





Programação Orientada a Objetos

Ms. Eng. Claudio Ferreira de Carvalho



AULA 06

Operadores relacionais e lógicos – Estruturas de decisão

Estruturas de decisão

- ✓ Permitem decidir em função de parâmetros e instruções existentes no programa ou de interações com os usuários, executar diferentes grupos de comandos.
 - ✓ Como exemplo uma aluno pode ser considerado aprovado se o cálculo de sua média for igual superior a 7 ou reprovado se a média for inferior a 7
- ✓ O C#, assim como a maioria das linguagens de programação possui 3 estruturas de decisão:
 - ✓ If.....else
 - ✓ If.....else if
 - ✓ Switch.....case

Operadores relacionais e Operadores lógicos

- ✓ Operadores relacionais:
 - ✓ Permitem relacionar valores ou conteúdos

Descrição	Operador
Igual	==
Diferente	!=
Menor que	<
Maior que	>
Menor ou igual	<=
Maior ou igual	>=

- ✓ Operadores lógicos:
 - ✓ Utilizados para executar operações em estruturas de decisão

Operação	Operador
E	&&
OU	
Não	!

Estrutura if...Else

✓ Objetivo

- ✓ Executar o bloco de comandos que vem após a “condição”, quando essa “condição” for verdadeira. Caso a “condição” não seja verdadeira (seja falsa), serão executados os comandos após a palavra else.

✓ Sintaxe

```
if (condição)
```

```
{
```

```
    <comando>;
```

```
    <comando>;
```

```
    :
```

```
    :
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    <comando>;
```

```
    <comando>;
```

```
    :
```

```
    :
```

```
}
```

```
}
```

Comandos executados se a condição for verdadeira

Comandos executados se a condição não for verdadeira

Importante

Caso seja colocado somente um comando após a (condição) ou após o else, as chaves abrindo e fechando os comandos podem ser dispensadas.

É possível utilizar o comando If sem a clausula else, neste caso os comandos somente serão executados se a condição for verdadeira, caso contrário nenhum comando será executado.

Estrutura if...Else – Exemplo

```
1. namespace Aula06_Ex01
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara Variáveis
8.             double nota;
9.
10.            // Recebe a nota
11.            Console.Write("Forneça sua nota: ");
12.            nota = double.Parse(Console.ReadLine());
13.
14.            // Verifica se o usuário está aprovado ou em exame
15.            if (nota >= 7)
16.                Console.WriteLine("Você está aprovado");
17.            else
18.                Console.WriteLine("Você está em exame");
19.
20.            //Mantem a tela aberta esperando a digitação de uma tecla
21.            Console.ReadKey();
22.
23.        }
24.    }
25. }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto

Forneça sua nota: 9
Você está aprovado

Observar que se a nota for maior ou igual a 7 este comando será executado

Observar que se o programa chegou neste item a nota é menor que 7

Estrutura if com a clausula else if

✓ Objetivo

- ✓ Pode acontecer de existirem mais desmembramentos nas condições testadas por uma estrutura if. No exemplo dado, com média inferior a 7, mas superior a 3 o aluno tem direito a recuperação, e somente com nota inferior a 3 é que ele está reprovado.
- ✓ Então, nova condição deve ser testada. Para tanto, utiliza-se o comando else if.

✓ Sintaxe

```
if (condição)
{
    <comando>;
    <comando>;
    :
    :
}
else if (condição)
{
    <comando>;
    <comando>;
    :
    :
}
else
{
    <comando>;
    <comando>;
    :
    :
}
}
```

Comandos executados se a primeira condição if for verdadeira

Comandos executados se a segunda condição if for verdadeira

Comandos executados se nenhuma condição for verdadeira

Estrutura if com a clausula else if – Exemplo

