

# Fluxogramas

Vanessa Braganholo  
vanessa@ic.uff.br

# Antes de vermos Fluxogramas...

---

... vamos conhecer um pouco sobre a evolução das linguagens de programação

# Paradigmas de Programação

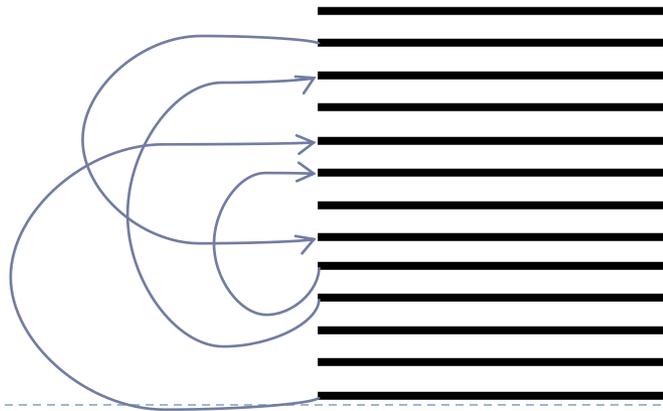
---

- ▶ Definem regras básicas que linguagens pertencentes a esse paradigma devem seguir
- ▶ Exemplos (dentre outros):
  - ▶ Não estruturado } *Não é mais usado*
  - ▶ Estruturado } *Veremos nesta aula*
  - ▶ Procedimental } *Veremos mais para frente no curso*
  - ▶ Orientado a Objetos }

# Paradigma não estruturado

---

- ▶ Anos 50
- ▶ Linhas de código sequenciais em um mesmo arquivo
- ▶ Sequência de execução alterada com uso do GOTO
- ▶ Código macarrônico (ilegível para sistemas grandes)
- ▶ Exemplos: primeiras versões de BASIC e COBOL



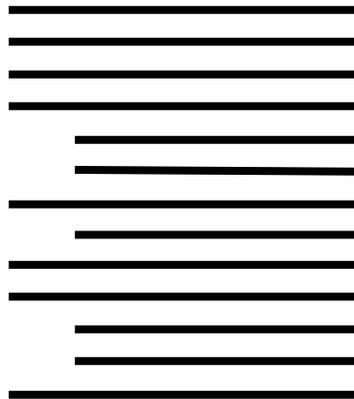
*Imaginem isso  
para 1.000.000 de  
linhas de código!*

# Paradigma estruturado

---

- ▶ Anos 60
- ▶ Artigo “Go To Statement Considered Harmful” de Edsger Dijkstra
- ▶ Eliminação de GOTO e adoção de três estruturas básicas:

- ▶ Sequência (de cima para baixo)
- ▶ Decisão (se...então...senão)
- ▶ Repetição (loops)



*Foco da 1ª metade do curso*

# Paradigma estruturado

---

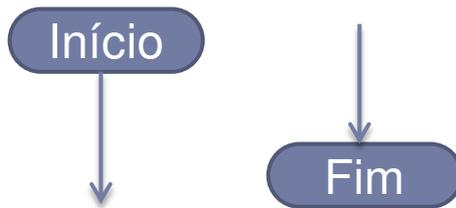
- ▶ Qualquer função computável pode ser descrita usando sequência, decisão e repetição
- ▶ Cada um desses elementos pode ser representado graficamente
  - ▶ Representação gráfica denominada Fluxograma (Flowchart)
  - ▶ Facilita o projeto e o entendimento de programas



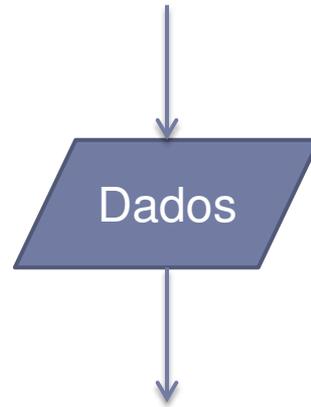
# Fluxograma (sequência)

