

# Linguagem de Programação C#

---

Prof. Ricardo Frohlich da Silva

# Linguagem C#

---

- C# é uma linguagem de programação, multiparadigma, de tipagem forte, desenvolvida pela Microsoft como parte da plataforma .NET.
- A sua sintaxe orientada a objetos foi baseada no C++ mas inclui muitas influências de outras linguagens de programação, como Object Pascal e, principalmente, Java.
- O código fonte é compilado para Common Intermediate Language (CIL) que é interpretado pela máquina virtual Common Language Runtime (CLR).
- C# é uma das linguagens projetadas para funcionar na Common Language Infrastructure da plataforma .NET Framework.

# Características da Linguagem C#

---

- É uma linguagem compilada:
  - código fonte → código objeto → código executável (linguagem de máquina).
- Todo comando termina em ';' (ponto e vírgula).

# Características da Linguagem C#

---

- Outras características:
  - Início e final de blocos representados por { e }
  - Comentários entre /\* \*/ ou //para comentar 1 linha

Comentários são trechos de texto que o compilador irá ignorar, pois são somente “comentários” que colocamos no código

# Tipos de dados da Linguagem C#

---

- Tipos básicos de dados
  - char (caractere).
    - Ex: **char** letra1 = 'A';
  - int (inteiro).
    - Ex: **int** valor1 = 54;
  - float (ponto flutuante).
    - Ex: **float** valor2 = 43.6778;
  - double (ponto flutuante de precisão dupla).
    - Ex: **double** valor3 = 32.45345346;

# Tipos de dados da Linguagem C#

Tipo	Tamanho aprox. em bits	Faixa
char	8	-127 a 127
int	16	-32.767 a 32.767
unsigned int	16	0 a 65.535
short int	O mesmo que int	
long int	32	-2.147.483.647 a 2.147.483.647
unsigned long int	32	0 a 2.294.967.265
float	32	6 dígitos de precisão
double	64	10 dígitos de precisão
long double	80	10 dígitos de precisão

# Variáveis na Linguagem C#

---

- Variável é uma posição nomeada de memória, que possui um nome para identificação e corresponde a um tipo de dado.

- Declaração de variáveis na linguagem C#:
  - **TIPO\_DE\_DADO** nome\_var1, nome\_var2;

- É possível na declaração definir um valor inicial da variável.

# Variáveis na Linguagem C#

---

- Exemplos de declaração:
  - `int x, y, z=10;`
  - `float a = 3, c = 4.23, d;`
  - `char i, j = 'a';`



# Operadores na Linguagem C#

---

- Operadores são símbolos utilizados para realizar operações lógicas e aritméticas sobre operandos.
- Operadores aritméticos atuam sobre variáveis, constantes e funções numéricas e produzem um resultado numérico.
- Operadores possuem prioridades em relação aos outros, ou seja, qual operação será executada primeiro em relação as demais.

# Operadores na Linguagem C#

---

- A tabela abaixo apresenta os operadores da linguagem C# e suas prioridades:

Prioridade	Operador	Operação
1	-	Inversão de sinal
2	*	Multiplicação
2	/	Divisão
2	%	Resto de Divisão
3	+	Adição
3	-	Subtração

# Operadores na Linguagem C#

---

- Os operadores que possuem mesma prioridade são executados na ordem em que aparecem quando a expressão é lida da esquerda para a direita.
- O uso dos parênteses define uma ordem de execução prioritária em relação à prioridade dos operadores da linguagem.
- Exemplos:
  - $x + 5 * y - 4$ 
    - a primeira operação realizada é a multiplicação, em seguida é executada a soma e por fim a subtração.
  - $(x+5) * (y-4)$ 
    - para estabelecer uma precedência diferente, podem ser utilizados os parênteses.
    - Neste caso, primeiro é realizado a soma, depois a diferença, e por fim a multiplicação.

