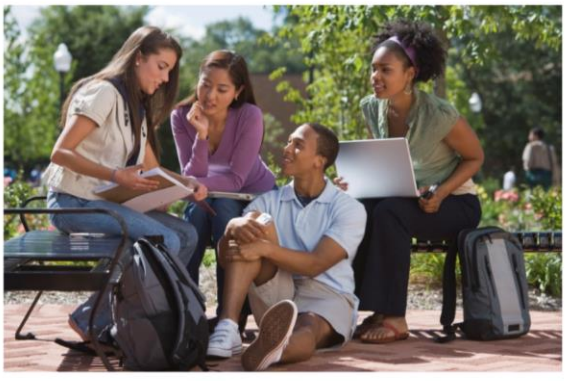
ORACLE

Academy

Java Foundations

3-3

Dados Textuais



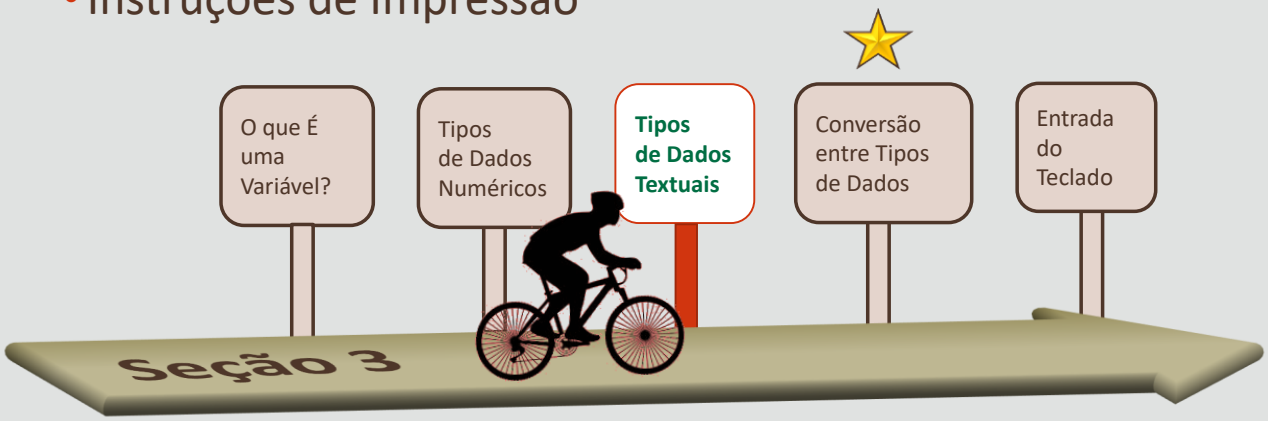
ORACLE
Academy

Objetivos

- Esta lição abrange os seguintes objetivos:
  - Usar o tipo de dados char
  - Usar Strings
  - Concatenar Strings
  - Entender sequências de escape
  - Entender melhor as instruções de impressão

Tópicos

- **Caracteres e Strings**
- Concatenação de Strings
- Combinando Strings e Números
- Instruções de Impressão



ORACLE Academy

Tipo Primitivo Textual

- O único tipo de dados textual primitivo é char
- Ele é usado para um único caractere (16 bits)

- Exemplo:
  - char shirtSize = 'M';

As aspas simples devem ser usadas com valores literais de caractere

Outro tipo de dados é usado para armazenar e manipular dados como um único caractere. O tipo primitivo `char` tem 16 bits de tamanho. Quando atribui um valor literal a uma variável `char`, você deve usar marcações de aspas simples ao redor do caractere, conforme mostrado no exemplo do código.

