Variáveis e Operações

Prof. Thiago Felski Pereira, MSc.

Tipos básicos de variáveis

C# Type	Valores possíveis de se armazenar
bool	Verdadeiro ou Falso (Valores booleandos)
byte	0 a 255 (8 bits)
sbyte	-128 a 127 (8 bits)
char	Um caractere (16 bits)
decimal	±1.0 × 10−28 a ±7.9 × 1028 (128 bits)
double	±5.0 × 10−324 a ±1.7 × 10308 (64 bits)
float	±1.5 × 10−45 a ±3.4 × 1038 (32 bits)
int	-2,147,483,648 a 2,147,483,647 (32 bits)

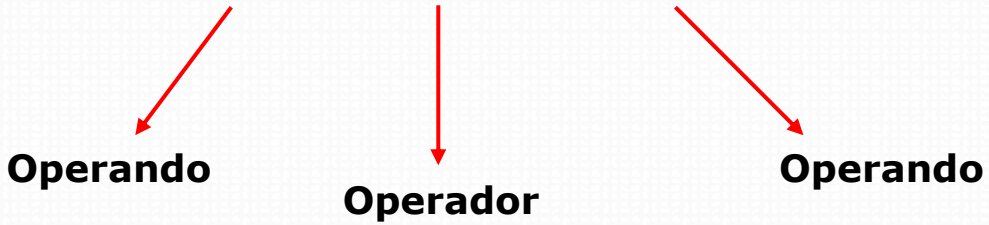
Tipos básicos de variáveis

C# Type	Valores possíveis de se armazenar
uint	0 a 4,294,967,295 (32 bits)
long	–9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 (64 bits)
ulong	0 a 18,446,744,073,709,551,615 (64 bits)
object	Qualquer tipo.
short	-32,768 a 32,767 (16 bits)
ushort	0 a 65,535 (16 bits)
string	Sequência de caracteres (16 bits por caractere)

Operadores e expressões

- **OPERADORES**
- Uma expressão é composta por um ou mais operandos, que se combinam entre si mediante operadores, produzindo um resultado.

- Exemplo: X + 15



UNIVALI

Operadores e expressões

- Nas linguagens é possível formar **expressões** utilizando variáveis constantes e operadores matemáticos: + (adição), - (subtração), / (divisão) e % (resto de divisão inteira, ou mod).
- Essas expressões são utilizadas onde é preciso utilizar um valor, que será o que resultará da expressão. Exemplo:
  - X + 30 //devolve x + 30.
  - X == 55 // verifica se são iguais e retorna 1 ou 0.
  - X = 10 //atribuição: retorna o valor atribuído.
- As expressões não são iguais a sentenças. As **sentenças** indicam ao compilador que realize alguma tarefa (e terminam com ;), enquanto as expressões indicam um cálculo. Uma sentença pode conter várias expressões.

Prioridade de operadores

- **OPERADORES**
- Precedência (prioridade): indica a prioridade do operador mediante outros no cálculo de uma expressão.

- Associatividade: determina a ordem que operandos do mesmo tipo se associam na ausência de parênteses.
  - Podem ser pela direita (D-E) – operadores binários, ternários e de atribuição, ou pela esquerda (E-D) – operadores binários. Em alguns operadores a associatividade não faz sentido, como no caso do operador sizeof().

Prioridade de operadores

- **OPERADORES**
- A tabela a seguir mostra a prioridade entre os operadores. O grupo 1, por exemplo, possui maior prioridade que o grupo 2. Seguem outras regras:
  - Se dois operandores são aplicados no mesmo operando, o operador com maior prioridade é aplicado primeiro.
  - Todos os operadores do mesmo grupo possuem prioridade de associatividade iguais.
  - Se dois operadores possuem prioridade igual, primeiro se aplica o operador com prioridade mais alta.
  - Os parênteses possuem prioridade máxima, e alteram qualquer ordem.

Prioridade de operadores

Prioridade	Operadores	Associatividade
1	:: (Resolução de Escopo) * -> (Direções) [] (Indexação Vetorial) () (Chamada de Funções)	E-D
2	++ -- (Incremento e Decremento) ~ (Manipulação de bits) ! (Lógico) - + (Aritméticos – negativo ou positivo) & * (Endereço) sizeof (tamanho de bytes)	D-E
3	.* ->*	E-D
4	* / % (Aritméticos)	E-D
5	+ - (Aritméticos)	E-D
6	<< >> (Deslocamento de bits)	E-D
7	< <= > >= (Relacionais)	E-D
8	== != (Relacionais)	E-D

Prioridade de operadores

Prioridade	Operadores	Associatividade
9	& (Manipulação de bits)	E-D
10	^ (Manipulação de bits)	E-D
11	|| (Manipulação de bits)	E-D
12	&& ou and (Lógico)	E-D
13	|| ou or (Lógico)	E-D
14	?: (Expressão Condicional)	D-E
15	= *= /= %= += -= <<= >>= &= ||= ^= (Atribuição)	D-E
16	, (Vírgula)	E-D

