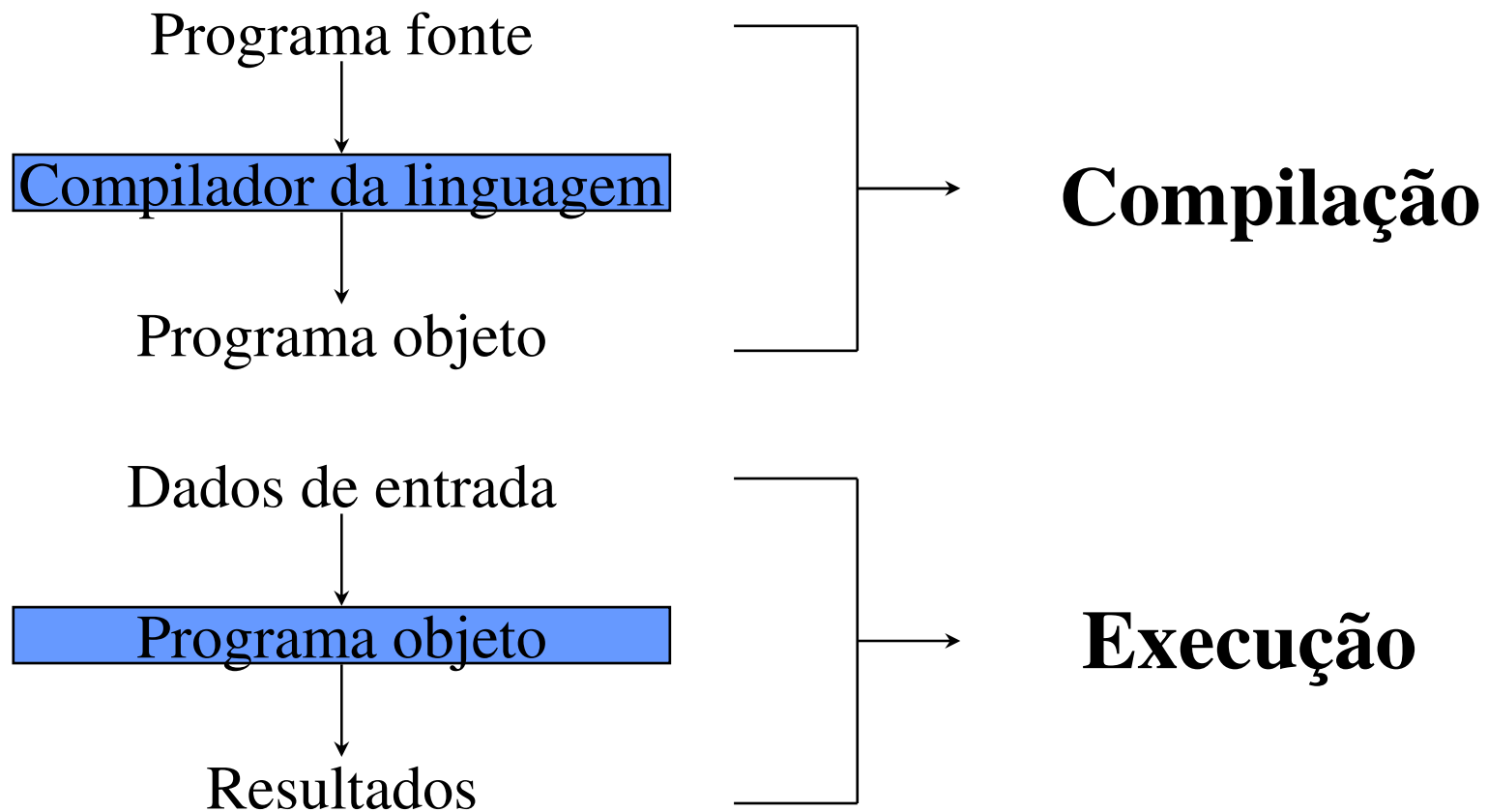
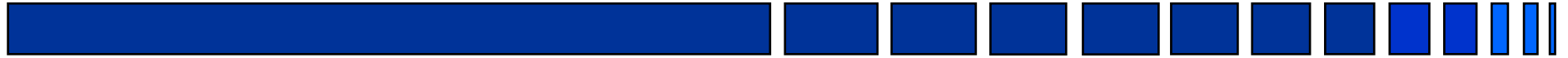
- **Algoritmo** é um conjunto de passos lógicos bem organizado e suficientemente detalhado para se resolver um problema.
- A elaboração do algoritmo pressupõe o conhecimento prévio dos métodos, estratégias e fórmulas para a solução do problema.
- A versão inicial descreve, em linhas gerais, as principais funções a serem executadas.
- Em seguida, o algoritmo deve ser refinado até que a sequência de instruções resultante esteja bem clara e detalhada.