Unindo Caracteres

- Você pode usar caracteres juntos para criar frases
- Esta é uma maneira ineficiente de fazer isso
- É preciso haver uma linha de código para cada letra em uma sentença

```
char letter1 = 'H';
char letter2 = 'e';
char letter3 = 'l';
char letter4 = 'l';
char letter5 = 'o';
//É difícil codificar sentenças longas
System.out.println(letter1 +letter2 +letter3 +letter4 +letter5);
```

Unindo Caracteres de Forma Eficiente

- Esta é uma maneira melhor
  - Só é necessária uma linha para a sentença inteira:

```
String greeting = "Hello World!";
//Observe as aspas duplas
System.out.println(greeting);
```

Caracteres x Strings

- chars são para um único caractere
  - Use aspas simples

```
char shirt1Size = 'S';
char shirt2Size = 'M';
char shirt3Size = 'L';
```

- chars não pode tratar vários caracteres

```
char shirt4Size = 'XL';
char shirt5Size = "XXL";
```

Caracteres x Strings

- Uma String pode tratar vários caracteres
  - Use aspas duplas

```
String shirt6Size = "XXXL";
```

Primitivas

Tipo	Comprimento	Dados
boolean	1 bit	true / false
byte	8 bits	Valores inteiros
short	16 bits	Valores inteiros
int	32 bits	Valores inteiros
long	64 bits	Valores inteiros
float	32 bits	Números de pontos flutuantes
double	64 bits	Números de pontos flutuantes
char	16 bits	Caracteres individuais

Onde estão as Strings?

ORACLE Academy

Vamos investigar

- Podemos identificar outras diferenças entre char e String?

```
char shirt3Size = 'L';
String shirt6Size = "XXXL";
```

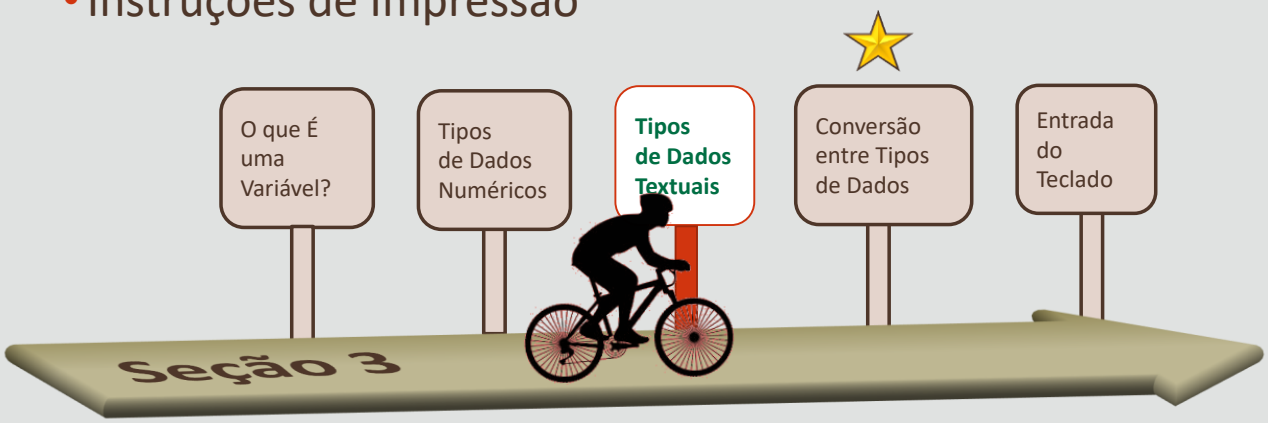
- 1. char fica azul
  - char é uma palavra-chave de um tipo de dados primitivo
  - As palavras-chave ficam azuis no NetBeans
- 2. String é capitalizada
  - As strings são um objeto, e não uma primitiva
  - Por convenção, os tipos de objetos são capitalizados

Strings São Objetos

- O Java vem com uma classe String que fornece detalhes sobre o seguinte:
  - Propriedades da string
  - Comportamentos da string
- As strings são objetos especiais
  - As strings são tratadas de maneira um pouco diferente que a maioria dos objetos
- Abordaremos mais estes pontos nas próximas seções:
  - Os objetos podem ter primitivas como propriedades
  - Os objetos podem ter objetos como propriedades, como Strings
  - Os objetos são armazenados de maneira diferente das primitivas na memória

Tópicos

- Caracteres e Strings
- **Concatenação de Strings**
- Combinando Strings e Números
- Instruções de Impressão



ORACLE Academy

Declaração e Inicialização de Strings

- Declare e atribua valores de String como faria com qualquer outra primitiva

```
//Uma variável declarada e inicializada
int intVar = 300;
String stringVar = "Three Hundred";

//Muitas variáveis declaradas
int x, y, z;
String xString, yString, zString;

//Uma variável declarada é inicializada mais tarde
x = 1;
xString = "One";
```

