





Programação Orientada a Objetos

Ms. Eng. Claudio Ferreira de Carvalho



AULA 04

Variáveis, tipos e utilizações, conceito de alias e constantes

Variáveis

- ✓ Valores que são atribuídos e serão manipulados durante a programação
- ✓ Possui:
 - ✓ Identificador => nome da variável
 - ✓ Conteúdo => valor da variável

Identificador

- ✓ Iniciar com uma letra.
- ✓ Pode utilizar letras e números.
- ✓ Não utilizar caracteres especiais.
- ✓ Não deixar espaço entre as letras.
- ✓ Não colocar letras ou palavras sublinhadas.
- ✓ Iniciar preferivelmente com letras minúsculas.
- ✓ Se representar duas palavras a segunda palavra deve iniciar com maiúscula.

Conteúdo

- ✓ Numérico: => Inteiro ou Decimal com ponto flutuante.
- ✓ Caractere => Aceita letras e números (se for número não pode ser operado).
- ✓ Boleano => Aceita só dois valores (Verdadeiro ou Falso) => True – False .

Tipos de Variáveis em C#

| Nome C# | Tipo | Valores possíveis de armazenar | Tamanho |
|---------|----------|--|----------|
| bool | Boleano | Verdadeiro ou Falso | 8 bits |
| byte | Inteiro | 0 a 255 | 8 bits |
| sbyte | Inteiro | -128 a 127 | 8 bits |
| ushort | Inteiro | 0 a 65,535 | 16 bits |
| short | Inteiro | -32,768 a 32,767 | 16 bits |
| uint | Inteiro | 0 a 4,294,967,295 | 32 bits |
| int | Inteiro | -2,147,483,648 a 2,147,483,647 | 32 bits |
| ulong | inteiro | 0 a 18,446,744,073,709,551,615 | 64 bits |
| long | inteiro | 9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 | 64 bits |
| float | Real | $\pm 1.5 \times 10^{-45}$ a $\pm 3.4 \times 10^{38}$ | 32 bits |
| double | Real | $\pm 5.0 \times 10^{-324}$ a $\pm 1.7 \times 10^{308}$ | 64 bits |
| decimal | Real | $\pm 1.0 \times 10^{-28}$ a $\pm 7.9 \times 10^{28}$ | 128 bits |
| char | Caracter | Um caracter | 16 bits |
| string | Caracter | Sequência de caracter (16 bits por caráter) | 16 bits |

| Nome C# | Tipo | Valores possíveis de armazenar | Precisão | Tamanho |
|---------|------|--|-----------------|----------|
| Float | Real | $\pm 1.5 \times 10^{-45}$ a $\pm 3.4 \times 10^{38}$ | 7 dígitos | 32 bits |
| Double | Real | $\pm 5.0 \times 10^{-324}$ a $\pm 1.7 \times 10^{308}$ | 15 a 16 dígitos | 64 bits |
| Decimal | Real | $\pm 1.0 \times 10^{-28}$ a $\pm 7.9 \times 10^{28}$ | 28 a 29 dígitos | 128 bits |

float e double são armazenados na forma binária, enquanto que dados do tipo decimal, são armazenados na forma decimal. Isto faz com que, embora utilizando mais memória os dados na forma decimal ficam mais precisos.

Declaração de Variáveis

✓ Objetivo

- ✓ Declarar o tipo de variável e o identificador da variável.
- ✓ É possível ao declarar a variável também atribuir a ela um conteúdo.
- ✓ O conteúdo também pode ser atribuído em linhas seguintes após a declaração.

✓ Sintaxe

// Declaração de variáveis

bool n1;

string n2, n3, n4;

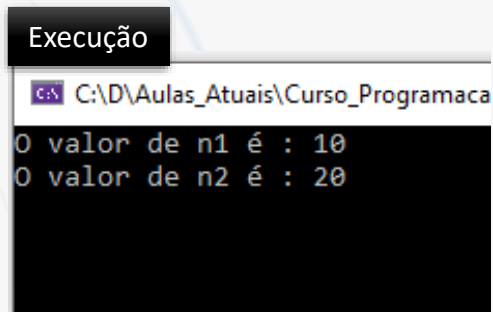
double a2;

int a1 = 10;

// Atribuir valores a variáveis já declaradas

a2 = 5,2;

✓ Exemplo



```
Execução
C:\D\Aulas_Atuais\Curso_Programaca
O valor de n1 é : 10
O valor de n2 é : 20
```

```
1. 1 namespace Aula04_Ex01
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de Variáveis do tipo Inteiro
8.             byte n1;
9.             byte n2 = 20;
10.
11.             // Atribui valor
12.             n1 = 10;
13.
14.             // Mostra os valores
15.             Console.WriteLine("O valor de n1 é : " + n1);
16.             Console.WriteLine("O valor de n2 é : " + n2);
17.
18.             // Mantem a tela aberta esperando a digitação de uma tecla
19.             Console.ReadKey();
20.         }
21.     }
```

