

Capítulo 24

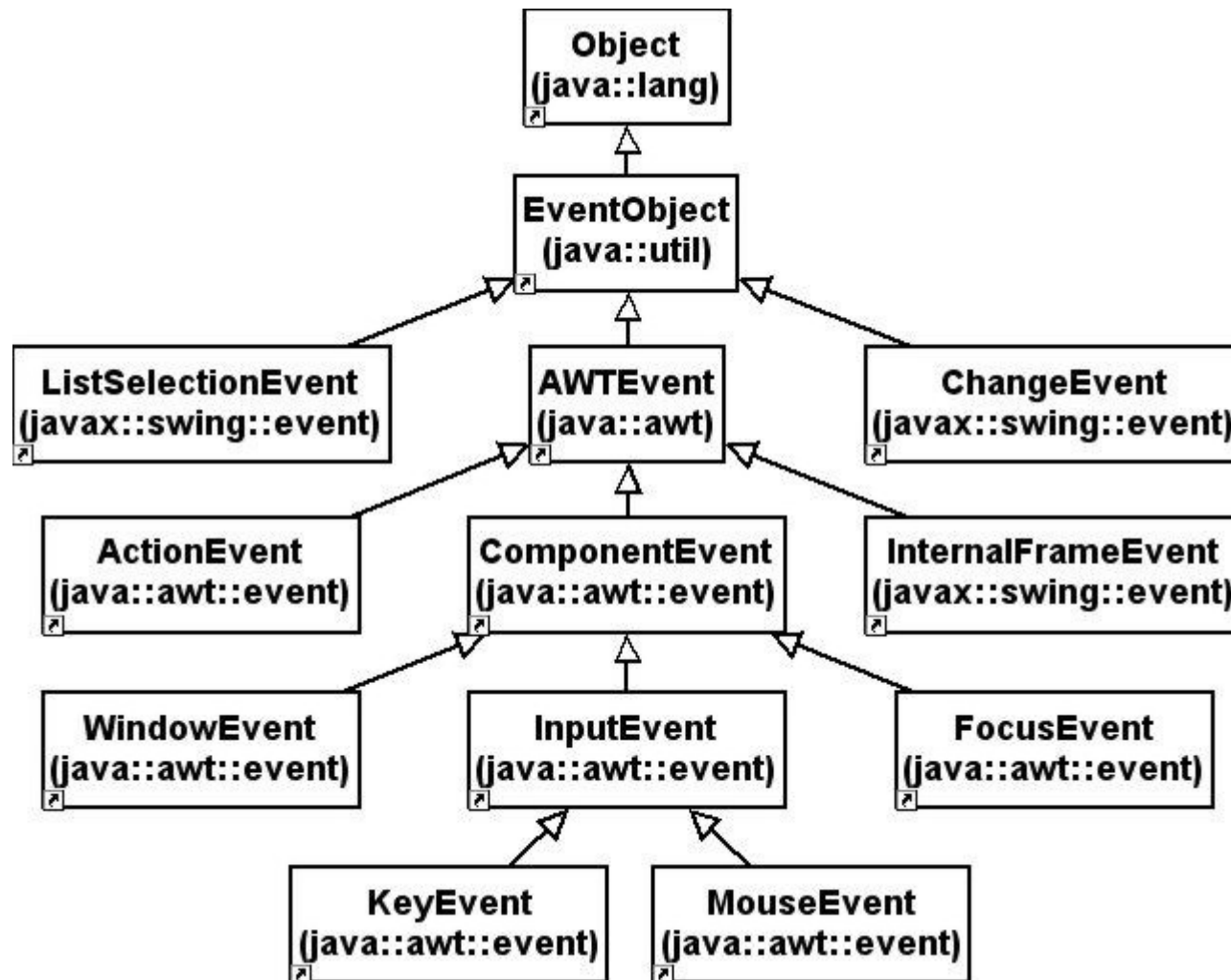
Tratamento de Eventos

Objetivos do Capítulo

- ❑ **Apresentar o funcionamento do mecanismo de tratamento de eventos aplicável à construção de GUIs.**
- ❑ **Explorar as principais classes e interfaces que devem ser empregadas no tratamento de eventos.**
- ❑ **Indicar diferentes caminhos que podem ser seguidos para realizar o tratamento de determinado tipo de eventos.**
- ❑ **Analisar exemplos concretos de interceptação e tratamento de eventos de janela, de ação, de componente, de foco, de teclado e de mouse.**