Operadores Aritméticos

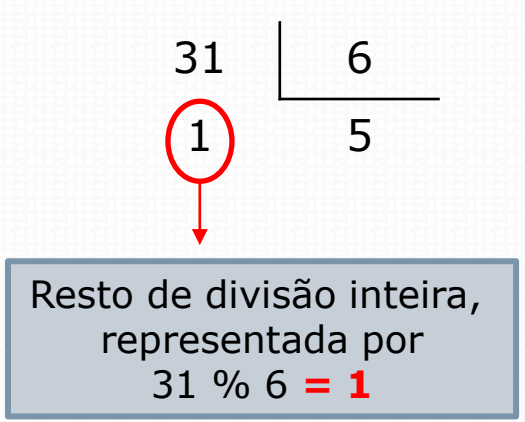
- Realizam operações aritméticas básicas.
- Prioridade de avaliação:
  - 1. Parênteses (mudam a ordem de prioridade).

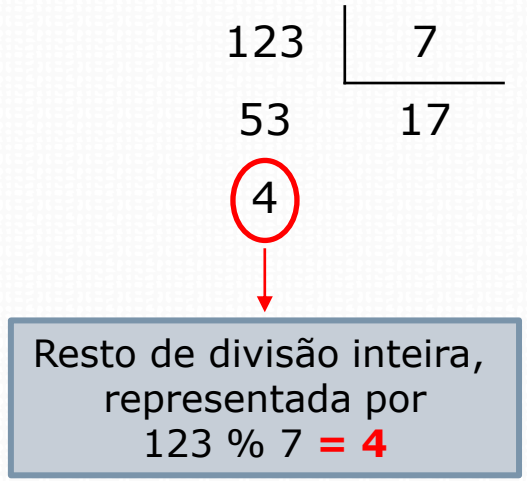
Operador	Operação	Prioridade
+, -	+25, - 6.475	2
*, /, %	5*5 é 25	3
	25/5 é 5	
	25%6 é 1	
+, -	2+3 é 5	4
	2-3 é -1	

- Quando de mesma prioridade, a associatividade é da esquerda para a direita.

Prioridade de operadores

- O que faz o operador %?
  - Este não é um operador de porcentagem, como na calculadora. É um operador de resto de divisão inteira. Exemplo:

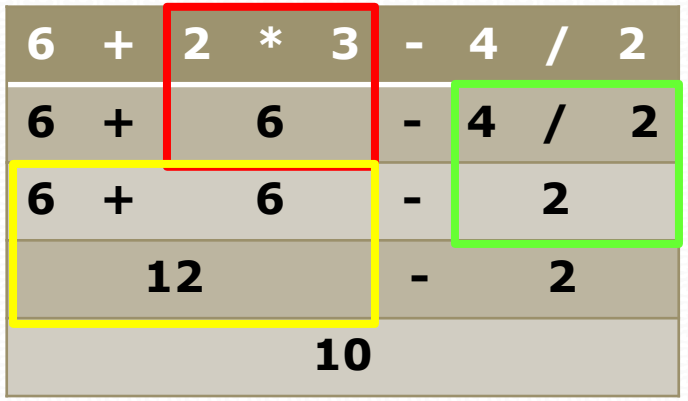


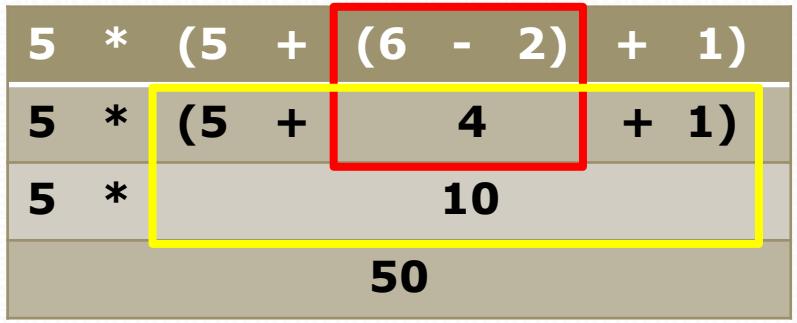


UNIVALI

Prioridade de operadores

- Veja as expressões abaixo, resolvidas de acordo com a prioridade.





- Sempre que abrir um parênteses feche-o, pois senão será gerado um erro de compilação. E cuide, pois a utilização dos parênteses de forma incorreta modificam todo o valor da expressão.

UNIVALI

Exercícios

- Questão 1: 5 + 6 * 4
- Resposta: _____.

- Questão 2: 5 + 6 - 3
- Resposta: _____.

- Questão 3: 12.0 / 6.0 * 4
- Resposta: _____.

- Questão 4: 8 + 3 * 5
- Resposta: _____.

- Questão 5: 4 % (5 - 3)
- Resposta: _____.

Exercícios

- Questão 1: 5 + 6 * 4
- Resposta: 29.

- Questão 2: 5 + 6 - 3
- Resposta: 8.

- Questão 3: 12.0 / 6.0 * 4
- Resposta: 8.

- Questão 4: 8 + 3 * 5
- Resposta: 23.

- Questão 5: 4 % (5 - 3)
- Resposta: 0.

