



Aula: Estruturas Condicionais

Introdução a Programação

Túlio Toffolo & Puca Huachi
<http://www.toffolo.com.br>

Departamento de Computação
Universidade Federal de Ouro Preto

Aula de Hoje

- 1 Fluxogramas
- 2 Operadores relacionais
- 3 Operadores lógicos
- 4 Comandos de decisão
- 5 Exercícios

Aula de Hoje

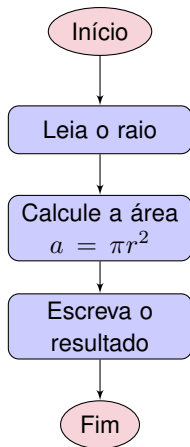
- 1 Fluxogramas
- 2 Operadores relacionais
- 3 Operadores lógicos
- 4 Comandos de decisão
- 5 Exercícios

Exemplo 1:

Faça um programa em C/C++, para calcular a área de um círculo. A área de um círculo é dada pela seguinte fórmula $a = \pi r^2$. O valor do raio r será digitado pelo usuário.

Solução do Exemplo 1:

```
1  /* Programa que calcula a área de um círculo
2  */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      // declaração da constante Pi
9      const double PI = 3.141592;
10     double raio;
11
12     printf("Digite o raio do círculo: ");
13     scanf("%lf", &raio);
14
15     if ( raio >= 0){
16         // calculando e imprimindo a área
17         double area = PI * raio * raio;
18         printf("\nÁrea do círculo: %lf\n", area);
19     }
20     return 0;
21 }
```



Dúvida?

- Não existe área negativa.
- Portanto, o programa não pode calcular a área se o valor do raio for negativo.
- Como saber se o valor do raio digitado é positivo?
 - *Responderemos nesta da aula...*

Aula de Hoje

- 1 Fluxogramas
- 2 Operadores relacionais**
- 3 Operadores lógicos
- 4 Comandos de decisão
- 5 Exercícios

Tomada de decisões

- Permite a um programa realizar uma ação alternativa, a partir de um resultado **verdadeiro** ou **falso** produzido por uma condição.
- As **condições** são formadas utilizando-se os operadores de igualdade e os operadores relacionais.
- Ambos operadores de igualdade têm o mesmo nível de precedência, o qual é inferior ao dos operadores relacionais, e associam-se da esquerda para a direita.

Operadores de igualdade e relacionais

Operador algébrico de igualdade ou relacional padrão	Operador de igualdade ou relacional em C++	Exemplo de condição em C++	Significado da condição em C++
<i>Operadores relacionais</i>			
>	>	<code>x > y</code>	x é maior que y
<	<	<code>x < y</code>	x é menor que y
≥	>=	<code>x >= y</code>	x é maior que ou igual a y
≤	<=	<code>x <= y</code>	x é menor que ou igual a y
<i>Operadores de igualdade</i>			
=	==	<code>x == y</code>	x é igual a y
≠	!=	<code>x != y</code>	x não é igual a y

Erro Comum em Programação

- Confundir o operador de igualdade ==
- Com o operador de atribuição =

`total` = `x + 10;`

avalia a expressão (direita)
e atribui o resultado
à variável (esquerda)

`total` == `x + 10;`

verifica se a expressão da
direita é IGUAL a expressão
da esquerda (vice-versa).

Aula de Hoje

- 1 Fluxogramas
- 2 Operadores relacionais
- 3 Operadores lógicos**
- 4 Comandos de decisão
- 5 Exercícios

Expressões booleanas

- No século 18, *George Boole*, matemático e filósofo britânico, desenvolveu um sistema algébrico lógico, que passou a ser conhecido como **Álgebra de Boole**.
 - Base para a lógica dos computadores digitais modernos.
 - Expressões lógicas (**expressões booleanas**) possuem o valor `true` ou `false`.
- Em C os inteiros também são usados como valores booleanos: qualquer valor não nulo (**1**) representa `true` e **0** representa `false`.

Expressões booleanas compostas

- na matemática, podemos restringir uma temperatura a um intervalo fechado, $0 \leq temp \leq 100$
- em C não podemos representar essa expressão por:
 $0 \leq temp \leq 100$
- embora ela seja uma expressão C válida!
- por exemplo, suponha que $temp = 150$; (não está no intervalo definido, logo esperamos que o resultado da expressão seja falso).