---

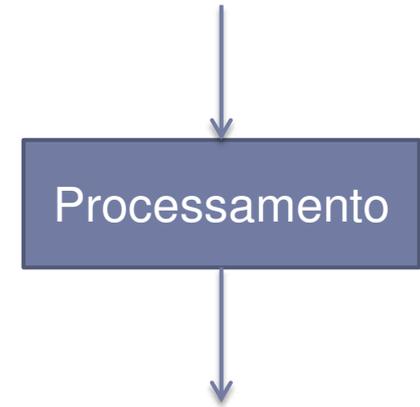
- ▶ Elementos básicos para representar sequência em fluxogramas:



Marcação de  
início/fim



Entrada e saída  
de dados



Operações de  
processamento

# Problema puramente sequencial

---

- ▶ Calcule o volume e o peso de uma esfera oca, onde o diâmetro, a espessura da parede e a densidade são informados
- ▶ Em português estruturado:

Leia o diâmetro, a espessura e a densidade

Calcule o raio externo

Calcule o raio interno

Calcule o volume

Calcule o peso

Escreva o volume e o peso

# Problema puramente sequencial

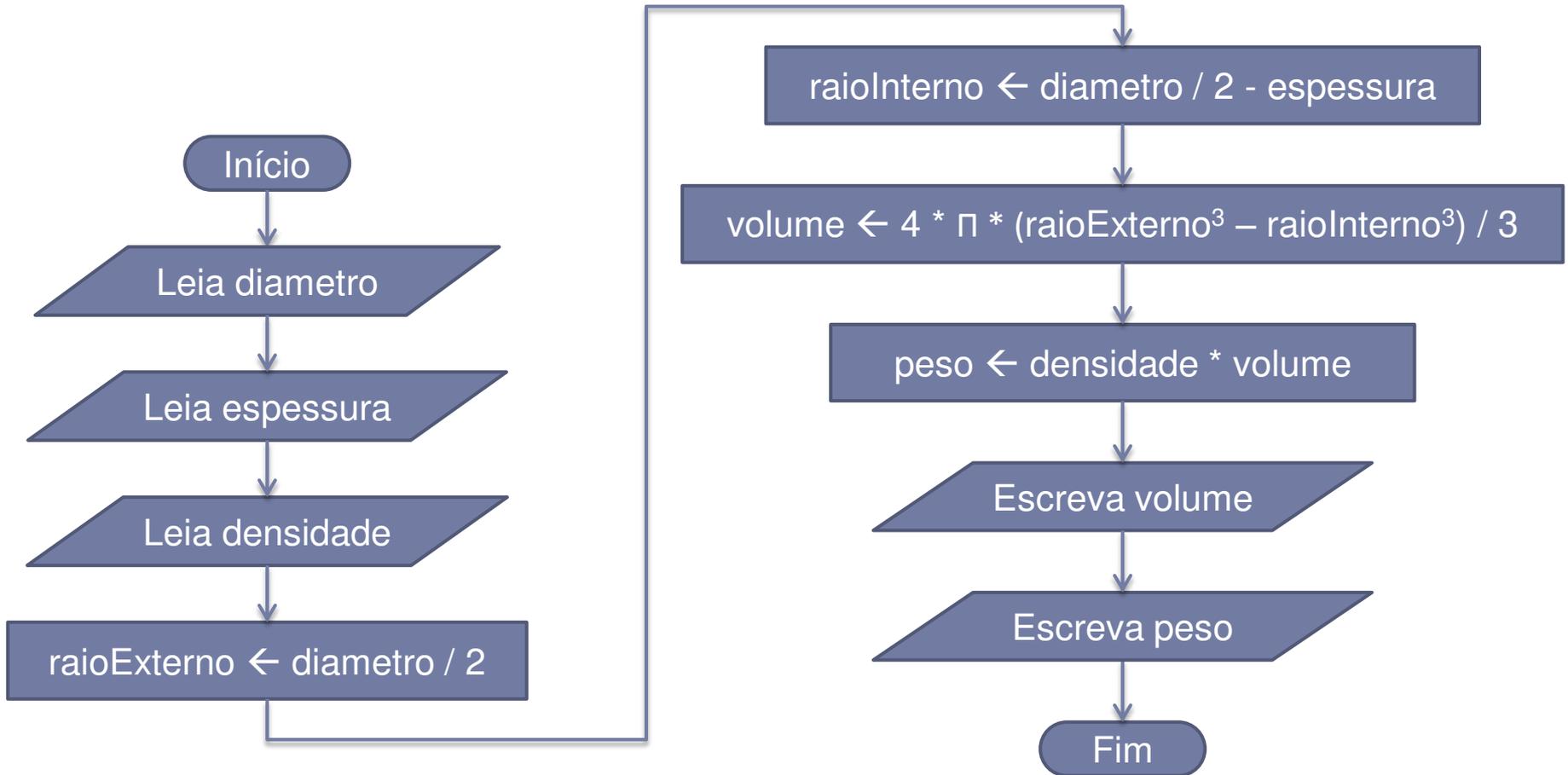
---

## ▶ Em pseudocódigo:

```
Leia diametro, espessura, densidade
raioExterno ← diametro / 2
