UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE PROCESSAMENTO DE ENERGIA ELÉTRICA

Revisão lógica programação usando C# 2

Curso de Engenharia de Controle e Automação

DPEE1090 - Programação orientada a objetos para automação

Prof. Renan Duarte

1º semestre de 2024

Sumário

Revisão lógica programação usando C#

- Estudo dos operadores aritméticos, de atribuição, comparativos e lógicos
- Conversão implícita e casting de tipos de dados
- Implementação de estruturas condicionais: if-else e operador ternário
- Utilização de laços de repetição: while, do-while, for e foreach

Relembrando

Convenção de nomeclatura

Camel Case - lastName

- Parâmetros de métodos, variáveis dentro de métodos

Pascal Case - LastName

- *Namespaces*, classe, *properties* e métodos

Padrão _lastName

- Atributos "internos" da classe

Relembrando

Restrições de nomes de variáveis

- Não podem começar com dígito: use uma letra ou _
- Não usar acentos ou til
- Não podem ter espaço em branco
- Sugestão: use nomes que **tenham um significado**

Errado	Correto	Não recomendado

`int 5minutos;`
`int salário;`
`int salario do funcionario;`
	`int _5minutos;`
	`int salario;`
	`int salarioDoFuncionario;`
		`int variavel1;`
		`int variavel2;`
		`int x;`

Relembrando

Variável tipo valor *vs* tipo referência

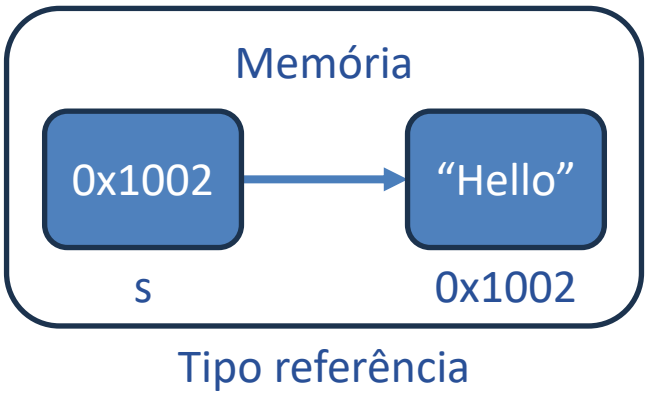
Tipo valor

- Valor da variável é salvo diretamente na memória
- `int x = 20;`

Tipo referência

- Endereço (referência) da variável final é salvo na memória
- `string s = “Hello”;`

Tipo valor



Tipo referência

Operadores aritméticos

Executam operações matemáticas

Operador	Significado
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da divisão)

Precedência:

*/% tem precedência maior que +-

Exemplo:
3 + 4 *2 → Resultado 11
(3 + 4) * 2 → Resultado 14

Obs: Operador ^ não realiza exponenciação. Para isso, deve-se usar o método **Math.Pow(x, exp)** do namespace **System.Math**

https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/arithmetic-operators

Operadores de atribuição

Usados para atribuir valores

Atribui um valor ao operando à sua esquerda baseado no valor do operando à direita

Operador	Exemplo	Significado
=	a = 10	a recebe valor 10
+=	a += 10	a recebe valor a + 10
-=	a -= 10	a recebe valor a – 10
*=	a *= 10	a recebe valor a * 10
/=	a /= 10	a recebe valor a / 10
%=	a %= 3	a recebe valor a % 10

https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/equality-operators

Operadores postfix e prefix

Usados para alterar valores

Prefix: o valor é incrementado/decrementado e então retornado

Postfix: o valor é retornado e depois incrementado/decrementado

Operador	Exemplo	Significado
++	a++	Retorna a. Depois faz a = a + 1
++	++a	Faz a = a + 1. Depois retorna a
--	a--	Retorna a. Depois faz a = a - 1
--	--a	Faz a = a - 1. Depois retorna a

https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/arithmetic-operators#increment-operator-

Operadores comparativos

Usados para comparar valores

Operador	Significado
>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou igual
==	Igual
!=	Diferente

https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/comparison-operators
https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/equality-operators

Operadores lógicos

Funcionamento idêntico ao de portas lógicas

Usados com variáveis booleanas

Operador	Significado
&&	E (and)
||	Ou (or)
!	Não (not)
^	Ou exclusivo (xor)

Precedência:

! > ^ > && > ||

Exemplo:
true || false && !false → Resultado true
(true || false) && !true → Resultado false

https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/boolean-logical-operators

Conversão implícita e casting

Usados para atribuir a uma variável um valor de outro tipo

Conversao implícita possível pois tamanho de *double* é maior que de *float*:

```
float n1 = 1.25F;
```
→ 4 bytes

```
double n2 = n1;
```
→ 8 bytes

Conversao inválida pois um *float* não consegue armazenar todos os bytes de um *double*

```
float n3 = n2
```

Casting de variável. Forçamos n3 a "caber" em 4 bytes. Pode haver perda de informações

```
float n3 = (float)n2;
```

https://learn.microsoft.com/pt-br/dotnet/csharp/programming-guide/types/casting-and-type-conversions

Conversão implícita e casting

Necessidade de casting

```
int a = 10;
int b = 3;

double c = a / b;
Console.WriteLine(c);
```

Resultado = 3
Pois int/int = int

```
int a = 10;
int b = 3;

double c = (double)a / b;
Console.WriteLine(c);
```

