Aula 5: Fluxogramas e Desvio Condicional

Prof. Sérgio Montazzolli Silva
smsilva@uel.br

Sumário

- Programação estruturada
- Representação através de fluxogramas
- Desvio condicional
  - IF-ELSE

Paradigmas de programação

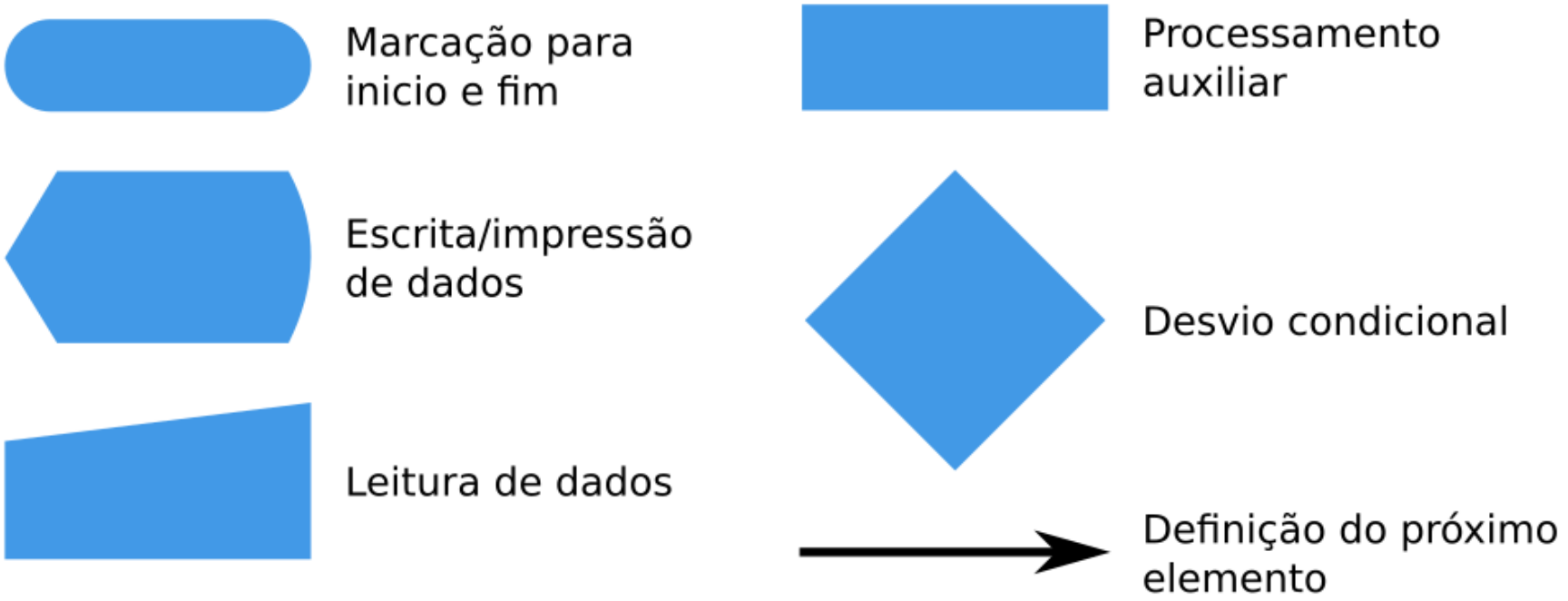
- Não-estruturado
- **Estruturado**
- Orientado a Objectos

- A grosso modo, podemos representar da seguinte forma:

Programação estruturada

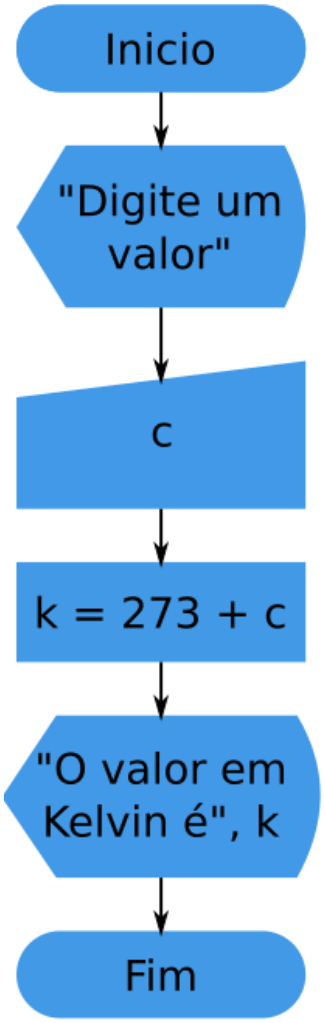
- Depende de um controle de fluxo estruturado
- Basicamente utiliza elementos de:
  - Decisão
  - Repetição
  - Blocos e sub-rotinas
- Pode ser representada graficamente por um **fluxograma**
- Fluxogramas facilitam o entendimento de programas ou rotinas mais complexas

Elementos de um fluxograma

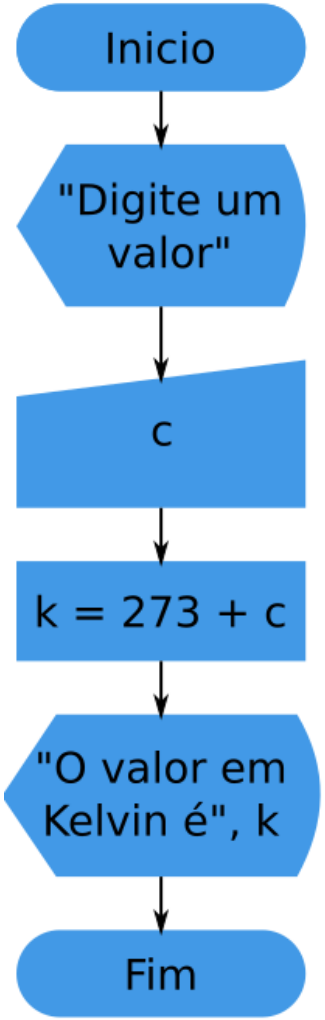


Rotina sequencial

- Supondo a seguinte rotina:
  - Ler entrada em graus célsius
  - Converter entrada para graus Kelvin
  - Mostrar em graus Kelvin
- Como poderia ser descrito através de um fluxograma?



Fluxograma para pseudocódigo



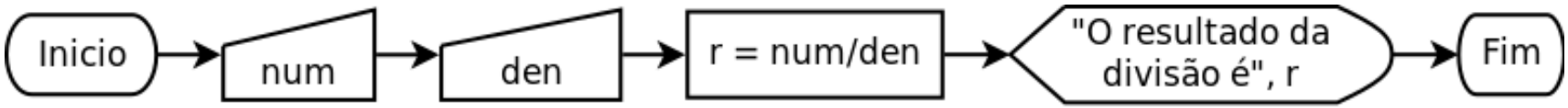
- Como representar em pseudocódigo?

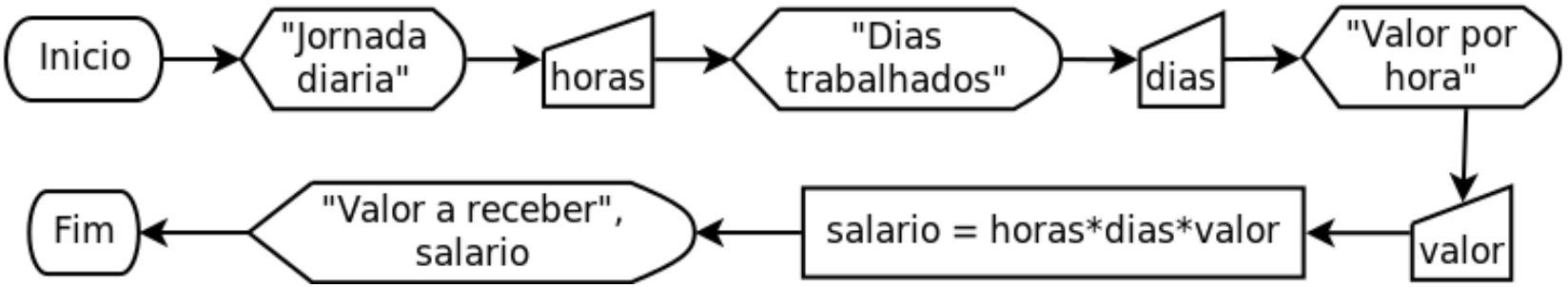
```
Algoritmo principal

Inicio
    imprimir "Digite um valor"
    ler c
    k ← 273 + c
    imprimir "O valor em Kelvin é"
    imprimir k
Fim
```