# Codificação do programa

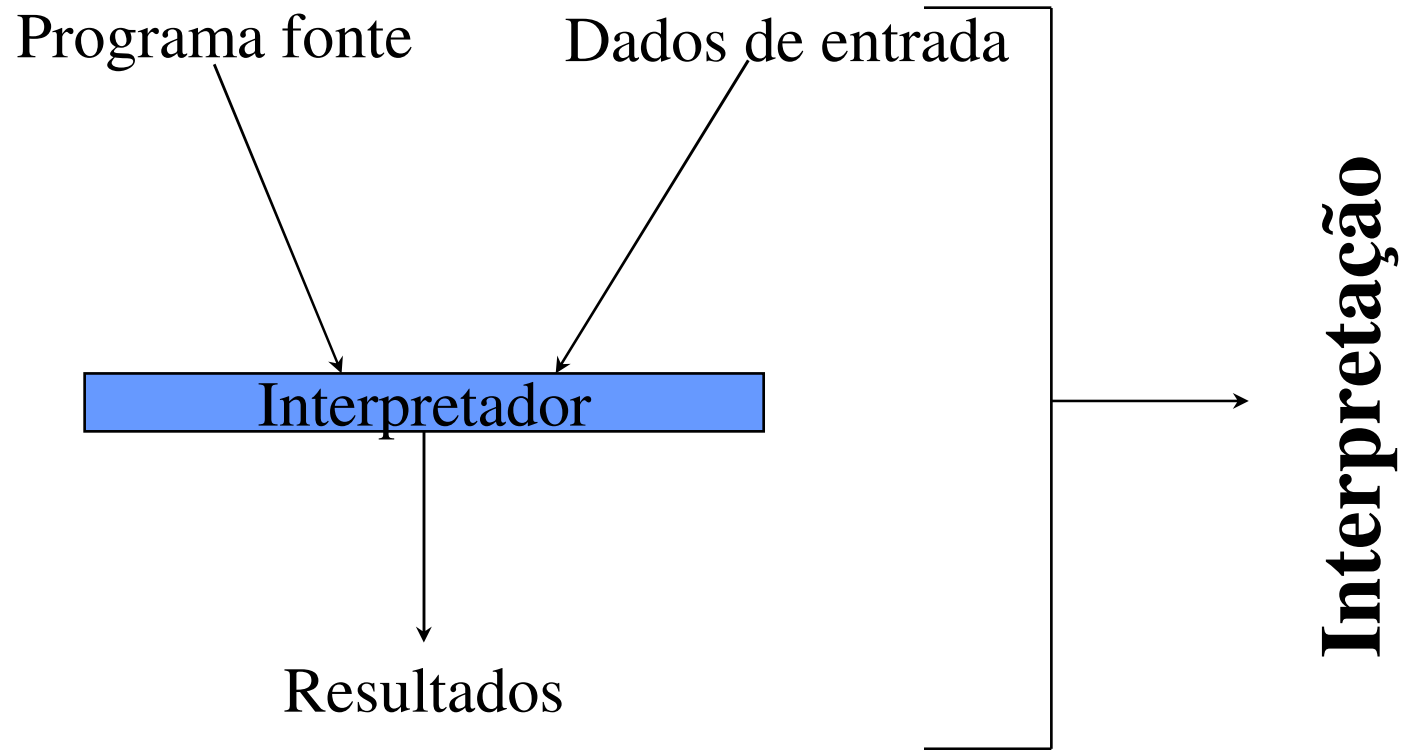
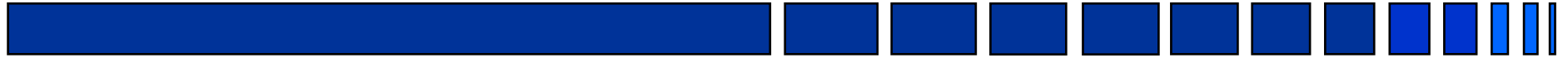


