

# I18N em Java

INE5661 - Tópicos Especiais  
I18N e L10N de Software  
Prof. Jose Eduardo De Lucca ([delucca@inf.ufsc.br](mailto:delucca@inf.ufsc.br))  
Baseado em material produzido por  
Fernando Rosa ([rosa@inf.ufsc.br](mailto:rosa@inf.ufsc.br))

## Internacionalização em Java

- Elementos de interface
  - mensagens e textos
  - carregados dinamicamente
- Suporte a novos idiomas não exige que o software seja recompilado

## Recursos para a I18N em Java

- Utiliza-se
  - *Locales para definir aspectos "culturais" locais*
  - Máscaras para a formatação de moedas, data, hora e números
  - Classe *Collator* para a ordenação de palavras
  - *Resource Bundles* para carregar dinamicamente as strings localizadas

## Classe *Locale*

- "Encapsula" uma língua, país e sistema operacional
- Locales pré-definidos
  - » Com formato de data, hora, moeda e números
- Parâmetro "língua" é obrigatório
- Na ausência de país e S.O. adota-se o "mais próximo" - Ex: *pt\_BR* / *pt*
- Parâmetro incorreto não gera erro (usa default)

## Classe *Locale*

- A classe *Locale* está no pacote *java.util.Locale*
- *getDefault()*
  - Locale padrão da máquina.
- *setDefault( objeto\_Locale )*
  - Para definir um Locale padrão.
- Exemplos
  - *en, en\_US, en\_GB\_UNIX*
  - *pt, pt\_PT, pt\_BR\_WIN*

## Locale em ação

```
import java.util.*;
public class Local
{
    public static void main(String args[])
    {
        String lingua = new String("pt");
        String pais = new String("BR");
        Locale local = new Locale(lingua,pais);
        Locale.setDefault(new Locale("en","US"));

        System.out.println(local);
        System.out.println(Locale.getDefault());
        Locale.setDefault(local);
        System.out.println(Locale.getDefault());
    }
}
```

```
C:\PROGRA-1>
pt_BR
en_US
pt_BR
Press any ke
```

## Classe Resource Bundle

- Classe Abstrata
- Classe de apoio onde são armazenadas as strings localizadas
- Implementação concreta é o *PropertyResourceBundle*
- Permite acesso dinâmico aos arquivos onde estão armazenadas as strings
  - Método getString

## Classe Resource Bundle

- Arquivos com traduções seguem formato:
  - palavra-chave=string\_localizado
- Codificação do arquivo deve ser ANSI
- Nomes dos arquivos devem seguir padrão
  - nome seguido do locale a que se destina
- Extensão do arquivo é *.properties*
- *Exemplo*
  - ArquivoDeMensagens\_en\_GB.properties

## Arquivo .properties

- Arquivo .properties:

```
ArquivoDeMensagens_en_GB.properties -
File Edit Format View Help
botaoOK = Ok Button!!!
botaoCancela = Canceled Button!!!
```

## Resource Bundle em ação

```
1 import java.util.Locale;
2 import java.util.ResourceBundle;
3
4 public class ExemploResourceBundle
5 {
6
7
8     public static void main(String args[])
9     {
10         Locale local = new Locale("en", "GB");
11         ResourceBundle mensagem;
12
13         mensagem = ResourceBundle.getBundle("ArquivoDeMensagens", local);
14
15         System.out.println(mensagem.getString("botaoOK"));
16         System.out.println(mensagem.getString("botaoCancela"));
17     }
18 }
```

```
ArquivoDeMensagens_en_GB.properties  C:\Program Files\Creator LE\GE2001.exe
File Edit Format View Help
botaoOK = Ok Button!!!
botaoCancela = Canceled Button!!!
Press any key to continue...
```

## Classe NumberFormat

- Diversas possibilidades
- Programa deve criar um "formatador", informando o locale.
  - O formatador é aplicado sempre que necessário
  - Ao aplicar o formato ao número, o resultado será um String formatado, pronto para ser apresentado.

