



Java Foundations

5-1

Expressões Booleanas e Construções if/else

ORACLE
Academy



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

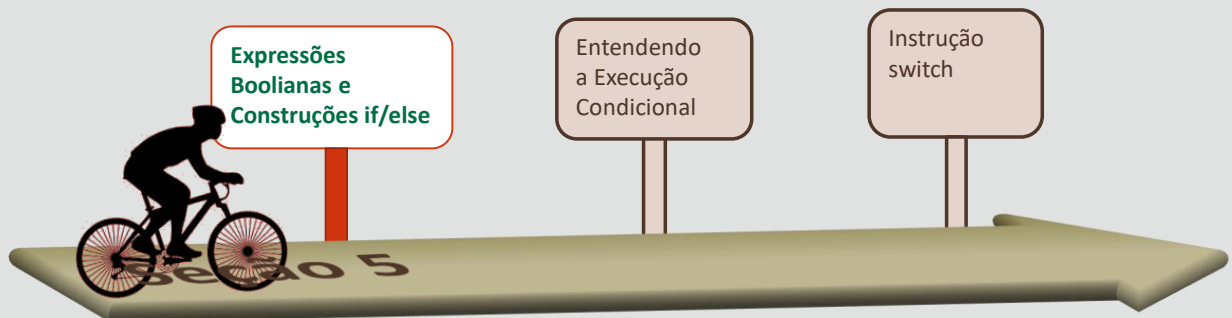
Objetivos

- Esta lição abrange os seguintes objetivos:
 - Declarar, inicializar e usar variáveis booleanas
 - Comparar expressões booleanas usando operadores relacionais
 - Criar uma instrução if
 - Criar construções if/else
 - Comparar Strings



Tópicos

- **Tomando Decisões**
- Operadores Relacionais
- Instrução if
- Construções if/else
- Comparando Variáveis de String



ORACLE
Academy

JFo5-1
Expressões Booleanas e Construções
if/else

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

4

Tomando Decisões

- Até agora nas lições anteriores, você viu diferentes tipos de dados suportados em Java
- O tipo boolean é outro tipo de dados no Java que ajuda a adicionar lógica a um programa
- Ele ajuda a tomar decisões

Tomando Decisões



ORACLE
Academy

JFo5-1
Expressões Booleanas e Construções
if/else

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

Tomando Decisões

- Suponhamos que você esteja indo para a escola e se depara em um cruzamento
- Agora você tem que tomar uma decisão lógica:
 - Se virar à esquerda, esse caminho me levará à escola?
 - Se for reto, esse caminho me levará à escola?
 - Se virar à direita, esse caminho me levará à escola?
- Só existem duas respostas para cada uma dessas perguntas:
 - sim ou não

Tipo de Dados boolean do Java

- É basicamente o mesmo que acontece no Java, onde valores booleanos informarão ao programa qual é o melhor curso de ação a tomar
- No Java, os valores do tipo de dados boolean são true e false, em vez de sim e não
- Você declara o tipo de dados boolean usando a palavra-chave boolean

Usando um Tipo de Dados boolean do Java: Exemplo

- Observação: o valor de uma variável booleana é exibido como true ou false

```
public static void main(String args[]) {  
  
    boolean passed, largeVenue, grade; — Declarando  
                                         variáveis booleanas  
  
    passed = true;  
    largeVenue = false;  
    grade = passed; } — Atribuindo valores a  
                      variáveis booleanas  
  
    System.out.println(passed);  
    System.out.println(largeVenue);  
    System.out.println(grade); } — Imprimindo valores de  
    } // fim do método main      variáveis booleanas
```

Lembre-se de que um tipo de dados `boolean` só pode ter dois valores possíveis: verdadeiro e falso.

Observação: verdadeiro e falso nunca estão entre aspas porque são valores `booleanos`, e não `Strings`.