- O processo de **codificação** se resume no mapeamento das instruções de um algoritmo em instruções de uma linguagem de programação.
- Para um melhor entendimento do programa são essenciais:
  - documentação;
  - indentação;
  - escolha representativa dos identificadores.
- A documentação consiste de:
  - cabeçalho;
  - comentários.

# Compilação e execução




# Interpretação

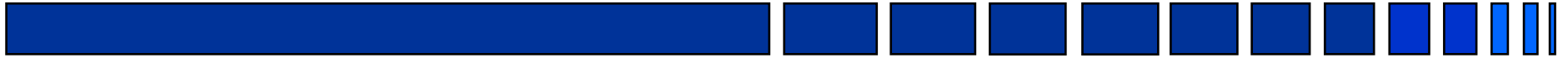




# Teste e depuração

- 
- **Teste** é o processo de executar um programa com a finalidade de encontrar erros.
  - **Depuração** é o processo de localizar e corrigir os erros encontrados.
  - Possíveis tipos de erros são:
    - Erros de compilação:
      - » escrever errado um comando;
      - » usar dados inadequados a uma operação.
    - Erros de execução:
      - » divisão por zero;
      - » *loop* infinito.
    - Erros de lógica:
      - » erro nos limites de uma comparação;
      - » erro nas fórmulas e expressões.

# Refinamentos sucessivos

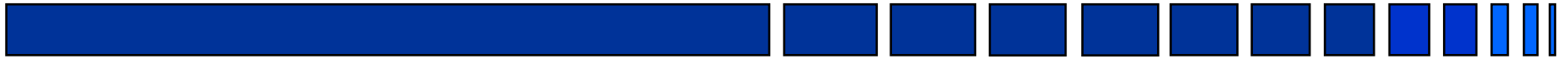


**Problema:** cálculo da área de um triângulo

Versão inicial do algoritmo:

- Forneça a base e a altura do triângulo
- Realize o cálculo da área do triângulo

# Refinamentos sucessivos



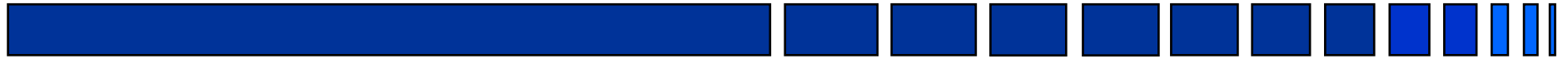
- Ref. Forneça a base e a altura do triângulo  
Leia base, altura
  
- Ref. Realize o cálculo da área do triângulo  
Se a base e a altura forem positivos  
Então  
    Calcule a área do triângulo  
    Apresente o valor calculado  
Senão  
    Apresente uma mensagem de erro  
Fim Se

# Refinamentos sucessivos



- Ref. Base e altura forem positivos  
(base > 0) e (altura > 0)
- Ref. Calcule a área do triângulo  
área ← (base × altura) / 2
- Ref. Apresente o valor calculado  
Escreva área
- Ref. Apresente uma mensagem de erro  
Escreva "mensagem de erro"

# Refinamentos Sucessivos



## Algoritmo final:

Leia base, altura

Se (base > 0) e (altura > 0)