- E em C?

Traduza para C

- Traduza cada fluxograma para um programa na linguagem C





Desvio condicional

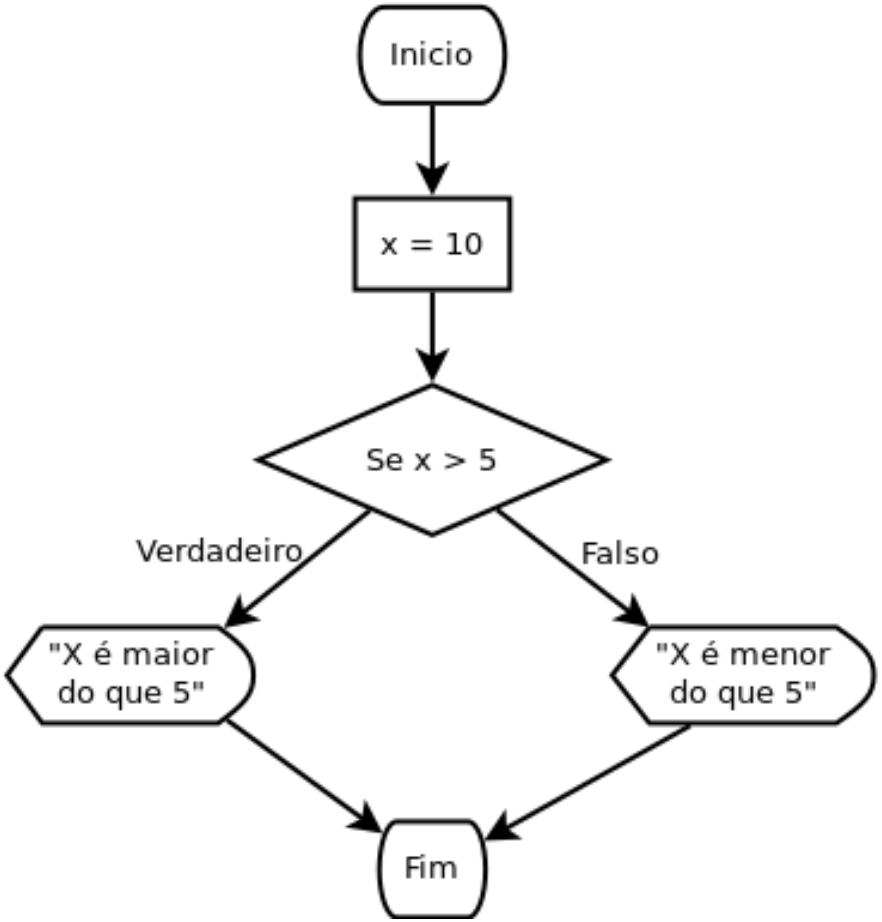
Desvio em fluxogramas

- Dado pelo elemento losango
- Possui uma condição e dois caminhos possíveis:

  - Um caminho para verdadeiro, e outro para falso

Fluxograma com desvio

- Exemplo



Desvio condicional (pseudocódigo)

- Usa-se as palavras **se**, **então**, **senão**, **fim-se**;
- Entre as palavras **se** e **então**, deve vir a **condição** a ser testada;
- A condição é dada por uma expressão relacional;
- Os comandos a serem executados caso a condição seja verdadeira devem aparecer entre **então** e **senão**;
- Os comandos a serem executados caso a condição seja falsa devem aparecer entre **senão** e **fim-se**

