

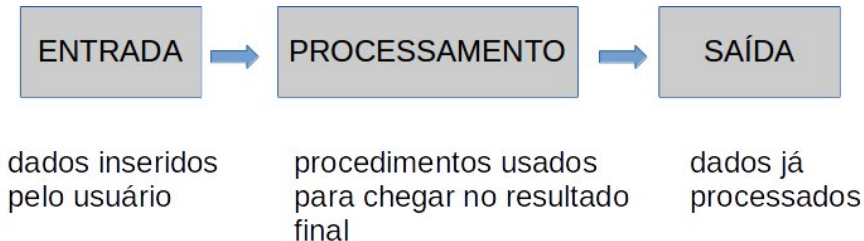
Fluxogramas e variáveis

Patrícia de Siqueira Ramos

UNIFAL-MG, *campus* Varginha

19 de Fevereiro de 2020

Fases de um algoritmo



O que é necessário para programar

- fluxograma: representação gráfica
- pseudocódigo: português estruturado
- linguagem de programação: linguagem desenvolvida para viabilizar a programação de computadores
- ambiente de programação: conjunto de tecnologias que dão suporte à programação de computadores (sistema operacional, editor, compilador)

Fluxograma

Forma padronizada e eficaz para representar a sequência de instruções, facilitando a visualização dos passos



Início/fim



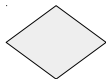
Entrada de dados (geralmente via teclado)



Processamento de dados (cálculos, atribuições)



Saída (visual) de dados

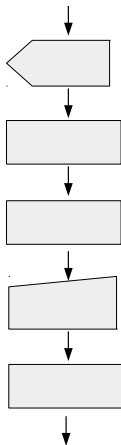


Decisão a ser tomada, condição avaliada com possibilidade de desvio

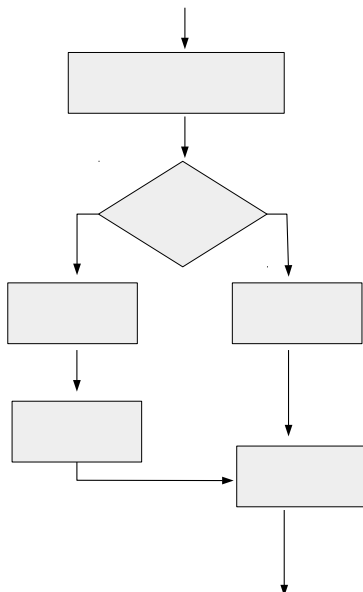


Conector usado quando é preciso particionar o diagrama, permitindo o desvio para outro ponto

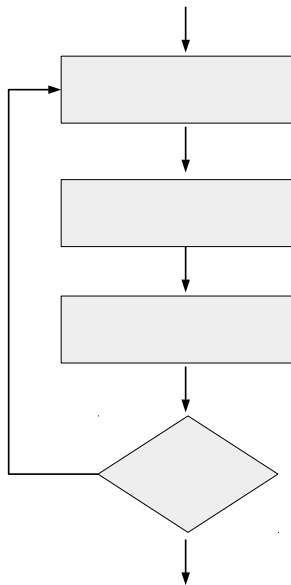
Fluxograma - estrutura de controle sequencial



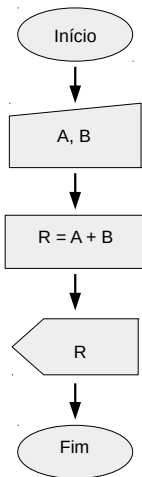
Fluxograma - estrutura de controle condicional



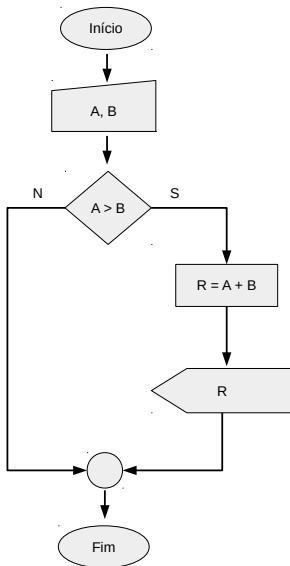
Fluxograma - estrutura de controle repetitiva



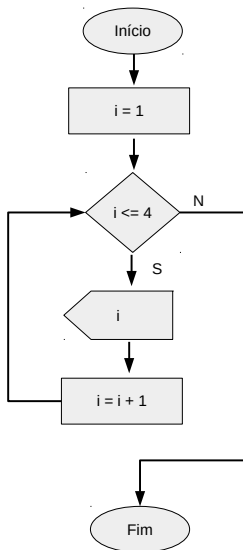
Ex.: fluxograma - estrutura de controle sequencial



Ex.: fluxograma - estrutura de controle condicional



Ex.: fluxograma - estrutura de controle repetitiva



Tipos de dados

- **Numéricos:**

- Inteiros (positivos e negativos):

- Ex: 1, -3, 0, 8990

- Reais:

- Ex.: 3.41, -111.333333, 0.567834, pi

Tipos de dados

- **Caracteres:** delimitados por aspas simples ou duplas (letras, números símbolos), mensagens ao usuário.

Ex.: 'Insira um número:', "Apto 202", 'O valor inserido foi:', "44"

Tipos de dados

- **Lógicos:** também chamados de valores *booleanos*, indicam presença/ausência.