Então

    área ← (base × altura) / 2

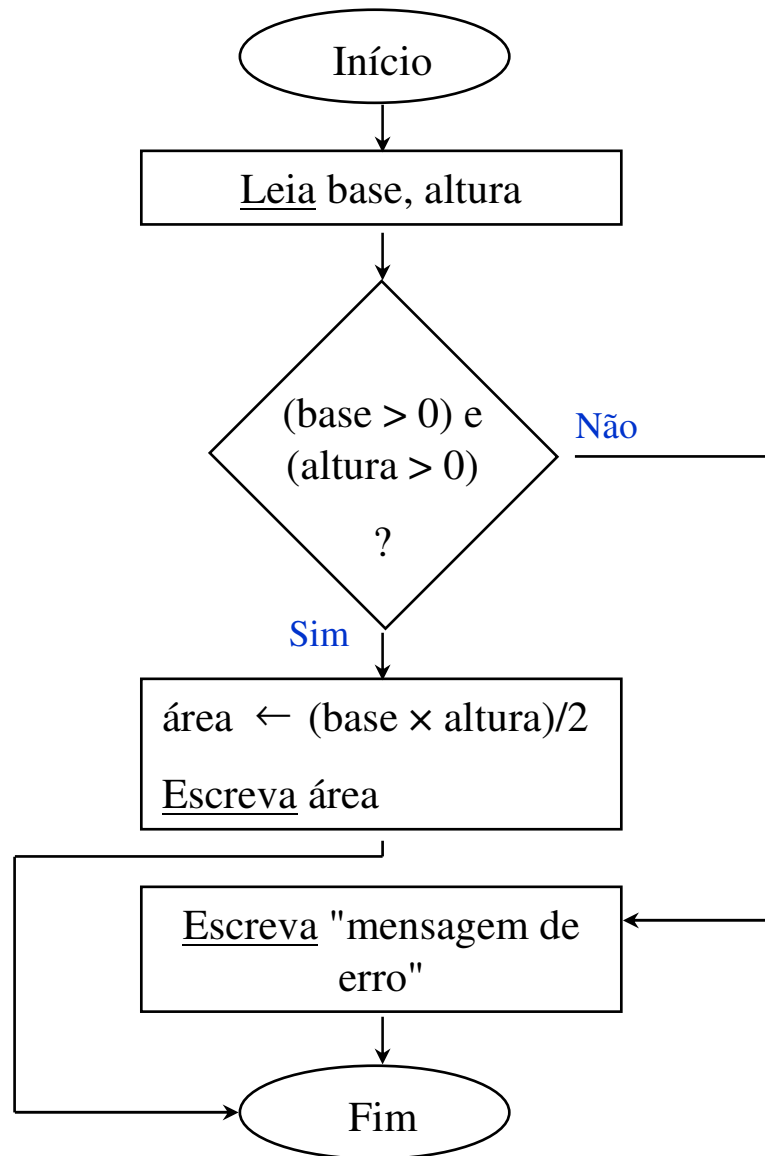
Escreva área

Senão

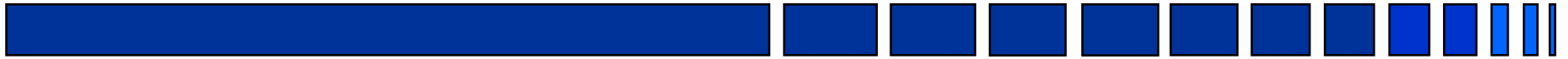
Escreva "mensagem de erro"

Fim Se

Diagrama de blocos ou fluxograma:



# Refinamentos sucessivos

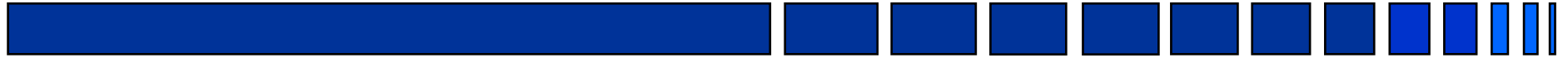


**Problema:** apresentar o menor dentre três valores numéricos.

Versão inicial do algoritmo:

- Forneça os três números
- Determine o menor número
- Apresente o menor número

# Refinamentos sucessivos



- Ref. Forneça os três números

Leia A, B, C

- Ref. Determine o menor número

Se  $(A < B)$  e  $(A < C)$

Então menor  $\leftarrow$  A

Senão Determine o menor dentre B e C

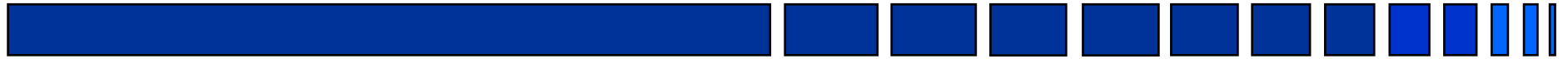
Fim Se

- Ref. Apresente o menor número

Escreva menor



# Refinamentos sucessivos



- Ref. Determine o menor dentre B e C

Se  $(B < C)$

Então

menor  $\leftarrow B$

Senão

menor  $\leftarrow C$

Fim Se

# Refinamentos sucessivos

**Algoritmo final:**

Leia A, B, C

Se (A < B) e (A < C)