Variável Real do tipo float

- ✓ Objetivo
 - ✓ Declarar variável do tipo float

- ✓ Sintaxe

float Identificador_da_variável; // (cria a variável sem atribuir valor

float Identificador_da_variável = valor numérico; // cria a variável e atribui um valor

- ✓ Exemplo

```
1. namespace Aula04_Ex02
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de Variáveis do tipo float
8.             float f1;
9.             float f2 = 60.229F;
10.
11.             // Atribui valor
12.             f1 = 50.1249F;
13.
14.             // Mostra os valores
15.             Console.WriteLine("O valor de f1 é: {0:F2} ", f1);
16.             Console.WriteLine("O valor de f2 é: {0:F2} ", f2);
17.             Console.ReadKey();
18.         }
19.     }
20. }
```

Por padrão, um literal numérico real no lado direito do operador de atribuição é tratado como [double](#). Portanto, para inicializar uma variável float, use o sufixo **F** ou **f**,

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orie
O valor de f1 é: 50,12
O valor de f2 é: 60,23
```

Variável Real do tipo double

✓ Objetivo

- ✓ Declarar variável do tipo double.

✓ Sintaxe

double Identificador_da_variável; *// (cria a variável sem atribuir valor)*

double Identificador_da_variável = valor numérico; *// cria a variável e atribui um valor*

✓ Exemplo

```
1. namespace Aula04_Ex03
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração Variáveis do tipo double
8.             double d1;
9.             double d2 = 40.238;
10.
11.             // Atribui valor
12.             d1 = 30.1;
13.
14.             // Mostra os valores
15.             Console.WriteLine("O valor de d1 é: {0:F2} ", d1);
16.             Console.WriteLine("O valor de d2 é: {0:F2} ", d2);
17.
18.             // Mantem a tela aberta esperando a digitação de uma tecla
19.             Console.ReadKey();
20.         }
21.     }
22. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto:
O valor de d1 é: 30,10
O valor de d2 é: 40,24
```


Variável Real do tipo decimal

✓ Objetivo

- ✓ Declarar variável do tipo decimal

✓ Sintaxe

decimal Identificador_da_variável; // *(cria a variável sem atribuir valor)*

decimal identficador_da_variável = valor numéricoM; // *cria a variável e atribui um valor*

✓ Exemplo

```
1. namespace Aula04_Ex04
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração Variáveis do tipo decimal
8.             decimal dc1;
9.             decimal dc2 = 50.4367M;
10.
11.             // Atribui valor
12.             dc1 = 40.4348M;
13.
14.             // Mostra os valores
15.             Console.WriteLine("O valor de dc1 é: {0:F2} ", dc1);
16.             Console.WriteLine("O valor de dc2 é: {0:F2} ", dc2);
17.
18.             // Mantem a tela aberta esperando a digitação de uma tecla
19.             Console.ReadKey();
20.         }
21.     }
22. }
```

decimal indica um tipo de dados de 128-bits. Comparado aos tipos de ponto flutuante, o tipo decimal tem mais precisão e um intervalo pequeno, o que torna apropriado para cálculos financeiros e monetários. Para decimal utilizar sufixo m ou M,

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Ol
O valor de dc1 é: 40,43
O valor de dc2 é: 50,44
```

Variáveis tipo char e string

✓ Objetivo

- ✓ Variáveis tipo char só recebem um caracter e tipo string podem receber diversos caracteres.

✓ Sintaxe

```
char Identificador_da_variável; // cria a variável tipo char  
string Identificador_da_variável; // cria a variável do tipo string
```

Importante

A atribuição de conteúdo tipo char deve ser feita entre aspas simples.

A atribuição de conteúdo tipo string deve ser feita entre aspas duplas.

Valores tipo char ou string, mesmo que sejam numéricos não podem ser operados.

Variáveis tipo char e string

✓ Objetivo

- ✓ Variáveis tipo char e string

✓ Sintaxe

char Identificador_da_variável;
string Identificador_da_variável;

✓ Exemplo

Importante

A atribuição de valores para variáveis tipo char deve ser feita entre aspas simples.
A atribuição de valores para variáveis tipo string deve ser feita entre aspas duplas.
Valores tipo char devem ser caracteres e valores tipo string devem ser strings.