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

```
int a float b float c float d
```

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3			

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	=a*3		

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	=3*3		

# Exemplo completo em C#

---

```
static void Main(string[] args)
{
    int a;
    float b, c, d;
    a = 3;
    b = a * 3;
    c = b / 2;
    d = a + c;
    d = c - (d/2) + c * d;
    c = a % 2;
    c = c + d;
}
```

int a	float b	float c	float d
3	9		



# Exemplo completo em C

---

```
static void Main(string[] args)
{
    int a;
    float b, c, d;
    a = 3;
    b = a * 3;
    c = b / 2;
    d = a + c;
    d = c - (d/2) + c * d;
    c = a % 2;
    c = c + d;
}
```

int a	float b	float c	float d
3	9	=b/2	

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	=9/2	

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	4.5	

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	4.5	=a+c

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	4.5	=3+4.5

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	4.5	7.5

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	4.5	$=c - (d/2) + c*d$

# Exemplo completo em C#

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	4.5	$=4.5 - (7.5/2) + 4.5 * 7.5$

Como foi utilizado o próprio valor de  $d$  na fórmula, é utilizado então o valor atual de  $d$  antes da execução da operação, ou seja, 7.5



# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	4.5	34.5

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	=a % 2	34.5

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	=3 % 2 ??????	34.5

# Exemplo completo em C#

```
static void Main(string[] args)
{
    int a;
    float b, c, d;
    a = 3;
    b = a * 3;
    c = b / 2;
    d = a + c;
    d = c - (d/2) + c * d;
    c = a % 2;
    c = c + d;
}
```

int a	float b	float c	float d
3	9	=3 % 2 ??????	34.5

O símbolo % representa o resto da divisão.

Ou seja, neste caso, se dividirmos 3 por 2:

$$\begin{array}{r} 3 \overline{) 2} \\ \underline{2} \phantom{0} \\ 1 \end{array}$$

este é o resto da divisão!

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	1	34.5

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	=c+d	34.5

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	=1+34.5	34.5

# Exemplo completo em C#

---

```
static void Main(string[] args)
```

```
{
```

```
    int a;
```

```
    float b, c, d;
```

```
    a = 3;
```

```
    b = a * 3;
```

```
    c = b / 2;
```

```
    d = a + c;
```

```
    d = c - (d/2) + c * d;
```

```
    c = a % 2;
```

```
    c = c + d;
```

```
}
```

int a	float b	float c	float d
3	9	=35.5	34.5



# Comandos de entrada e saída

---

- Para permitir a interação do usuário com os programas, existem os comandos que fazem a entrada (leitura) e saída (escrita) de dados.
- O dispositivo padrão de entrada é o teclado e o dispositivo padrão de saída de dados é o monitor de vídeo.

# Comando de saída

---

- **Console.WriteLine** (**string de controle**)
  - Na string de controle iremos colocar os dados que serão impressos e, se necessário, o formato que estes serão exibidos.

# Comando de saída

---

- Exemplo 1: mostrar um texto na tela:

- `static void Main(string[] args)`

- `{`

`Console.WriteLine("Eita mundo sô!");`

- `}`

- Quando queremos mostrar somente um texto fixo, sem variáveis, basta colocar o texto entre aspas duplas

# Comando de saída

---

- Exemplo 2: mostrar um texto e o valor de uma variável na tela:

```
static void Main(string[] args)
{
    int x;
    x = 5;
    Console.WriteLine("Você digitou: "+x);
}
```

# Comando de saída

---

- Exemplo 3: mostrar várias variáveis junto a um texto na tela:

```
static void Main(string[] args)
{
    int x = 5, y = 10;
    double z;
    z = 3.1415;
    Console.WriteLine("Os valores são "+x+", "+y+" e "+z);
}
```

---

```
static void Main(string[] args)
{
    int idade1, idade2, diferenca;
    char letra1, letra2 = 'b';
    double valor1 = 5.788, valor2;
    letra1 = 'z';
    idade1 = 20;
    idade2 = 45;
    diferenca = idade2 - idade1;
    valor2 = valor1 - 2.088;
    Console.WriteLine("este é o meu primeiro algoritmo com saída de dados na tela");
    Console.WriteLine("a idade 1 é: "+idade1);
    Console.WriteLine("a idade 2 é: "+idade2);
    Console.WriteLine("a letra 1 é: "+letra1);
    Console.WriteLine("a letra 2 é: "+letra2);
    Console.WriteLine("o valor 1 é: "+valor1);
    Console.WriteLine("o valor 2 é: "+valor2);
    Console.WriteLine("a diferenca do valor1 para o valor 2 é: "+(valor1-valor2));
    Console.WriteLine("a diferença de idade é:"+diferenca);
    Console.WriteLine("a diferença de idade é: "+(idade2 - idade1));
    Console.WriteLine("mostrando tudo: "+idade1+", "+idade2+", "+letra1+", "+letra2+", "+valor1+", "+valor2);
}
```

# Comando de entrada

---

- **Console.ReadLine()**
- Na linguagem C#, a leitura irá diferir para cada tipo de dado.
- No `scanf` antes dos nomes das variáveis a serem lidas pelo teclado deve-se inserir um `&` (que indica o endereço da variável).