Então

menor ← A

Senão

Se (B < C)

Então

menor ← B

Senão

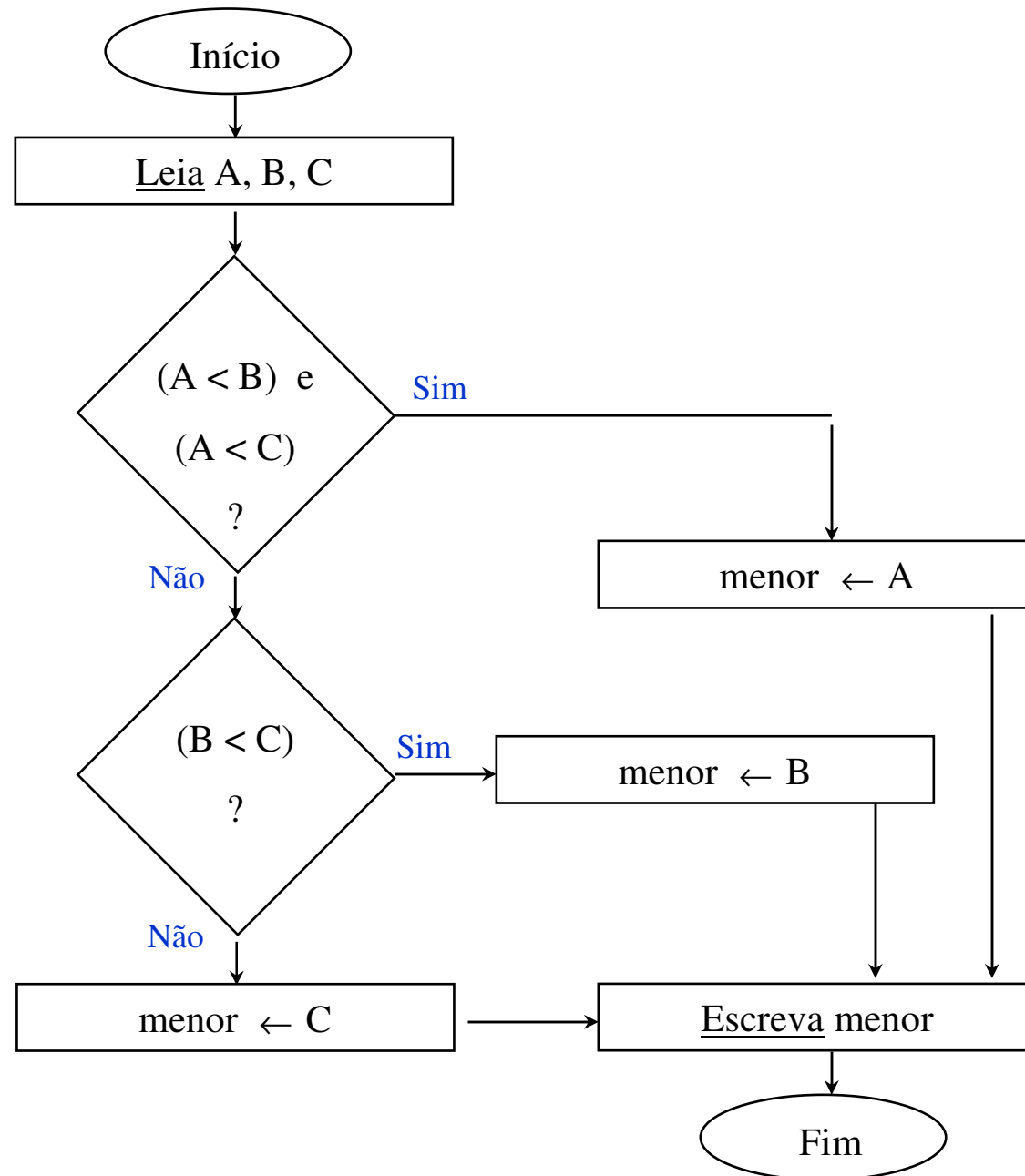
menor ← C

Fim Se

Fim Se

Escreva menor

Diagrama de blocos ou fluxograma:



# O Comando if

```
if <condição> then
    <bloco "então">
else
    <bloco "senão">
end
```


# Cláusula else vazia

```
if <condição> then
  <bloco "então">
else
  // Nenhum comando aqui
end
```

```
if <condição> then
  <bloco "então">
end
```

# Equações de Segundo Grau: Programa Scilab – Eq2g\_3.sce - 2

```
//Cálculo e impressão das raízes
delta = b^2 - 4*a*c;
if delta > 0 then
    r1 = (-b+sqrt(delta))/(2*a);
    r2 = (-b-sqrt(delta))/(2*a);
    printf("Raízes: %g e %g.", r1, r2);
else
    printf("Raízes complexas.")
end
```



# Partes de um comando If

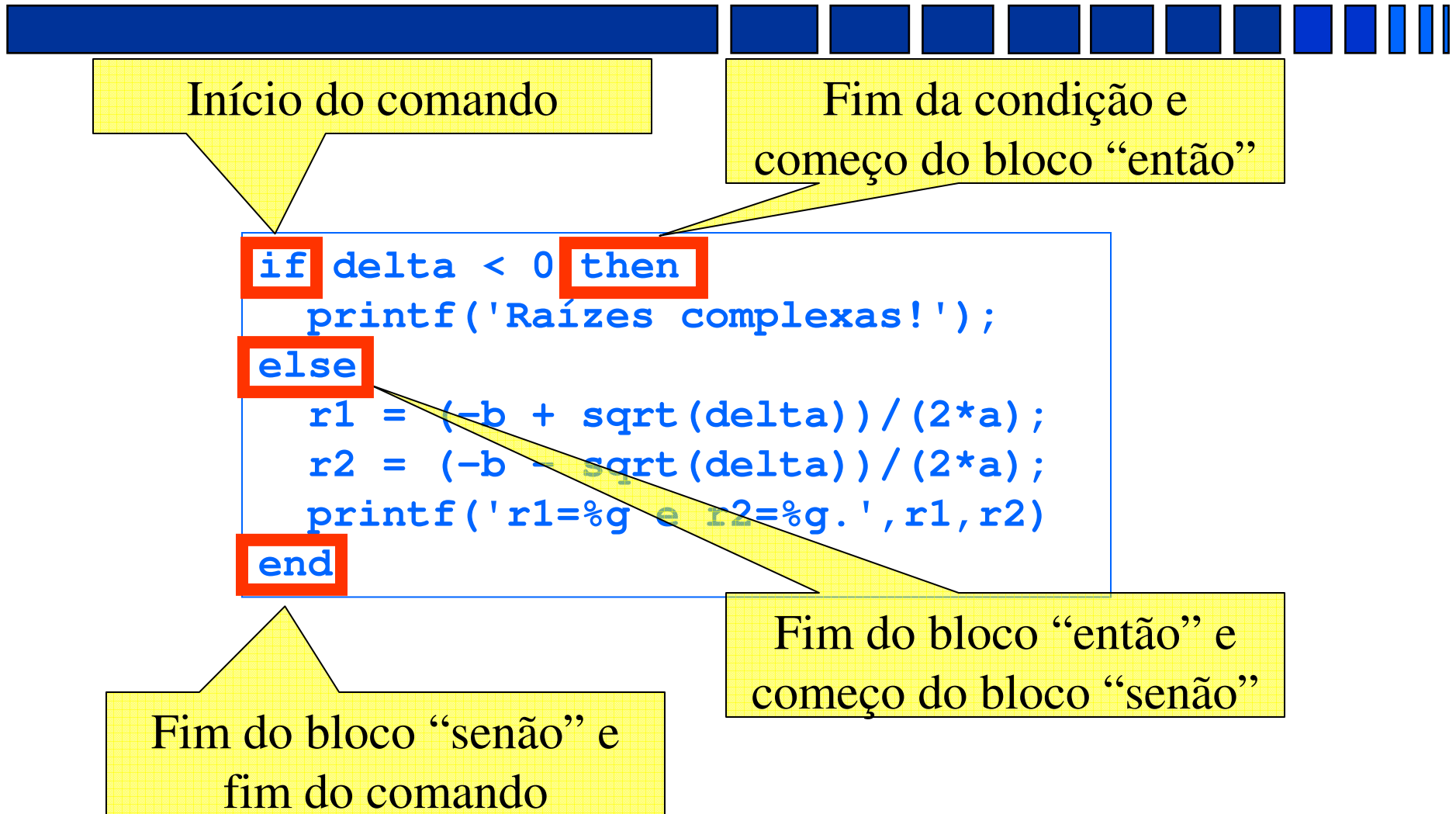
<condição>

<bloco “então”>

```
if delta >= 0 then
    r1 = (-b+sqrt(delta))/(2*a)
    r2 = (-b-sqrt(delta))/(2*a)
    printf("As raízes são %g e %g", r1, r2)
else
    printf("As raízes são complexas")
end
```

<bloco “senão”>

# Palavras-chave de um Comando if





# Operadores Relacionais

>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual a
==	igual a
<> ou ~=	diferente de

# Tipos primitivos



- Inteiro
  - Ex.: 2, 25, -30, 255
- Ponto-flutuante – reais
  - Ex.: 2.5, 0.27, 2.456D-5, 2.7D+5
- String – Cadeia de caracteres
  - “abcdefghij”
  - ‘abcdefghij’
  - “Cadeia de caracteres”
  - ‘Programação de computadores’
  - “Programação ““de”” computadores”

# Exercícios

- Criar um algoritmo e um programa em Scilab que leia dois valores para as variáveis A e B, que efetue a troca dos valores de forma que a variável A passe a ter o valor da variável B e que a variável B passe a ter o valor da variável A. Apresente os valores trocados.