Ex.: V e F (*True* e *False*), Sim e Não, 1 e 0

Variável

- Uma variável é a representação de uma região da memória para armazenar/acessar um valor. Ela pode ser numérica, caractere ou lógica.
- O nome de uma variável é usado para identificá-la e representá-la em um programa:

Variável

- Uma variável é a representação de uma região da memória para armazenar/acessar um valor. Ela pode ser numérica, caractere ou lógica.
- O nome de uma variável é usado para identificá-la e representá-la em um programa:
 - O primeiro caractere de identificação de uma variável não pode ser um número ou um símbolo (\$, #, ;, (, [, etc.), deve ser sempre alfabético.

- Exemplos de nomes válidos de variáveis:
Nome, nome, n1, x, Y4, nome_usuario etc.

- Exemplos de nomes válidos de variáveis:
Nome, nome, n1, x, Y4, nome_usuario etc.
- Não usar espaços em branco para dividir o nome da variável.
Ex. de nomes inválidos de variáveis:
aluno a
valor 1

Variável

- Exemplos de nomes válidos de variáveis:
Nome, nome, n1, x, Y4, nome_usuario etc.
- Não usar espaços em branco para dividir o nome da variável.
Ex. de nomes inválidos de variáveis:
aluno a
valor 1
- Não utilizar um comando da linguagem usada para um nome de variável.
Ex.: for = 5 ou if = 8 não podem ser utilizadas.

Comandos básicos

- Existem alguns comandos básicos utilizados nas linguagens de programação
- O comando mais utilizado é o de atribuição que pode ser:

```
variavel = valor
```

ou

```
variavel <- valor
```

Ex.:

```
x = 4      ou      x <- 4
```

Operadores

- **aritméticos:** $+$, $-$, $/$, $*$, `sqrt()`, `**` (potenciação), divisão inteira (`//`), resto da divisão (`%`), arredondamento etc.

Operadores

- **aritméticos:** $+$, $-$, $/$, $*$, `sqrt()`, `**` (potenciação), divisão inteira (`//`), resto da divisão (`%`), arredondamento etc.
- **lógicos:** `and`, `or`, `not` (e, ou, não)

Operadores

- **aritméticos:** $+$, $-$, $/$, $*$, `sqrt()`, `**` (potenciação), divisão inteira (`//`), resto da divisão (`%`), arredondamento etc.
- **lógicos:** **and**, **or**, **not** (e, ou, não)
- **relacionais:** utilizados para testar condições:
 - `==` (o mesmo que `=`)
 - `!=` (o mesmo que \neq)
 - `<=` (o mesmo que \leq)
 - `>=` (o mesmo que \geq)
 - `<`
 - `>`

Expressões booleanas

and	V	F
V	V	F
F	F	F

Expressões booleanas

and	V	F
V	V	F
F	F	F

or	V	F
V	V	V
F	V	F

Expressões booleanas

and	V	F
V	V	F
F	F	F

or	V	F
V	V	V
F	V	F

not	V	F
	F	V

Precedência das operações

- parênteses: `()`
- potenciação: `**`
- multiplicação ou divisão: `*`, `/`, `//`, `%`
- adição ou subtração: `+`, `-`
- relacional: `==`, `!=`, `<=`, `>=`, `>`, `<`
- lógicos: `not`, `and`, `or`

Exercícios

1. Como escrever a seguinte expressão aritmética na forma computacional?

$$x = 9 + \{43 \cdot [55 / (30 - 2, 5)]\}$$

2. Quais os resultados das seguintes expressões?

- a) $1 + 7 * 2 * * 2 - 1$
- b) $3 * (1 - 2) + 4 * 2$
- c) *round*(8.7)
- d) $20 == 6$
- e) $4 ! = 6$
- f) $(10 + 2) >= 12$
- g) $(5 >= 7) \text{ or } (8 <= 10)$
- h) $(3 >= 5) \text{ and } (5 <= 3)$
- i) *not* $7 == 5$

Exercícios

3. Quais os valores das variáveis?

$A = 4.6$

$B = \text{sqrt}(\text{int}(A))$ obs.: *int* retorna a parte inteira do número

$C = \text{round}(A) * B$

$D = (A * B) / C$

$E = B ** 3$

4. Como escrever

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

em linguagem computacional?

Exercícios

5. Sabendo que $a = 3$, $b = 7$ e $c = 4$, informe se as expressões são V ou F:

- a) $(a + c) > b$
- b) $(b + a) \leq c$
- c) $b \geq (a + 2)$
- d) $(c + a) \geq b$
- e) $c == (b - a)$
- f) $b * 2 > c * 4$
- g) $b // c == a$ $//$: divisão inteira
- h) $b \% c == a$ $\%$: resto da divisão

6. Sabendo que $sal = 1000$ e $salliq = 900$, informe se as expressões são V ou F:

- a) $salliq \geq 900$
- b) $salliq < 900$
- c) $salliq == (sal - 10)$

7. Sabendo que $A = 5$, $B = 4$, $C = 3$ e $D = 6$, informe se as expressões são V ou F:

- a) $(A > C)$ and $(C \leq D)$
- b) $((A + B) > 10)$ or $((A + B) == (C + D))$
- c) $(A \geq C)$ and $(D \leq B)$
- d) not $A == 5$
- e) not $(C + D) == (A * B)$