



# Java: Eventos do Mouse

Levemente modificado da apresentação  
“Interfaces gráficas utilizando a JFC” de  
Frederico Costa Guedes Pereira © 2000





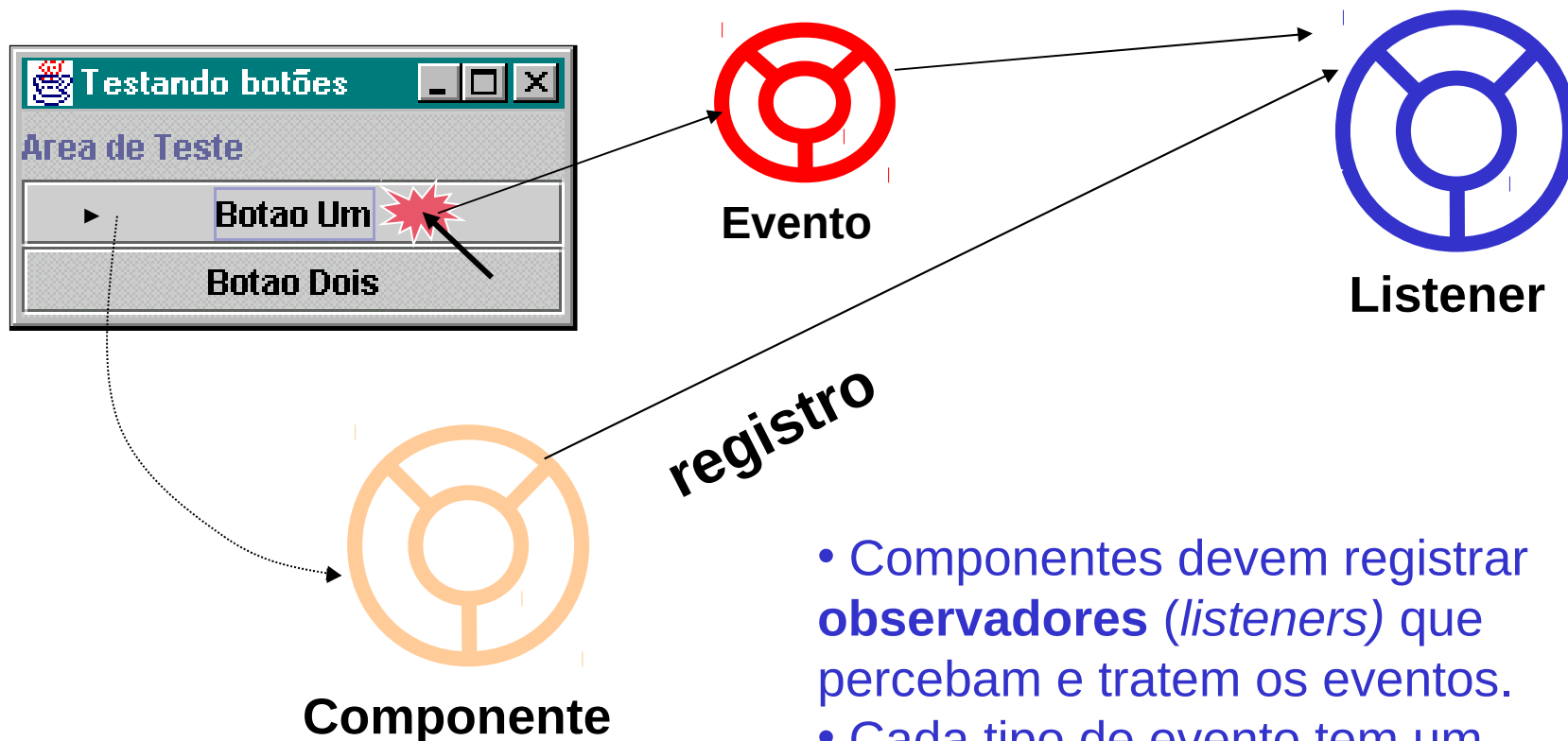
# Modelo de Eventos de Java

---

- ◆ GUIs são orientadas a eventos, i. e., geram eventos quando o usuário interage com seus componentes.
- ◆ Interação = clicar o mouse, clicar um botão, digitar num campo de texto, fechar uma janela, mover o ponteiro,...
- ◆ A cada interação um evento é gerado e enviado ao programa.