raioInterno ← diametro / 2 - espessura
volume ← 4 * Π * (raioExterno3 - raioInterno3) / 3
peso ← densidade * volume
Escreva volume, peso
```

# Problema puramente sequencial

## ► Em fluxograma:



# Características observáveis

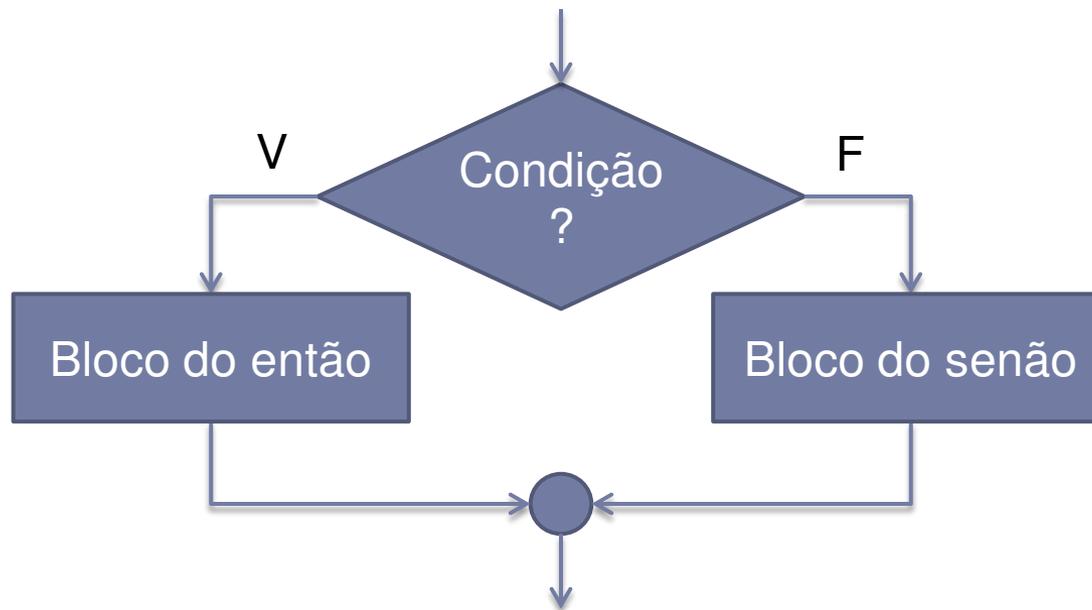
---

- ▶ Português estruturado
  - ▶ Baixo grau de formalismo
  - ▶ Fácil de fazer
  - ▶ Alto grau de ambiguidade
- ▶ Pseudocódigo
  - ▶ Nível um pouco maior de formalismo
  - ▶ Notação textual
- ▶ Fluxograma
  - ▶ Definição clara dos tipos de ação
  - ▶ Notação gráfica
- ▶ Quando existem símbolos de dados ou de processamento contíguos no fluxograma, **um único símbolo** pode ser utilizado com os comandos em diferentes linhas