# Comando de entrada

---

- Exemplo 1: leitura de uma variável:

```
static void Main(string[] args)
{
    int x;
    x = int.Parse(Console.ReadLine());
}
```



# Comando de entrada

---

- Exemplo 2: leitura de uma variável:

```
static void Main(string[] args)
{
    int x;
    Console.WriteLine("Digite um numero: ");
    x = int.Parse(Console.ReadLine());
}
```

- Neste exemplo, antes de ler o valor para x, mostramos um texto informativo, pedindo para que o usuário digite um valor.
- Torna nossa interface mais amigável 😊

# Comando de entrada

---

- Exemplo 2: leitura de uma variável:

```
static void Main(string[] args)
{
    int x;
    Console.WriteLine("Digite um numero: ");
    x = int.Parse(Console.ReadLine());
    Console.WriteLine("Você digitou o numero "+x);
}
```

- E também podemos mostrar o que foi digitado anteriormente...

# Comando de entrada

---

- E para os outros tipos de dados ?
  - Para char -> `char.Parse(Console.ReadLine());`
  - Para double -> `double.Parse(Console.ReadLine());`

# Comando de entrada

---

- Exemplo 3: leitura de variáveis:

```
static void Main(string[] args)
{
    int x;
    char ch;
    double n2;

    Console.WriteLine("Digite um numero inteiro: ");
    x = int.Parse(Console.ReadLine());

    Console.WriteLine("Digite uma letra: ");
    ch = char.Parse(Console.ReadLine());

    Console.WriteLine("Digite um numero, pode ser com decimal: ");
    n2 = double.Parse(Console.ReadLine());

    Console.WriteLine("Você digitou o numero "+x);
    Console.WriteLine("Você digitou a letra " + ch);
    Console.WriteLine("Você o numero " + n2);
}
```

# Comando de entrada

---

- E para os outros tipos de dados ?
  - Para char -> `char.Parse(Console.ReadLine());`
  - Para double -> `double.Parse(Console.ReadLine());`
  - Deu tudo certo?

# Comando de entrada

---

- E para os outros tipos de dados ?
  - Para char -> `char.Parse(Console.ReadLine());`
  - Para double -> `double.Parse(Console.ReadLine(),  
CultureInfo.InvariantCulture);`

# Comando de entrada

---

- O que é `CultureInfo.InvariantCulture`?

# Comando de entrada

---

```
static void Main(string[] args)
{
    int x;
    char ch;
    double n2;

    Console.WriteLine("Digite um numero inteiro: ");
    x = int.Parse(Console.ReadLine());

    Console.WriteLine("Digite uma letra: ");
    ch = char.Parse(Console.ReadLine());

    Console.WriteLine("Digite um numero, pode ser com decimal: ");
    n2 = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

    Console.WriteLine("Você digitou o numero "+x);
    Console.WriteLine("Você digitou a letra " + ch);
    Console.WriteLine("Você o numero " + n2);
}
```



# Atividade

---

- Instale o Visual Studio 2022  
(<https://visualstudio.microsoft.com/pt-br/vs/>)
- Como instalar?  
<https://www.youtube.com/watch?v=2IIEDkIddeM>
- Replique os códigos aqui nestes slides

# Atividade

---

- Desenvolva os seguintes exercícios:
  - Faça a leitura de dois números e apresenta a soma, subtração, multiplicação e divisão
  - Faça um algoritmo para o calculo da área de um retângulo.
  - Faça um algoritmo para calcular a área de um triângulo equilátero.
  - Escreva um algoritmo para calcular a área de um círculo, com base em um raio digitado pelo usuário.
  - Escreva um algoritmo que leia a idade de uma pessoa, e então mostre na tela o ano em que ela nasceu.