# O Modelo de Eventos de Java



- Componentes devem registrar **observadores** (*listeners*) que percebam e tratem os eventos.
- Cada tipo de evento tem um **observador** específico.



# O Modelo de Eventos de Java

- ◆ Criação do objeto *observador* (no construtor!):  
**TrataBotoes tratador = new TrataBotoes();**
- ◆ Registra *observador* para os botões (idem):  
**botao1.addActionListener(tratador);**  
...  
**botao2.addActionListener(tratador);**
- ◆ Método do *observador* que trata o evento:  
**public void actionPerformed(ActionEvent e){**  
...  
**}**



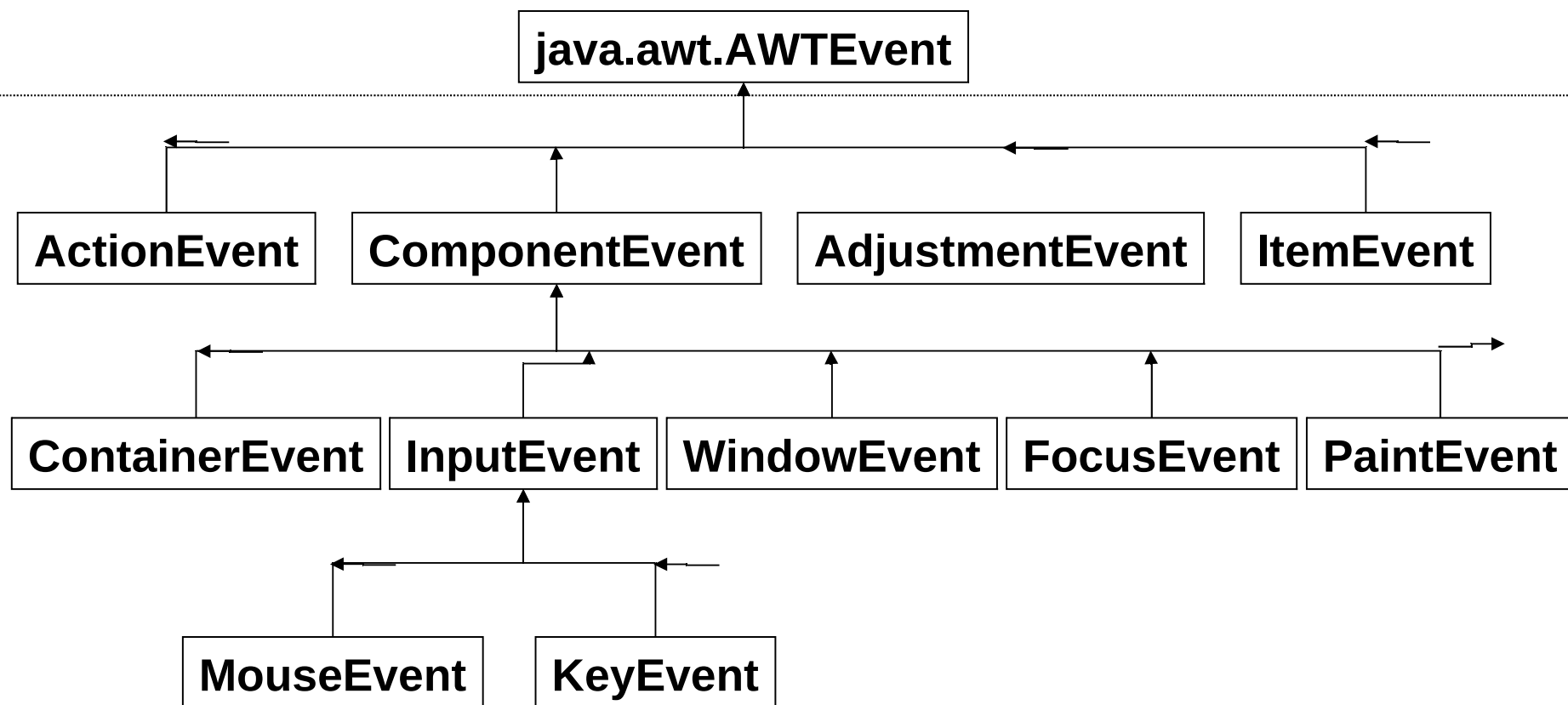
# Modelo de Eventos de Java

---

- ◆ Eventos são classes, observadores são interfaces! (que devem ser implementados)
- ◆ Criar um observador de eventos para um componente consiste em criar uma classe que implemente a interface *observadora* do evento a ser tratado.
- ◆ Cada uma das interfaces observadoras define um ou mais métodos tratadores de eventos.