# Fluxograma (decisão)

---

- ▶ Elemento básico para representar decisão em fluxogramas:



# Problema com decisão

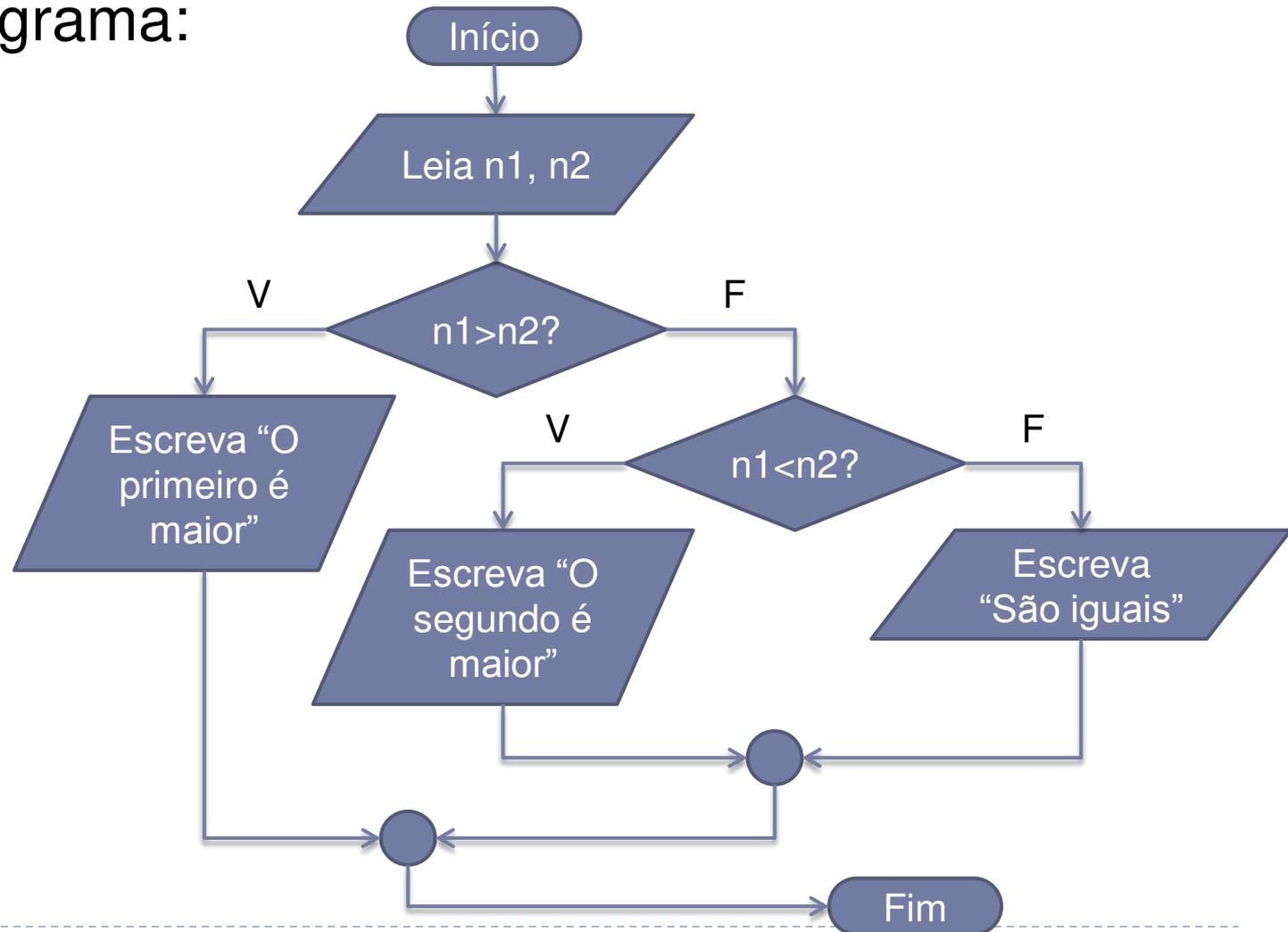
---

- ▶ Informe o maior número entre dois números N1 e N2 informados
- ▶ Em pseudocódigo:

```
Leia n1, n2
Se n1 > n2 então
    Escreva "O primeiro é maior"
Senão se n1 < n2 então
    Escreva "O segundo é maior"
Senão
    Escreva "São iguais"
```