```
1. namespace Aula06_Ex02
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara Variáveis
8.             double nota;
9.
10.            // Recebe a nota
11.            Console.WriteLine("Forneça sua nota: ");
12.            nota = double.Parse(Console.ReadLine());
13.
14.            if (nota >= 7)
15.                // Verifica se o usuário está aprovado
16.                Console.WriteLine("Você está aprovado");
17.
18.            else if (nota >= 3)
19.                // Verifica se o usuário está em recuperação
20.                // Observar que a nota com certeza não é maior ou igual a 7
21.                Console.WriteLine("Você está de recuperação");
22.
23.            else
24.                // Informa que o usuário está reprovado
25.                // Observar que a nota com certeza não é maior ou igual a 7
26.                // Nem maior ou igual a 5
27.                Console.WriteLine("Você reprovado");
28.
29.            //Mantem a tela aberta esperando a digitação de uma tecla
30.            Console.ReadKey();
31.        }
32.    }
33. }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2
Forneça sua nota: 6
Você está de recuperação

Observar que se a execução chegar no else if é porque o valor é menor que 7

Observar que se a execução chegar if é porque o valor é menor que 7 e menor que 3

Estrutura if com operadores lógicos

✓ Objetivo

- ✓ Pode haver necessidade de um teste ser feito dentro de um intervalo ou com intervalos alternativos. Neste caso torna-se necessário a utilização de operadores lógicos na clausula if, de maneira que mais de uma condição seja testada.

✓ Sintaxe

```
if ((condição1) && (condição2)) <=
{
    <comando>
    <comando>
    :
    :
}

if (condição3) || (condição4)<=
{
    <comando>
    <comando>
    :
    :
}
```

Neste caso os comandos após o if só serão executados se as duas condições forem verdadeiras

Neste caso os comandos após o if serão executados se qualquer uma das duas condições for verdadeira

Estrutura if com operadores lógicos – Exemplo

```
1. namespace Aula06_Ex03
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara Variáveis
8.             double nota;
9.
10.            // Recebe a nota
11.            Console.WriteLine("Forneça sua nota: ");
12.            nota = double.Parse(Console.ReadLine());
13.
14.            if ((nota < 0) || (nota > 10))
15.                // Executado se a nota for menor que zero ou maior que 10
16.                Console.WriteLine("Forneça um valor entre 0 e 10");
17.
18.            else if (nota >= 7)
19.                // Verifica se o usuário está aprovado
20.                Console.WriteLine("Você está aprovado");
21.
22.            else if (nota >= 3)
23.                // Verifica se o usuário está em recuperação
24.                Console.WriteLine("Você está de recuperação");
25.
26.            else
27.                // Informa que o usuário está reprovado.
28.                // Observar que neste caso, a nota.
29.                // não é menor que zero nem maior que 10.
30.                // não está entre 7 e 10 e
31.                // não está entre 3 e 7)
32.                Console.WriteLine("Você está reprovado");
33.
34.            //Mantem a tela aberta esperando a digitação de uma tecla
35.            Console.ReadKey();
36.
37.        }
38.    }
39. }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2

```
Forneça sua nota: 6
Você está de recuperação
```

Observar que este if (se a nota for menor que zero ou maior que 10) testa a digitação para evitar notas impossíveis. Caso ele seja executado o restante do programa não é executado.

Comandos ifs aninhados

✓ Objetivo

- ✓ Aninhar comandos consiste em colocar comandos dentro de comandos, no caso de “comandos ifs”, consiste em colocar “estruturas ifs” dentro de “estruturas ifs”..

✓ Sintaxe

```
if condição1)
{
    if (condição 2)
    {
        <comando>
        <comando>
        :
        :
    }
}
```

Observar que este if (com a condição 2) está dentro do outro if (com a condição 1). O if interno está aninhado ao externo.

Comandos ifs aninhados – Exemplo