Variável da String x Literal da String

```
String stringVariable = "Esta é uma literal de String.";
```

Variável

Literal

- É possível criar uma String combinando literais de Strings:

```
String combinedLiterals = "Quero" + " comprar uma camisa.";
```

- É possível criar uma String combinando variáveis de Strings:

```
String var1 = "Esta camisa tem";
String var2 = " muitos botões.";
String combinedVariables = var1 + var2;
```

Concatenação de Strings

- A combinação de várias Strings é denominada concatenação
- As strings não podem ser combinadas usando o operador +
  - stringVariable1 + stringVariable2
  - stringVariable1 + “String literal”
  - stringVariable1 + “String literal” + stringVariable2

```
String greet1 = "Hello";
String greet2 = "World";
String message1 = greet1 + " " + greet2 + "!";
String message2 = greet1 + " " + greet2 + " " + 2020 + "!";
```

ORACLE Academy

A combinação de várias Strings é denominada "concatenação". Você pode concatenar uma variável de String com outra variável de String. Também pode concatenar uma literal de String com uma variável de String.

Você pode concatenar qualquer número de variáveis e literais de String para alcançar seu objetivo.

Saída de Concatenação da String

- Exemplo de concatenação:

```
String greet1 = "Hello";
String greet2 = "World";
String message1 = greet1 + " " + greet2 + "!";
String message2 = greet1 + " " + greet2 + " " + 2020 + "!";
```

- Você pode concatenar Strings dentro de uma instrução de impressão:

```
System.out.println(message2);
System.out.println(greet1 + " " + greet2 + " " + 2020 + "!");
```

- Saída:

```
Hello World 2020!
Hello World 2020!
```

ORACLE Academy

Nos exemplos do slide, você vê duas variações de impressão de dados de String usando System.out.println.

- No primeiro exemplo, a variável mensagem2 que você viu no slide anterior será impressa.
- No segundo exemplo, a expressão que contém a concatenação de variáveis, mais as literais de String, pode ser usada dentro dos parênteses de método. A concatenação é concluída pelo mecanismo de run-time antes de o método println ser executado.
- Como você pode ver, a saída das duas invocações do método é a mesma.

Cenário do Exercício 1

- Pense novamente no exemplo do catálogo de roupas da loja Duke's Choice:
- O cenário incluiu uma classe ShoppingCart
- Algumas propriedades e alguns comportamentos de ShoppingCart são analisados superficialmente neste exercício
- Propriedades de ShoppingCart:
  - A quem ela pertence
  - Os itens que ela contém
  - Uma mensagem/descrição do carrinho
- Comportamentos de ShoppingCart:
  - Imprime sua mensagem

Representado como Strings neste exercício

ORACLE Academy

O cenário do catálogo da Duke's Choice foi analisado na Seção 2, Lição 3.

Exercício 1, Parte 1

- Importe e edite o projeto `ShoppingCart01`
- Declare e inicialize a variável de String custName
- Declare e inicialize a variável de String itemDesc
- Declare uma variável de String message