## Formatação de Números

```
1 import java.util.Locale;
2 import java.text.NumberFormat;
3
4 public class ExemploFormataNumero
5 {
6     public static void main(String args[])
7     {
8         Locale regioao = new Locale("pt", "BR");
9         Locale regioao2 = new Locale("en", "US");
10        Double numero = new Double(1234567.89);
11        NumberFormat formataNumero = NumberFormat.getInstance(regiao);
12        String numeroFormatado = formataNumero.format(numero);
13        System.out.println("Numero Formatado : " + numeroFormatado + "\n\n" +
14        "Regiao do Formato: " + regioao.toString());
15
16        System.out.println("\n");
17
18        formataNumero = NumberFormat.getInstance(regiao2);
19        numeroFormatado = formataNumero.format(numero);
20        System.out.println("Numero Formatado : " + numeroFormatado + "\n\n" +
21        "Regiao do Formato: " + regioao2.toString());
22    }
23 }
```

## Formatação de Números

```
Locale regioao = new Locale("pt", "BR");
Locale regioao2=new Locale ("en", "US");
Double numero = new Double(1234567.89);
NumberFormat formataNumero = NumberFormat.getNumberInstance(regiao);
String numeroFormatado = formataNumero.format(numero);
System.out.println("Numero Formatado : " + numeroFormatado + "\r\n"+
"Regiao do Formato: " + regioao.toString());
```

```
Numero Formatado : 1.234.567,89
Regiao do Formato: pt_BR
```

```
Numero Formatado : 1,234,567.89
Regiao do Formato: en_US
Press any key to continue...
```

## DecimalFormat

```
1 import java.text.DecimalFormatSymbols;
2 import java.text.DecimalFormat;
3 import java.util.Locale;
4 public class ExemploFormataNumeroMascara
5 {
6     public static void main(String args[])
7     {
8         Locale regioao = new Locale("pt", "BR");
9         DecimalFormatSymbols simbolos = new DecimalFormatSymbols(regiao);
10        String mascara = "###,###.#";
11        DecimalFormat formato = new DecimalFormat(mascara, simbolos);
12        String valor = formato.format(12345.678);
13        System.out.println("Numero Formatado: "+valor+
14        "\r\n"+"Mascara Usada : "+mascara);
15    }
16 }
```

Podem ser definidas máscaras para formatar números

```
Numero Formatado: 12.345,7
Mascara Usada : ###,###.#
Press any key to continue...
```

.678 arredondado

Ponto ou vírgula?

## Formatação de Moedas

- A formatação de moedas é feita do mesmo modo da formatação básica de números
- Também é possível utilizar máscara

## Formatação de Moedas

```
1 import java.util.Locale;
2 import java.text.NumberFormat;
3
4 public class ExemploFormatoMoedaLocale
5 {
6     public static void main(String args[])
7     {
8         Locale regioao = new Locale("pt", "BR");
9         Locale regioao2=new Locale ("en", "US");
10        Double numero = new Double(1234567.89);
11        NumberFormat formataNumero = NumberFormat.getCurrencyInstance(regiao);
12        String numeroFormatado = formataNumero.format(numero);
13        System.out.println("Numero Formatado : " + numeroFormatado + "\r\n"+
14        "Regiao do Formato: " + regioao.toString());
15
16
17        System.out.println("\n");
18
19        formataNumero = NumberFormat.getCurrencyInstance(regiao2);
20        numeroFormatado = formataNumero.format(numero);
21        System.out.println("Numero Formatado : " + numeroFormatado + "\r\n"+
22        "Regiao do Formato: " + regioao2.toString());
23
24 }
```

## Formatação de Moedas

```
Locale regioao = new Locale("pt", "BR");
Locale regioao2=new Locale ("en", "US");
Double numero = new Double(1234567.89);
NumberFormat formataNumero = NumberFormat.getCurrencyInstance(regiao);
String numeroFormatado = formataNumero.format(numero);
System.out.println("Numero Formatado : " + numeroFormatado + "\r\n"+
"Regiao do Formato: " + regioao.toString());
```

```
Numero Formatado : R$ 1.234.567,89
Regiao do Formato: pt_BR
```

```
Numero Formatado : $1,234,567.89
Regiao do Formato: en_US
Press any key to continue...
```