Introdução

□ Hierarquia de classes



Introdução

Evento	Interface	Métodos
ActionEvent	ActionListener	actionPerformed()
AdjustmentEvent	AdjustmentListener	adjustmentValueChanged()
ItemEvent	ItemListener	itemStateChanged()
TextEvent	TextListener	textValueChanged()
ComponentEvent	ComponentListener	componentMoved() componentHidden() componentResized() componentShown()
ContainerEvent	ContainerListener	componentAdded() componentRemoved()
FocusEvent	FocusListener	focusGained() focusLost()

Introdução

Evento	Interface	Métodos
KeyEvent	KeyListener	keyPressed() keyReleased() keyTyped()
MouseEvent	MouseListener	mousePressed() mouseReleased() mouseEntered() mouseExited() mouseClicked()
WindowEvent	WindowListener	windowClosing() windowOpened() windowIconified() windowDeiconified() windowClosed() windowActivated()

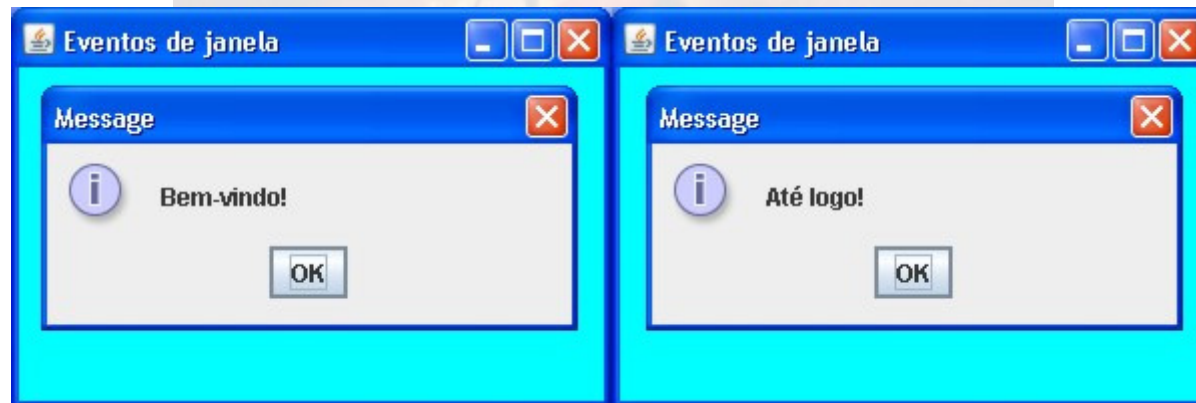
Introdução

□ Conceitos fundamentais

- **Fonte: componente gerador do evento**
- **Evento**
 - ❖ Representado por uma classe
 - ❖ Interface: métodos a serem implementados
- **Ouvinte**
 - ❖ Objeto a ser notificado
 - ❖ Responsável pelo tratamento
 - ❖ Deve realizar a interface correspondente ao evento
 - ❖ Suporte a múltiplos ouvintes
- **Classe adaptadora**
 - ❖ Realiza a interface de um tipo de evento
 - ❖ Disponível para eventos com 2 ou mais métodos
 - ❖ Ex.: WindowEvent -> WindowAdapter
- **Registro**
 - ❖ Adição de objeto à lista de ouvintes
 - ❖ Ex.: `botao.addActionListener(<ouvinte>)`