Exercício 1, Parte 2

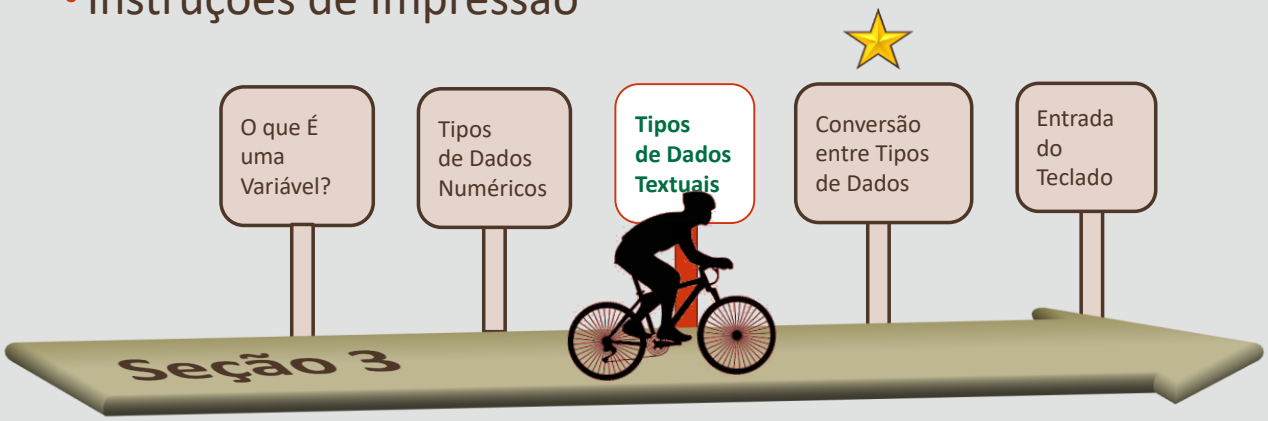
- Atribua à variável message um valor concatenado que inclua custName, itemDesc e uma literal de String que resulte em uma sentença completa:
  - (exemplo: "**Alex quer comprar uma Camisa**")
- Imprima a mensagem

- Seu programa deve produzir a seguinte saída:

```
Alex quer comprar uma Camisa
```

Tópicos

- Caracteres e Strings
- Concatenação de Strings
- **Combinando Strings e Números**
- Instruções de Impressão



ORACLE Academy

Combinando Strings e Números

- As strings podem conter números:

```
String totalPrice = "Total: $" +3;
System.out.println(totalPrice);      //Total: $3
```

- Cuidado quando tentar fazer cálculos:

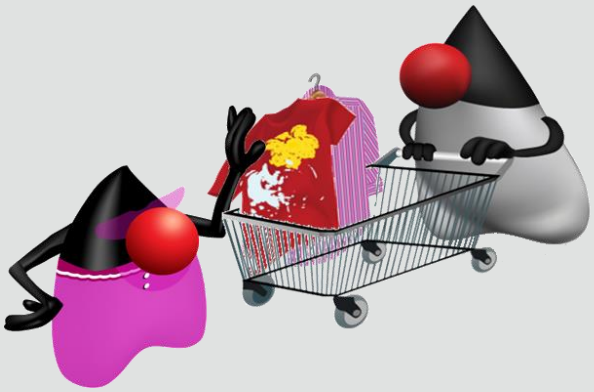
```
String totalPrice = "Total: $" +3 +2 +1;
System.out.println(totalPrice);      //Total: $321
```

- Use parênteses para os números:

```
String totalPrice = "Total: $" +(3 +2 +1);
System.out.println(totalPrice);      //Total: $6
```

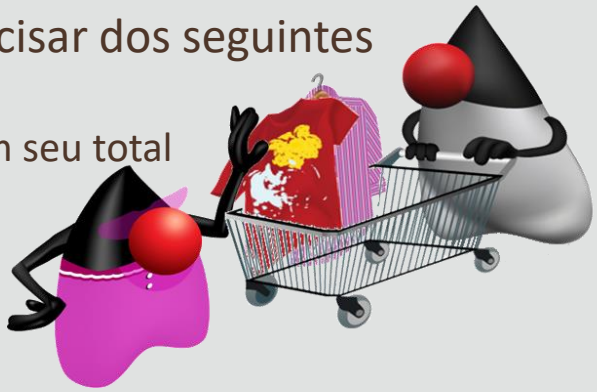
Cenário do Exercício 2

- Pergunta:
  - se os clientes encherem o carrinho, quanto eles pagarão?
- Precisamos representar os itens do carrinho mais detalhadamente para responder isso



Cenário do Exercício 2

- Um ShoppingCart pode precisar saber as seguintes propriedades:
  - Preço do item
  - Valor do imposto sobre vendas
  - Quantidade de itens
  - Preço total calculado de todos os itens no carrinho
- Um ShoppingCart pode precisar dos seguintes comportamentos:
  - Imprimir uma mensagem com seu total



Exercício 2, Parte 1

- Importe e edite o projeto `ShoppingCart02`
- Declare e inicialize campos numéricos:
  - preço (double)
  - imposto (double)
  - quantidade (int)
- Declare um totalPrice duplo:
  - Atribua um valor calculado de preço , imposto e quantidade

