

Operadores Aritméticos

- $+ - * /$ → Mudam a ordem, só usam parenteses
- Com inteiros → Caso o valor seja real, o valor será arredondado

Operadores de Atribuição

- $=$ → Atribuição → inteiro = 5;
- Short hand → $x += 5$; → igual a $x = x + 5$;
- $++$ e $--$ → Também poupa código.

Atribuição

Funciona
p/ todos
os outros

Operadores de Comparação

- Retorna sempre → true ou false
- $==$ igual
- $!=$ diferente
- $> < >= <=$ → compare JA!!!

Qualquer tipo
de dados

Operadores Lógicos

- Usados em condicionais

- $\&\&$ → Todas true
- $\|\|$ → ou
- $!$ → negação

Sempre retorna um Bool,

Condicionais

- IF () → condição, podemos possuir todos os operadores
- {
- // Bloco
- }
- else if ()
- {
- // Bloco
- }
- else
- {
- // Bloco
- }

(ASO o Bloco tem somente
uma linha, pode omitir as chaves

- Podemos também usar o return, que sai da função.

Switch

- Chaveamentos de tomadas de decisão
- precisamos passar cada chaveamento com break;
- switch (var) → podemos ser qualquer tipo
- {
- teste
- CASE "João": (console.WriteLine; break;
- CASE "João": (console.WriteLine; break;
- CASE "João": (console.WriteLine; break;
- DEFAULT: ... ; break;
- }
- exec

Laços de Repetição

- For

```
Contador ↑      Condição ↑      incremento →
for (vna i=0; i <= 5; i++)
{
    // Repete aqui até bater a condição.
}
```

- While

- Checa antes
- Precisa só de uma condição.

while (condição)

```
{
    // Executa aqui até a condição for falsa
    // Precisa colocar algo p/ mudar a condição
}
```

} a ordem

Do

- Já faz antes de checar

```
do
{
```

```
    // faz isso pelo menos uma vez
```

```
}
```

```
while (condição);
```

Funções e métodos

- Possui → retorno, nome e parâmetros.

- Sempre começa em maiúsculos.

- Criação de um método

static void Name (int placeholder)

```
{
```

```
    Console.WriteLine(placeholder)
```

```
}
```

→ tipo do ↓

↑ pode separar por vírgula invocando

→ void pois retorna nada

→ passando o parâmetro p/ o método

Name (5); // saída 5

→ tipo de retorno da função

- Podemos ter muitos métodos com o mesmo nome mas retornando ou com parâmetros diferentes //

- Função com retorno

static string retornaNome (string nome, string sobrenome)

```
{
```

```
    return nome + " " + sobrenome;
```

```
}
```

OBS

Em alguma parte tem que retornar o que foi pedido.

String

- Não podemos ter métodos, dentro de métodos.

- Parâmetros opcionais //

static string retornaNome (string nome, string sobrenome, int idade = 99)

↑ precisa

opcional

e o parâmetro opcional precisa vir por último nos parâmetros.