Expressões booleanas compostas

- os operadores relacionais são associativos a esquerda. A expressão será processada da seguinte forma:

$$\underbrace{0 \leq 150}_{(a)} \leq 100$$

- a expressão (a) resulta em `true`, que é representado em C pelo inteiro 1 (ou outro inteiro diferente de 0).
- Assim, na segunda etapa da avaliação, a expressão resulta em:

$$1 \leq 100$$

- que também é verdadeira e resulta em `true`.
- Entretanto, a expressão original deveria resultar falso, como na matemática.

Expressões booleanas compostas

- Para solucionar esse problema, reescrevemos a desigualdade como:

`(temp >= 0) && (temp <= 100)`

- onde `&&` é um operador lógico.
- Utilizamos os operadores lógicos para combinar expressões booleanas formando, assim, expressões booleanas compostas.

Operadores lógicos

Operador	Expressão	Nome	Descrição
!	!p	NÃO (negação)	!p é falso, se p é verd.; !p é verd., se p é falso.
&&	p && q	E (conjunção)	p && q é verdadeiro, se ambos, p e q são verd.; e falso, caso contrário.
	p q	OU (disjunção)	p q é verdadeiro, se p , q ou ambos é verd.; e falso, caso contrário.

Operadores lógicos – Tabela verdade

p	$!p$
true	false
false	true

p	q	$p \ \&\& \ q$	$p \ \ q$
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Operadores lógicos – Precedência

Operator	Priority	Associativity
!, ~	highest	Right
/, *, %		Left
+, -		Left
<, >, <=, >=		Left
==, !=		Left
&		Left
^		Left
		Left
&&		Left
		Left
=, +=, *=, ...	lowest	Right

Aula de Hoje

- 1 Fluxogramas
- 2 Operadores relacionais
- 3 Operadores lógicos
- 4 Comandos de decisão**
- 5 Exercícios

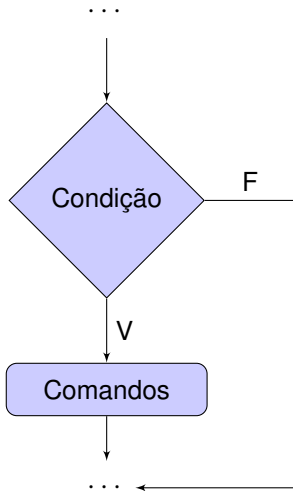
Tomada de decisão

- Permite a um programa realizar uma ação alternativa, a partir de um resultado **verdadeiro** ou **falso** produzido por uma condição.
- As condições são formadas utilizando-se os operadores de igualdade e os operadores relacionais.
- Ambos operadores de igualdade têm o mesmo nível de precedência, o qual é inferior ao dos operadores relacionais, e associam-se da esquerda para a direita.

Tomada de decisão

- Comando `if`
 - consiste de uma palavra-chave `if` seguida de uma expressão de teste entre parênteses. A instrução será executada apenas se a expressão de teste for **verdadeira**.
 - O corpo de um comando `if` pode conter **uma única instrução terminada por ponto-e-vírgula** ou **várias instruções entre chaves**.

Tomada de decisão



Sintaxe do comando `if`

```
1 if ( <expressão_de_teste> )  
2     instrução_única;
```

OU

```
1 if ( <expressão_de_teste> )  
2 {  
3     instrução1;  
4     instrução2;  
5     instrução3;  
6     ...  
7 }
```

Exemplo usando o comando `if`

Exemplo 1 (resolvido anteriormente):

Faça um programa em C/C++, para calcular a área de um círculo. A área de um círculo é dada pela seguinte fórmula $a = \pi r^2$. O valor do raio r será digitado pelo usuário.

- Altere o programa anterior para calcular a área somente se o valor do raio for positivo.


```
1  /* Programa que calcula a área de um círculo
2  */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      // declaração da constante Pi
9      const double PI = 3.141592;
10     double raio;
11
12     printf("Digite o raio do círculo: ");
13     scanf("%lf", &raio);
14
15     // calculando e imprimindo a área do círculo
16     if (raio >= 0) {
17         double area = PI * raio * raio;
18         printf("\nÁrea do círculo: %lf\n", area);
19     }
20
21     return 0;
22 }
```

Aula de Hoje

- 1 Fluxogramas
- 2 Operadores relacionais
- 3 Operadores lógicos
- 4 Comandos de decisão
- 5 Exercícios**

Exercícios

Exercício 1: Maior número

Faça um programa que leia dois números inteiros e verifique qual deles é maior.

- Imprima uma mensagem informando qual deles é o maior.
- Exemplo de execução caso o usuário digite 10 e 20:

```
1 Digite dois números: 10 20
2 20 é o maior número
```