A saída do exemplo mostrado no slide:

verdadeiro

falso

verdadeiro

Tipo de Dados boolean: Cenário

- E se você estivesse dirigindo um carro com um sistema GPS instalado sendo executado em Java?
- Antes de sair de casa, você pede ao sistema GPS para levar você para a escola
- Que código simples você escreve para ajudar a decidir qual caminho virar?

Com base nesse código simples, o carro irá na direção que tem uma variável `booleana` com um valor verdadeiro.

Observação: mais adiante neste lição, veremos como é possível aprimorar este exemplo com mais código.

Tipo de Dados boolean: Cenário

- Vamos começar

```
public static void main(String args[]) {  
    String left = "museu";  
    String straight = "ginástica";  
    String right = "restaurante";  
    boolean isLeft = false;  
    boolean isStraight = true;  
    boolean isRight = false;  
    System.out.println("Seguir em frente");  
}//fim do método main
```

Com base nesse código simples, o carro irá na direção que tem uma variável `booleana` com um valor verdadeiro.

Observação: mais adiante neste lição, veremos como é possível aprimorar este exemplo com mais código.

Expressões e Variáveis

- As expressões matemáticas podem ser...
 - Impressas
 - Atribuídas a uma variável `int` ou `double`

```
System.out.println(2 + 2);
```

```
int x = 2 + 2;
```

Use o sinal de igual (=) para fazer uma atribuição e use o sinal == para fazer uma comparação e retornar um valor `booleano`.

Expressões e Variáveis

- As expressões booleanas podem ser...
 - Impressas
 - Atribuídas a uma variável booleana

```
System.out.println(x == 5);  
boolean isFive = x == 5;
```

Use o sinal de igual (=) para fazer uma atribuição e use o sinal == para fazer uma comparação e retornar um valor `booleano`.

Igualdade e Atribuição

- `==` é um operador relacional
- Este operador verifica se os dois lados de uma expressão booliana são iguais
- Uma expressão booliana retorna um valor `true` ou `false`

```
x == 5
```

Igualdade e Atribuição

- = é um operador de atribuição
- Esse operador atribui um valor a uma variável
- É possível atribuir a uma variável booliana qualquer valor que uma expressão booliana retorne

```
int x = 4;  
boolean isFive = x == 5;
```

Valores em Expressões Booleanas

- Use == para testar a igualdade entre valores das primitivas
- As expressões booleanas podem conter variáveis ou valores submetidos a hard-code

```
boolean res1 = 24 == 15;  
System.out.println("res1: " + res1);  
int value1 = 15;  
int value2 = 24;  
boolean res2 = value1 == value2;  
System.out.println("res2: " + res2);
```

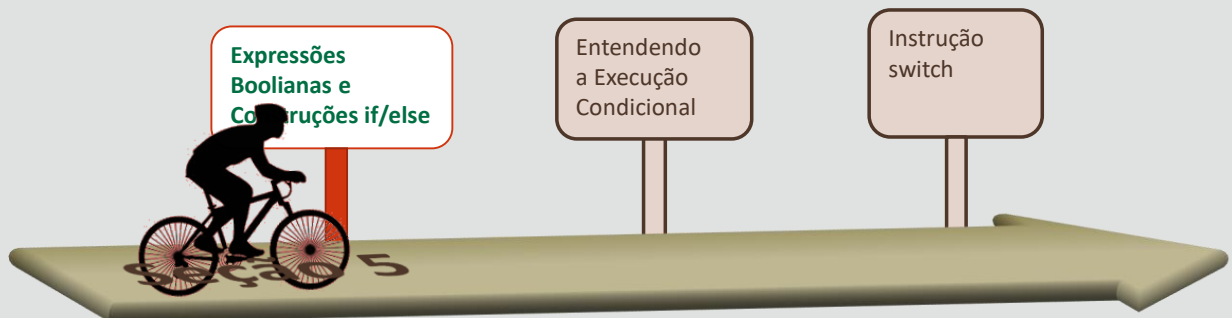

Valores em Expressões Booleanas

- As duas expressões abaixo retornam o mesmo valor:
- Se value1 e value2 contiverem o mesmo valor, a expressão retornará um resultado true
- Caso contrário, a expressão retornará false

```
boolean res1 = 24 == 15;  
System.out.println("res1: " + res1);  
int value1 = 15;  
int value2 = 24;  
boolean res2 = value1 == value2;  
System.out.println("res2: " + res2);
```

