

The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by a thin black border, with dark gray horizontal bars at the top and bottom.

ORACLE

Academy

Java Foundations

5-2

Entendendo a Execução Condicional

ORACLE
Academy



Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

Objetivos

- Esta lição abrange os seguintes objetivos:
 - Descrever a execução condicional
 - Descrever operadores lógicos
 - Entender a avaliação de operadores lógicos de “curto-circuito”
 - Criar construções if encadeadas



Quando Várias Condições se Aplicam

- E se determinada ação precisar ser tomada apenas se várias condições forem verdadeiras?
- Considere o cenário em que um aluno estará qualificado a receber uma bolsa de estudos se as duas condições a seguir forem atendidas:
 - Nota ≥ 88
 - Número de faltas = 0

Tratando Várias Condições

- Operadores relacionais são uma boa opção quando você está verificando uma única condição
- Você pode usar uma sequência de instruções if para testar mais de uma condição

```
if (grade >= 88) {  
    if (numberDaysAbsent == 0) {  
        System.out.println("Você está qualificado para uma "  
                           + "bolsa de estudo.");  
    }  
}  
}
```

Tratando Várias Condições Exemplo

- Como é demonstrado no exemplo:
 - A sequência de instruções if é difícil de ser escrita, mais difícil de ser lida e torna-se ainda mais difícil quando você adiciona mais condições
 - Felizmente, o Java tem uma maneira fácil de tratar várias condições: operadores lógicos

Operadores Lógicos Java

- Você pode usar três operadores lógicos Java para combinar várias expressões booleanas em uma única expressão booleana

Operador Lógico	Significado
&&	E
	OU
!	NÃO

Três Operadores Lógicos

Operação	Operador	Exemplo
Se uma condição E outra condição	&&	<code>int i = 2;</code> <code>int j = 8;</code> <code>((i < 1) && (j > 6))</code>
Se uma das condições OU as duas condições	 	<code>int i = 2;</code> <code>int j = 8;</code> <code>((i < 1) (j > 10))</code>
NÃO	!	<code>int i = 2;</code> <code>!(i < 3)</code>

A tabela no slide lista os operadores lógicos na linguagem de programação Java. Todos os exemplos geram um resultado booleano falso.

Aplicando Operadores Lógicos

- Você pode escrever o exemplo anterior usando o operador lógico AND como:

```
grade >= 88 && numberDaysAbsent == 0
```

Expressão
Booliana 1

Operador
Lógico

Expressão
Booliana 2

- O operador lógico permite que você teste várias condições mais facilmente, e o código é mais legível

Neste exemplo, você usa o operador lógico `&&` porque as duas expressões booleanas devem ser verdadeiras para tornar o aluno qualificado a receber uma bolsa de estudos.

Operador lógico `&&`:

- A condição combinada será verdadeira se e apenas se as duas expressões booleanas forem verdadeiras.
- A condição combinada será falsa se uma ou as duas expressões booleanas forem falsas.

Operador Lógico E: Exemplo

```
public static void main(String[] args) {  
    int numberDaysAbsent = 0;  
    int grade = 95;  
    if (grade >= 88 && numberDaysAbsent == 0) {  
        System.out.println("Você está qualificado para uma "  
            + " bolsa de estudos.");  
    }  
    else {  
        System.out.println("Você não está qualificado para uma "  
            + "bolsa de estudos.");  
    }  
} //fim if  
} //fim do método main
```

Será avaliado como verdadeiro se as duas expressões booleanas forem verdadeiras

Este exemplo ilustra o operador lógico E. Para que a saída seja exibida como "Você está qualificado para uma bolsa de estudos", as duas condições devem ser verdadeiras. Ou seja: a nota deve ser maior ou igual a 88, e o número de faltas deve ser igual a zero.

Operadores Lógicos OU

- Considere o cenário em que um aluno estará qualificado para participar de uma equipe esportiva se uma destas condições for atendida:
 - Nota ≥ 70
 - Número de faltas < 5
- Neste caso, você pode usar o operador lógico OU para unir as várias expressões booleanas

```
grade >=70 || numberDaysAbsent < 5
```

Expressão
Booleana 1

Operador
Lógico

Expressão
Booleana 2

A condição combinada será verdadeira se uma ou as duas expressões booleanas forem verdadeiras.

A condição combinada será falsa se as duas expressões booleanas forem falsas.

Operadores Lógicos OU: Exemplo

```
public static void main(String[] args) {  
    int numberDaysAbsent = 3;  
    int grade = 85;  
    if (grade >= 70 || numberDaysAbsent < 5) {  
        System.out.println("Você está qualificado para uma"  
            + " equipe esportiva");  
    }  
    else {  
        System.out.println("Você não está qualificado para"  
            + " uma equipe esportiva");  
    }  
} //fim if  
} //fim do método main
```

Será avaliada como verdadeira se uma das expressões booleanas for verdadeira

Este exemplo ilustra como usar o operador lógico OU. Neste exemplo, a saída "Você está qualificado para uma equipe esportiva" será exibida mesmo se uma das condições for verdadeira. Ou seja, a nota precisa ser ≥ 70 ou o número de faltas deve ser menor que cinco.