```
1. namespace Aula06_Ex04
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara Variáveis
8.             double nota;
9.
10.            // Recebe a nota
11.            Console.WriteLine("Forneça sua nota: ");
12.            nota = double.Parse(Console.ReadLine());
13.
14.            if ((nota >= 0) && (nota <= 10))
15.            {
16.                // Executado se a nota for maior ou igual a zero
17.                // e menor ou igual a 10
18.                if (nota >= 7)
19.                {
20.                    // Verifica se o usuário está aprovado
21.                    Console.WriteLine("Você está aprovado");
22.
23.                else if (nota >= 3)
24.                {
25.                    // Verifica se o usuário está em recuperação
26.                    Console.WriteLine("Você está de recuperação");
27.
28.                else
29.                {
30.                    // Informa que o usuário está reprovado.
31.                    // Observar que neste caso, a nota.
32.                    // é menor que 7 e menor que 3.
33.                    Console.WriteLine("Você está reprovado");
34.                }
35.            }
36.            else
37.            {
38.                // Executado se a nota não estiver entre 0 e 10
39.                Console.WriteLine("Forneça um valor entre 0 e 10");
40.
41.            //Mantem a tela aberta esperando a digitação de uma tecla
42.            Console.ReadKey();
43.        }
44.    }
45. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto>
Forneça sua nota: 19
Forneça um valor entre 0 e 10
```

Teste de digitação com valores numéricos

✓ Objetivo

- ✓ Quando um usuário deve digitar um valor numérico este valor precisa ser recebido com o comando `Console.ReadLine()`.
- ✓ Como este comando só aceita valores string, a digitação do usuário deverá ser recebida, convertida para um valor numérico tal como byte, int, double.
- ✓ Se no momento da digitação o usuário colocar algum caractere especial ou letra, o programa apresentará um erro de runtime se for convertido diretamente com instruções do tipo `double.Parse`.
- ✓ Uma das maneiras de evitar este tipo de erro é receber os valores com instruções do tipo:

✓ Sintaxe

✓ `digi = double.TryParse(Console.ReadLine(), out valor);`

digi é do tipo bool, recebe true ou false

valor deve ser numérico. Recebe o número digitado

Teste de digitação com valores numéricos

```
1. namespace Aula06_Ex05
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara Variáveis
8.             double valor;
9.             bool digi;
10.
11.             // Recebe valores digitados no teclado
12.             // Se a digitação for numérica, atribui true para a variável digi
13.             // Se a digitação não for numérica, atribui false para a variável digi
14.             Console.WriteLine("Forneça um número: ");
15.             digi = double.TryParse(Console.ReadLine(), out valor);
16.
17.             // Testa se o valor digitado é numérico ou possui caracteres
18.             if (digi == false)
19.             {
20.                 Console.WriteLine("Você não digitou um número, então:");
21.             }
22.             else
23.             {
24.                 Console.WriteLine("Você digitou número, então:");
25.             }
26.             Console.WriteLine("O valor atribuído à variável digi (tipo bool) foi: " + digi);
27.             Console.WriteLine("O valor atribuído à variável valor (tipo double) foi: " + valor);
28.
29.             //Mantem a tela aberta esperando a digitação de uma tecla
30.             Console.ReadKey();
31.         }
32.     }
33. }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projetos_POO\Aula06

```
Forneça um número: A
Você não digitou um número, então:
O valor atribuído à variável digi (tipo bool) foi: False
O valor atribuído à variável valor (tipo double) foi: 0
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projetos_POO\Aula06

```
Forneça um número: 9
Você digitou número, então:
O valor atribuído à variável digi (tipo bool) foi: True
O valor atribuído à variável valor (tipo double) foi: 9
```


Comando switch....case

✓ Objetivo

- ✓ Deve-se dar preferência ao comando “switch.. .case”, quando existem mais que duas (várias) decisões a serem tomadas.
- ✓ Por exemplo deseja-se criar um programa para fornecer uma mensagem para cada dia da semana, como existem 7 opções possíveis, é melhor utilizar o comando “switch...case”, embora o programa também pudesse ser implementado com os comandos “if...else if...else”.

✓ Sintaxe