Tópicos

- Tomando Decisões
- **Operadores relacionais**
- Instrução if
- Construções if/else
- Comparando Variáveis de String



ORACLE
Academy

JFo5-1
Expressões Booleanas e Construções
if/else

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

18

Operadores Relacionais

- Use operadores relacionais em expressões booleanas que são utilizados para avaliar instruções if/else

Veja a seguir uma lista mais completa dos operadores relacionais. A tabela lista as diferentes condições que você pode testar usando operadores relacionais. O resultado de todos os operadores relacionais é um valor `booleano`. Todos os exemplos geram um resultado `booleano` verdadeiro.

Operadores Relacionais

Condição	Operador	Exemplo
É igual a	<code>==</code>	<code>int i=1;</code> <code>(i == 1)</code>
Não é igual a	<code>!=</code>	<code>int i=2;</code> <code>(i != 1)</code>
É menor que	<code><</code>	<code>int i=0;</code> <code>(i < 1)</code>
É menor que ou igual a	<code><=</code>	<code>int i=1;</code> <code>(i <= 1)</code>
É maior que	<code>></code>	<code>int i=2;</code> <code>(i > 1)</code>
É maior que ou igual a	<code>>=</code>	<code>int i=1;</code> <code>(i >= 1)</code>

ORACLE
Academy

JFo5-1
Expressões Booleanas e Construções
if/else

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

Veja a seguir uma lista mais completa dos operadores relacionais. A tabela lista as diferentes condições que você pode testar usando operadores relacionais. O resultado de todos os operadores relacionais é um valor `booleano`. Todos os exemplos geram um resultado `booleano` verdadeiro.

Operadores Relacionais: Exemplo

- Observação: use o sinal de igual (=) para fazer uma atribuição e use o sinal == para fazer uma comparação e retornar um valor booleano

```
public static void main(String args[]) {  
    int a = 10;  
    int b = 20;  
    System.out.println(a == b);  
    System.out.println(a != b);  
    System.out.println(a > b);  
    System.out.println(a < b);  
    System.out.println(b >= a);  
    System.out.println(b <= a);  
} //fim do método main
```

Para os valores
das primitivas, ==
verifica o teste de
igualdade

ORACLE
Academy

JFo5-1
Expressões Booleanas e Construções
if/else

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

21

Teste de Igualdade:

Para objetos, use o método de igualdade para testar a igualdade.

Saída:

falso

verdadeiro

falso

verdadeiro

verdadeiro

falso

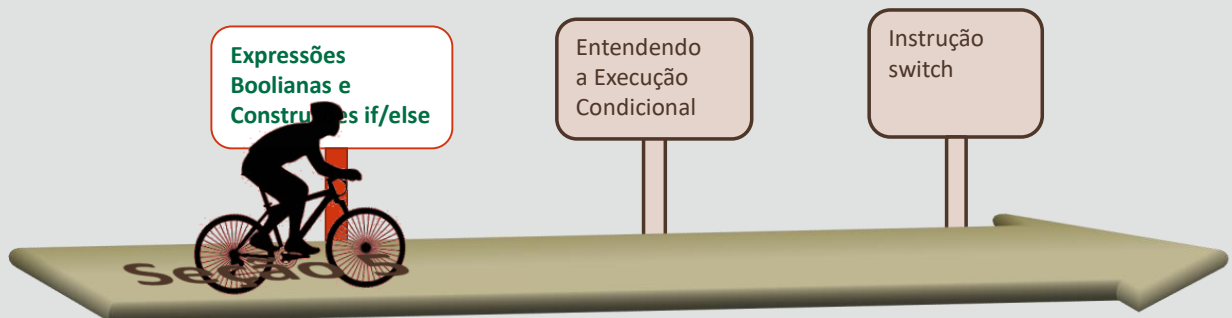


Exercício 1

- Importe e abra o projeto `IfElseEx`
- Modifique `AgeValidity.java` para implementar o seguinte:
 - Faça com que os usuários informem sua idade
 - Declare uma variável booleana: `drivingUnderAge`
 - Inicialize `drivingUnderAge` como falso
 - Escreva uma expressão booleana para verificar se a idade inserida pelo usuário é menor ou igual a 18
 - Em seguida, defina `drivingUnderAge` como `true`
 - Imprima o valor de `drivingUnderAge`