Operadores Lógicos NÃO

- Considere o cenário em que um aluno estará qualificado para monitoria gratuita se as duas condições a seguir forem atendidas:
 - Nota < 88
 - Número de faltas < 3
- Use o operador lógico !

```
!madeFreeTutor && numberDaysAbsent < 3
```

Operador
Lógico

Expressão
Booliana 1

Expressão
Booliana 2

Operadores Lógicos NÃO

```
public static void main(String args[]) {  
    int numberDaysAbsent = 2;  
    int grade = 65;  
    boolean madeFreeTutor = grade >= 88;  
    if (!madeFreeTutor && numberDaysAbsent < 3) {  
        System.out.println("Você está qualificado a receber"  
            + " monitoria gratuita");  
    }  
}  
}
```

Este exemplo ilustra o operador lógico !. Como a nota é igual a 65, `!madeFreeTutor` é verdadeiro porque `madeFreeTutor` é falso.

A expressão combinada é avaliada como verdadeira e a seguinte saída é exibida: "Você está qualificado a receber monitoria gratuita."

Exercício 1

- Crie um novo projeto e adicione o arquivo `WatchMovie.java` a ele
- Modifique `WatchMovie.java` para assistir um filme que atenda às duas condições a seguir:
- O preço do ingresso é maior ou igual a US\$ 12
- A classificação do filme é igual a 5
 - Exiba a saída como **"Estou interessado em assistir o filme."**
 - Exiba também a saída como **"Não estou interessado em assistir o filme."**

Ignorando o Segundo Teste E

- Os operadores && e || são operadores de curto-circuito
- Se a primeira expressão (à esquerda) for falsa, não haverá necessidade de calcular a segunda (à direita)

```
b = (x != 0) && ((y / x) > 2);
```

Expressão à Esquerda Expressão à Direita

Essa avaliação é feita da esquerda para a direita e é interrompida assim que o resultado é conhecido. Isso significa que a expressão à direita não será calculada se não for necessário.

Ignorando o Segundo Teste E

```
b = (x != 0) && ((y / x) > 2);
```

Expressão à
Esquerda

Expressão
à Direita

- Se x for 0, então (x != 0) será falso
- Para o operador &&, como não importa se ((y/x)>2) é true ou false, o resultado dessa expressão é false
- Então, o Java não continua a calcular ((y/x)>2)

Essa avaliação é feita da esquerda para a direita e é interrompida assim que o resultado é conhecido. Isso significa que a expressão à direita não será calculada se não for necessário.

Ignorando o Segundo Teste OU

- Se a primeira expressão (à esquerda) for true, não haverá necessidade de calcular a segunda (à direita)
- Considere este exemplo:

```
boolean b = (x <= 10) || (x > 20);
```

Expressão
à Esquerda

Expressão
à Direita

- Se $(x \leq 10)$ for verdadeiro, $(x > 20)$ não será calculado porque não importa se $(x > 20)$ é verdadeiro ou falso
- O resultado desta expressão será true

O que É um Operador Condicional Ternário?

Operação	Operador	Exemplo
Se a condição for verdadeira, atribua resultado = valor1 Caso contrário, atribua resultado = valor2 Observação: valor1 e valor2 devem ser do mesmo tipo de dados	?:	resultado = condição ? valor1 : valor2 Exemplo: <code>int x = 2, y = 5, z = 0;</code> <code>z = (y < x) ? x : y;</code>

Instruções equivalentes

```
z = (y < x) ? x : y;
```

```
if(y<x){  
    z=x;  
}  
else{  
    z=y;  
}//endif
```

ORACLE
Academy

JFo 5-2
Entendendo a Execução Condicional

Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

19

O operador ternário é um operador condicional que requer três operandos. Ele tem uma sintaxe mais compacta que uma instrução `if/else`.

Use o operador ternário, em vez de uma instrução `if/else`, se quiser tornar seu código mais curto. Existem três operandos no exemplo do slide:

- `(y < x)` : Essa expressão booliana (condição) está sendo avaliada.
- `? x` : Se `(y < x)` for verdadeiro, o valor de `x` será atribuído a `z`.
- `: y` : Se `(y < x)` for verdadeiro, o valor de `y` será atribuído a `z`.

No exemplo do slide, `z = 5`.

Operador Condicional Ternário: Cenário

- Suponha que você esteja jogando futebol e esteja controlando os gols da seguinte forma:

```
public static void main(String args[]) {  
    int numberOfGoals = 5;  
    String s;  
    if (numberOfGoals == 1) {  
        s = "gol";  
    }  
    else {  
        s = "gols";  
    } //fim if  
    System.out.println("Marquei " + numberOfGoals + " " + s);  
} //fim do método main
```

Com base no número de gols marcados, esses exemplos imprimirão a forma singular ou plural da palavra "gol". A operação é compacta porque ela pode gerar somente dois resultados com base em uma expressão booleana.