```
switch (valorEsperado)
{
    case valorTestado1:
        <comando>;
        break;
    case valorTestado2:
        <comando>;
        break;
    case valorTestado3:
        <comando>;
        break;
    :
    :
    :
}
```

Vantagens

A variável a ser testada é lida uma só vez;
Uma mesma condição ou condições pode ou podem ser utilizadas para diversas clausulas (exemplo será mostrado a seguir);

Desvantagens

Com o comando “switch...case” só é possível testar igualdade, ou seja, diferentemente da sintaxe do if e else...if não é possível verificar se o valor a ser comparado é maior, menor, diferente etc.
O comando “switch...case” só opera com variáveis Booleana (bool), Inteiras (byte, sbyte, short, etc) e caracteres (char e string). Não aceita variáveis reais tais como float e double etc.

Comando switch.....case - Exemplo

```
1. namespace Aula06_Ex06
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara variáveis
8.             char dia;
9.
10.            // Recebe o dia da semana
11.            Console.WriteLine("Qual dia da semana você deseja saber
cardápio? ");
12.            Console.WriteLine("Forneça 1 para domingo");
13.            Console.WriteLine("Forneça 2 para segunda feira");
14.            Console.WriteLine("Forneça 3 para terça feira");
15.            Console.WriteLine("Forneça 4 para quarta feira");
16.            Console.WriteLine("Forneça 5 para quinta feira");
17.            Console.WriteLine("Forneça 6 para sexta feira");
18.            Console.WriteLine("Forneça 7 para sábado");
19.            dia = char.Parse(Console.ReadLine());
20.            // Fornece o prato do dia
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Proje
Qual dia da semana você deseja saber cardápio?
Forneça 1 para domingo
Forneça 2 para segunda feira
Forneça 3 para terça feira
Forneça 4 para quarta feira
Forneça 5 para quinta feira
Forneça 6 para sexta feira
Forneça 7 para sábado
6
Sexta feira: Bacalhau
```

```
20. // Fornece o prato do dia
21. switch (dia)
22.     {
23.         case ('1'):
24.             Console.WriteLine("Domingo: Macarrão a Bolonhesa");
25.             break;
26.         case ('2'):
27.             Console.WriteLine("Segunda feira: Virado a Paulista");
28.             break;
29.         case ('3'):
30.             Console.WriteLine("Terça feira: Dobradinha");
31.             break;
32.         case ('4'):
33.             Console.WriteLine("Quarta feira: Feijoada");
34.             break;
35.         case ('5'):
36.             Console.WriteLine("Quinta feira: Rabada");
37.             break;
38.         case ('6'):
39.             Console.WriteLine("Sexta feira: Bacalhau");
40.             break;
41.         case ('7'):
42.             Console.WriteLine("Sábado: Feijoada especial");
43.             break;
44.     }
45.     // Mantem a tela aberta esperando a digitação de uma tecla
46.     Console.ReadKey();
47. }
48. }
49. }
```

Comando switch.....case com clausula default

✓ Objetivo

- ✓ Caso nenhum dos “cases” forem executados, devem ser executados os comandos existentes após a cláusula default.
- ✓ A cláusula default não é obrigatória, mas sempre que for utilizada deverá ser colocada após todos os cases.

✓ Sintaxe

```
switch (valorEsperado)
{
    case valorTestado1:
        <comando>;
        break;
    case valorTestado2:
        <comando>;
        break;
    case valorTestado3:
        <comando>;
        break;
    default:
        <comando>;
        Break;
}
```

Serão executados se nenhum caso for satisfeito

Comando switch....case com clausula default - Exemplo

```
1. namespace Aula06_Ex07
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             char dia;
8.             int dig;
9.
10.            Console.WriteLine("Dias da semana:");
11.            Console.WriteLine("1 => domingo.");
12.            Console.WriteLine("2 => segunda feira.");
13.            Console.WriteLine("3 => terça feria.");
14.            Console.WriteLine("4 => quarta feira.");
15.            Console.WriteLine("5 => quinta feira.");
16.            Console.WriteLine("6 => sexta feira.");
17.            Console.WriteLine("7 => sábado.");
18.            Console.Write("Qual dia da semana deseja saber o
cardápio: ");
19.            dig = Console.Read();
20.            dia = Convert.ToChar(dig);
21.
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projet
Dias da semana:
1 => domingo.
2 => segunda feira.
3 => terça feria.
4 => quarta feira.
5 => quinta feira.
6 => sexta feira.
7 => sábado.
Qual dia da semana deseja saber o cardápio: 9
Digite um número de 1 à 7
```