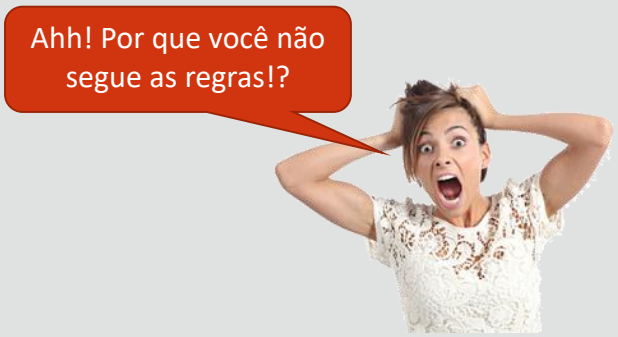
Exercício 2, Parte 2

- Mude a mensagem para incluir a quantidade:
  - (exemplo: "**Alex quer comprar duas Camisas**")
- Imprima outra mensagem mostrando o custo total
- Seu programa deve produzir uma saída semelhante:

```
Alex quer comprar duas Camisas
O custo total com o imposto é: $25.78
```

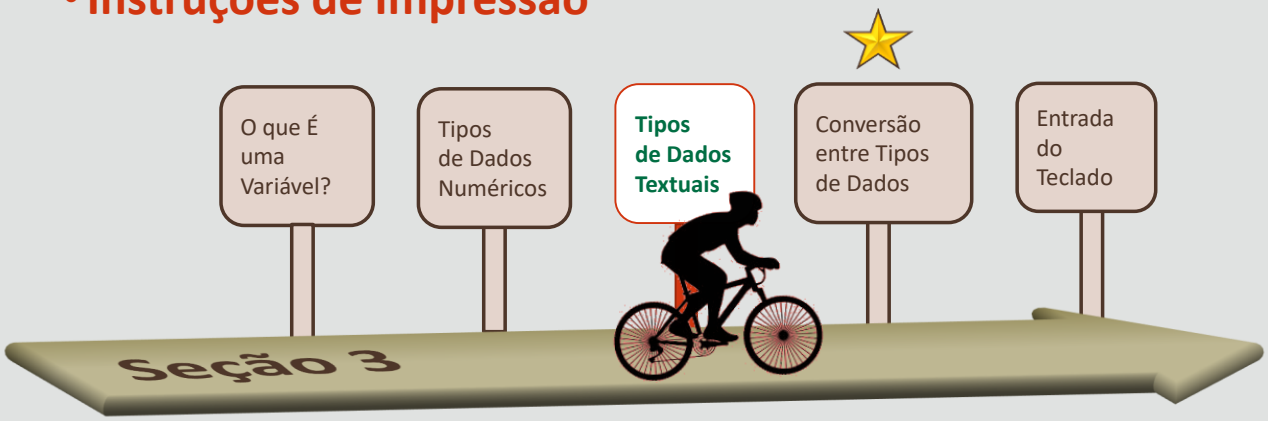
Observações do Exercício

- O ideal não é representar as propriedades e os comportamentos de objetos totalmente dentro do método main
- Violamos essa regra nesta seção para podermos focar na manipulação de dados
- Tentaremos fazer um trabalho melhor seguindo as regras na próxima seção



Tópicos

- Caracteres e Strings
- Concatenação de Strings
- Combinando Strings e Números
- **Instruções de Impressão**



ORACLE Academy

Caracteres Especiais nas Strings

- Você lembra de quando imprimimos o gato?
- Na verdade, as duas barras invertidas não foram impressas:
  - Só uma barra invertida foi impressa
  - Por quê?

```
3   public class Text01 {
4       public static void main(String[] args) {
5           System.out.println("   /\\         /\\   ");
6           System.out.println("  /  \\_______/  \\  ");
7           System.out.println(" /               \\ ");
8           System.out.println("(  /\\         /\\  )");
9           System.out.println("====     V     ====");
10          System.out.println("======(__|__)======");
11          System.out.println("  (             )  ");
12          System.out.println("   (___________)   ");
13      }
14  }
```

Sequência de Escape

- Um caractere precedido de uma barra invertida denomina-se sequência de escape e tem um significado especial para o compilador
- A tabela no próximo slide mostra as sequências de escape do Java