## Formatação de Moedas

```
Locale regioao = new Locale("pt", "BR");
Locale regioao2=new Locale ("en", "US");
Double numero = new Double(1234567.89);
NumberFormat formataNumero = NumberFormat.getCurrencyInstance(regiao);
String numeroFormatado = formataNumero.format(numero);
System.out.println("Numero Formatado : " + numeroFormatado + "\r\n"+
"Regiao do Formato: " + regioao.toString());
```

```
Numero Formatado : R$ 1.234.567,89
Regiao do Formato: pt_BR
```

```
Locale regioao = new Locale("pt", "BR");
Locale regioao2=new Locale ("en", "US");
Double numero = new Double(1234567.89);
NumberFormat formataNumero = NumberFormat.getNumberInstance(regiao);
String numeroFormatado = formataNumero.format(numero);
System.out.println("Numero Formatado : " + numeroFormatado + "\r\n"+
"Regiao do Formato: " + regioao.toString());
```

## Formatação de Data e Hora

- Cada locale apresenta uma variedade de formatos diferentes
- Formatos básicos prontos
  - DateFormat
    - » DEFAULT, SHORT, MEDIUM, LONG, FULL
- Mecanismo para formatação adicional
  - Pode-se criar uma máscara com uma convenção de letras
  - SimpleDateFormat

## Formatação de Data e Hora

```
1 import java.util.*;
2 import java.text.*;
3
4 public class ExemploData
5 {
6     public static void main(String args[])
7     {
8         Locale regioao = new Locale("fr", "FR");
9         Locale regioao2 = new Locale("pt", "BR");
10        DateFormat formataData =
11            DateFormat.getDateInstance(DateFormat.FULL, DateFormat.FULL, regioao);
12
13        Date hoje = new Date();
14        String dataFormatada = formataData.format(hoje);
15        System.out.println("Data Formatada: "+dataFormatada+"\r\n"+Regiao : "+
16            regioao);
17        System.out.println("\n");
18        formataData =
19            DateFormat.getDateInstance(DateFormat.FULL, DateFormat.FULL, regioao2);
20
21        hoje = new Date();
22        dataFormatada = formataData.format(hoje);
23        System.out.println("Data Formatada: "+dataFormatada+"\r\n"+
24            "Regiao : "+regiao2);
25    }
26 }
```

## Formatação de Data e Hora

```
Locale regioao = new Locale("fr", "FR");
Locale regioao2 = new Locale("pt", "BR");
DateFormat formataData =
    DateFormat.getDateInstance(DateFormat.FULL, DateFormat.FULL, regioao);

Date hoje = new Date();
String dataFormatada = formataData.format(hoje);
System.out.println("Data Formatada: "+dataFormatada+"\r\n"+Regiao : "+
    regioao);
```

```
Data Formatada: jeudi 13 novembre 2003 01 h 28 GMT-03:00
Regiao : fr_FR
```

```
Data Formatada: Quinta-feira, 13 de Novembro de 2003 01h28min25s GMT-03:00
Regiao : pt_BR
```

## Formatação de Data e Hora

```
1 import java.util.*;
2 import java.text.*;
3
4 public class ExemploDataMascara
5 {
6     public static void main(String args[])
7     {
8         Locale regioao = new Locale("pt", "BR");
9         Date hoje = new Date();
10        String resultado;
11        SimpleDateFormat formata;
12        String mascara="dd/MM/yyyy G D E w HH:mm:ss";
13        formata = new SimpleDateFormat(mascara, regioao);
14        resultado = formata.format(hoje);
15        System.out.println("Data Formatada: "+resultado+"\r\n"+Regiao : "+
16            regioao);
17        System.out.println("\n");
18    }
19 }
20
```

```
C:\Program Files\JCreator LE\GE2001.exe
Data Formatada: 13/11/2003 AD 317 Qui 46 01:23:35
Regiao : pt_BR
```

## Formatação de Data e Hora

- |   |                            |
|---|----------------------------|
| ■ G - Era                                   | ■ h - hora (12 hs)         |
| ■ y - Ano                                   | ■ H - hora (24 hs)         |
| ■ M - Mês                                   | ■ m - minuto               |
| ■ d - Dia do mês                            | ■ s - segundo              |
| ■ E - Dia da semana (texto)                 | ■ S - milissegundo         |
| ■ D - Dia do ano                            | ■ K - só a hora (12 hs)    |
| ■ w - Semana do ano                         | ■ k - só a hora (24 hs)    |
| ■ W - Semana no mês                         | ■ a - texto AM/PM          |
| ■ F - dia da semana no mês (2a. Qua do mês) | ■ z - fuso horário (texto) |