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projeto04\Program.cs
0 valor de c1 é : C
0 valor de c2 é : Certo
0 telefone é c3 concatenado com C4 :1391112345
```

```
1. namespace Aula04_Ex05
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara Variáveis do tipo Char e String
8.             char c1;
9.             string c2;
10.            string c3, c4;
11.
12.            // Atribui valores
13.            c1 = 'C';
14.            c2 = "Certo";
15.            c3 = "13";
16.            c4 = "91112345";
17.
18.            // Mostra os valores
19.            Console.WriteLine("O valor de c1 é : " + c1);
20.            Console.WriteLine("O valor de c2 é : " + c2);
21.            Console.WriteLine("O telefone é c3 concatenado com C4 : " + c3 + c4);
22.
23.            // Mantem a tela aberta esperando a digitação de uma tecla
24.            Console.ReadKey();
25.        }
26.    }
27. }
```

Variável tipo booleano

- ✓ Objetivo
 - ✓ Variáveis tipo booleano só podem receber dois valores Verdadeiro (True) ou Falso (False)
- ✓ Sintaxe
 - bool identificador; // cria a variável tipo booleano
- ✓ Exemplo

Variável tipo booleano

- ✓ Objetivo
 - ✓ Variáveis tipo booleano
- ✓ Sintaxe
 - bool identificador; // cria a variável
- ✓ Exemplo

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto>

```
O valor de b1 é 20
O valor de b2 é 10
A afirmação b1 >= b2 é True
A afirmação b1 <= b2 é False
```

```
1. namespace Aula04_Ex06
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de Variáveis
8.             byte b1, b2;
9.             bool compara;
10.
11.            // Atribuição de valores
12.            b1 = 20;
13.            b2 = 10;
14.            // Mostrar valores
15.            Console.WriteLine("O valor de b1 é " + b1);
16.            Console.WriteLine("O valor de b2 é " + b2);
17.
18.            // Atribui valores
19.            compara = b1 >= b2;
20.            // Mostrar valores
21.            Console.WriteLine("A afirmação b1 >= b2 é " + compara);
22.
23.            // Atribui valores
24.            compara = b1 <= b2;
25.            // Mostrar valores
26.            Console.WriteLine("A afirmação b1 <= b2 é " + compara);
27.
28.            // Mantém a tela aberta aguardando a digitação de uma tecla
29.            Console.ReadKey();
30.        }
31.    }
32. }
```

Alias

- ✓ Alias normalmente tende a ser um nome mais simples para referenciar algo que tem um nome ou endereço complicado ou muito grande.
- ✓ Ao declarar uma variável, no menu suspenso aparece bit e Byte, double e Double e também em outros tipos.
- ✓ O byte com “b” minúsculo é um tipo do C#, enquanto que Byte com “B” maiúsculo é uma referência à classe do .NET
- ✓ byte é construído em um tipo em C#.
- ✓ System.Byte é uma classe construída no frame .NET que representa um byte.
- ✓ O “byte” é um alias para a classe “System.Byte”.

O programa a seguir demonstra estas informações, pois declara variáveis do tipo byte e Byte e do tipo string e String. Em seguida ele compara os conteúdos e mostra que eles são diferentes, mas ao comparar os tipos mostra que eles são iguais. O programa pode ser repetido para outros tipos de variáveis, como float e Float, double e Double, string e String, etc.

Alias

- ✓ Alias normalmente tende a ser um nome ou endereço curto
- ✓ Ao declarar uma variável, você pode usar também em outros tipos.
- ✓ O byte com “b” minúscula é uma referência à classe `System.Byte`
- ✓ `byte` é construído em um alias
- ✓ `System.Byte` é uma classe
- ✓ O “byte” é um alias para `System.Byte`

```
1. namespace Aula04_Ex07
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // declaração de variáveis iniciando com minúscula e maiúscula
8.             byte b = 1;
9.             Byte B = 2;
10.
11.             // variáveis booleanas para executar a comparação
12.             bool comparavariaveis;
13.             bool comparatipos;
14.
15.             // Mostra os valores e os tipos de variáveis
16.             Console.WriteLine("O valor de b é " + b);
17.             Console.WriteLine("O tipo de b é: " + b.GetType());
18.             Console.WriteLine("\nO valor de B é " + B);
19.             Console.WriteLine("O tipo de B é: " + B.GetType());
20.
21.             // Compara as variáveis e os tipos de variáveis
22.             comparavariaveis = b == B;
23.             comparatipos = b.GetType() == B.GetType();
24.             Console.WriteLine("\nAs variáveis b e B são iguais: " + comparavariaveis);
25.             Console.WriteLine("As variáveis b e B são do mesmo tipo: " + comparatipos);
26.
27.             // Mantém a tela aberta aguardando a digitação de uma tecla
28.             Console.ReadKey();
29.
30.         }
31.     }
32. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projeto>dotnet run
O valor de b é 1
O tipo de b é: System.Byte
O valor de B é 2
O tipo de B é: System.Byte
As variáveis b e B são iguais: False
As variáveis b e B são do mesmo tipo: True
```

Constantes

✓ Objetivo

- ✓ Constantes são variáveis declaradas precedidas da palavras const. Elas tem um tipo mas seus conteúdos não poderão ser alterados durante a programação ou execução do programa.

✓ Sintaxe

const byte valor = 10;

