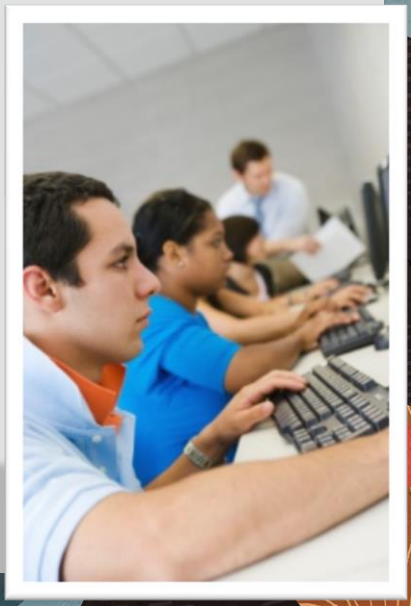
ORACLE

Academy

Java Foundations

5-2
Entendendo a Execução Condicional



ORACLE
Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

Tópicos

- **Operadores Lógicos**
- Avaliação de Curto-Circuito
- Operador Ternário
- Criar Construções if Encadeadas

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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.");
   }//fim if
}//fim if
```


Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

Operadores Lógicos Java

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

Operador Lógico	Significado
&&	E
||	OU
!	NÃO

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

Três Operadores Lógicos

Operação	Operador	Exemplo

Se uma condição E outra condição	&&	int i = 2;
		int j = 8;
		((i < 1) && (j > 6))

Se uma das condições OU as duas condições	||	int i = 2;
		int j = 8;
		((i < 1) || (j > 10))

NÃO	!	int i = 2; (!(i < 3))

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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

Operador lógico `E`:

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

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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 boolianas

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

Expressão Booliana 1 — **Operador Lógico** — **Expressão Booliana 2**

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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

A condição combinada será falsa se as duas expressões boolianas 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 boolianas for verdadeira

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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**

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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."`

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");
    }//fim if
 }//fim do método main
```

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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."`

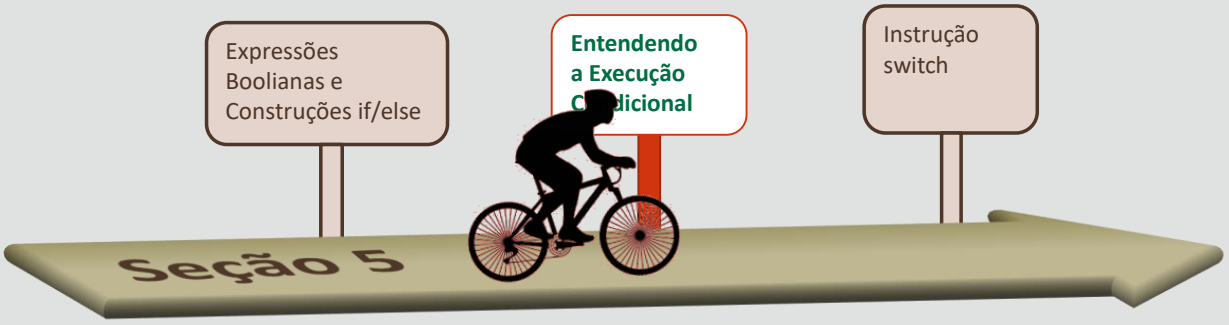
Exercício 1

- Importe e abra o projeto `ConditionalEx`
- 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.**"

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

Tópicos

- Operadores Lógicos
- **Avaliação de Curto-Circuito**
- Operador Ternário
- Criar Construções if Encadeadas



ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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)

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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**

