

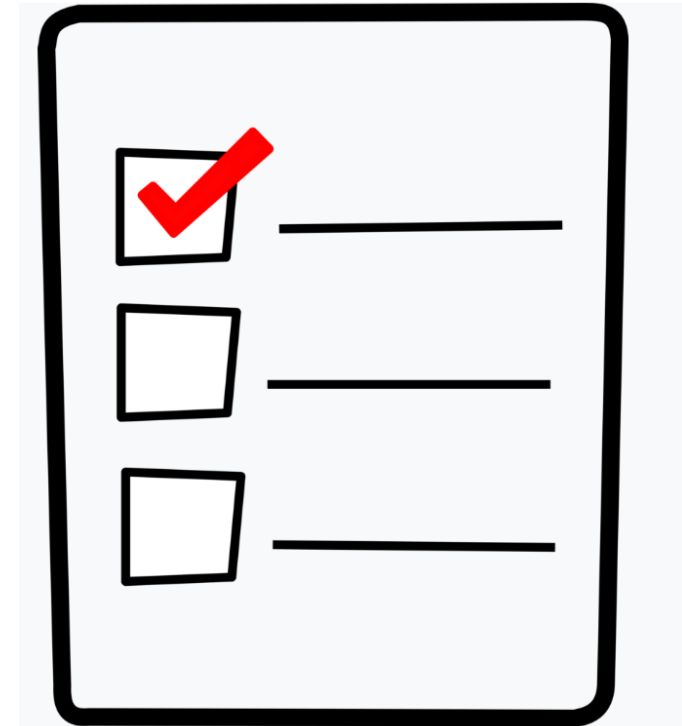
Programa de Mentoria em Lógica de Programação e Algoritmos

Mentores

Amanda Guerra, Anna Ladewig, Lícia Verêda e Raphael Silva.

Programação

- Operadores Relacionais
- Tabela-Verdade
- Operadores Lógicos
- Prioridade dos Operadores
- Operadores de Incremento e Decremento
- Exercícios



Operadores Relacionais

Operador	Nome	Exemplo	Resultado
==	Igual	x == 10	
!=	Diferente	3 != 2	true
<	Menor	10 < 18	true
>	Maior	10 > 18	false
>=	Maior ou igual	3 >= 3	true
<=	Menor ou igual	7 <= 6	false

Operadores Relacionais

Operador	Nome	Exemplo	
==	Igual	x == 10	
!=	Diferente	3 != 2	
<	Menor	10 < 18	true
>	Maior	10 > 18	false
>=	Maior ou igual	3 >= 3	true
<=	Menor ou igual	7 <= 6	false

Não confundir
= Atribuição
== Igualdade

Expressões Lógicas

O uso dos operadores relacionais permite a criação de **expressões lógicas**, usadas em **condições** para o computador tomar decisões com base em tipos primitivos de dados.

Uma expressão lógica deve ter ao menos um operador relacional e resulta sempre em um valor lógico (**verdadeiro ou falso**).

Condição	Resultado
Verificar se o número é positivo.	<code>numero > 0</code>
Verificar se o resultado é par.	<code>resultado % 2 == 0</code>
Verificar se uma altura é igual ou superior a 1,80m.	<code>altura >= 1.8</code>

Operadores Relacionais

Uso desses operadores:

Eles são bastante usados nas **estruturas condicionais** e de repetição.

As **condições** muitas vezes são **expressões lógicas**. Exemplo: **(idade >= 18)**

```
package MentorLogicaDeProgramação;  
  
public class EstruturaIfElse {  
    public static void main(String[] args) {  
        if (condition) {  
            // ...  
        } else if (condition) {  
            // ...  
        } else {  
            // ...  
        }  
    }  
}
```

```
package MentorLogicaDeProgramação;  
  
public class EstruturaWhile {  
    public static void main(String[] args) {  
        while (condition) {  
            // ...  
        }  
    }  
}
```

O uso dessas estruturas será assunto dos próximos encontros.

Tabela-Verdade

Tabela do “E”

Simbologia: **&&**

Só dá verdadeiro se **todas** as proposições são verdadeiras.

P	Q	P && Q
V	V	V
V	F	F
F	V	F
F	F	F

Tabela-Verdade

Tabela do “E”

Simbologia: **&&**

Só dá verdadeiro se **todas** as proposições são verdadeiras.

P	Q	P && Q
V	V	V
V	F	F
F	V	F
F	F	F

Tabela-Verdade

Tabela do “OU”

Simbologia: \vee

Só dá falso se **todas** as proposições são falsas.