Eventos de Janela

- ❑ **Código 24.1 – EventoJanela1.java**
- ❑ **Código 24.2 – EventoJanela2.java**
- ❑ **Código 24.3 – EventoJanela3.java**
- ❑ **Código 24.4 – EventoJanela4.java**
- ❑ **Código 24.5 – EventoJanela5.java**



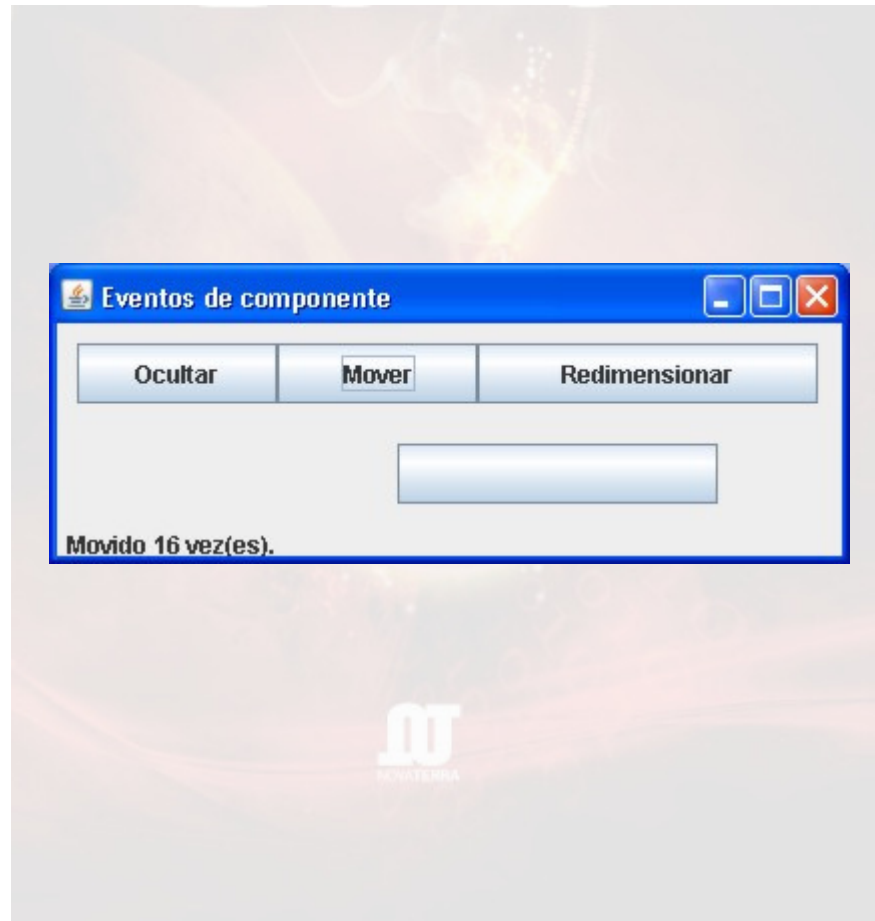
Eventos de Ação

❑ Código 24.6 – EventoAcao.java



Eventos de Componente

❑ Código 24.7 – EventoComponente.java



Eventos de Foco

❑ Código 24.8 – EventoFoco.java



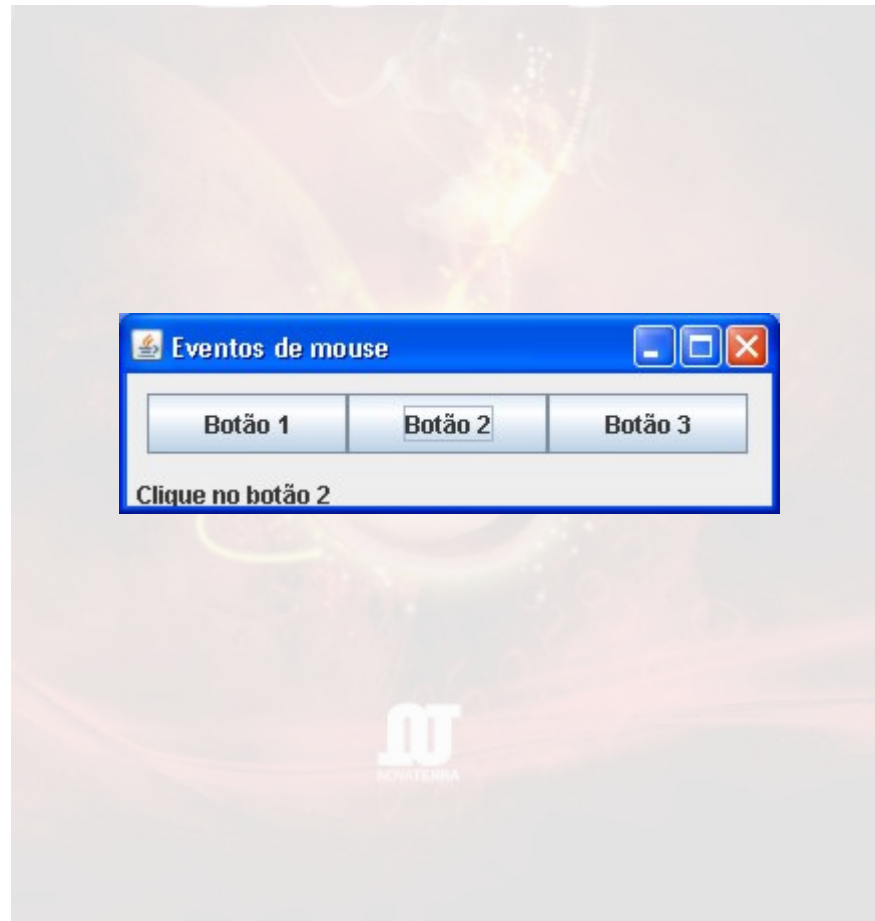
Eventos de Teclado

❑ Código 24.9 – EventoTeclado.java



Eventos de Mouse

❑ Código 24.10 – EventoMouse.java



Exercícios

- ❑ **A resolução de cada um dos exercícios que são propostos a seguir envolve a construção de uma janela e o tratamento de algum tipo de evento.**
 - **O primeiro e o segundo exercício envolvem o tratamento de eventos de ação.**
 - **O terceiro exercício envolve o tratamento de eventos de mouse.**
 - **O quarto exercício envolve o tratamento de eventos de teclado.**
 - **O quinto exercício envolve o tratamento de eventos de foco e de ação.**
- ❑ **Em cada exercício, há uma figura que apresenta a aparência esperada para a janela que deve ser construída.**
 - **Procure construir a sua própria janela de modo a se aproximar deste modelo.**
 - **Além disso, procure seguir à risca todas as especificações acerca do funcionamento da mesma.**

Exercício 1

- ❑ **Crie uma nova janela, chamada TrocaFundo, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela deve ter 300 pixels de largura e 150 pixels de altura e deve iniciar com a cor de fundo padrão, ou seja, cinza claro.**
- ❑ **Cada vez que um dos botões desta janela for pressionado, sua cor de fundo deve ser alterada para uma tonalidade clara da cor correspondente ao texto do botão.**



