

C#-5- Control Flow_v1

1. Operações
2. Estruturas de controle

1. Operadores

O C# fornece vários operadores. Muitos deles têm suporte dos [tipos internos](#) e permitem que você execute operações básicas com valores desses tipos. Esses operadores incluem os seguintes grupos:

- [Operadores aritméticos](#) que executam operações aritméticas com operandos numéricos. Veja exemplos no [Link](#).
- [Operadores de comparação](#) que comparam operandos numéricos. Veja exemplos no [Link](#).
- [Operadores lógicos booleanos](#) que executam operações lógicas com operandos. Veja exemplos no [Link](#).
- [Operadores de tecla e bit de alternância](#) que executam operações de bit ou de Shift com operandos dos tipos integrais. Veja exemplos no [Link](#).
- [Operadores de igualdade](#) que verificam se os operandos são iguais ou não. Veja exemplos no [Link](#).

2. Estruturas de controle

No meio dos anos 60, matemáticos provaram que qualquer programa, não importa o quão complicado ele seja, pode ser construído usando uma ou mais de apenas três estruturas, que são: sequência, seleção e iteração.

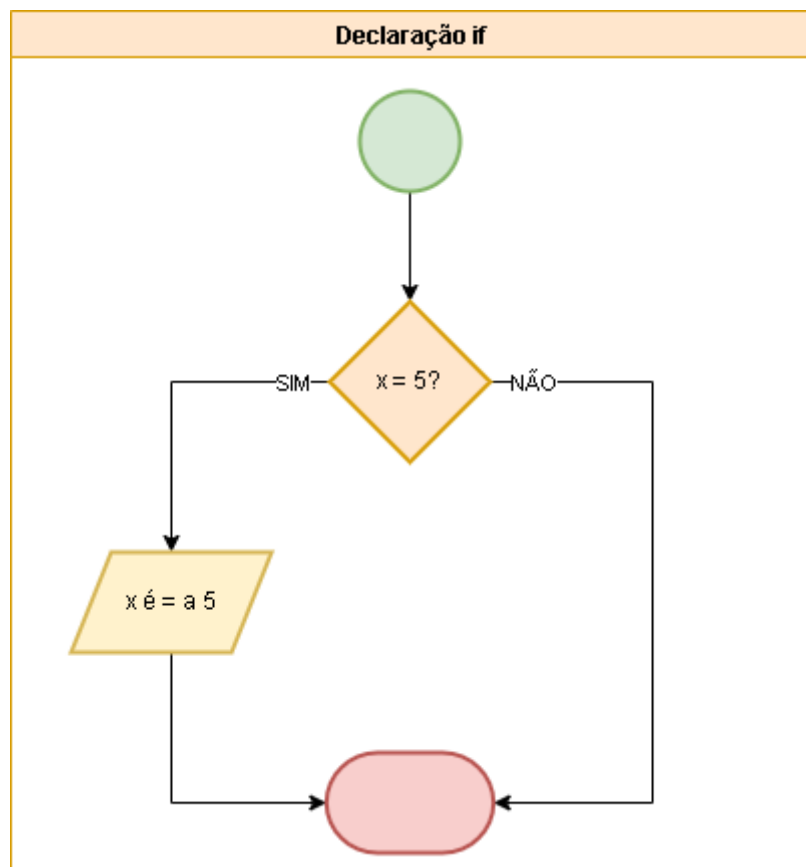
Na sequência, as ações são feitas uma após a outra e o fluxo do programa é linear. Uma vez que você começa uma série de ações em uma sequência, você deve continuar passo-a-passo até que a sequência termine.

Estrutura if

Ela simplesmente recebe uma condição entre parênteses, e se ela for verdadeira, o código que está dentro das chaves é executado. Uma **if** instrução sem **else** uma parte executará seu corpo somente se uma expressão booleana for avaliada como **true**, como mostra o exemplo a seguir:

```
int x = 5;
if (x == 5)
{
    Console.WriteLine("x é igual a 5");
}
```

Fluxograma:

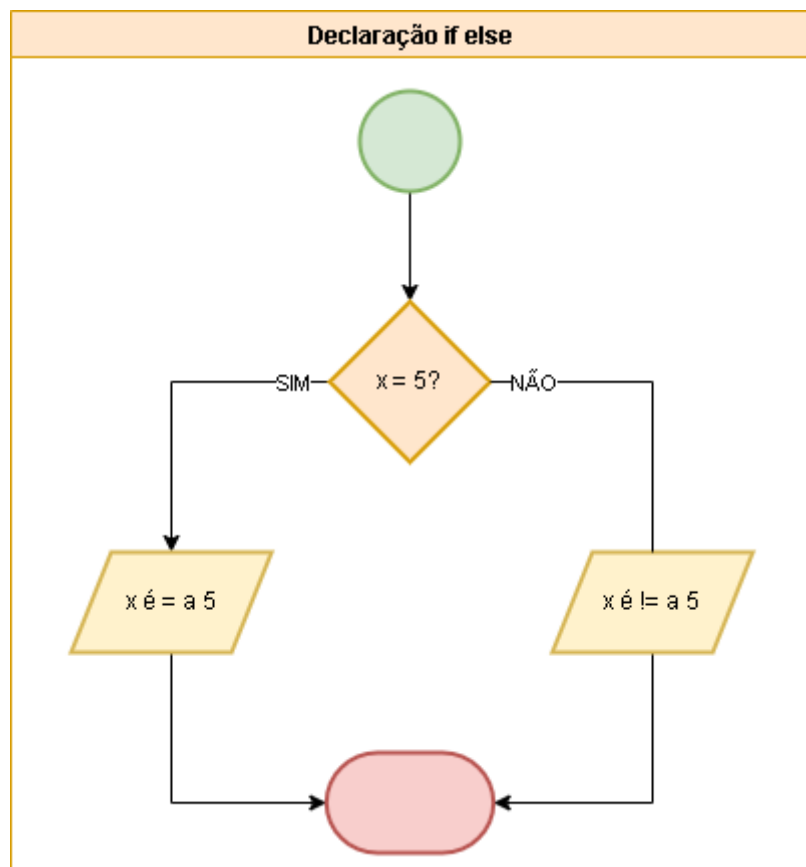


Estrutura if else

Ainda no exemplo anterior, o que faríamos se quiséssemos mostrar uma mensagem caso a variável `x` não fosse igual a 5? Para isso, existe uma estrutura de controle chamada **else**. Ela deve ser colocada imediatamente depois de um **if** ou de um **else if**, que você vai ver daqui a pouco. Veja como ficaria o código com o **if else**:

```
int x = 5;
if (x == 5)
{
    Console.WriteLine("x é igual a 5");
}
else
{
    Console.WriteLine("x é diferente de 5");
}
```

Fluxograma:

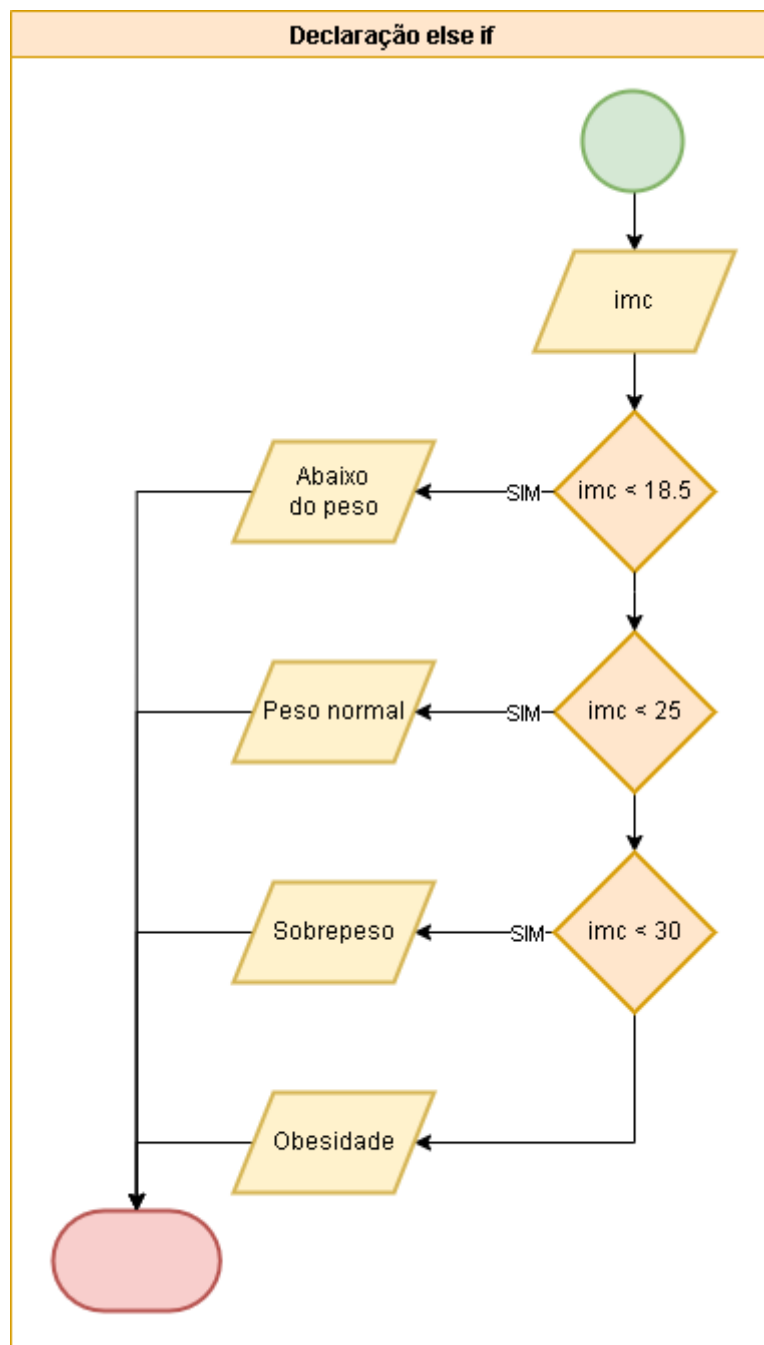


Estrutura else if

Essa estrutura precisa vir imediatamente após um **if** ou um **else if**. Ela é executada quando a estrutura anterior não tem o seu código executado. Até esse ponto ela é igual ao **if else**. A diferença é que nem sempre o **if else** se será executado, porque ele tem uma condição. Se ela for verdadeira, ele é executado. Nesse ponto, ele é igual ao **if**. Então, podemos dizer que essa estrutura pega um pouco das duas estruturas. Inclusive, o nome dela indica isso.

```
double imc;
imc = Console.ReadLine();
if (imc < 18.5) {
    Console.WriteLine("Abaixo do peso");
}
else if (imc < 25) {
    Console.WriteLine("Peso normal");
}
else if (imc < 30) {
    Console.WriteLine("Sobrepeso");
}
else {
    Console.WriteLine("Obesidade");
}
```

Fluxograma:



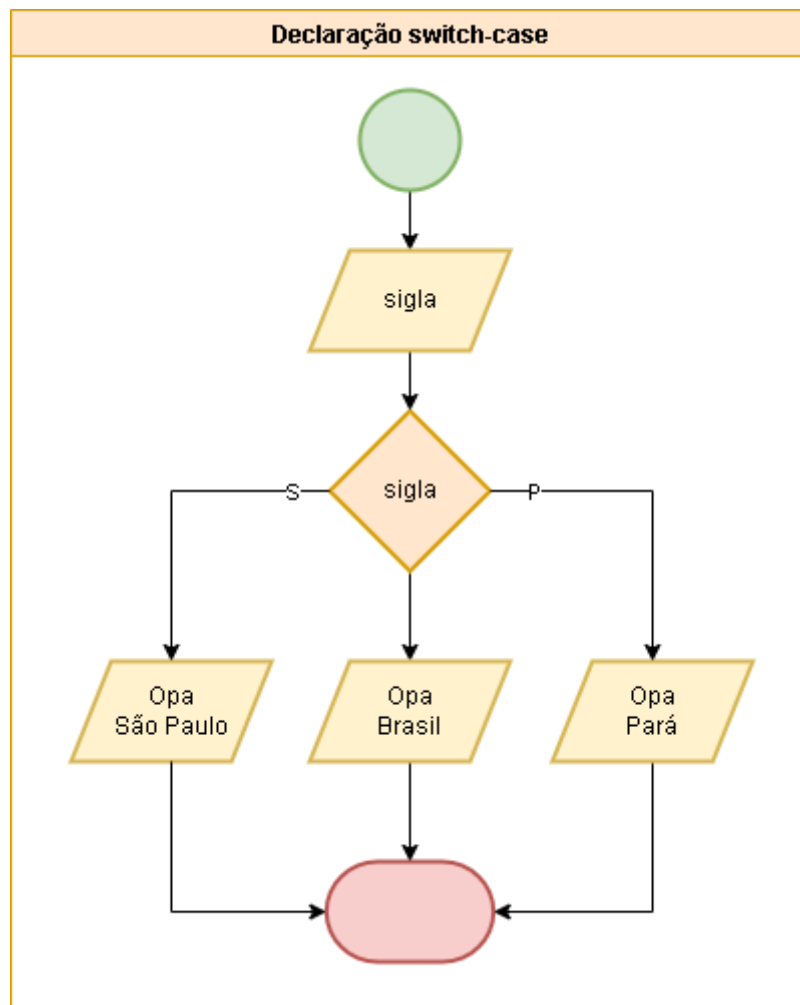
Estrutura switch-case

A instrução **switch** seleciona uma lista de instrução a ser executada com base em uma combinação de padrões, como mostra o exemplo a seguir:

```
char sigla
sigla = Console.ReadLine();
switch (sigla)
{
    case 'S':
        Console.WriteLine("Opa São Paulo");
        break;
    case 'P':
        Console.WriteLine("Opa Pará");
        break;
    default:
        Console.WriteLine("Opa Brasil");
        break;
}
```

O exemplo anterior também demonstra o default caso. O default caso especifica instruções a ser executadas quando uma expressão de match não corresponder a nenhum outro padrão de caso.

Fluxograma:



Para refinar seu conteúdo acesse a documentação do C#, segue o Link da [Documentação](#).