P	Q	$P \vee Q$
V	V	V
V	F	V
F	V	V
F	F	F

Tabela-Verdade

Tabela do “OU”

Simbologia: \vee

Só dá falso se **todas** as proposições são falsas.

P	Q	$P \vee Q$
V	V	V
V	F	V
F	V	V
F	F	F

Tabela-Verdade

Tabela do “Não”

Simbologia: !

É a negação da proposição.
Então você **troca o valor** da proposição.

P	!P
V	F
F	V

Operadores Lógicos

Operador	Nome	Exemplo	Resultado
&&	E	(2 > 0) && (10 < 5)	false
 	OU	(10 < 11) (10 > 12)	true
!	NÃO	! (1 == 4)	true

Operadores Lógicos

Uso desses operadores:

Eles são muito usados nas **estruturas condicionais e de repetição**.

A **condição** tem que ser **VERDADEIRA** e por isso a importância de saber a tabela-verdade

```
package MentoriaLogicaDeProgramação;  
  
public class EstruturaIfElse {  
    public static void main(String[] args) {  
        if (condition) {  
        } else if (condition) {  
        } else {  
        }  
    }  
}
```

```
package MentoriaLogicaDeProgramação;  
  
public class EstruturaWhile {  
    public static void main(String[] args) {  
        while (condition) {  
        }  
    }  
}
```

O uso dessas estruturas será assunto dos próximos encontros.

Prioridade dos Operadores

Prioridade	Operador
1º	Aritmético
2º	Relacional
3º	Não
4º	E
5º	OU

Para alterar a prioridade são utilizados **parênteses “()”**.
Não podem ser usados colchetes e chaves.

Prioridade dos Operadores

$4 + 1 < 7 \ \&\& \ 9 - 5 < 3$ | $6 > 10 - 5 \ || \ 5 > 3 * 5 \ \&\& \ 12 / 2 > 8$ | $5 < 7 \ \&\& \ !(6 * 2 < 4 + 10 \ || \ 7 > 10)$

$4 + 1 < 7$ $\ \&\& \$ $9 - 5 < 3$

$5 < 7 \ \&\& \ 4 < 3$

$V \ \&\& \ F$

FALSO

Prioridade dos Operadores

$4 + 1 < 7 \ \&\& \ 9 - 5 < 3$ | $6 > 10 - 5 \ || \ 5 > 3 * 5 \ \&\& \ 12 / 2 > 8$ | $5 < 7 \ \&\& \ !(6 * 2 < 4 + 10 \ || \ 7 > 10)$

$4 + 1 < 7$ $\ \&\& \$ $9 - 5 < 3$ | $6 > 10 - 5$ $\ || \$ $5 > 3 * 5$ $\ \&\& \$ $12 / 2 > 8$ |

$5 < 7 \ \&\& \ 4 < 3$

$6 > 5 \ || \ 5 > 15 \ \&\& \ 6 > 8$

$V \ \&\& \ F$

$V \ || \ F \ \&\& \ F$

FALSO

$V \ || \ F$

VERDADEIRO

Note que, nesse exemplo, se você não usasse a ordem de precedência, o resultado daria FALSO.

Prioridade dos Operadores

$4 + 1 < 7 \ \&\& \ 9 - 5 < 3$	$6 > 10 - 5 \ \ 5 > 3 * 5 \ \&\& \ 12 / 2 > 8$	$5 < 7 \ \&\& \ !(6 * 2 < 4 + 10 \ \ 7 > 10)$
<u>$4 + 1 < 7$</u> $\ \&\& \ $ <u>$9 - 5 < 3$</u>	$6 > $ <u>$10 - 5$</u> $ \ \ $ $5 > $ <u>$3 * 5$</u> $ \ \&\& \ $ <u>$12 / 2 > 8$</u>	$5 < 7 \ \&\& \ !($ <u>$6 * 2 < 4 + 10$</u> $ \ \ $ <u>$7 > 10$</u> $)$
$5 < 7 \ \&\& \ 4 < 3$	$6 > 5 \ \ 5 > 15 \ \&\& \ 6 > 8$	$5 < 7 \ \&\& \ !(12 < 14 \ \ 7 > 10)$
$V \ \&\& \ F$	$V \ \ F \ \&\& \ F$	$V \ \&\& \ !(V \ \ F)$
FALSO	$V \ \ F$	$V \ \&\& \ !(V)$
	VERDADEIRO	$V \ \&\& \ F$
		FALSO

Note que, nesse exemplo, se você não usasse a ordem de precedência, o resultado daria FALSO.