Tópicos

- Tomando Decisões
- Operadores Relacionais
- **Instrução if**
- Construções if/else
- Comparando Variáveis de String



ORACLE
Academy

JFo5-1
Expressões Booleanas e Construções
if/else

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

23

Instruções Condicionais

- As instruções condicionais permitem escolher quais instruções serão executadas a seguir
- Essas decisões baseiam-se em expressões booleanas (ou condições) que resultam em `true` ou `false`
- Estas são as instruções condicionais em Java:
 - Instrução **if**
 - Instrução **if/else**
 - Instrução **switch**

Entendendo a Instrução if

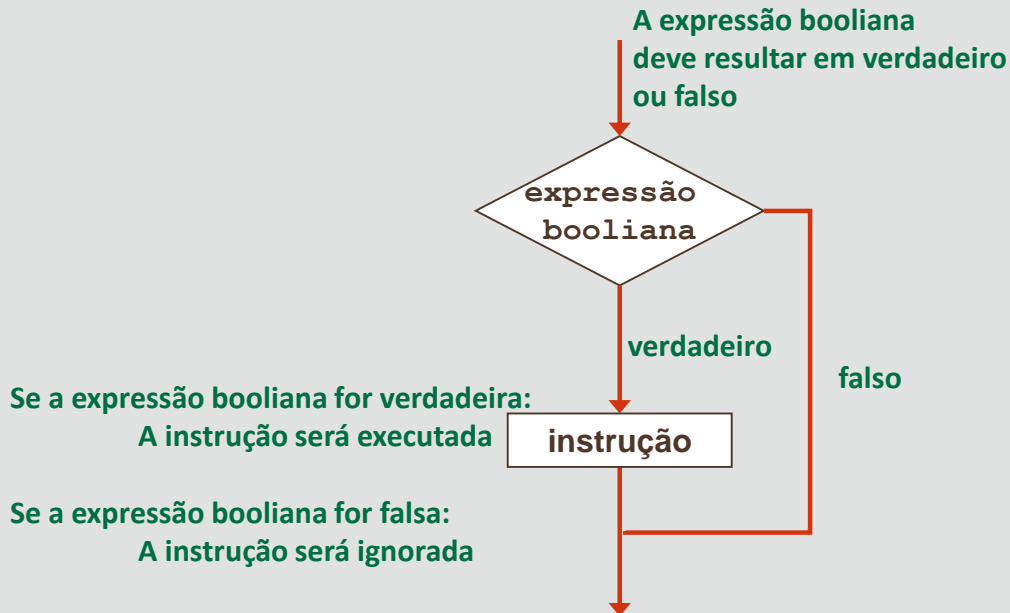
- Uma instrução if consiste em uma expressão booliana seguida de uma ou mais instruções
- Sintaxe:

expressão booliana

```
if ( <alguma condição é verdadeira> ){  
    //Instruções serão executadas se a expressão  
    //booliana for verdadeira  
} //fim if
```

A condição a ser avaliada fica entre parênteses. Ela é referida como uma expressão `booliana` porque deve resultar em verdadeiro ou falso.