Operador Condicional Ternário: Exemplo

- Um resultado semelhante é obtido com o operador ternário substituindo toda a instrução if/else por uma única linha

```
int numberOfGoals = 1;  
  
System.out.println("Marquei " + numberOfGoals + " "  
                  + (numberOfGoals == 1 ? "gol" : "gols")  
);
```

Operador Condicional Ternário: Exemplo

- Vantagem: insere a operação diretamente dentro de uma expressão

```
int numberOfGoals = 1;  
  
System.out.println("Marquei " + numberOfGoals + " "  
                  + (numberOfGoals == 1 ? "gol" : "gols")  
                  );
```

- Desvantagem: pode ter apenas dois resultados possíveis

```
(numberOfGoals==1 ? "gol" : "gols" : "More Xgols");  
  └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘  
  boolean verdadeiro falso ???
```

Como você pode ver, o operador ternário pode ser de grande utilidade para reduzir o número de linhas de código, mas também pode dificultar a leitura do código. Portanto, ele não é ideal para instruções aninhadas.

Exercício 2

- Adicione o arquivo `TernaryOperator.java` ao projeto que você criou para o exercício 1
- Modifique `TernaryOperator.java` para duplicar a lógica fornecida na instrução `if/else` usando o operador ternário

Tratando Condições Complexas com uma Construção if Encadeada

- A instrução if encadeada:
 - Conecta várias condições em uma única construção
 - Tende a ser confusa de ser lida e difícil de ser mantida

Encadeando Construções if/else

- Você pode encadear construções if e else juntas para definir vários resultados para diversas expressões diferentes
- Sintaxe:

```
if (<condition1>) {  
    //code_block1  
}  
else if (<condition2>) {  
    // code_block2  
}  
else {  
    // default_code  
} //fim if
```

A sintaxe de uma construção if/else é mostrada no exemplo do slide:

- Cada uma das condições é uma expressão booliana.
- `bloco_código1` representa as linhas de código que serão executadas se a condição1 for verdadeira.
- `bloco_código2` representa as linhas de código que serão executadas se a condição1 for falsa e a condição2 for verdadeira.
- `código_padrão` representa as linhas de código que serão executadas se as duas condições forem falsas.

Observação: é possível avaliar várias instruções `else if`. A instrução `else` é opcional.

Encadeando Construções if/else: Exemplo

```
public static void main(String args[]) {  
    double income = 30000, tax;  
  
    if (income <= 15000) {  
        tax = 0;  
    }  
    else if (income <= 25000) {  
        tax = 0.05 * (income - 15000);  
    }  
    else {  
        tax = 0.05 * (income - (25000 - 15000));  
        tax += 0.10 * (income - 25000);  
    }  
} //fim if  
} //fim do método main
```

Este exemplo demonstra construções `if/else` encadeadas para testar várias condições. A instrução `else` será executada se todas as condições forem falsas.

É Possível Aninhar Instruções if?

- Em Java, uma instrução if pode estar presente dentro do corpo de outra instrução if

```
if (tvType == "em cores") {  
    if (size == 14) {  
        discPercent = 8;  
    }  
    else {  
        discPercent = 10;  
    }  
}  
}
```

- Neste exemplo, a instrução else é combinada com a instrução if (size==14)

Em uma instrução `if` aninhada:

É muito importante ter certeza da construção `else` que acompanha a construção `if`. Esse recuo torna o código bem mais claro para o leitor.

Neste exemplo, se a instrução `if` externa for verdadeira, a instrução `if` interna será executada.

Entendendo Instruções if Aninhadas

- Neste exemplo, a instrução else é combinada com a instrução if externa (TVType=="em cores")

```
if (tvType == "em cores") {  
    if (size == 14) {  
        discPercent = 8;  
    }//endif  
}  
else {  
    discPercent = 10;  
}//fim if
```

Exercício 3

- Adicione o arquivo `ComputeFare.java` ao projeto que você criou para o exercício 1
- Examine `ComputeFare.java`
- Implemente o seguinte usando construções `if/else`:
 - Declare uma variável inteira: `age`
 - Faça com que o usuário digite o valor da idade
- Usando uma construção `if` encadeada, calcule a tarifa com base na idade de acordo com estas condições:
 - Se a idade for menor que 11, a tarifa será igual a US\$ 3
 - Se a idade for maior que 11 e menor que 65, a tarifa será igual a US\$ 5
 - Para todas as demais idades, a tarifa será igual a US\$ 3

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Descrever a execução condicional
 - Descrever operadores lógicos
 - Entender a avaliação de operadores lógicos de “curto-circuito”
 - Criar construções if encadeadas



The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by a thin black border, with dark gray horizontal bars at the top and bottom.

ORACLE

Academy