

### **Engenharia da Computação – 3ª série**

## **Internacionalizar em Java** **(L1/1, L2/1 e L3/1)**

**2025**

# ECM251 – Linguagens de Programação I

## Aula 15 – L1/1, L2/1 e L3/1

### Horário

Terça-feira: 2 x 2 aulas/semana

- L1/1 (07h40min-09h20min): *Prof. Calvetti*;
- L1/2 (09h30min-11h10min): *Prof. Calvetti*;
- L2/1 (07h40min-09h20min): *Prof. Menezes*;
- L2/2 (11h20min-13h00min): *Prof. Calvetti*;
- L3/1 (09h30min-11h10min): *Prof. Evandro*;
- L3/2 (11h20min-13h00min): *Prof. Evandro*.

### Tópico

- Internacionalização de uma aplicação

## *Internacionalização de uma aplicação*

### Definição



- Internacionalizar uma aplicação é apresentar todas as informações, fornecidas ao usuário, em um idioma selecionado;
- Para internacionalizar uma aplicação em Java é necessário cumprir as etapas apresentadas a seguir.

### Definição



1. Criar um arquivo de configuração, chamado **Bundle**, para cada idioma/país utilizado na aplicação, seguindo o padrão:

`<nomearquivo>_<linguagem>_<país>.properties`

### Definição



2. Fazer a carga desses arquivos através da classe:

*ResourceBundle*

### Definição



3. Invocar o método *getString()* da classe *ResourceBundle* para cada informação que será apresentada na tela;

### Definição



4. O conteúdo da aplicação sempre deverá seguir o padrão abaixo para cada uma das informação que serão internacionalizadas, observando que a chave deverá ser a mesma em toda a aplicação:

<chave> = <valor>



### Definição



5. Para que a aplicação seja internacionalizada é necessário substituir o valor da informação pela chamada do método:

*getString("<chave>")*

Ao invés de se colocar um texto fixo, deverá ser feita uma chamada ao método *getString()*, informando a chave do arquivo de **Bundle** que contém o valor que deverá ser apresentado no local;

### Definição



6. A carga do ***Bundle*** será feita com as classes *ResourceBundle* e *Locale*;

### Definição



7. Os arquivos de ***Bundle*** devem estar no *classpath* para que sejam carregados (e.g. na pasta de classes Java);

# ECM251 – Linguagens de Programação I

## Internacionalizar em Java

### Exemplo



```
1 import java.awt.Container;
2 import java.awt.FlowLayout;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.Locale;
6 import java.util.ResourceBundle;
7 import javax.swing.JButton;
8 import javax.swing.JFrame;
9 import javax.swing.JLabel;
10 import javax.swing.JOptionPane;
11 import javax.swing.JTextField;
12
13 public class TelaExemplo extends JFrame implements ActionListener
14 { private JButton bt;
15   private JTextField tx;
16   private JLabel rt;
17   private ResourceBundle bn = null;
18
19   public TelaExemplo()
20   { // Menu simples de escolha de idioma
21     int op = Integer.parseInt(JOptionPane.showInputDialog(
22       "Idioma - Language - Langue\n\n1- Português\n2- English\n3- Française\n "
23       )
24   );
```

# ECM251 – Linguagens de Programação I

## Internacionalizar em Java

### Exemplo



```
25 // Carga dos arquivos de internacionalização
26 switch(op)
27 { case 1: bn = ResourceBundle.getBundle("ex1", new Locale("pt", "BR"));
28     break;
29     case 2: bn = ResourceBundle.getBundle("ex1", Locale.US);
30     break;
31     case 3: bn = ResourceBundle.getBundle("ex1", Locale.FRANCE);
32     break;
33     default: bn = ResourceBundle.getBundle("ex1");
34     break;
35 }
36 // Escolhe Layout do container
37 Container cx = getContentPane();
38 cx.setLayout(new FlowLayout());
39 // Instânciação dos objetos
40 bt = new JButton(bn.getString("tela1.botao.calcular"));
41 rt = new JLabel(bn.getString("tela1.rotulo.valor") + ":");
42 tx = new JTextField(10);
43 // Inclusão no container
44 cx.add(rt);
45 cx.add(tx);
46 cx.add(bt);
47 // Registro no listener dos objetos controlados
48 bt.addActionListener(this);
49 // Ajustes finais do frame
50 setTitle(bn.getString("tela1.titulo"));
51 setSize(350, 100);
52 setVisible(true);
53 }
54
```

