

Lógica de Programação

Unidade 10 – Operadores Lógicos



QI ESCOLAS E FACULDADES
Curso Técnico em Informática

SUMÁRIO

INTRODUÇÃO	3
OPERADORES LÓGICOS	3
UTILIZANDO OPERADORES LÓGICOS	3
<i>Exemplos de Condicionais</i>	<i>3</i>
TABELA VERDADE.....	4
EXEMPLO DE SINTAXE COM OPERADORES LÓGICOS	5
<i>Diagrama de Classe.....</i>	<i>5</i>
<i>Sintaxe da Classe Numero.....</i>	<i>6</i>
<i>Classe Main</i>	<i>6</i>
<i>Resultado no terminal.....</i>	<i>6</i>

INTRODUÇÃO

Existe a possibilidade de que uma expressão lógica faça mais de uma pergunta ao mesmo tempo. De modo que essas perguntas têm de estar relacionadas de alguma forma. Um exemplo é sua aprovação em qualquer disciplina da escola, para que você seja aprovado na disciplina você precisa ter uma média e uma frequência mínimas, apenas a média não basta. Para isso utilizaremos os operadores lógicos.

OPERADORES LÓGICOS

Para que possamos fazer mais de uma pergunta lógica ao mesmo tempo para o processador, devemos sempre utilizar um dos operadores abaixo.

Operador	Significado
&&	e (multiplicação lógica)
	ou (adição lógica)
!	não (inversor lógico)

Utilizando Operadores Lógicos

Iremos utilizar operadores lógicos dentro das condicionais e dentro de loops.

Exemplos de Condicionais

***Para um aluno ser aprovado sua média e frequência devem ser mínimas.**

```
if(this.media>=6 && this.frequencia>=75){
    return "Aprovado";
}
```

No exemplo acima vemos que um aluno será aprovado somente se tiver uma média maior ou igual a 6 **e** uma frequência maior ou igual a 75, qualquer uma das situações sendo contrária, o aluno é reprovado.

***Verificando se o número é par positivo:**

```
if(this.numero>0 && this.numero%2==0){
    return "Par Positivo";
}
```

No exemplo acima vemos que um número só será Par e Positivo se ele for maior que 0 e divisível por 2, qualquer uma das situações sendo contrária, o número não será considerado um “Par Positivo”.

* Verificando se o número está entre 1 e 7:

```
if(this.numero>=1 && this.numero<=7){
    return "O número está no intervalo de 1 à 7.";
}
```

*Verificando se o número está fora do intervalo de 1 à 7:

```
if(this.numero<1 || this.numero>7){
    return "O número está fora do intervalo de 1 à 7.";
}
```

No exemplo acima constamos que para estabelecer que o número esteja fora do intervalo de 1 a 7, foi utilizado o || (ou), pois o número para estar fora do intervalo, ou ele é menor que 1 ou maior que 7.

TABELA VERDADE

Para podermos conferir se o algoritmo está executando corretamente, retornando as respostas certas, temos que entender como funciona o programa. Utilizamos a tabela verdade para testar as condicionais do nosso programa. Testar quando será exibida uma resposta ou outra.

A	B	A&&B	A B	!A	!B
V	V	V	V	F	F
F	V	F	V	V	F
V	F	F	V	F	V
F	F	F	F	V	V

Para entender a tabela, imagine que o A seja uma condição (um teste), e o B outra.

Exemplo 1 - &&:

A
B

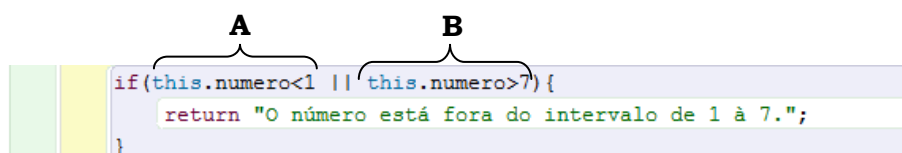
```
if(this.media>=6 && this.frequencia>=75){
    return "Aprovado";
}
```

Neste exemplo o A é a resposta da condição da média do aluno e B resposta da condição de frequência. Para que o aluno seja aprovado ambos devem retornar verdadeiro, ou seja, a média tem que ser maior igual a 6 e a frequência tem que ser

maior igual a 75. Se um aluno obtiver média 8 e frequência 90, por exemplo, esse será aprovado.

Portanto: V (verdade) && V (verdade) retorna **V** (verdade), qualquer uma das condições sendo F (falsa) o aluno é Reprovado.

Exemplo 2 - ||:



```

    A      B
    {      {
if (this.numero < 1 || this.numero > 7) {
    return "O número está fora do intervalo de 1 à 7.";
}
    
```

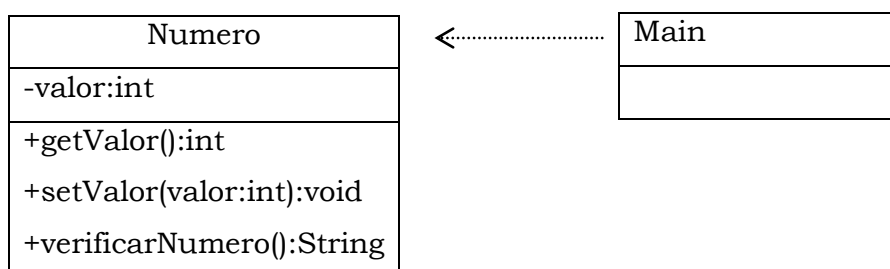
Neste outro exemplo o A é a resposta da condição do número menor que 1 e B resposta da condição maior que 7. Para que retorne a resposta “O número está fora do intervalo de 1 a 7, basta o A ou o B retornarem verdade.

Portanto no ||, tendo uma das condições retornando **V** (verdade), o resultado final será V.

Exemplo de Sintaxe com Operadores Lógicos

Enunciado: Desenvolva um programa que leia um número e retorne se o mesmo é PAR POSITIVO, PAR NEGATIVO, ÍMPAR POSITIVO, ÍMPAR NEGATIVO OU NULO.

Diagrama de Classe



Sintaxe da Classe Numero

```

1 public class Numero{
2     private int valor;
3
4     public int getValor(){
5         return this.valor;
6     }
7     public void setValor(int valor){
8         this.valor = valor;
9     }
10
11     public String verificarNumero(){
12         if(this.valor%2==0 && this.valor>0){
13             return "PAR POSITIVO";
14         }else if(this.valor%2==0 && this.valor<0){
15             return "PAR NEGATIVO";
16         }else if(this.valor%2!=0 && this.valor>0){
17             return "ÍMPAR POSITIVO";
18         }else if(this.valor%2!=0 && this.valor<0){
19             return "ÍMPAR NEGATIVO";
20         }else{
21             return "NULO";
22         }
23     }
24 }

```

Classe Main

```

1 import java.util.Scanner;
2 public class Main{
3     public static void main(String args[]){
4         Scanner ler = new Scanner (System.in);
5         Numero n1 = new Numero();
6
7         System.out.println("Digite um número: ");
8         n1.setValor(ler.nextInt());
9
10        System.out.println(n1.verificarNumero());
11    }
12 }

```

Resultado no terminal