Operadores de Incremento e Decremento

Operador	Chamado de	Expressão	Explicação
++	pré-incremento	++a	Incrementa “a” em 1. Depois utiliza o novo valor de “a” na expressão em que “a” reside.
++	pós-incremento	a++	Utiliza o valor atual de “a” na expressão em que “a” reside, depois incrementa “a” um 1.
--	pré-decremento	-- a	Decrementa “a” em 1, depois utiliza o novo valor de “a” na expressão em que “a” reside.
--	pós-decremento	a --	Utiliza o valor atual de “a” na expressão em que “a” reside, depois decrementa “a” em 1.

Operadores de Incremento e Decremento

Exemplos em expressões:

`int a, resultado;`

`a = 4;`

`resultado = 3 * (++a);`

Qual o valor em a?

Qual o valor em resultado?

`int a, resultado;`

`a = 4;`

`resultado = 3 * (a++);`

Qual o valor em a?

Qual o valor em resultado?

Operadores de Incremento e Decremento

Exemplos em expressões:

```
int a, resultado;
```

```
a = 4;
```

```
resultado = 3 * (++a);
```

Qual o valor em a? **5**

Qual o valor em resultado? **15**

```
int a, resultado;
```

```
a = 4;
```

```
resultado = 3 * (a++);
```

Qual o valor em a? **5**

Qual o valor em resultado? **12**

Operadores de Incremento e Decremento

Uso desses operadores:

Nas **estruturas de repetição**, eles são muito usados como **contadores**.

contador++ é o mesmo que **contador = contador + 1**

```
package MentoriaLogicaDeProgramação;

public class EstruturaWhile {

    public static void main(String[] args) {

        int contador = 0;
        boolean condition = true;

        while (condition) {

            contador++;

        }

    }

}
```

```
package MentoriaLogicaDeProgramação;

public class EstruturaFor {

    public static void main(String[] args) {

        for (int i = 0; i < args.length; i++) {

        }

    }

}
```

O uso dessas estruturas será assunto dos próximos encontros.

Exercícios

Faça um programa que leia o nome de 3 candidatos e a quantidade de votos recebidos por cada um dos candidatos. Após, de acordo com a quantidade de votos, diga se haverá ou não o 2º turno.

>> Para não haver 2º turno, um dos candidatos deve ter mais de 50% dos votos. <<

```
package EstruturaDeDecisao;

import java.util.Scanner;

public class Exercício02 {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);

        String candidato1, candidato2, candidato3;
        int votos1, votos2, votos3;

        System.out.println("Qual o nome do 1º candidato? ");
        candidato1 = teclado.next();
        System.out.println("Qual a quantidade de votos do 1º candidato? ");
        votos1 = teclado.nextInt();

        System.out.println("Qual o nome do 2º candidato? ");
        candidato2 = teclado.next();
        System.out.println("Qual a quantidade de votos do 2º candidato? ");
        votos2 = teclado.nextInt();

        System.out.println("Qual o nome do 3º candidato? ");
        candidato3 = teclado.next();
        System.out.println("Qual a quantidade de votos do 3º candidato? ");
        votos3 = teclado.nextInt();

        if (votos1 > votos2 + votos3 || votos2 > votos1 + votos3 || votos3 > votos1 + votos2) {
            System.out.printf("Não haverá segundo turno.");
        } else {
            System.out.println("Haverá segundo turno.");
        }
        teclado.close();
    }
}
```

```
package EstruturaDeDecisao;

import java.util.Scanner;

public class Exercício02 {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);

        String candidato1, candidato2, candidato3;
        int votos1, votos2, votos3;

        System.out.println("Qual o nome do 1º candidato? ");
        candidato1 = teclado.next();
        System.out.println("Qual a quantidade de votos do 1º candidato? ");
        votos1 = teclado.nextInt();
```

```
(votos1 > votos2 + votos3 || votos2 > votos1 + votos3 || votos3 > votos1 + votos2)
System.out.printf("Não haverá segundo turno.");
```

```
        System.out.println("Qual o nome do 3º candidato? ");
        candidato3 = teclado.next();
        System.out.println("Qual a quantidade de votos do 3º candidato? ");
        votos3 = teclado.nextInt();

        if (votos1 > votos2 + votos3 || votos2 > votos1 + votos3 || votos3 > votos1 + votos2) {
            System.out.printf("Não haverá segundo turno.");
        } else {
            System.out.println("Haverá segundo turno.");
        }
        teclado.close();
    }
}
```


Exercícios

Faça um programa que receba a data de nascimento (dd/mm/aaaa) de uma pessoa e faça uma crítica para que:

- só seja aceito até 31 dias;
- só sejam aceitos 12 meses;
- não seja aceito o dia 31, nos meses que só tenham 30 dias;
- no mês de fevereiro, não sejam aceitos os dias 30 e 31.