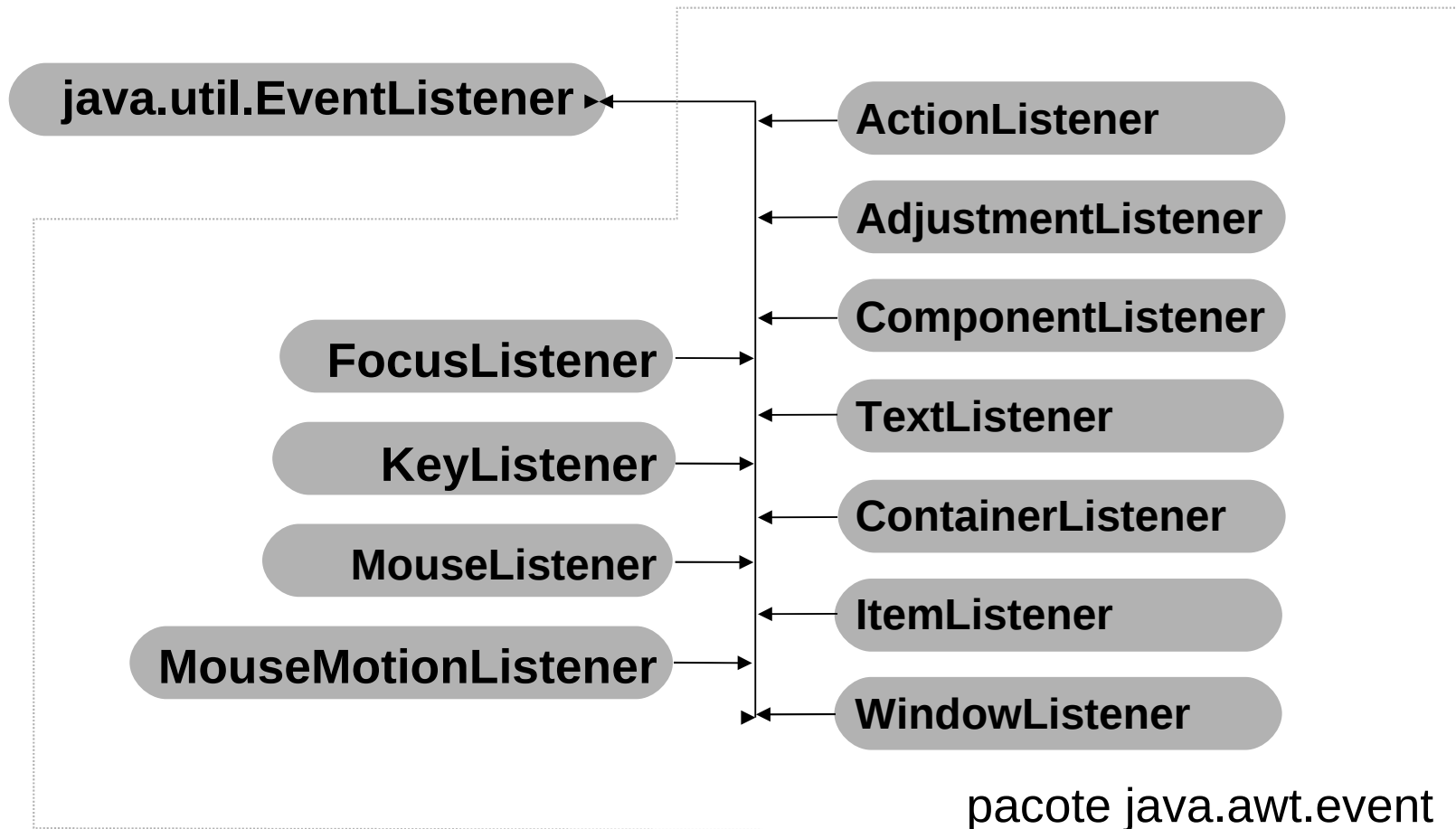
# Hierarquia de Eventos



pacote java.awt.event



# Hierarquia de Observadores





# Interface MouseListener

- ◆ Eventos capturados:
  - ⇒ MouseEvent
- ◆ Método(s) definidos:
  - ⇒ void mousePressed(MouseEvent e)
  - ⇒ void mouseClicked(MouseEvent e)
  - ⇒ void mouseReleased(MouseEvent e)
  - ⇒ void mouseEntered(MouseEvent e)
  - ⇒ void mouseExited(MouseEvent e)
- ◆ Método usado para registrar:
  - ⇒ addMouseListener(this)
  - ⇒ No caso de applets dentro do método init()