- Criar um algoritmo e um programa em Scilab para calcular e apresentar o valor do volume de uma lata de óleo, utilizando a fórmula:
$$V = 3.14159 \times R^2 \times h$$
  - onde  $V$  é o volume,  $R$  é o raio e  $h$  é a altura.
- Criar um algoritmo e um programa em Scilab que leia um valor de hora (hora:minutos) e informe (calcule) o total de minutos se passaram desde o início do dia (0:00h).

# Refinamentos Sucessivos



**Problema:** apresentar a soma dos números pares de 100 até 200, inclusive.

Versão inicial do algoritmo:

- Realize a soma desejada
- Apresente o valor da soma

Ref. Realize a soma desejada

Inicialize a soma com zero

Inicialize o número par com 100

Enquanto o número par for menor ou igual a 200 Faça

    Adicione à soma o valor do número par

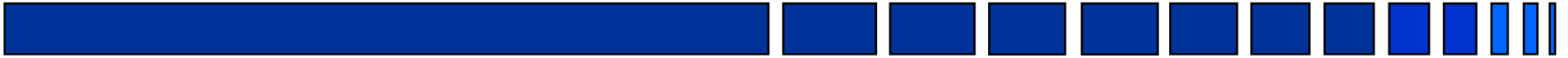
    Determine o próximo número par

fim enquanto

Ref. Apresente o valor da soma

Escreva soma

# Refinamentos Sucessivos



Ref. Inicialize a soma com zero

$soma \leftarrow 0$

Ref. Inicialize o número par com 100

$par \leftarrow 100$

Ref. Adicione à soma o valor do número par

$soma \leftarrow soma + par$

Ref. Determine o próximo número par

$par \leftarrow par + 2$

# Refinamentos Sucessivos



## Algoritmo final:

soma  $\leftarrow$  0

par  $\leftarrow$  100

Enquanto par  $\leq$  20 Faça

| soma  $\leftarrow$  soma + par

| par  $\leftarrow$  par + 2

Fim enquanto

Escreva soma

# Refinamentos Sucessivos

**Diagrama de blocos ou fluxograma:**