# Problema com decisão

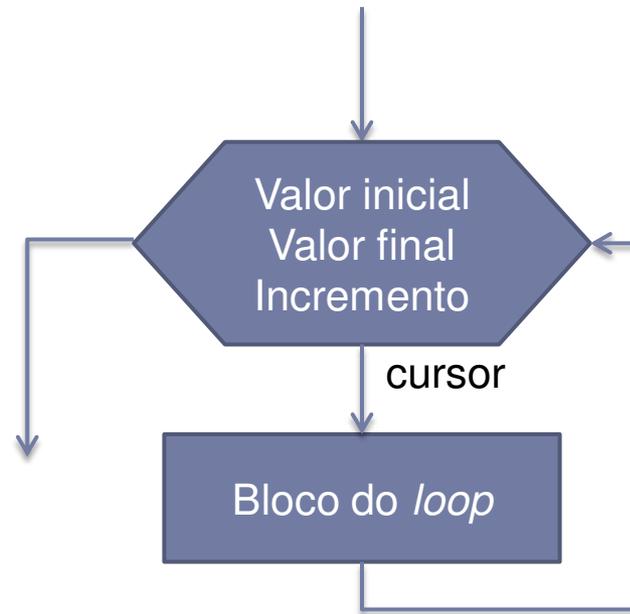
► Em fluxograma:



# Fluxograma (repetição contável)

---

- ▶ Elemento básico para representar repetição contável em fluxogramas:



# Problema com repetição contável

---

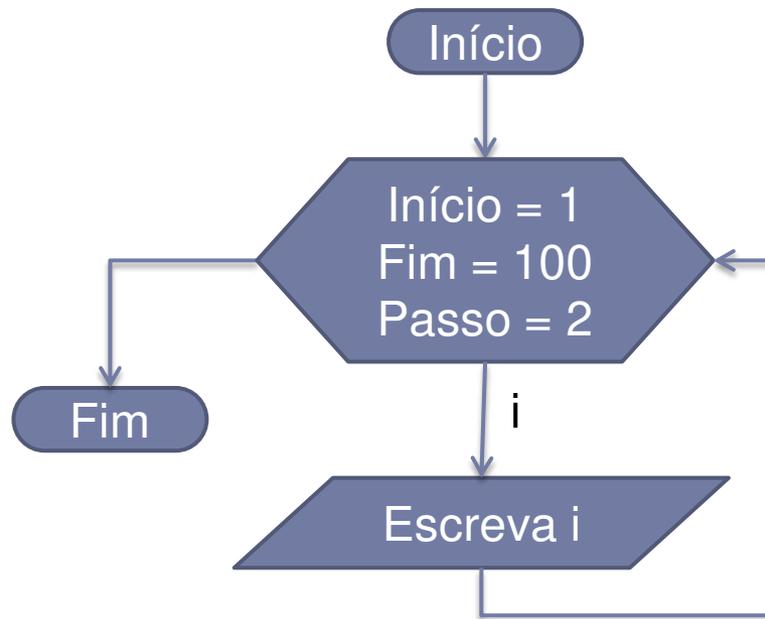
- ▶ Liste todos os números ímpares de 1 a 100
- ▶ Em pseudocódigo:

```
Para i variando de 1 a 100 com passo 2  
  Escreva i
```

# Problema com repetição contável

---

- ▶ Em fluxograma:



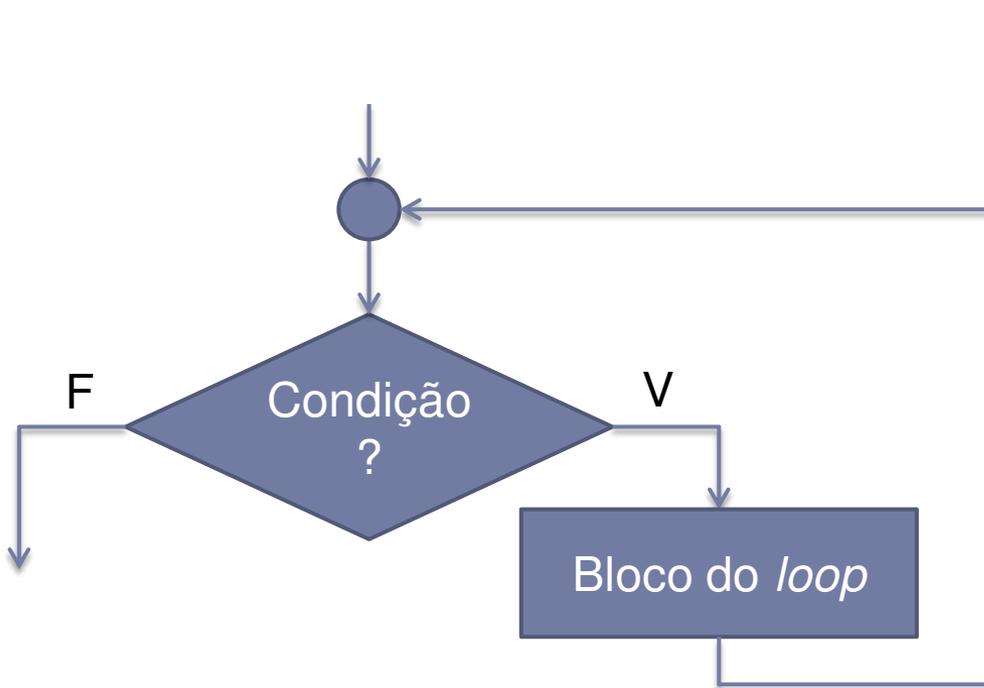
# Características observáveis

---

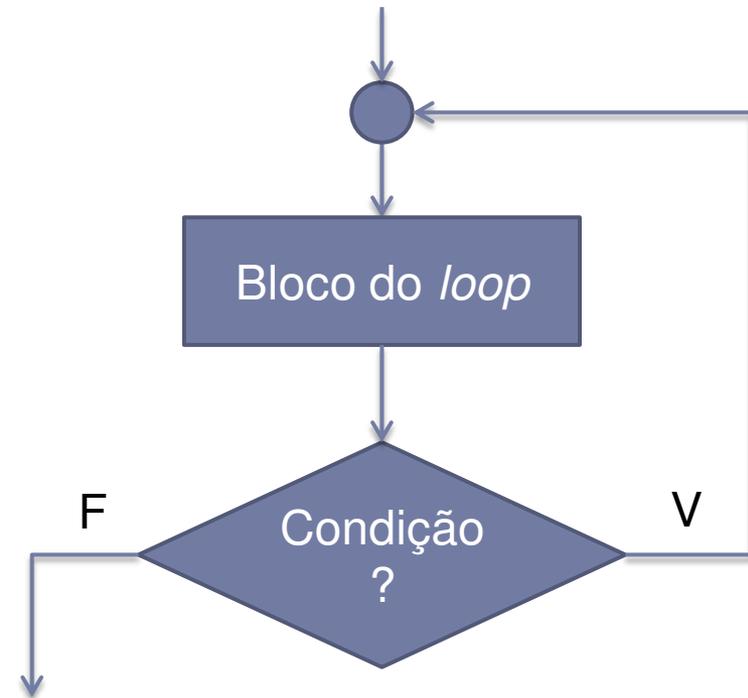
- ▶ Só é possível utilizar repetição contável se conseguirmos determinar de antemão quantas repetições são necessárias
- ▶ O computador controlará as repetições
  - ▶ Total de  $(\text{fim} - \text{início}) / \text{passo} + 1$  repetições
- ▶ Normalmente se deseja acessar o valor da variável que controla a repetição, a cada iteração