Sequência de Escape

Sequência de Escape	Descrição
\t	Insere uma nova tabulação
\b	Insere um caractere de retorno
\n	Insere uma nova linha
\r	Insere um retorno de carro
\f	Insere um avanço de página
\'	Insere um caractere de aspas simples
\"	Insere um caractere de aspas duplas
\\	Insere um caractere de barra invertida

Sequência de Escape: Exemplo

- Se você quiser inserir aspas dentro de aspas, deverá usar a sequência de escape, \", nas aspas internas
  - Para imprimir a sequência...

```
O gato fez "Miau!" para mim.
```

  - Você escreveria

```
System.out.println("O gato fez \"Miau!\" para mim.");
```

Instruções de Impressão

- Escrever um texto em uma nova linha pode não imprimi-lo em uma nova linha:

```
System.out.println("Esta é a primeira linha."
        + " Esta NÃO é a segunda linha.");
```

Saída:
Esta é a primeira linha. Esta NÃO é a segunda linha.

- As sequências de escape podem ajudar a formatar sua saída:

```
System.out.println("Esta é a primeira linha. \n"
                 + "Esta é a segunda linha.");
```

Saída:
Esta é a primeira linha.
Esta é a segunda linha.

Mais Instruções de Impressão

- Existem dois métodos importantes para impressão:

```
System.out.println("Método Imprimir linha");
System.out.print("Método Imprimir");
```

- println funciona como se você estivesse inserindo \n no fim da instrução
- As duas instruções a seguir produzem resultados equivalentes:

```
System.out.println("Imprimindo ");
System.out.print("Imprimindo \n");
```

println() x print()

- println() cria automaticamente uma linha:

```
System.out.println("Esta é a primeira linha.");
System.out.println("Esta é a segunda linha.");
```

Saída:
Esta é a primeira linha.
Esta é a segunda linha.

- print() não cria automaticamente uma linha:

```
System.out.print("Esta é a primeira linha.");
System.out.print("Esta NÃO é a segunda linha.");
```

Saída:
Esta é a primeira linha. Esta NÃO é a segunda linha.

Atalho do NetBeans

Método Imprimir	Com que Frequência Usarei Esse Método?
`System.out.println()`	Frequentemente
`System.out.print()`	Raramente

- System.out.println() é usado com bastante frequência
- Mas requer muita digitação para configuração
- O Netbeans oferece um atalho:
  - Digite sout

```
sout
```

  - Pressione Tab

```
System.out.println("");
```

Imprimir um Grande Volume de Texto, Opção 1

- Dependendo do que você esteja tentando imprimir, pode ser que prefira:
  - Dividir uma única instrução de impressão em muitas linhas no NetBeans:

```
System.out.println("Esta é a primeira linha."
                + "Esta ainda é a primeira linha."
                + "É apenas uma linha muito longa "
                + "e eu não posso ver isso tudo no NetBeans."
                + "\n" + "Esta é a segunda linha."
                + "\n" + "Esta é a terceira linha.");
```

  - OU...

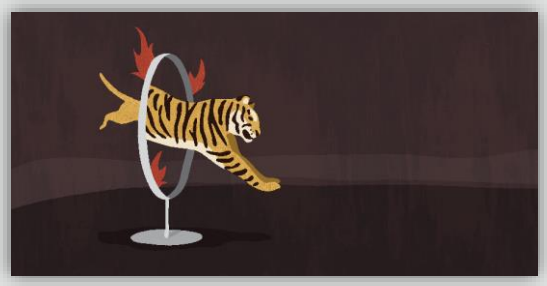
Imprimir um Grande Volume de Texto, Opção 2

- Usar muitas instruções de impressão:

```
System.out.println("Esta é a primeira linha.");
System.out.println("Esta é a segunda linha.");
System.out.println("Esta é a terceira linha.");
System.out.println("Esta é a quarta linha.");
```

Resumo

- Nesta lição, você deverá ter aprendido a:
  - Usar o tipo de dados char
  - Usar Strings
  - Concatenar Strings
  - Entender sequências de escape
  - Entender melhor as instruções de impressão



ORACLE Academy


ORACLE
Academy