

VIANNA JUNIOR
INSTITUTO



Algoritmos

Estruturas Condicionais

Professor: Camillo Falcão

- Operadores relacionais e lógicos
- Alternativa simples
- Alternativa dupla
- Alternativa múltipla escolha

Operadores relacionais e lógicos

- Os operadores relacionais em C# são:
 - < menor que
 - > maior que
 - <= menor ou igual que
 - >= maior ou igual que
 - == igual a
 - != diferente de
- Estes operadores comparam dois valores.
- O resultado produzido por um operador relacional é *true* (verdadeiro) ou *false* (falso).

Operadores relacionais e lógicos

- Os operadores lógicos combinam expressões lógicas (ou booleanas).
- Operadores:
 - && operador binário E (AND)
 - || operador binário OU (OR)
 - ! operador unário de NEGAÇÃO (NOT)
- Expressões compostas por && ou || são avaliadas da esquerda para a direita.
- A avaliação termina assim que o resultado for conhecido.

Operadores relacionais e lógicos

- Esses operadores são normalmente utilizados para tomada de decisões.
- Podem ser utilizados para atribuir valores a variáveis.
- Exemplo:

```
int a = 13;
```

```
int b = a + 5;
```

```
bool c, d, e;
```

```
c = (b > 20);
```

falso

```
d = (a < 15) || (b > a);
```

verdadeiro

```
e = (a < 15) && (b > a);
```

verdadeiro

Operadores relacionais e lógicos

- Tabela Verdade para operadores lógicos.

a	b	a && b	a b
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

a	! a
V	F
F	V

Operadores relacionais e lógicos

- Operadores lógicos E e OU.
- O resultado de uma expressão com o operador **E** só é verdadeiro quando ambos operandos são verdadeiros.

$(15 > 10) \ \&\& \ (15 < 30)$

- O resultado de uma expressão com o operador **OU** é verdadeiro sempre que pelo menos um operando é verdadeiro.

$(10 < 20) \ || \ (-1 > 5)$

- Utilizada quando a execução de uma ação depender de uma inspeção ou teste de uma condição (expressão lógica).
- Pode ser:
 - Alternativa Simples
 - Alternativa Dupla
 - Alternativa Múltipla Escolha

Alternativa simples: Sintaxe

```
if ( condição )  
{  
    comandos;  
}
```

Alternativa simples:

Exemplo 1

```
x = 10;  
max = 5;  
  
if (x > max)  
{  
    max = x;  
}
```

Alternativa simples:

Exemplo 2

```
x = 10;  
max = 20;  
  
if (x > max)  
{  
    max = x;  
}
```

Alternativa simples:

Exemplo 2

```
x = 10;  
max = 20;  
  
if (x > max)  
    max = x;
```

- **Observação:** Se houver só um comando no bloco não é necessário usar chaves.

Alternativa simples:

Exemplo 3

- Imprimir o MAIOR entre A e B

```
static void Main(string[] args)
{
    int a, b, maior;
    a = 9;
    b = 2;
    maior = b;
    if (a > maior)
        maior = a;

    Console.WriteLine("\nMAIOR = {0}", maior);
}
```

Alternativa dupla: Sintaxe

```
if (condicao)
{
    comandos1;
}
else
{
    comandos2;
}
```

Alternativa dupla: Exemplo

```
static void Main(string[] args)
{
    int a, b, maior;
    a = 9;
    b = 2;
    maior = b;
    if (a > maior)
        maior = a;

    Console.WriteLine("\nMAIOR = {0}", maior);
}
```


Alternativa dupla: Exemplo

```
static void Main(string[] args)
{
    int a, b, maior;
    a = 9;
    b = 2;
    if (a > b)
    {
        maior = a;
    }
    else
    {
        maior = b;
    }
    Console.WriteLine("\nMAIOR = {0}", maior);
}
```

Alternativa múltipla escolha: Sintaxe

```
switch (expressão)
{
    case V1 : C1;
                break;
    case V2 : C2;
                break;
    . . .
    case Vn : Cn;
                break;
    default : Cn+1;
                break;
}
```

Alternativa múltipla escolha

```
static void Main(string[] args)
{
    Console.WriteLine("Digite o numero do trimestre:");
    int epoca = Convert.ToInt32(Console.ReadLine());
    switch (epoca)
    {
        case 1:
            Console.WriteLine("verao");
            break;
        case 2:
            Console.WriteLine("outono");
            break;
        case 3:
            Console.WriteLine("inverno");
            break;
        case 4:
            Console.WriteLine("primavera");
            break;
        default:
            Console.WriteLine("Trimestre invalido");
            break;
    }
}
```

Problema exemplo

- Construir um algoritmo para ler os coeficientes A , B e C de uma equação do segundo grau e se:
 - **Delta for negativo:** imprimir a mensagem “Não há solução real”.
 - **Delta for maior ou igual a zero:** calcular e imprimir as raízes da equação.