# Fluxograma (repetição condicional)

- ▶ Elementos básicos para representar repetição condicional em fluxogramas:



Tipo enquanto...faça



Tipo faça...enquanto

# Problema com repetição condicional

---

- ▶ Some todos os números informados até que o número zero seja informado

```
soma ← 0
```

```
Leia n
```

```
Enquanto n ≠ 0 faça
```

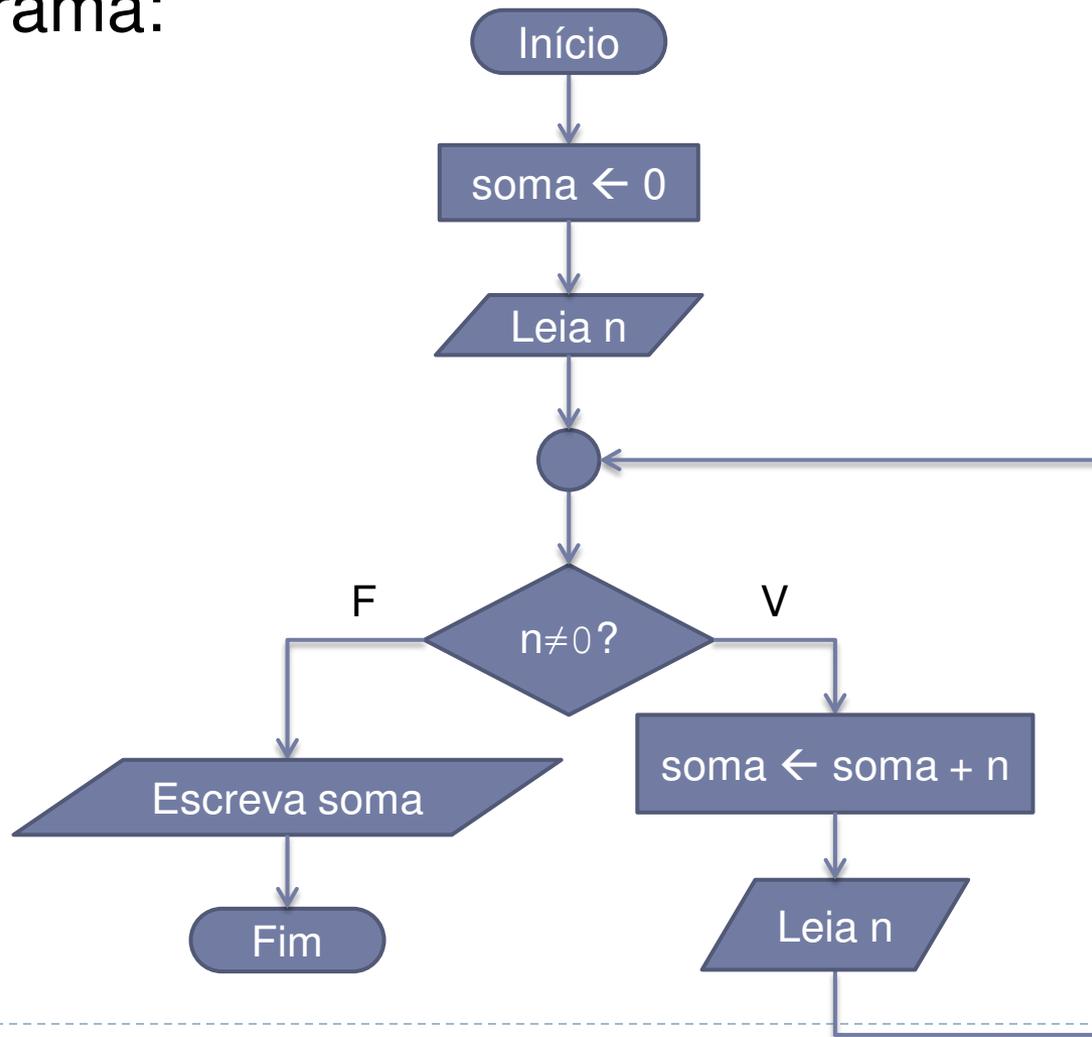
```
    soma ← soma + n
```

```
    Leia n
```

```
Escreva soma
```