```
22. switch (dia)
23.     {
24.         case ('1'):
25.             Console.WriteLine("Domingo: Macarrão a
bolonhesa.");
26.             break;
27.         case ('2'):
28.             Console.WriteLine("Segunda: Virado a
paulista.");
29.             break;
30.         case ('3'):
31.             Console.WriteLine("Terça: Dobradinha.");
32.             break;
33.         case ('4'):
34.             Console.WriteLine("Quarta: Feijoada.");
35.             break;
36.         case ('5'):
37.             Console.WriteLine("Quinta: Rabada.");
38.             break;
39.         case ('6'):
40.             Console.WriteLine("Sexta: Bacalhau.");
41.             break;
42.         case ('7'):
43.             Console.WriteLine("Sábado: Feijoada
especial.");
44.             break;
45.         default:
46.             Console.WriteLine("Digite um número de 1 à
7");
47.             break;
48.     }
49.     // Mantem a tela aberta esperando a digitação de uma
tecla
50.     Console.ReadKey();
51.
52. }
53. }
54. }
```

Diversos cases devem executar os mesmos comandos

✓ Objetivo

- ✓ Uma cláusula case não pode testar mais que um valor, entretanto, caso se deseje que os mesmos comandos sejam executados para diversas clausulas cases, basta deixar os cases em sequência e sem comandos (inclusive sem break), atribuindo os comandos desejados somente ao último case da série. Desta maneira, será executado somente o case que tem comandos (que deve ser o último da sequência) e consequentemente um break ao final.
- ✓ Para entender este funcionamento, vamos supor que se deseja fazer um programa que fornece três mensagens para os 7 dias da semana, conforme a seguir:
- ✓ Para segunda e terça a mensagem é “Força a semana está começando”.
- ✓ Para quarta, quinta e sexta a mensagem é “Aguente, o fim de semana já vai chegar”.
- ✓ Para sábado e domingo a mensagem é “Que bom, estamos no fim de semana”.
- ✓ Nota: Neste exemplo está sendo utilizado também o comando ToLower() que converte letras maiúsculas para letra minúsculas.

Comando switch.....case com clausula default - Exemplo

```
1. namespace Aula06_Ex08
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de variáveis
8.             char dia;
9.             // Recebe o dia da semana
10.            Console.WriteLine("Que dia da semana é hoje? ");
11.            Console.WriteLine("Forneça s para segunda feira");
12.            Console.WriteLine("Forneça t para terça feira");
13.            Console.WriteLine("Forneça q para quarta feira");
14.            Console.WriteLine("Forneça i para quinta feira");
15.            Console.WriteLine("Forneça e para sexta feira");
16.            Console.WriteLine("Forneça a para sábado");
17.            Console.WriteLine("Forneça d para domingo");
18.            dia = char.Parse(Console.ReadLine().ToLower());
19.        }
20.    }
21. }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_202

```
Que dia da semana é hoje?
Forneça s para segunda feira
Forneça t para terça feira
Forneça q para quarta feira
Forneça i para quinta feira
Forneça e para sexta feira
Forneça a para sábado
Forneça d para domingo
e
Aguente, o fim de semana já vai chegar
```

```
20. // Fornece a mensagem
21.     switch (dia)
22.     {
23.         case 's':
24.         case 't':
25.             Console.WriteLine("Força a semana está começando");
26.             break;
27.         case 'q':
28.         case 'i':
29.         case 'e':
30.             Console.WriteLine("Aguente, o fim de semana já vai chegar");
31.             break;
32.         case 'a':
33.         case 'd':
34.             Console.WriteLine("Que bom, estamos no fim de semana");
35.             break;
36.         default:
37.             Console.WriteLine("Por favor digite s,t,q,i,e,a ou d");
38.             break;
39.     }
40.     // Mantem a tela aberta esperando a digitação de uma tecla
41.     Console.ReadKey();
42.
43. }
44.
45. }
```