- Enunciado está entendido?
- Para resolver o problema é necessário saber a equação de 2º grau. Esta equação tem a seguinte forma:

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

- onde

$$\Delta = b^2 - 4ac$$

- Quais variáveis serão necessárias?
- A, B, C e DELTA serão as variáveis reais que comporão a fórmula da equação acima.
- Serão necessárias mais duas variáveis reais X_1 e X_2 que serão as raízes da equação.

- Quais métodos serão utilizados?
- Por se tratar de uma condição, utilizaremos a estrutura de controle de alternativa.
- **if-else**

```
static void Main(string[] args)
{
    double a, b, c, delta, x1, x2;
    Console.WriteLine("Digite os coeficientes da equação (A B C): ");
    a = Convert.ToDouble(Console.ReadLine());
    b = Convert.ToDouble(Console.ReadLine());
    c = Convert.ToDouble(Console.ReadLine());
    delta = (b*b - 4*a*c);

    if (delta >= 0)
    {
        x1 = (-b + Math.Sqrt(delta)) / (2*a);
        x2 = (-b - Math.Sqrt(delta)) / (2*a);
        Console.WriteLine("Raízes da equação: {0}, {1}", x1, x2);
    }
    else
    {
        Console.WriteLine("Não existem raízes reais");
    }
}
```


- Testar o algoritmo.
- Por exemplo, faça o teste de mesa para a seguinte entrada
 - $A = 1$
 - $B = 5$
 - $C = 4$
- Saída esperada: -1 e -4.

TESTE DE MESA

Digite os coeficientes (A B C):

TESTE DE MESA

Digite os coeficientes (A B C):

1


```
1 static void Main(string[] args)
2 {
3     double a, b, c, delta, x1, x2;
4     Console.WriteLine("Digite os coeficientes (A B C):");
5     a = Convert.ToDouble(Console.ReadLine());
6     b = Convert.ToDouble(Console.ReadLine());
7     c = Convert.ToDouble(Console.ReadLine());
8     delta = (b*b - 4*a*c);
9
10    if (true)
11    {
12        x1 = (-b + Math.Sqrt(delta)) / (2*a);
13        x2 = (-b - Math.Sqrt(delta)) / (2*a);
14        Console.WriteLine("Raízes: {0}, {1}", x1, x2);
15    }
16    else
17    {
18        Console.WriteLine("Não existem raízes reais");
19    }
20 }
```

TESTE DE MESA

linh a	a	b	c	delt a	x 1	x2
3	?	?	?	?	?	?
4	?	?	?	?	?	?
5	1	?	?	?	?	?
6	1	5	?	?	?	?
7	1	5	4	?	?	?
8	1	5	4	9	?	?
10	1	5	4	9	?	?

```
Digite os coeficientes (A B C):
1
5
4
```

```
1 static void Main(string[] args)
2 {
3     double a, b, c, delta, x1, x2;
4     Console.WriteLine("Digite os coeficientes (A B C):");
5     a = Convert.ToDouble(Console.ReadLine());
6     b = Convert.ToDouble(Console.ReadLine());
7     c = Convert.ToDouble(Console.ReadLine());
8     delta = (b*b - 4*a*c);
9
10    if (delta >= 0)
11    {
12        x1 = (-b + Math.Sqrt(delta)) / (2*a);
13        x2 = (-b - Math.Sqrt(delta)) / (2*a);
14        Console.WriteLine("Raízes: {0}, {1}", x1, x2);
15    }
16    else
17    {
18        Console.WriteLine("Não existem raízes reais");
19    }
20 }
```