# Problema com repetição condicional

- ▶ Em fluxograma:



# Características observáveis

---

- ▶ A condição de controle do *loop* deve alternar para falso em algum momento
  - ▶ Sem isso, o programa entrará em *loop* infinito
- ▶ Antes de dar como terminado o algoritmo, é importante testar!
  - ▶ Faça uso do Método Chinês
  - ▶ Atenção especial para os extremos (início e término do *loop*)

# Método Chinês

---

- ▶ Ler o algoritmo do início ao fim, obedecendo cada uma das instruções
- ▶ Anotar os valores de cada variável para cada passo do algoritmo
- ▶ Ao final, será possível identificar se alguma instrução está se comportando de forma inapropriada

# Exercício

---

- ▶ Construa um algoritmo (pseudocódigo e fluxograma) para dizer se um número inteiro informado pelo usuário é par ou ímpar
- ▶ Utilize o Método Chinês para verificar se o algoritmo está correto

# Exercício

---

- ▶ Construa um algoritmo (pseudocódigo e fluxograma) para montar a tabela de multiplicação de números de 1 a 10 (ex.:  $1 \times 1 = 1$ ,  $1 \times 2 = 2$ , etc.)
- ▶ Utilize o Método Chinês para verificar se o algoritmo está correto

# Exercício

---

- ▶ Construa um algoritmo (pseudocódigo e fluxograma) para determinar o número de dígitos de um número informado
- ▶ Utilize o Método Chinês para verificar se o algoritmo está correto

# Exercício

---

- ▶ Construa um algoritmo (pseudocódigo e fluxograma) para listar todos os divisores de um número ou dizer que o número é primo caso não existam divisores
- ▶ Ao final, verifique se o usuário deseja analisar outro número
  
- ▶ Utilize o Método Chinês para verificar se o algoritmo está correto

# Exercício

---

- ▶ Construa um algoritmo (pseudocódigo e fluxograma) para determinar as raízes de uma equação de 2º grau:  $ax^2 + bx + c = 0$  (recordar que o discriminante  $\Delta = b^2 - 4ac$ , e que a raiz  $r = (-b \pm \sqrt{\Delta})/2a$ )
- ▶ Utilize o Método Chinês para verificar se o algoritmo está correto

# Exercício

---

- ▶ Construa um algoritmo (pseudocódigo e fluxograma) para calcular a série de Fibonacci para um número informado pelo usuário, sendo  $F(0) = 0$ ,  $F(1) = 1$  e  $F(n) = F(n-1) + F(n-2)$ 
  - ▶ Por exemplo, caso o usuário informe o número 9, o resultado seria: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.
- ▶ Utilize o Método Chinês para verificar se o algoritmo está correto

# Exercício

---

- ▶ Construa um algoritmo (pseudocódigo e fluxograma) para indicar, a partir de um valor informado em centavos, a menor quantidade de moedas que representa esse valor.
- ▶ Considere moedas de 1, 5, 10, 25 e 50 centavos, e 1 real. Exemplo: para o valor 290 centavos, a menor quantidade de moedas é 2 moedas de 1 real, 1 moeda de 50 centavos, 1 moeda de 25 centavos, 1 moeda de 10 centavos e 1 moeda de 5 centavos
- ▶ Utilize o Método Chinês para verificar se o algoritmo está correto

# Referências

---

- ▶ Slides de Leonardo Murta
- ▶ Slides baseados no curso de C da Prof. Vanessa Braganholo
- ▶ Alguns exercícios extraídos do livro Furlan, M., Gomes, M., Soares, M., Concilio, R., 2005, “Algoritmos e Lógica de Programação”, Editora Thomson.

# Fluxogramas

Vanessa Braganholo  
vanessa@ic.uff.br