

Manipulando Eventos em Java



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
PROF. VITOR YANO

Eventos



- Na programação orientada a objetos, pode não haver uma ordem determinada de execução de um programa;
- Janelas e interfaces gráficas não servem para nada sem os eventos;
- Quando um usuário interage com os componentes da tela, estes geram eventos;
- O tratamento desses eventos (o que fazer quando ocorrem) é feito por tratadores de eventos.

Eventos



- Exemplos de eventos:
 - Pressionar um botão;
 - Clicar com o mouse;
 - Arrastar uma janela;
 - Fechar uma janela;
 - Digitar um caractere.

Ouvintes de evento

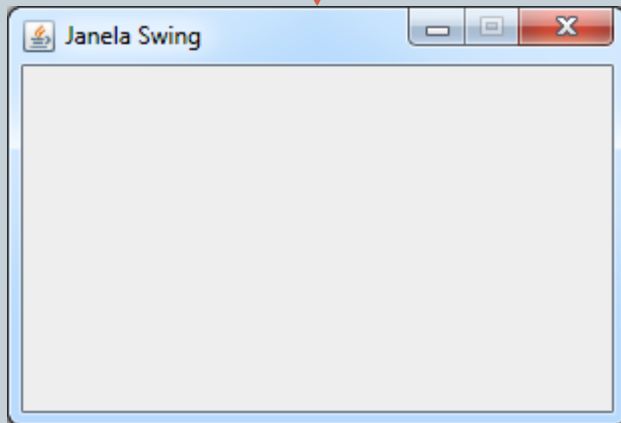


- O Swing trata eventos através de interfaces chamadas “ouvintes de eventos” (**EventListener**);
- Quando um ouvinte é associado a um componente, ele passa a responder aos eventos ocorridos;
- O sistema operacional (no caso do Java, a máquina virtual), monitora todas as ações que são executadas, seja pelo usuário ou por outros programas;
- Quando um evento ocorre, ele “avisa” o ouvinte;
- Exemplo: **WindowListener**, visto na aula passada.

Exemplo



Máquina virtual Java



WindowListener

```
...  
public void windowClosing(WindowEvent we) {  
    System.exit(0);  
}  
...
```

Ouvintes de evento



- Os ouvintes de evento básicos são interfaces, todas contidas no pacote **java.awt.event**:
 - ActionListener – eventos de ação
 - AdjustmentListener – eventos de ajuste
 - FocusListener – eventos de foco
 - ItemListener – eventos de item
 - KeyListener – eventos do teclado
 - MouseListener – eventos do mouse
 - MouseMotionListener – eventos de movimento do mouse
 - TextListener – eventos de texto
 - WindowListener – eventos de janela

Configurando componentes



- Para associar um ouvinte de evento, deve-se usar um dos seguintes métodos no componente:
 - addActionListener()
 - addFocusListener()
 - addItemListener()
 - addKeyListener()
 - addMouseListener()
 - addMouseMotionListener()
 - addWindowListener()

ActionListener



```
public interface ActionListener extends EventListener
{
    public void actionPerformed(ActionEvent ae);
}
```


FocusListener



```
public interface FocusListener extends EventListener {  
  
    public void focusGained(FocusEvent fe);  
  
    public void focusLost(FocusEvent fe);  
}
```

ItemListener



```
public interface ItemListener extends EventListener {  
  
    public void itemStateChanged(ItemEvent ie);  
  
}
```

- Para determinar em que item ocorreu o evento, use o método `getItem()` no objeto `ie` (`ItemEvent`).

KeyListener



```
public interface KeyListener extends EventListener {  
  
    public void keyTyped(KeyEvent ke);  
  
    public void keyPressed(KeyEvent ke);  
  
    public void keyReleased(KeyEvent ke);  
}
```

- Para determinar a tecla associada ao evento, usa-se o método `getKeyChar()` no objeto `ke` (`KeyEvent`).

MouseListener



```
public interface MouseListener extends EventListener {  
  
    public void mouseClicked(MouseEvent me);  
  
    public void mousePressed(MouseEvent me);  
  
    public void mouseReleased(MouseEvent me);  
  
    public void mouseEntered(MouseEvent me);  
  
    public void mouseExited(MouseEvent me);  
  
}
```

MouseEventListener



```
public interface MouseEventListener extends EventListener {  
  
    public void mouseDragged(MouseEvent me);  
  
    public void mouseMoved(MouseEvent me);  
}
```

- Alguns métodos que podem ser usados no objeto `MouseEvent` são:
 - `int getClickCount()` – retorna quantas vezes o mouse foi clicado
 - `Point getPoint()` – retorna as coordenadas x,y dentro do componente onde ocorreu o evento do mouse
 - `int getX()` – retorna a posição x
 - `int getY()` – retorna a posição y

TextListener



```
public interface TextListener extends EventListener {  
  
    public void textValueChanged(TextEvent te);  
  
}
```

WindowListener



```
public interface WindowListener extends EventListener {  
  
    public void windowOpened(WindowEvent we);  
  
    public void windowClosing(WindowEvent we);  
  
    public void windowClosed(WindowEvent we);  
  
    public void windowIconified(WindowEvent we);  
  
    public void windowDeiconified(WindowEvent we);  
  
    public void windowActivated(WindowEvent we);  
  
    public void windowDeactivated(WindowEvent we);  
  
}
```

Exemplos



- Livro “Aprenda Java 2 em 21 dias”, Rogers Cadenhead e Laura Lemay
- Capítulo 10 - montando uma interface Swing
- Capítulo 11 - organizando componentes em uma interface com o usuário
- Capítulo 12 - respondendo à entrada do usuário