TESTE DE MESA

linh a	a	b	c	delt a	x 1	x2
3	?	?	?	?	?	?
4	?	?	?	?	?	?
5	1	?	?	?	?	?
6	1	5	?	?	?	?
7	1	5	4	?	?	?
8	1	5	4	9	?	?
10	1	5	4	9	?	?
12	1	5	4	9	- 1	?

```
Digite os coeficientes (A B C):
1
5
4
```

```
1 static void Main(string[] args)
2 {
3     double a, b, c, delta, x1, x2;
4     Console.WriteLine("Digite os coeficientes (A B C):");
5     a = Convert.ToDouble(Console.ReadLine());
6     b = Convert.ToDouble(Console.ReadLine());
7     c = Convert.ToDouble(Console.ReadLine());
8     delta = (b*b - 4*a*c);
9
10    if (delta >= 0)
11    {
12        x1 = (-b + Math.Sqrt(delta)) / (2*a);
13        x2 = (-b - Math.Sqrt(delta)) / (2*a);
14        Console.WriteLine("Raízes: {0}, {1}", x1, x2);
15    }
16    else
17    {
18        Console.WriteLine("Não existem raízes reais");
19    }
20 }
```

TESTE DE MESA

linh a	a	b	c	delt a	x 1	x2
3	?	?	?	?	?	?
4	?	?	?	?	?	?
5	1	?	?	?	?	?
6	1	5	?	?	?	?
7	1	5	4	?	?	?
8	1	5	4	9	?	?
10	1	5	4	9	?	?
12	1	5	4	9	- 1	?
13	1	5	4	9	- 1	-4

```
Digite os coeficientes (A B C):
1
5
4
```

```
1 static void Main(string[] args)
2 {
3     double a, b, c, delta, x1, x2;
4     Console.WriteLine("Digite os coeficientes (A B C):");
5     a = Convert.ToDouble(Console.ReadLine());
6     b = Convert.ToDouble(Console.ReadLine());
7     c = Convert.ToDouble(Console.ReadLine());
8     delta = (b*b - 4*a*c);
9
10    if (delta >= 0)
11    {
12        x1 = (-b + Math.Sqrt(delta)) / (2*a);
13        x2 = (-b - Math.Sqrt(delta)) / (2*a);
14        Console.WriteLine("Raízes: {0}, {1}", x1, x2);
15    }
16    else
17    {
18        Console.WriteLine("Não existem raízes reais");
19    }
20 }
```

TESTE DE MESA

linh a	a	b	c	delt a	x 1	x2
3	?	?	?	?	?	?
4	?	?	?	?	?	?
5	1	?	?	?	?	?
6	1	5	?	?	?	?
7	1	5	4	?	?	?
8	1	5	4	9	?	?
10	1	5	4	9	?	?
12	1	5	4	9	-1	?
13	1	5	4	9	-1	-4
14	1	5	4	9	-1	-4

```
Digite os coeficientes (A B C):
1
5
4
Raízes: -1, -4
```

Resolução – Procedimento

- Exemplo de implementação do problema como um procedimento.

```
static void equacao2grau(double a, double b, double c)
{
    double delta, x1, x2;
    delta = (b*b - 4*a*c);
    if (delta >= 0)
    {
        x1 = (-b + Math.Sqrt(delta)) / (2*a);
        x2 = (-b - Math.Sqrt(delta)) / (2*a);
        Console.WriteLine("Raízes da equação: {0}, {1}", x1, x2);
    }
    else
    {
        Console.WriteLine("Não existem raízes reais");
    }
}
```

Alternativas aninhadas

- As vezes pode ser preciso usar estruturas de alternativa aninhadas

```
if (condicao)
{
    comandos1;
    if (outraCondicao)
    {
        comandos2;
    }
}
else
{
    comandos3;
}
```

Alternativas aninhadas

- Ou ainda usar o comando **else if**:

```
if (condicao)
{
    comandos1;
}
else if (condicao2)
{
    comandos2;
}
else if (condicao3)
{
    comandos3;
}
else
{
    comandos4;
}
```

Exemplo

- Determinar se um número inteiro é zero, negativo ou positivo.

```
static void Main(string[] args)
{
    int num;
    Console.WriteLine("Digite um numero inteiro: ");
    num = Convert.ToInt32(Console.ReadLine());
    if(num == 0)
        Console.WriteLine("{0} zero", num);
    else
    {
        if(num > 0)
            Console.WriteLine("{0} positivo", num);
        else
            Console.WriteLine("{0} negativo", num);
    }
}
```


