

Números por extenso

Transcrição

Entrei novamente na página da Alura, para dar uma olhada nos valores.

The screenshot shows the Alura checkout process. The header has the Alura logo and 'Sua trilha na tecnologia'. The main title is 'Informações da sua compra'. Below it are three steps: 1. Suas informações (Nome completo), 2. Cartão (with logos for American Express, Mastercard, Visa, and PayPal), and 3. Confirmação (12x R\$ 75.00 (ou R\$ 900.00 à vista)).

O valor que aparece do lado direito da tela está escrito em algarismos. É interessante ter também esse valor por extenso, para acrescentar uma informação ao usuário. Há quem prefira fazer a leitura dos números dessa maneira. Essa forma também é muito comum em contratos, o que é uma peculiaridade do Brasil.

Podemos deixar o número por extenso com o Java. Começaremos criando uma nova classe, com o botão direito sobre o pacote e em `New > Class`. O nome da classe será `NumeroPorExtenso`. Como das outras vezes, começaremos criando o método `o`.

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){

    }

}
```

Usaremos uma instância do Stella para nos ajudar, a `NumericToWordsConverter(formato)`. Ela recebe o formato para o qual você deseja converter. Escolheremos o `new InteiroSemFormato()`, que é um construtor para o formato.

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        new NumericToWordsConverter(new InteiroSemFormato());
    }

}
```

Com o conversor, precisamos criar a instância efetivamente

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter coversor = NumericToWordsConverter(new InteiroSemFormato());
    }
}
```

Precisamos agora inserir o número a ser transformado. Usaremos o `double` .

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter coversor = NumericToWordsConverter(new InteiroSemFormato());
        double valor = 75,00;
    }
}
```

Com o `coversor` e o valor, `75,00` , aplicaremos um método ao conversor, o `to Words()` .

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter coversor = NumericToWordsConverter(new InteiroSemFormato());
        double valor = 75,00;
        coversor.toWords(valor);
    }
}
```

Com isso feito, seria interessante receber o número por extenso dentro de uma string, e já pediremos para imprimir.

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter coversor = NumericToWordsConverter(new InteiroSemFormato());
        double valor = 75,00;
        String valorPorExtenso = coversor.toWords(valor);
        System.out.println(valorPorExtenso);
    }
}
```

Ao salvar, console nos retorna o seguinte:

```
setenta e cinco
```

Ficou faltando especificar do que se trata. São setenta e cinco o quê? Reais. Vamos concatenar, portanto, com a palavra "Reais".

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter conversor = NumericToWordsConverter(new InteiroSemFormato());
        double valor = 75,00;
        String valorPorExtenso = conversor.toWords(valor);
        System.out.println(valorPorExtenso + " Reais");
    }
}
```

Atente para o espaço inserido dentro das aspas antes de " Reais". Ele garante que o número não fique grudado na unidade. Veja o que o console nos mostra quando rodamos:

```
setenta e cinco Reais
```

Usamos o `double` para inserir o valor, entretanto, ele não é a melhor opção pois não tem muita precisão em multiplicações. Para entender melhor, recomendo [esse artigo \(http://blog.caelum.com.br/arredondamento-no-java-do-double-ao-bigdecimal/\)](http://blog.caelum.com.br/arredondamento-no-java-do-double-ao-bigdecimal/), que discorre sobre arredondamento com `double` e com `BigDecimal`. Substituiremos, portanto, o `double` pelo `BigDecimal`, pois este nos serve melhor. Note que ele deve receber o valor como string.

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter conversor = NumericToWordsConverter(new InteiroSemFormato());
        BigDecimal valor = new BigDecimal("75,00");
        String valorPorExtenso = conversor.toWords(valor);
        System.out.println(valorPorExtenso + " Reais");
    }
}
```

Mas o `toWords` precisa receber o valor em `double`. Podemos resolver isso dentro de seu argumento, mantendo o valor em `BigDecimal`.

```
package br.com.alura;

public class NumeroPorExtenso {
```

```
public static void main(String[] args){
    NumericToWordsConverter conversor = NumericToWordsConverter(new InteiroSemFormato());
    BigDecimal valor = new BigDecimal("75,00");
    String valorPorExtenso = conversor.toWords(valor.doubleValue());
    System.out.println(valorPorExtenso + " Reais");

}
}
```

Para conferir se deu certo, vamos rodar novamente.

setenta e cinco Reais

Um valor inteiro é sempre agradável de se trabalhar, mas como se trata de dinheiro, nem sempre é o que teremos em mãos. O `InteiroSemFormato` nos limita um pouco. Seria interessante poder trabalhar com um formato de moeda. Para a nossa sorte, temos `FormatoDeReal`.

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter conversor = NumericToWordsConverter(new FormatoDeReal());
        BigDecimal valor = new BigDecimal("75,00");
        String valorPorExtenso = conversor.toWords(valor.doubleValue());
        System.out.println(valorPorExtenso + " Reais");

    }
}
```

Como ele é bem específico, já insere a unidade "reais" para nós na string. Assim, podemos tirar a concatenação.

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter conversor = NumericToWordsConverter(new FormatoDeReal());
        BigDecimal valor = new BigDecimal("75,00");
        String valorPorExtenso = conversor.toWords(valor.doubleValue());
        System.out.println(valorPorExtenso);

    }
}
```

Vejamos como ficou, ao rodarmos.

setenta e cinco reais

E se for apenas 1 real?

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter conversor = NumericToWordsConverter(new FormatoDeReal());
        BigDecimal valor = new BigDecimal("1");
        String valorPorExtenso = conversor.toWords(valor.doubleValue());
        System.out.println(valorPorExtenso);

    }
}
```

Ao rodar, temos:

```
um real
```

O `FormatoDeReal` identificou o singular, então não precisamos nos preocupar com o singular. Se tivéssemos mantido a concatenação, teria ficado sem concordância.

E se precisarmos lidar com centavos?

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter conversor = NumericToWordsConverter(new FormatoDeReal());
        BigDecimal valor = new BigDecimal("75.99");
        String valorPorExtenso = conversor.toWords(valor.doubleValue());
        System.out.println(valorPorExtenso);

    }
}
```

Salvando e rodando, temos:

```
setenta e cinco reais e noventa e nove centavos
```

O `FormatoDeReal` lida bem com os centavos também. Vamos testar mais um valor redondo.

```
package br.com.alura;

public class NumeroPorExtenso {

    public static void main(String[] args){
        NumericToWordsConverter conversor = NumericToWordsConverter(new FormatoDeReal());
        BigDecimal valor = new BigDecimal("900.00");
        String valorPorExtenso = conversor.toWords(valor.doubleValue());
        System.out.println(valorPorExtenso);
    }
}
```

```
}  
}
```

Rodando, vemos no console:

```
novecentos reais
```

No caso da página da Alura, o que programamos nos serviria bem, pois consegue comunicar os "75.00" e os "900.00 à vista"

The mockup shows a checkout page for Alura. At the top is the Alura logo with the tagline 'Sua trilha na tecnologia'. The main heading is 'Informações da sua compra'. Below this are three steps: 1. Suas informações, 2. Cartão, and 3. Confirmação. Step 1 has a label 'Nome completo' and an empty text input field. Step 2 shows logos for various payment methods including Mastercard, Visa, and PayPal. Step 3 displays the payment amount: '12x R\$ 75.00 (ou R\$ 900.00 à vista)'.

Lembrando que esse framework é livre e está no GitHub, você pode contribuir, usando a documentação caso tenha alguma dúvida. Há também o fórum para resolver questões e compartilhar experiências. Até a próxima!