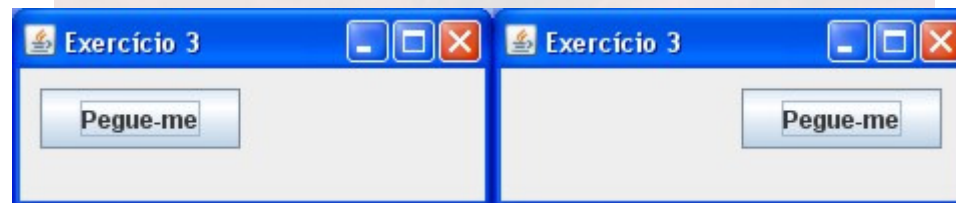
Exercício 2

- ❑ **Crie uma nova janela, chamada SorteioFundo, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela deve ter apenas um botão, rotulado como “Sortear Cor”.**
 - **Cada vez que este botão for pressionado, deve ser aplicada uma nova cor para o fundo da janela.**
- ❑ **A escolha da nova cor deve ser realizada de forma aleatória.**
 - **Lembre-se que uma cor é representada por uma instância da classe `java.awt.Color` e que ela pode ser criada utilizando um construtor que exige três números inteiros como argumentos.**
 - **Cada um destes argumentos representa, respectivamente, a intensidade de vermelho, de verde e de azul que compõe a cor. Estas intensidades podem variar de 0 a 255.**



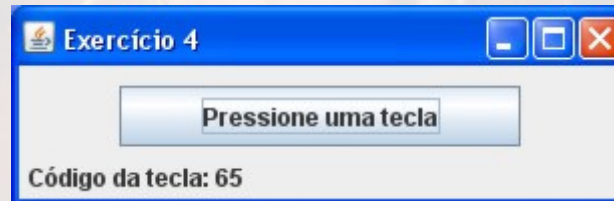
Exercício 3

- ❑ **Crie uma nova janela, chamada BotaoFugitivo, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela deve ter apenas um botão.**
 - **Inicialmente, este botão deve ser disposto do lado esquerdo da janela.**
 - **Cada vez que o mouse for movido para dentro da área deste botão, ele deve deslocar-se imediatamente para o lado oposto da janela.**
 - **Este botão não deve permitir que você estacione o mouse sobre ele, provocando a impressão de ele está fugindo do mouse.**



Exercício 4

- ❑ **Crie uma nova janela, chamada `CodigoTecla`, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela deve ter apenas um botão e um rótulo.**
 - **O botão deve ser disposto na parte superior da janela e o rótulo deve ser disposto na parte inferior.**
 - **Cada vez que uma tecla for pressionada sobre o botão, o código da tecla pressionada deve ser apresentado no rótulo.**



Exercício 5

- ❑ **Crie uma nova janela, chamada NavegadorVetor, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela deve criar um vetor e nele armazenar 1.000 números. Utilize o método `nextInt()` da classe `java.util.Random` para gerar, aleatoriamente, os números a serem gravados neste vetor.**
 - **Os números gerados não devem estar no intervalo de 10.000 a 50.000.**



Exercício 5

- ❑ **Os componentes gráficos desta janela devem permitir que você percorra o vetor supracitado para inspecionar o número contido em cada posição.**
 - **O primeiro botão mudará para a primeira posição do vetor.**
 - **O segundo botão mudará para a posição anterior à atual.**
 - **O terceiro botão mudará para a posição posterior à atual.**
 - **O último botão mudará para a última posição do vetor.**
 - **Os rótulos que estão em negrito servem para apresentar a posição atual do vetor e o número que se encontra nela.**
- ❑ **Utilize uma variável para representar a posição atual em que o navegador encontra-se no vetor.**
 - **Quando a janela for aberta, posicione o navegador na primeira posição do vetor e exiba o seu conteúdo.**
 - **Sempre que um dos botões for pressionado, mude a posição do navegador e altere a posição e o número que estão sendo apresentados.**
- ❑ **Os botões devem ser apresentados com a cor de fundo branca.**
 - **Sempre que um deles receber o foco, ela deve ser alterada para uma tonalidade clara de verde.**
 - **Quando perder o foco, sua cor de fundo deve voltar a ser branca.**

Contato

Com o autor:

Rui Rossi dos Santos

E-mail: livros@ruirossi.pro.br

Web Site: <http://www.ruirossi.pro.br>

Com a editora:

Editora NovaTerra

Telefone: (21) 2218-5314

Web Site: <http://www.editoranovatterra.com.br>