Entendendo a Instrução if



Usando Expressões booleanas em Instruções if

```
public static void main(String args[]) {  
    String left = "museu";  
    String straight = "ginástica";  
    String right = "restaurante";  
  
    if (left == "ginástica") {  
        System.out.println("Virar à Esquerda");  
    } //fim if  
  
    if (straight == "ginástica") {  
        System.out.println("Seguir em Frente");  
    } //fim if  
  
    if (right == "ginástica") {  
        System.out.println("Virar à Direita");  
    } //fim if  
} //fim do método main
```

Este bloco é executado

No exemplo do slide, a expressão booleana na segunda instrução if retorna verdadeiro. Por isso, "Seguir em Frente" é impresso no console.

Executando um Bloco de Código

1. Um bloco de código não é necessário para uma instrução ser executada por uma instrução `if`
 - Veja um exemplo a seguir:

```
daysInFeb = 28;  
if(isLeapYear)    Só essa instrução é  
    daysInFeb = 29; executada  
    System.out.println(year + "é um ano bissexto");
```

1

No exemplo 1, você quer que as duas instruções sejam executadas quando a condição for verdadeira. Como não existe um bloco de código, só a primeira instrução será executada.

No exemplo 2, as duas instruções são executadas quando a condição é verdadeira porque existe um bloco de código para a instrução `if`.

Para evitar esse erro, você deve usar blocos de código mesmo que haja apenas uma instrução para ser executada no `if` bloco.

Executando um Bloco de Código

- No entanto, sempre é recomendado que você use blocos de código, mesmo que só haja uma instrução para ser executada no bloco

```
daysInFeb = 28;  
if(isLeapYear){  
    daysInFeb = 29;  
    System.out.println(year + " é um ano bissexto");  
} //endif
```

2

Este bloco é executado

No exemplo 1, você quer que as duas instruções sejam executadas quando a condição for verdadeira. Como não existe um bloco de código, só a primeira instrução será executada.

No exemplo 2, as duas instruções são executadas quando a condição é verdadeira porque existe um bloco de código para a instrução `if`.

Para evitar esse erro, você deve usar blocos de código mesmo que haja apenas uma instrução para ser executada no bloco `if`.

Instrução if: Exemplos

```
public static void main(String args[]) {  
    int grade = 85;  
  
    if (grade > 88) {  
        System.out.println("Você formou-se com Louvor.");  
    } //fim if  
  
    if (grade <=88) {  
        System.out.println("Você está apto a ensinar.");  
    }//fim if  
  
} //fim do método main
```

Segunda
instrução if

- Saída:

Você está apto a ensinar.

No slide do exemplo, existem duas instruções `if`: A primeira testa os valores maiores que 88; a segunda testa os valores menores ou iguais que 88.

As duas instruções `if` são avaliadas, mesmo que a primeira seja verdadeira.

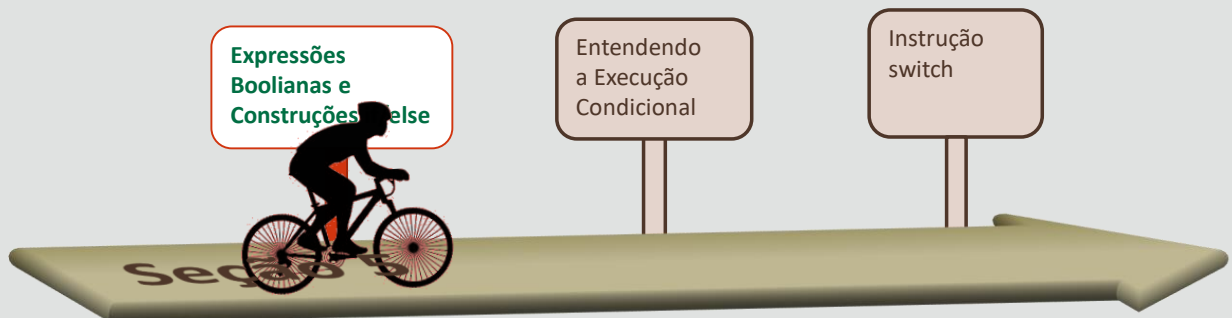


Exercício 2

- Importe e abra o projeto `IfElseEx`
- Modifique `ChkOddEven.java` para implementar o seguinte:
 - Insira um número entre 1 e 10
 - Use instruções `if`
 - Teste se um número é par ou ímpar
- Seu programa deve gerar a seguinte saída:
 - Insira um número: 7
 - O número é 7, ímpar