## Formatação de Data e Hora

```
1 import java.util.*;
2 import java.text.*;
3
4 public class ExemploDataMascara
5 {
6     public static void main(String args[])
7     {
8         Locale regioao = new Locale("pt", "BR");
9         Date hoje = new Date();
10        String resultado;
11        SimpleDateFormat formata;
12        String mascara="dd/MM/yyyy G D E w HH:mm:ss";
13        formata = new SimpleDateFormat(mascara, regioao);
14        resultado = formata.format(hoje);
15        System.out.println("Data Formatada: "+resultado+"\r\n"+Regiao : "+
16            regioao);
17        System.out.println("\n");
18    }
19 }
20
```

Repetição dos códigos

Pontuação

```
C:\Program Files\JCreator LE\GE2001.exe
Data Formatada: 13/11/2003 AD 317 Qui 46 01:23:35
Regiao : pt_BR
```

## Exemplos de máscaras

- dd.MM.yy                    ■ 09.04.98
- yyyy.MM.dd G 'at'  
hh:mm:ss z                    ■ 1998.04.09 AD at  
                                  06:15:55 PDT
- EEE, MMM d, "yy            ■ Thu, Apr 9, '98
- h:mm a                        ■ 18:15:00
- H:mm                         ■ 18:15:00
- H:mm:ss:SSS                ■ 18:15:55:624
- K:mm a,z                     ■ 6:15 PM,PDT
- yyyy.MMMMMM.dd GGG      ■ 1998.April.09 AD 06:15  
hh:mm aaa                    PM

## Outros problemas

- Mensagens compostas
  - Excluir mensagens com mais de 120 dias.
  - O disco *Disco1* possui 157 arquivos.
  - Hay 157 ficheros en el disco *Disco1*.
- Mensagens variáveis
  - There are no files on Xdisk
  - There is 1 file on Xdisk
  - There are 2 files on Xdisk
  - Processando 1 arquivo, Processando 2 arquivos

## Outros problemas (ii)

- Comparações
- Ordenação
- Limites de caracteres, palavras e frases
  - Casa em árabe contém 3 letras و بيت
  - Mas são 6 caracteres Unicode!
- Processamento de textos
  - Delimitação de frases nem sempre é feita com ponto
  - Ponto nem sempre indica fim de frase.

## Para consultar

- Tutorial Java I18N
  - [java.sun.com/docs/books/tutorial/i18n](http://java.sun.com/docs/books/tutorial/i18n)
- FAQ Java I18N
  - [java.sun.com/j2se/corejava/intl/reference/faq](http://java.sun.com/j2se/corejava/intl/reference/faq)
- [www.i18nfaq.com/java.html](http://www.i18nfaq.com/java.html)

## Collator

- Classe para ordenação das palavras.
- Pode ser usada diretamente com um Locale (se for suportado pela versão)
- Ou se pode criar “regras” para a classe Collator, definindo a ordem dos caracteres para a ordenação

## Collator

```
1 import java.util.*;
2 import java.text.*;
3
4 public class ExemploCollator
5 {
6     public static void main(String args[])
7     {
8         Collator fr_FRCollator=Collator.getInstance(new Locale("fr","FR"));
9         Collator en_USCollator=Collator.getInstance(new Locale("en","US"));
10        String[] words={"sun", "peach", "peach"};
11        sortStrings(fr_FRCollator.words);
12        printStrings(words);
13        System.out.println("\n");
14        sortStrings(en_USCollator.words);
15        printStrings(words);
16    }
17
18    public static void sortStrings(Collator collator,String[] words)
19    {
20        String tap;
21        for(int i=0;i<words.length;i++)
22        {
23            for(int j=i+1;j<words.length;j++)
24            {
25                if(collator.compare(words[i],words[j])>0)
26                {
27                    tmp=words[i];
28                    words[i]=words[j];
29                    words[j]=tmp;
30                }
31            }
32        }
33    }
34
35    public static void printStrings(String[] words)
36    {
37        for (int i=0;i<words.length;i++)
38        {
39            System.out.println(words[i]);
40        }
41    }
42 }
```

