

Escopo

- Importações
- namespace → separação lógica
- Classe
- método principal

```
1 using System;
2
3 namespace MeuApp;
4
5 // references
6 class Program
7 {
8     // references
9     static void Main(string[] args)
10     {
11         Console.WriteLine("Hello, World!");
12     }
13 }
```

Namespaces → Pastas são divisões físicas

USADO p/ melhor organização, como bloqueando o uso de classes com o mesmo nome em um namespace
→ Acompanham os nomes da pasta.

→ Usados para importações

- Unificam tudo (uma dll) sobrando somente a divisão lógica.

Using

- Importações de libs,
- começo do programa.
- Usando using.
- Organização.

Variações

- Armazena informações. Exemplos
- Pode variar.
- Instanciar == criar.
- Primeira letra em minúscula

```
int idade; // começa com zero
int idade = 19;
var idade = 19;
var idade; // errado
```

→ usado só p/ não usar o nome do tipo.

Constantes

- Armazena dados também, mas é uma informação que não pode variar.
- Exemplo → `const <type> <NAME>`
→ em maiúsculo, separados por _

Palavras Reservadas ou Keywords

→ palavras que não podemos utilizar na criação de var/const...

Comentários

→ não são interpretados pelo compilador

// Comentário de uma linha

/* */ Múltipla linhas

/// Notação XML

Tipos Primitivos / Built-in - Tipos complexos são feitos com os primitivos.

- Armazena o valor e não ref
- Simple Types
- Enums
- Structs
- Nullable Types

Cada tipo tem sua capacidade.

System → Tipo base, o resto vem daqui. Uso já implícito.

Tipos

- Byte de fato ou sbyte que permite valores negativos. 8 bits 0 até 255
sbyte -128 até 127

Inteiros

- short / ushort
 - int / uint
 - long / ulong
- } Remove o SINAL

Definições

- short (16-bit)
 - -32,768 to 32,767
- ushort (16-bit)
 - 0 até 65,535
- int (32-bit)
 - -2,147,483,648 até 2,147,483,647
- uint (32-bit)
 - 0 até 4,294,967,295
- long (64-bit)
 - -9,223,372,036,854,775,808 até 9,223,372,036,854,775,807
- ulong (64-bit)
 - 0 até 18,446,744,073,709,551,615

Reais

- float (F notation) $\rightarrow 32$
 - double $\rightarrow 64$
 - Decimal (D notation) $\rightarrow 128$
- } Sem Unsigned

`float salario = 2.500f;` \rightarrow F notation

`decimal salariozao = 2030210.232132m;` \rightarrow m notation

`double salriozin = 2.000;`

Bool

- Apenas True ou False \rightarrow 8 bits

CHAR

- Somente um caractere (Unicode)
- 16 bits

String

- Coleção de chars (lista)

Var

- Substitui o nome do tipo
- Pegando o tipo do primeiro item.
- Usaremos mais quando criarmos os tipos complexos.