Tópicos

- Tomando Decisões
- Operadores Relacionais
- Instrução if
- **Construções if/else**
- Comparando Variáveis de String



ORACLE
Academy

JFo5-1
Expressões Booleanas e Construções
if/else

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

32

Escolhendo entre Duas Alternativas

- Se você quiser escolher entre duas alternativas, use a instrução `if/else`
- Sintaxe:

```
                expressão booliana
if ( <alguma condição for verdadeira> ) {
    // fazer algo
}
else {
    // fazer algo diferente
} //fim if
```

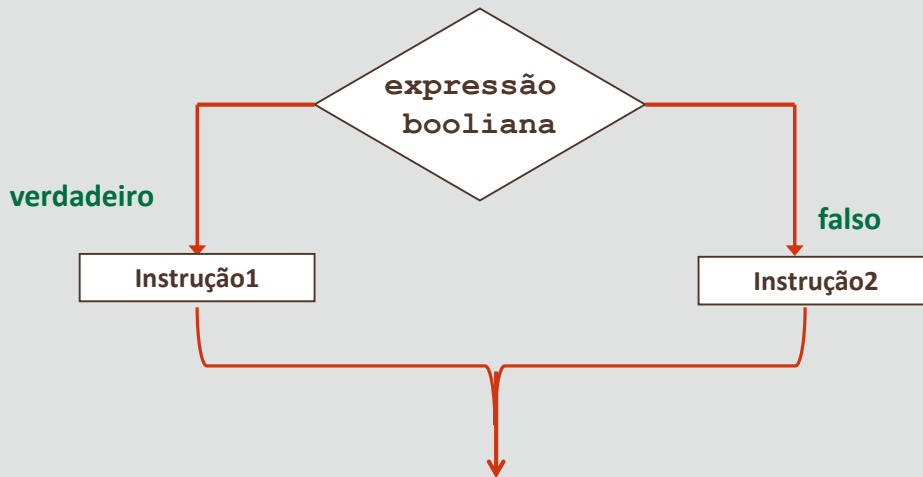
bloco if

bloco else

A instrução `if/else` é uma maneira de separar seu código em função de alguma condição. Ela usa duas palavras-chave Java: `if` e `else`.

- Se alguma condição for verdadeira, execute o código dentro do bloco `if`.
- Se essa condição for falsa, execute o código no bloco `else`.

Entendendo Instruções if/else



Se a expressão booliana for verdadeira: a Instrução1 será executada
Se a expressão booliana for falsa: a Instrução2 será ignorada

Instruções if/else: Exemplo 1

```
String forecast;  
double temperature = getTemperature();  
  
if (temperature <= 32.0) {  
    forecast = "NEVE";  
}  
else {  
    forecast = "CHUVA";  
} //fim if
```

Este bloco é executado



30.3 °F

Este slide demonstra um exemplo da instrução if/else. O método `getTemperature()` retorna a temperatura como 30,3. Como a temperatura é menor que 32 graus, a expressão booleana `(temperature <= 32.0)` retorna verdadeiro, e o bloco if é executado.

Instruções if/else: Exemplo 2

```
String forecast;  
double temperature = getTemperature();  
  
if (temperature <= 32.0) {  
    forecast = "NEVE";  
}  
else {  
    forecast = "CHUVA";  
} //fim if
```



40,2 °F

Este bloco é executado

Este slide demonstra um exemplo da instrução if/else. O método `getTemperature()` retorna a temperatura como 40,2. Na instrução if/else, como `temperatura <= 32` graus, a expressão booleana `temperatura <= 32.0` retorna falso, e o bloco else é executado.