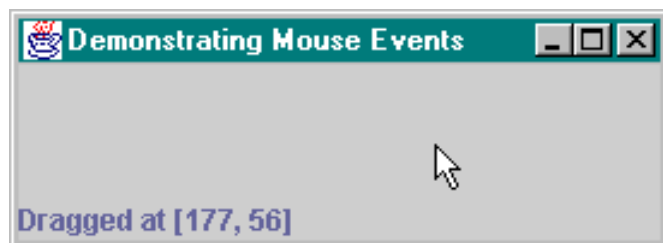
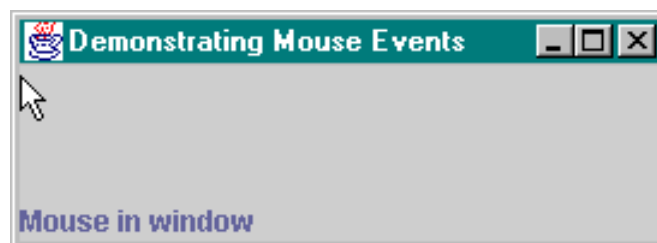
# Interface

## MouseListener

- ◆ Eventos capturados:
  - ⇒ MouseEvent
- ◆ Método(s) definidos:
  - ⇒ void mouseDragged(MouseEvent e)
  - ⇒ void mouseMoved(MouseEvent e)
- ◆ Método usado para registrar o listener:
  - ⇒ addMouseListener(this)
  - ⇒ No caso de applets dentro do método init()



# Exemplo





```
public class MouseTracker extends JFrame
    implements MouseListener,
                MouseMotionListener {
    private JLabel statusBar;

    public MouseTracker(){
        super("Demonstrating Mouse Events");

        statusBar = new JLabel();
        getContentPane().add( statusBar,
                                BorderLayout.SOUTH );

        this.addMouseListener(this);
        this.addMouseMotionListener(this);

        setSize(275, 100);
        show();
    } //obs: a classe não acaba aqui!
```



```
// Tratadores de MouseListener
```

```
public void mouseClicked( MouseEvent e ){
    statusBar.setText( "Clicked at [" + e.getX() +
                        ", " + e.getY() + "]" );
}
public void mousePressed( MouseEvent e ){
    statusBar.setText( "Pressed at [" + e.getX() +
                        ", " + e.getY() + "]" );
}
public void mouseReleased( MouseEvent e ){
    statusBar.setText( "Released at [" + e.getX() +
                        ", " + e.getY() + "]" );
}
public void mouseEntered( MouseEvent e ){
    statusBar.setText( "Mouse in window" );
}
public void mouseExited( MouseEvent e ){
    statusBar.setText( "Mouse outside window" );
}
```



```
// Tradadores de MouseMotionListener
```

```
public void mouseDragged( MouseEvent e ){
    statusBar.setText( "Dragged at [" + e.getX() +
                      ", " + e.getY() + "]" );
}
public void mouseMoved( MouseEvent e ){
    statusBar.setText( "Moved at [" + e.getX() +
                      ", " + e.getY() + "]" );
}
public static void main( String args[] ){
    MouseTracker app = new MouseTracker();
    app.addWindowListener(
        new WindowAdapter() {
            public void windowClosing( WindowEvent e ){
                System.exit( 0 );
            }
        }
    );
}
```



# Classe MouseEvent

- ◆ Alguns métodos:
  - ⇒ `int getX()`: Devolve a coordenada x onde o MouseEvent ocorreu.
  - ⇒ `int getY()`: Devolve a coordenada y onde o MouseEvent ocorreu.
  - ⇒ `int getClickCount()`: informa quantas vezes o mouse foi clicado.
  - ⇒ `boolean isMetaDown()`: Botão da direita do mouse foi usado para clicar.
  - ⇒ `boolean isAltDown()`: Botão do meio do mouse foi usado para clicar.