Exercícios

- Questão 6: 12 % 5 / 3
- Resposta: _____.

- Questão 7: 2 + (5 * (3 + 2))
- Resposta: _____.

- Questão 8: 4 + ((10 / 2) * 4)
- Resposta: _____.

- Questão 9: 12 % 7
- Resposta: _____.

- Questão 10: 125 / 3 % 7 + 3 * 5 - 4
- Resposta: _____.

Exercícios

- Questão 6: 12 % 5 / 3
- Resposta: 0. Alguém sabe porque não é 0.66?

- Questão 7: 2 + (5 * (3 + 2))
- Resposta: 27.

- Questão 8: 4 + ((10 / 2) * 4)
- Resposta: 24.

- Questão 9: 12 % 7
- Resposta: 5.

- Questão 10: 125 / 3 % 7 + 3 * 5 - 4
- Resposta: 17.

Leitura e Escrita no console

- Algo importante a ser lembrado é que o usuário do nosso programa raramente terá acesso ao código que estivermos escrevendo.
  - Portanto é essencial que nos comuniquemos de forma correta com o usuário do nosso programa

- O método mais simples de ler e escrever informações é utilizando o console.
  - ESCRITA: `Console.WriteLine()` e `Console.Write()`
  - LEITURA: `Console.Read()`

Leitura e Escrita no console

- Exemplos de ESCRITA no console

- Escrevendo o texto: Olá Mundo!

```
1 using System;
2 public class Program {
3     public static void Main() {
4         Console.Write("Olá Mundo!");
5         Console.Write("Olá Mundo!");
6     }
7 }
```

  - Escreve o texto entre aspas e mantém o cursor para continuar a escrita logo depois do texto escrito. Assim se você utilizar o comando 2x ele irá escrever:
  - `Olá Mundo!Olá Mundo!`

Leitura e Escrita no console

- Exemplos de ESCRITA no console

- Escrevendo o texto: Olá Mundo!

```
1 using System;
2 public class Program {
3     public static void Main() {
4         Console.WriteLine("Olá Mundo!");
5         Console.WriteLine("Olá Mundo!");
6     }
7 }
```

  - Escreve o texto entre aspas e ajusta o cursor para continuar a escrita na linha seguinte do texto escrito. Assim se você utilizar o comando 2x ele irá escrever:
  - `Olá Mundo!`
  - `Olá Mundo!`

Leitura e Escrita no console

- Exemplos de ESCRITA no console

- Escrevendo o conteúdo de uma variável

```
1 using System;
2 public class Program {
3     public static void Main() {
4         int variavel = 10;
5         Console.WriteLine(variavel);
6     }
7 }
```

  - Escreverá o conteúdo de variável, ficando a escrita da seguinte forma
  - 10

Leitura e Escrita no console

- Exemplos de ESCRITA no console

- Escrevendo várias coisas com um único comando de escrita
  - Para escrever várias coisas com um único comando você precisa separar as várias escritas com o sinal de +

```
1 using System;
2 public class Program {
3     public static void Main() {
4         int v1 = 10, v2 = 20;
5         Console.WriteLine("v1 = " + v1 + " e v2 = " + v2);
6     }
7 }
```

  - Ficando o texto escrito da seguinte forma
  - `v1 = 10 e v2 = 20`

Leitura e Escrita no console

- Exemplos de ESCRITA no console

- Editores de escrita
  - Você pode verificar todas as possibilidades em:
    - https://learn.microsoft.com/pt-br/dotnet/api/system.console.writeline?view=net-7.0
  - Por exemplo: é possível escrever comandos nos textos entre aspas como
  - \n para nova linha
  - \t para TAB

```
1 using System;
2 public class Program {
3     public static void Main() {
4         Console.WriteLine("\n\n\t\tOlá Mundo!");
5     }
6 }
```

- O texto será escrito 2 linhas depois e 2 TABS a frente

Leitura e Escrita no console

- Exemplos de LEITURA no console

- Quando queremos que o usuário informe algo para o programa é desejável que informemos isso por meio de uma escrita:
  - `Console.WriteLine("Informe seu nome: ");`

  - `Console.WriteLine("Informe sua idade: ");`

  - `Console.WriteLine("Informe sua altura: ");`

Leitura e Escrita no console

- Exemplos de LEITURA no console

- Para ler textos do Console, basta utilizar o comando `Console.ReadLine()`, como no exemplo a seguir:

```
string nome;
Console.WriteLine("Informe seu nome: ");
nome = Console.ReadLine();
```

- Como `string` guarda variáveis de texto e `Console.ReadLine()` o retorna o texto digitado, não teremos que fazer conversão nenhuma.

Leitura e Escrita no console

- Exemplos de LEITURA no console

- Como tudo que lemos no console vem para o programa no formato de texto, cabe ao programador converter (`Parse`) o texto recebido para o formato desejado.

```
int idade;
float altura;
Console.WriteLine("Informe sua idade: ");
idade = int.Parse(Console.ReadLine());
Console.WriteLine("Informe sua altura: ");
altura = float.Parse(Console.ReadLine());
```

Obrigado pela atenção

contato: felski@univali.br