Desvio condicional (pseudocódigo)

- Exemplo em pseudocódigo do fluxograma anterior:

```
Algoritmo principal

Inicio
   x ←10
   se x > 5 então
      imprimir "X é maior do que 5"
   senão
      imprimir "X é menor do que 5"
   fim-se
Fim
```

Desvio condicional em C

- Em C, usa-se as palavras **if** e **else** (em português: "se" e "senão")
- Cada uma destas palavras devem ser seguida de chaves "{" e "}"
  - De maneira semelhante as chaves da função main
- Entre **if** e abre-colchete "{", deve haver uma expressão relacional dentro de parênteses ()
- Da seguinte forma:

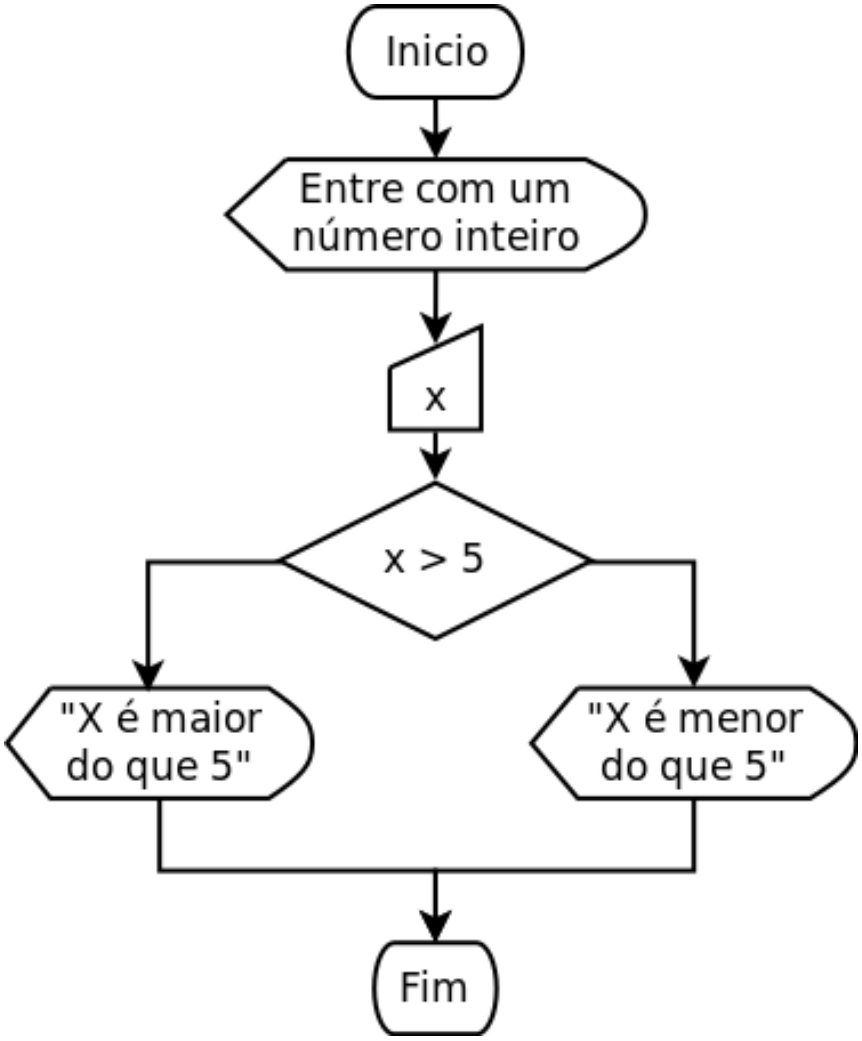
```
if (expressão) {
    ... comandos se expressão for verdadeira ....
} else {
    ... comandos se expressão for false ...
}
```

Desvio condicional em C

- Exemplo do fluxograma anterior:

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int x = 10;

    if (x > 5) {
        printf("X e maior do que 5");
    } else {
        printf("X e menor do que 5");
    }
}
```

Exercício em sala



Expressão Relacional

- Expressa a relação entre dois valores através de um operador
- Valores neste caso podem ser constantes e/ou variáveis
- Resulta sempre em um valor binário: verdadeiro ou falso
- Operadores relacionais em C:

Operador	Matemática	C
Igual a	$=$	==
Diferente de	$\neq$	!=
Maior que	$>$	>
Menor que	$<$	<
Maior ou igual a	$\geq$	>=
Menor ou igual a	$\leq$	<=

Expressão Relacional

- Exemplos:

Expressão	Matemática	C
a maior do que b	$a > b$	a >= b
10 menor do que 7	$10 < 7$	10 < 7
c igual a 36.1	$c = 36.1$	c == 36.1
d é diferente de 40	$d \neq 40$	d != 40

Exemplos expressões em C

- Supondo as variáveis:
  - int a = 10
  - int b = 30
  - int c = 10
  - float d = 32.2
  - float e = 7.8

Expressão relacional	Resultado
a == b	Falso
a != b	Verdadeiro
a > b	Falso
a < b	Verdadeiro
a < c	Falso
a <= c	Verdadeiro
a != c	Falso
a > d	Falso
e == b	Falso
e < d	Verdadeiro
d != e	Verdadeiro

Exercícios em sala

- Crie um programa que leia dois números reais **a** e **b**. Se **a** for maior ou igual a **b**, verifique também se **a** é igual a **b. Exemplos de saídas desejadas:**

Digite um numero: 10 Digite outro numero: 10 10 e maior ou igual a 10 10 e igual 10	Digite um numero: 63 Digite outro numero: -20 63 e maior ou igual a -20	Digite um numero: -78 Digite outro numero: -30 -78 e menor que -30

- Leia 1 número inteiro **a** e verifique se ele está no intervalo fechado [-20,100].

Operadores Lógicos

- Operadores lógicos relacionam valores binários
- Como expressões relacionais culminam em valores binários, então operadores lógicos podem relacionar uma ou mais expressões relacionais
- Existem 3 tipos principais de operadores lógicos:
  - E
  - Ou
  - Negação

Operadores Lógicos

- E / and / Λ:
  - retorna verdadeiro apenas se duas expressões também retornem verdadeiro, senão retorna falso:
  - (expressão1) Λ (expressão2)

- Ou / or / V:
  - retorna falso apenas quando ambas as expressões forem falsas, senão retorna verdadeiro:
  - (expressão1) V (expressão2)

- Negação / neg / ¬:
  - Retorna verdadeiro se expressão for falsa, ou falso se expressão for verdadeira
    - Apenas inverte o resultado da expressão
  - ¬ (expressão1)

Operadores Lógicos

- E e OU

Expressão1	Expressão2	Expressão1 ∧ Expressão2 (E)	Expressão1 ∨ Expressão2 (OU)
Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso	Verdadeiro
Falso	Verdadeiro	Falso	Verdadeiro
Falso	Falso	Falso	Falso

- Negação

Expressão	¬Expressão
Verdadeiro	Falso
Falso	Verdadeiro

Em linguagem C

- Operadores lógicos:
  - AND: representado por &&
  - OR: representado por ||
  - NEG: representado por !

- Combinando expressões
  - Combinando expressões com operador AND
    - if (expressao1 && expressão2 && … expressaoN) { // código }
  - Combinando expressões com operador OR
    - if (expressao1 || expressão2 || … expressaoN) { // código }

- Usando operador de negação:
  - if (!(expressao)) { // código }

Exercícios

- Novamente, escreva um programa que leia 3 números inteiros **a**, **b** e **c**, e verifique se **b** está no intervalo fechado entre **a** e **c**. Use apenas *um comando IF*.

- Para receber um certo beneficio, uma pessoa precisa ter ao menos 3 filhos, **ou** ter mais de 65 anos de idade. Crie um programa que leia a idade e o número de filhos de uma pessoa, e verifique se essa pessoa tem direito ou não ao beneficio.