Exemplo

- Determinar se um número inteiro é zero, negativo ou positivo.

```
static void Main(string[] args)
{
    int num;
    Console.WriteLine("Digite um numero inteiro: ");
    num = Convert.ToInt32(Console.ReadLine());
    if(num == 0)
        Console.WriteLine("{0} zero", num);
    else if(num > 0)
        Console.WriteLine("{0} positivo", num);
    else
        Console.WriteLine("{0} negativo", num);
}
```

Exercício 1

- Qual é a saída do programa:

```
static void Main(string[] args)
{
    int x = 10, y = 3;
    if (x < 10)
        Console.WriteLine("Saida 1");
    else
        if (y < 4)
            if (x > 10)
                Console.WriteLine("Saida 2");
            else Console.WriteLine("Saida 3");
}
```

Exercício 2

- Qual é a saída do programa:

```
static void Main(string[] args)
{
    int a = 2, b = 3, c = 4;
    if ((a<3) && ((b !=3) || (c == 4)))
        Console.WriteLine("saida 1");
    if ((c ==3) || (b ==4))
        Console.WriteLine("saida 2");
    if ((a < 3) && (b > 4))
        Console.WriteLine("saida 3");
    else if (a == 2)
    {
        if (b==5)
            Console.WriteLine("saida 4");
        else Console.WriteLine("saida 5");
    }
}
```

Exercício 2

- Qual é a saída do programa:

```
static void Main(string[] args)
{
    int a = 2, b = 3, c = 4;
    if ((a<3) && ((b !=3) || (c == 4)))
        Console.WriteLine("saida 1");
    if ((c ==3) || (b ==4))
        Console.WriteLine("saida 2");
    if ((a < 3) && (b > 4))
        Console.WriteLine("saida 3");
    else
        if (a == 2)
            if (b==5)
                Console.WriteLine("saida 4");
            else
                Console.WriteLine("saida 5");
}
```

Lembre-se, a indentação do código facilita o seu desenvolvimento e a leitura do código.

- 1) Ler dois números inteiros e informar se o primeiro valor lido é maior, menor ou igual ao segundo.
- 2) Ler um número inteiro e informar se ele é divisível por 2.
- 3) Altere o algoritmo anterior para que seja informado se o número é divisível por 2 e por 3 simultaneamente.

- 4) Altere o algoritmo anterior para que seja informado se o número é divisível por 2 e por 3, mas que não seja divisível por 5.

- 5) Desenvolver um algoritmo para ler o número do dia da semana e imprimir o seu respectivo nome por extenso. Considerar o número 1 como domingo, 2 para segunda etc. Caso o dia da semana não exista (menor do que 1 ou maior do que 7), exibir a mensagem "Dia da semana inválido".

Exercícios

- 6) Fazer um algoritmo para ler dois números e um dos símbolos das operações: +, -, * e /. Imprimir o resultado da operação efetuada sobre os números lidos.
- 7) Desenvolver um algoritmo para ler o valor inteiro da idade de uma pessoa e imprimir uma das mensagens: se $\text{idade} < 13$: Criança, se $13 \leq \text{idade} < 20$: Adolescente, se $20 \leq \text{idade} < 60$: Adulto e se $\text{idade} \geq 60$: Idoso.

- 8) Elaborar um algoritmo para ler o código de um produto e informar a sua origem:
- a) Código do produto entre 1 e 20: Europa
 - b) Código do produto entre 21 e 40: Ásia
 - c) Código do produto entre 41 e 60: América
 - d) Código do produto entre 61 e 80: África
 - e) Código do produto maior que 80: Paraguai

9) Para auxiliar os vendedores de uma loja na orientação aos clientes sobre as diversas formas de pagamento, desenvolver um algoritmo para:

a) Imprimir o seguinte menu:

```
Forma de pagamento:
- À vista.
- Cheque para trinta dias.
- Em duas vezes.
- Em três vezes.
- Em quatro vezes.
- A partir de cinco vezes.
Entre com sua opção:
```

b) Ler o código da opção de pagamento.

c) Imprimir uma das mensagens de acordo com a opção lida:

Opção = 1: Desconto de 20%

Opção = 2, 3 ou 4: Mesmo preço a vista

Opção = 5: Juros de 3% ao mês

Opção = 6: Juros de 5% ao mês

Opção <1 ou opção >6: Opção inválida

Estruturas de Controle Alternativa

Aula de Exercícios

Professor: Camillo Falcão

Comando Condicional - `if`

- O comando `if` é uma estrutura de decisão que decide se uma sequência de comandos será ou não executada. Sua sintaxe é:

```
if (expressão)
{
    sequencia de comandos;
}
```