Resultado = 3.3333
Pois double/int = double

Operadores

Exercício

Escreva um programa que recebe 3 números (inteiros ou racionais) digitados pelo usuário e, após, exibe se cada número é par ou ímpar, a soma dos três números e a média

https://bit.ly/498A5Ni

Estrutura condicional *if-else*

Executa código baseado em condição lógica

Simples

```
if (condição)
{
    comando 1;
    comando 2;
}
```

Composta

```
if (condição)
{
    comando 1;
    comando 2;
}
else
{
    comando 3;
    comando 4;
}
```

Encadeada

```
if (condição 1)
{
    comando 1;
    comando 2;
}
else if (condição 2)
{
    comando 3;
    comando 4;
}
else
{
    comando 3;
    comando 4;
}
```

Nota: se o bloco de comandos possuir apenas um comando, as chaves são opcionais

Operador ternário

Versão compacta de if-else

```
Condição ? Código se verdadeiro : Código se falso;
```

Exemplo: Verificar se número x é par

```
string par = x % 2 == 0 ? "par" : "impar";
```

Programa verifica se resto da divisão de x por 2 é nulo. Se sim, string *par* recebe o valor “par”. Senão, recebe o valor “impar”.

Obs: Evitar o encadeamento de operadores ternários

```
hora < 12 ? "Bom dia" : (hora < 18 ? "Boa tarde" : "Boa noite")
```

Estruturas condicionais

https://bit.ly/3vr5QTM

Exercício

Leia um valor com duas casas decimais, equivalente ao salário de uma pessoa. Em seguida, calcule e mostre o valor que esta pessoa deve pagar de Imposto de Renda, segundo a tabela abaixo:

Renda	Imposto devido
Até 2000.00	Isento
De 2000.01 até 3000.00	8%
De 3000.01 até 4500.00	18%
Acima de 4500.00	28%

Lembre-se que, se o salário for 3002.00 por exemplo, a taxa que incide é de 8% apenas sobre 1000.00, pois a faixa de salário de 0.00 até 2000.00 é isenta de Imposto de Renda. No exemplo, a taxa é de 8% sobre 1000.00 + 18% sobre 2.00, o que resulta em 80.36 no total. O valor deve ser impresso com duas casas decimais.

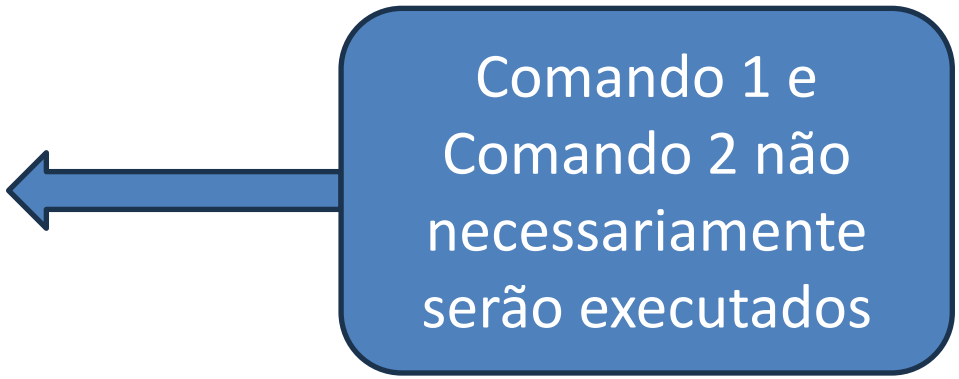
Laços de repetição

Estrutura repetitiva *while* – Enquanto

Regra

- Condição verdadeira: Executa comandos e volta
- Condição falsa: Não executa commandos

```
while (condição)
{
    comando 1;
    comando 2;
}
```



Laços de repetição

Estrutura repetitiva *do-while* – Faça enquanto

Regra

- Executa comando
- Condição verdadeira: Volta
- Falso: Não volta

```
do
{
    comando 1;
    comando 2;
} while (condição)
```

Comando 1 e Comando 2 serão executados **pelo menos** uma vez

Laços de repetição

Estrutura repetitiva for – Para

Executa enquanto condição for verdadeira, executando incremento ao final de cada laço

```
for (início; condição; incremento)
{
    comando 1;
    comando 2;
}
```

Início → Executa na primeira vez

Condição → Se verdadeira, executa e volta. Senão, encerra laço

Incremento → Executa toda vez após comandos

Laços de repetição

Estrutura repetitiva foreach – Para cada

Executa com base em um array ou vetor, selecionando um elemento por vez

```
int[] conjunto = {1, 4, 44, 12, 15, -1, 22, 0};
foreach (int n in conjunto)
{
    comando 1;
    comando 2;
}
```

Enquanto o array é percorrido, *n* assume o valor do elemento em questão de *conjunto*

Laços de repetição

https://bit.ly/3PD3sjE

Exercício

Leia um valor inteiro N, que representa o número de casos de teste que vem a seguir. Cada caso de teste consiste de 3 valores reais, cada um deles com uma casa decimal. Apresente a média ponderada para cada um destes conjuntos de 3 valores, sendo que o primeiro valor tem peso 2, o segundo valor tem peso 3 e o terceiro valor tem peso 5.

Exemplo:

Entrada	Saída
3	
6.5 4.3 6.2	5.7
5.1 4.2 8.1	6.3
8.0 9.0 10.0	9.3

Revisão

Próxima aula

Definição e Sintaxe de Classes

- Conceitos fundamentais de classes
- Estrutura e sintaxe de declaração de classes
- Atributos e construtores

Dúvidas?

renan.duarte@gedre.ufsm.br

GEDRE – Prédio 10 – CTLAB