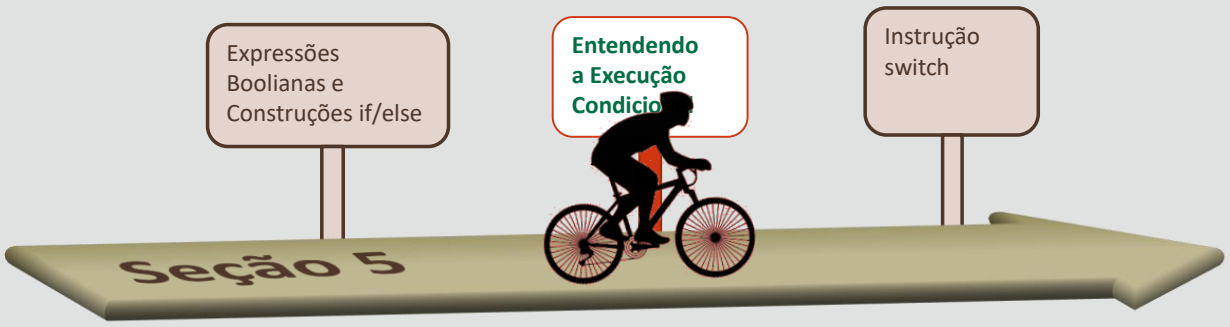
- Se (x<=10) for verdadeiro, (x>20) não será calculado porque não importa se (x>20) é verdadeiro ou falso
- O resultado desta expressão será true

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

Tópicos

- Operadores Lógicos
- Avaliação de Curto-Circuito
- **Operador Ternário**
- Criar Construções if Encadeadas



ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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:
		int x = 2, y = 5, z = 0;
		
		z = (y < x) ? x : y;

Instruções equivalentes

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

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

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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
```


Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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 booliana.

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")
);
```


Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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 gols");
```

booliano — verdadeiro — falso — ???

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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.

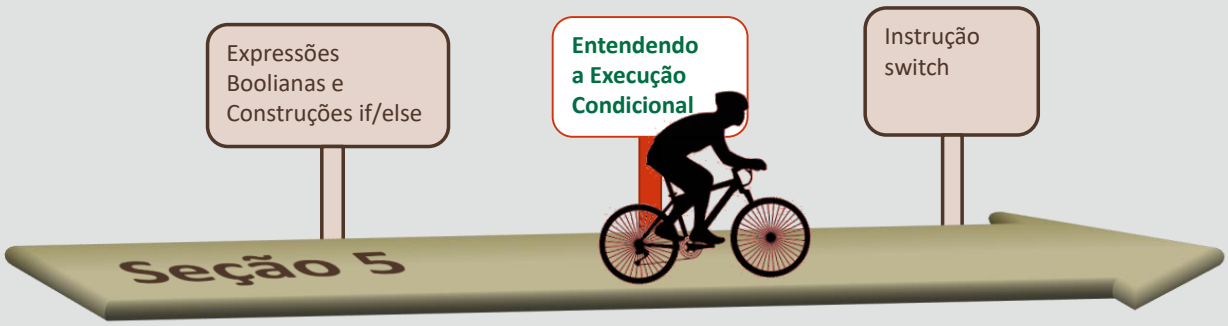
Exercício 2

- Importe e abra o projeto `ConditionalEx`
- Modifique `TernaryOperator.java` para duplicar a lógica fornecida na instrução if/else usando o operador ternário

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

Tópicos

- Operadores Lógicos
- Avaliação de Curto-Circuito
- Operador Ternário
- **Criar Construções if Encadeadas**



ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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
```

ORACLE Academy

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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
```


Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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;
   }//fim if
}//fim if
```

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

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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
```


Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

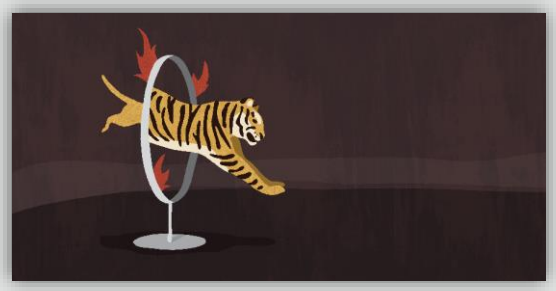
Exercício 3

- Importe e abra o projeto `ConditionalEx`
- 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

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

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



Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.


ORACLE
Academy