- Ou

```
if (expressão)
    unico comando;
```

Comando Condicional - `if`

```
if (expressão)  
{  
    sequencia de comandos;  
}
```

- A expressão sempre será avaliada, e o resultado lógico (verdadeiro ou falso).

if - Exemplos

- Programa para determinar o maior de dois números.

```
static void Main(string[] args)
{
    int a=9, b=2, maior;
    if (a > b)
    {
        maior = a;
    }
    else
    {
        maior = b;
    }
    Console.WriteLine("\nMAIOR = {0}", maior);
}
```

if - Exemplos

- Ler duas variáveis (x e y) e imprimi-las na ordem crescente.

```
static void Main(string[] args)
{
    double x, y, aux;
    Console.WriteLine("Digite os dois numeros");
    x = Convert.ToDouble(Console.ReadLine());
    y = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Valores originais (x, y) : {0}, {1} ", x, y);
    if (y < x)
    {
        aux = x;
        x = y;
        y = aux;
    }
    Console.WriteLine("Valores de x e y ordenados: {0}, {1}", x, y);
}
```

Condicional: if-else

- O comando `if` pode decidir entre duas sequências de comandos qual vai ser a executada e tem a seguinte sintaxe:

```
if (expressão)
{
    // caso a expressão retorne verdadeiro
    sequencia de comandos;
}
else
{
    // caso a expressão retorne falso
    sequencia de comandos;
}
```


if-else

- Exemplo: Verificar se um número é par.

```
static void Main(string[] args)
{
    int x;
    Console.WriteLine("Digite o numero: ");
    x = Convert.ToInt32(Console.ReadLine());
    if (x % 2 == 0)
        Console.WriteLine("{0} e' par", x);
    else
        Console.WriteLine("{0} e' impar", x);
}
```

- Obs.: No exemplo acima não são utilizadas chaves no `if` e no `else` pois há apenas um comando a ser executado. Em casos como esse pode-se ou não utilizar as chaves.

if else , if , ...

```
if (condição1)
{
    comandos if1;
    if (condição2)
    {
        comandos if2;
    }
    else
    {
        comandos else2;
    }
}
else
{
    comandos else1;
    if (condição3)
    {
        comandos if3;
    }
}
```

Comando switch

- Utilizado quando uma determinada variável pode ser igual a diferentes valores que se deseja avaliar
- Sintaxe:

```
switch (variavel)
{
    case constante1: comandos;
    break;
    case constante2: comandos;
    break;
    default: comandos;
}
```

Comando switch - Exemplo

```
static void Main(string[] args)
{
    int epoca;
    Console.WriteLine("Digite o trimestre do ano em que estamos: ");
    epoca = Convert.ToInt32(Console.ReadLine());
    switch (epoca)
    {
        case 1: Console.WriteLine("verao");
        break;
        case 2: Console.WriteLine("outono");
        break;
        case 3: Console.WriteLine("inverno");
        break;
        case 4: Console.WriteLine("primavera");
        break;
        default: Console.WriteLine("período inválido");
    }
}
```

1. Faça um programa que leia dois números inteiros e faça a divisão do primeiro pelo segundo (se o segundo for diferente de zero).
2. Faça um programa para ler dois números reais e verificar se ambos são maiores que zero. Caso positivo, informar “Valores são válidos”. Caso contrário, informar “Valores inválidos”.

3. Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um programa que calcule seu peso ideal, utilizando as seguintes fórmulas:
para homens: $(72.7 * h) - 58$
para mulheres: $(62.1 * h) - 44.7$
4. Faça um programa que leia 3 comprimentos (x , y e z) e responda se eles formam um triângulo, ou seja, se
 $x < y + z$ e $y < x + z$ e $z < x + y$.

5. Faça um programa para ler um caractere e imprimir as seguintes mensagens, segundo o caso:
- “Sinal de menor”
 - “Sinal de maior”
 - “Sinal de igual”
 - “Outro caractere”

6. Elabore um programa que, dada a idade de um nadador, classifique-o em uma das seguintes categorias:

- infantil A = 5 – 7 anos
- infantil B = 8 – 10 anos
- juvenil A = 11 – 13 anos
- juvenil B = 14 – 17 anos
- adulto = 18 – 30 anos
- sênior = maiores de 30 anos