



Curso de Java

by Antonio Rodrigues Carvalho Neto





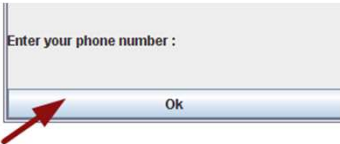
Eventos e Ouvintes



Events & Listeners

- Eventos são ações executadas por elementos internos ou externos ao sistema e que podem ser compreendidas pela aplicação. Neste caso lidaremos com eventos provocados pela ação do usuário.
- Quando um evento ocorre o sistema operacional verifica e aciona a aplicação avisando sobre o evento.

Events & Listeners

	Evento	Interface Listener	Método a ser implementado
	MouseEvent	MouseListener	mousePressed(MouseEvent e) mouseReleased(MouseEvent e) mouseClicked(MouseEvent e) mouseEntered(MouseEvent e) mouseExited(MouseEvent e)
	MouseWheelEvent	MouseWheelListener	mouseMoved(MouseWheelEvent e) mouseDragged(MouseWheelEvent e)
	KeyEvent	KeyListener	keyPressed(KeyEvent e) keyReleased(KeyEvent e) keyTyped(KeyEvent e)
	ActionEvent	ActionListener	actionPerformed(ActionEvent e)



Events & Listeners

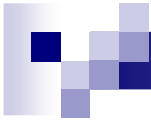
- Eventos são gerados e trafegam a todo momento entre as aplicações. Para trabalhar com eles precisamos apenas preparar nossa aplicação para escutá-los.
- Esta preparação envolve 5 etapas conforme a seguir:
 1. Preparar um código para ser executado quando o evento for acionado.
 2. Criar uma classe de ouvinte que implemente o Listener correspondente ao Evento.
 3. Implemente todos os métodos da interface, lembrando de colocar o código preparado na etapa 1 dentro do método correspondente ao Evento desejado.
 4. Crie um objeto do tipo da classe ouvinte.
 5. Adicione o objeto à fila de ouvintes do componente que irá se submeter ao evento.



Events & Listeners

- Exemplo, criar um ação para fazer o botão **ok** da janela mostrar a mensagem “Ola...” na console do JVM:
 1. Preparar um código para ser executado quando o evento for acionado.
System.out.println(“Ola”);
 2. Criar uma classe de ouvinte que implemente o Listener correspondente ao Evento.
// O Evento que estamos querendo capturar é o ActionEvent
// O Listener relativo a este evento é ActionListener
public class Ouvinte implements ActionListener {

}
 3. Implemente todos os métodos da interface, lembrando de colocar o código preparado na etapa 1 dentro do método correspondente ao Evento desejado.
public class Ouvinte implements ActionListener {
public void actionPerformed(ActionEvent e) {
System.out.println(“Ola”);
}
}
 4. Crie um objeto do tipo da classe ouvinte.
Ouvinte ouv = new Ouvinte();
 5. Adicione o objeto à fila de ouvintes do componente que irá se submeter ao evento.
btnOk.addActionListener(ouv);
// Estamos assumindo que este trecho está sendo inserido em um código que já contenha o botão btnOk criado.



Events & Listeners

```
public class Ouvinte implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.out.println("Ola ....");
    }
}

public class Teste {
    public static void main( String args[] ) {
        JFrame janela = new JFrame("Events & Listeners");
        JButton btnOk = new JButton("Ok");

        janela.getContentPane().add( btnOk );

        Ouvinte ouv = new Ouvinte ();
        btnOk.addActionListener( ouv );

        janela.setSize(600, 400);
        janela.setVisible(true);
    }
}
```

Exercício - 1

- Monte uma janela de teste que exiba um texto qualquer, e que contenha um botão **Sair** o qual fecha a janela e encerra a aplicação (conforme figura abaixo).



- **Dica :** Utilize o comando **System.exit(0)** para encerrar a aplicação java



Exercício - 2

- Crie uma classe chamada Play que herde a classe Container. Adicione dois atributos para esta classe (X e Y)
- Sobrescreva o método paint para pintar um círculo na cordenada X e Y com comprimento 20 e altura 20.
- Crie um construtor para a classe Play que inicie os atributos X e Y com o valor 10 (X = 10 e Y = 10).
- Faça com que a classe Play implemente a interface **KeyListener**. Sobrescreva os métodos keyPressed, keyReleased e keyType para implementar apenas { }
- Modifique o código keyPressed para fazer o seguinte:
 - ☐ Aumentar o valor de X quando o getKeyCode() == 39
 - ☐ Diminuir o valor de X quando o getKeyCode() == 37
 - ☐ Aumentar o valor de Y quando o getKeyCode() == 40
 - ☐ Diminuir o valor de Y quando o getKeyCode() == 38



Exercício - 3

- Modifique a calculadora feita na aula de Interface Gráfica para que os botões mostrem os números no display da calculadora.
- **Desafio** : Modifique o exercício para que a calculadora efetue cálculos.



Referências

- Java como programar 6ª edição

Capítulo 11

- Use a cabeça Java

Capitulo 12

pags. 254 a 257

pags. 263 a 272