Instruções if/else: Exemplo 3

- Você pode substituir as duas instruções if por uma instrução if/else
- A instrução if/else é mais eficiente porque só uma comparação está sendo feita

```
public static void main(String args[]) {  
    int grade = 85;  
    if (grade > 88) {  
        System.out.println("Você formou-se com Louvor.");  
    }  
    else {  
        System.out.println("Você passou.");  
    } //fim if  
} //fim do método main
```



Exercício 3

- Importe e abra o projeto `IfElseEx`
- Examine `AgeCheck.java`:
 - O programa tem um problema lógico
 - Para alguns valores, ele imprime a resposta errada
 - Encontre os problemas e corrija-os (Pode ser que você precise executar o programa algumas vezes e testar outros valores para ver quais falharão)
 - Substitua as duas instruções `if` por uma instrução `if/else`



Exercício 4

- Importe e abra o projeto IfElseEx
- Examine ShoppingCart.java
- Use uma instrução if/else para implementar o seguinte:
 - Declare e inicialize uma variável booliana, outOfStock
 - Se a quantidade for > 1 , altere a variável da mensagem para indicar plural
 - Se não houver um item no estoque, informe ao usuário que o item não está disponível
 - Else imprime a mensagem e o custo total

Tópicos

- Tomando Decisões
- Operadores Relacionais
- Instrução if
- Construções if/else
- **Comparando Variáveis de String**



ORACLE
Academy

JFo5-1
Expressões Booleanas e Construções
if/else

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

40

Comparando Variáveis

- Quando compara valores usando expressões booleanas, você precisa entender as nuances de determinados tipos de dados
- Operadores relacionais como `==` são...
 - Excelentes para comparar primitivas
 - Ruins para comparar Strings (e outros objetos)
- Vamos analisar por quê

Comparando Primitivas

- O valor z é definido como a soma de $x + y$
- Quando uma expressão booliana testa a igualdade entre z e a soma de $x + y$, o resultado é `true`

```
int x = 3;  
int y = 2;  
int z = x + y;  
  
boolean test = (z == x + y);  
System.out.println(test);           //verdadeiro
```

Observação para os Instrutores: a caixa de código neste slide deve estar na mesma posição para que pequenas diferenças tornem-se mais aparentes.

Comparando Strings

- O valor z é definido como sendo a concatenação de x + y
- Quando uma expressão booliana testa a igualdade entre z e a concatenação de x + y, o resultado é false

```
String x = "Ora";  
String y = "cle";  
String z = x + y;  
  
boolean test = (z == x + y);  
System.out.println(test);           //false
```

por quê?

Observação para os Instrutores: a caixa de código neste slide deve estar na mesma posição para que pequenas diferenças tornem-se mais aparentes.

Por que Há Resultados Contraditórios?

- As primitivas e os objetos são armazenados de forma diferente na memória
 - As Strings recebem um tratamento especial
 - Isso será discutido mais adiante no curso
- Como resultado...
 - == compara os valores das primitivas
 - == compara os locais dos objetos na memória
- É mais provável que você precise comparar o conteúdo de Strings e não suas localizações na memória

Como Você Deve Comparar Objetos String?

- Você nunca deve comparar Strings usando ==
- Em vez disso, compare Strings usando o método equals()
 - Esse método é parte da classe String
 - Ele aceita um argumento String, verifica se o conteúdo das Strings é igual e retorna um booleano
 - Também existe um método semelhante: equalsIgnoreCase()

```
String x = "Ora";  
String y = "cle";  
String z = x + y;  
boolean test = z.equals(x + y);  
System.out.println(test);           //verdadeiro
```



Exercício 5

- Importe e abra o projeto `IfElseEx`
- Examine `StringEquality.java`
- Use uma instrução `if` e uma instrução `if/else`:
 - Declare um nome de variável de `String`
 - Faça com que o usuário insira um valor para o nome
 - Verifique se o nome é "Moe" e imprima "Você é o rei do rock and roll"
 - Caso contrário, imprima "Você não é o rei"
 - Não use `==`

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Declarar, inicializar e usar variáveis booleanas
 - Comparar valores das primitivas usando operadores relacionais
 - Criar uma instrução if
 - Criar construções if/else
 - Comparar Strings