>> Meses que só tem 30 dias: abril, junho, setembro e novembro. <<

```
package MentoriaLogicaDeProgramação;

import java.util.Scanner;

public class MesComCritica {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);

        String data = "";
        int diaInt, mesInt, anoInt;

        do {
            System.out.println("Digite a data de seu nascimento (dd/mm/aaaa): ");
            data = teclado.next();

            diaInt = Integer.parseInt(data.substring(0,2));
            mesInt = Integer.parseInt(data.substring(3,5));
            anoInt = Integer.parseInt(data.substring(6));

            if (diaInt < 1 || diaInt > 31) {
                System.out.println("\nDia inválido. Tente Novamente.\n");
                continue;
            }
            if (mesInt < 1 || mesInt > 12) {
                System.out.println("\nMês inválido. Tente Novamente.\n");
                continue;
            }
            if (diaInt == 31 && (mesInt == 4 || mesInt == 6 || mesInt == 9 || mesInt == 11)) {
                System.out.println("\nDia inválido. Tente Novamente.\n");
                continue;
            }
            if (mesInt == 02 && (diaInt == 30 || diaInt == 31)) {
                System.out.printf("\nDia inválido. Tente Novamente.\n");
                continue;
            }
            break;
        } while (true);

        teclado.close();
    }
}
```

```
package MentoriaLogicaDeProgramação;
```

```
import java.util.Scanner;
```

```
(diaInt < 1 || diaInt > 31) {  
    System.out.println("\nDia inválido. Tente Novamente.\n")
```

```
    String data = "";  
    int diaInt, mesInt, anoInt;
```

```
(mesInt < 1 || mesInt > 12) {  
    System.out.println("\nMês inválido. Tente Novamente.\n")
```

```
    diaInt = Integer.parseInt(data.substring(0,2));  
    mesInt = Integer.parseInt(data.substring(3,5));  
    anoInt = Integer.parseInt(data.substring(6));
```

```
    if (diaInt < 1 || diaInt > 31) {
```

```
(diaInt == 31 && (mesInt == 4 || mesInt == 6 || mesInt == 9 || mesInt == 11))  
    System.out.println("\nDia inválido. Tente Novamente.");
```

```
        continue;
```

```
    }
```

```
    if (diaInt == 31 && (mesInt == 4 || mesInt == 6 || mesInt == 9 || mesInt == 11)) {  
        System.out.println("\nDia inválido. Tente Novamente.\n");
```

```
(mesInt == 02 && (diaInt == 30 || diaInt == 31)) {  
    System.out.printf("\nDia inválido. Tente Novamente.");
```

```
    }
```

```
    break;
```

```
    } while (true);
```

```
    teclado.close();
```

```
    }
```

```
}
```

Exercícios

Treino em expressões lógicas

Situações	Expressão Lógica	Entrada	Resultado	
Fazer uma crítica para <u>não</u> receber um número que não seja equivalente a um mês do ano.	nMes < 1 nMes > 12	nMes = 7;	falso	Não vai ter mensagem de erro Vai ter mensagem de erro
		nMes = 15;	verdadeiro	
Analisar se é o primeiro dia do mês e se a temperatura está negativa.	diaMes == 1 && temp < 0	diaMes = 1; temperatura = 6.0f;	falso	
Ver se o ano é do século 20. (De 1901 até 2000)	ano >= 1901 && ano <= 2000	ano = 1980;	verdadeiro	
Verificar se é o primeiro dia de uma quinzena.	nDia == 1 nDia == 15	nDia = 16;	falso	

Exercícios

Treino em expressões lógicas

Situações	Expressão Lógica	Entrada	Resultado
Ver se o número do dia da semana corresponde a um dia útil.	<code>nDia >= 2 && numDia <= 6</code>	<code>nDia = 7;</code>	falso
Analisar se a hora e os minutos informados são válidos.	<code>hora >=0 && hora <= 23 && minuto >= 0 && minuto <=59</code>	<code>hora = 24; min = 0;</code>	falso
Verificar se o número do dia corresponde a um final de semana em um mês par.	<code>(dia == 1 dia == 7) && mês % 2 == 0</code>	<code>dia = 6; mes = 10;</code>	falso

Obrigada!



Anna Maria Ladewig



/annaladewig



/annaladewig