# ECM251 – Linguagens de Programação I

## Internacionalizar em Java

### Exemplo



```
55 // Implementacao do metodo da interface ActionListener
56 public void actionPerformed(ActionEvent e)
57 { if(tx.getText().length() == 0)
58   { JOptionPane.showMessageDialog(
59     null,
60     bn.getString("mensagem.valor.nulo"),
61     bn.getString("tela1.erro.titulo"),
62     JOptionPane.ERROR_MESSAGE
63   );
64   }else
65   { int    n = Integer.parseInt(tx.getText());
66     n = n * n;
67     tx.setText(""+n);
68   }
69 }
70 }
```

# ECM251 – Linguagens de Programação I

## Internacionalizar em Java

### Exemplo



```
1 import javax.swing.JFrame;  
2  
3 public class TelaExemploTest  
4 {   public static void main(String args[])  
5     {   TelaExemplo exemplo = new TelaExemplo();  
6         exemplo.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
7     }  
8 }
```

### Exemplo



Arquivo *Ex1\_pt\_BR.properties*

```
1 tela1.botao.calcular=Valor ^ 2
2 tela1.rotulo.valor=Valor
3 tela1.titulo=Exemplo
4 tela1.erro.titulo=Mensagem de Erro
5 mensagem.valor.nulo=0 campo "valor" não pode ser vazio.
6
```



### Exemplo



Arquivo *Ex1\_en\_US.properties*

```
1 tela1.botao.calcular=Value ^ 2
2 tela1.rotulo.valor=Value
3 tela1.titulo=Example
4 tela1.erro.titulo=Error Message
5 mensagem.valor.nulo=The field "value" can not be empty.
6
```

### Exemplo



Arquivo *Ex1\_fr\_FR.properties*

```
1 tela1.botao.calcular=Valeur ^ 2
2 tela1.rotulo.valor=Valeur
3 tela1.titulo=Exemple
4 tela1.erro.titulo=Message d'erreur
5 mensagem.valor.nulo=Le domaine de la "valeur" ne peut pas être vide.
6
```

### Exercícios



1. Reescreva as classes **TelaExemplo** e **TelaExemploTest**, fornecidas a seguir, adaptando a interface com o usuário do idioma Inglês para os idiomas Português, Inglês, Francês e Italiano. Para a escolha do idioma, insira essa opção de escolha no Menu, através de itens, no próprio contexto do aplicativo; e
2. Internacionalizar o exercício proposto 2 (dois) da aula 14 (pg. 39 do material “IMT-2025-ECM251-Lab-Aula14-ManipulaçãoDeÍconesSenhas&Datas-ProfCalvetti.pdf”) para os idiomas Português e Inglês, escolhidos através de um menu, antes do usuário realizar o seu “login”.

### Bibliografia Básica



- MILETTO, Evandro M.; BERTAGNOLLI, Silvia de Castro. Desenvolvimento de software II: introdução ao desenvolvimento web com HTML, CSS, javascript e PHP (Tekne). Porto Alegre: Bookman, 2014. E-book. Referência Minha Biblioteca:  
<https://integrada.minhabiblioteca.com.br/#/books/9788582601969>
- WINDER, Russel; GRAHAM, Roberts. Desenvolvendo Software em Java, 3ª edição. Rio de Janeiro: LTC, 2009. E-book. Referência Minha Biblioteca:  
<https://integrada.minhabiblioteca.com.br/#/books/978-85-216-1994-9>
- DEITEL, Paul; DEITEL, Harvey. Java: how to program early objects. Hoboken, N. J: Pearson, c2018. 1234 p. ISBN 9780134743356.

*Continua...*

### Bibliografia Básica (continuação)



- HORSTMANN, Cay S; CORNELL, Gary. Core Java. SCHAFRANSKI, Carlos (Trad.), FURMANKIEWICZ, Edson (Trad.). 8. ed. São Paulo: Pearson, 2010. v. 1. 383 p. ISBN 9788576053576.
- LIANG, Y. Daniel. Introduction to Java: programming and data structures comprehensive version. 11. ed. New York: Pearson, c2015. 1210 p. ISBN 9780134670942.
- TURINI, Rodrigo. Desbravando Java e orientação a objetos: um guia para o iniciante da linguagem. São Paulo: Casa do Código, [2017]. 222 p. (Caelum).

# ECM251 – Linguagens de Programação I

## Aula 15 – L1/1, L2/1 e L3/1

### Bibliografia Complementar



- HORSTMANN, Cay. Conceitos de Computação com Java. Porto Alegre: Bookman, 2009. E-book. Referência Minha Biblioteca:  
<https://integrada.minhabiblioteca.com.br/#/books/9788577804078>
- MACHADO, Rodrigo P.; FRANCO, Márcia H. I.; BERTAGNOLLI, Silvia de Castro. Desenvolvimento de software III: programação de sistemas web orientada a objetos em java (Tekne). Porto Alegre: Bookman, 2016. E-book. Referência Minha Biblioteca:  
<https://integrada.minhabiblioteca.com.br/#/books/9788582603710>
- BARRY, Paul. Use a cabeça! Python. Rio de Janeiro: Alta Books, 2012. 458 p.  
ISBN 9788576087434.

*Continua...*

# ECM251 – Linguagens de Programação I

## Aula 15 – L1/1, L2/1 e L3/1

### Bibliografia Complementar (continuação)



- LECHETA, Ricardo R. Web Services RESTful: aprenda a criar Web Services RESTful em Java na nuvem do Google. São Paulo: Novatec, c2015. 431 p.  
ISBN 9788575224540.
- SILVA, Maurício Samy. JQuery: a biblioteca do programador. 3. ed. rev. e ampl. São Paulo: Novatec, 2014. 544 p.  
ISBN 9788575223871.
- SUMMERFIELD, Mark. Programação em Python 3: uma introdução completa à linguagem Python. Rio de Janeiro: Alta Books, 2012. 506 p.  
ISBN 9788576083849.

*Continua...*

# ECM251 – Linguagens de Programação I

## Aula 15 – L1/1, L2/1 e L3/1

### Bibliografia Complementar (continuação)



- YING, Bai. Practical database programming with Java. New Jersey: John Wiley & Sons, c2011. 918 p.
- ZAKAS, Nicholas C. The principles of object-oriented JavaScript. San Francisco, CA: No Starch Press, c2014. 97 p. ISBN 9781593275402.



# ECM251 – Linguagens de Programação I

## Aula 15 – L1/1, L2/1 e L3/1

FIM

### **Engenharia da Computação – 3ª série**

## **Internacionalização em Java** **(L1/2, L2/2 e L3/2)**

**2025**

# ECM251 – Linguagens de Programação I

## Aula 15 – L1/2, L2/2 e L3/2

### Horário

Terça-feira: 2 x 2 aulas/semana

- L1/1 (07h40min-09h20min): *Prof. Calvetti*;
- L1/2 (09h30min-11h10min): *Prof. Calvetti*;
- L2/1 (07h40min-09h20min): *Prof. Menezes*;
- L2/2 (11h20min-13h00min): *Prof. Calvetti*;
- L3/1 (09h30min-11h10min): *Prof. Evandro*;
- L3/2 (11h20min-13h00min): *Prof. Evandro*.

### Exercícios



- Terminar, entregar e apresentar ao professor para avaliação, os exercícios propostos na aula de teoria, deste material.

### Bibliografia (apoio)



- LOPES, ANITA. GARCIA, GUTO. Introdução à Programação: 500 algoritmos resolvidos. Rio de Janeiro: Elsevier, 2002.
- DEITEL, P. DEITEL, H. Java: como programar. 8 Ed. São Paulo: Prentice-Hall (Pearson), 2010;
- BARNES, David J.; KÖLLING, Michael. Programação orientada a objetos com Java: uma introdução prática usando o BlueJ . 4. ed. São Paulo: Pearson Prentice Hall, 2009.

# ECM251 – Linguagens de Programação I

## Aula 15 – L1/2, L2/2 e L3/2

FIM