

Programação Orientada a Objetos

1

BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

PROF. LUCIO AGOSTINHO ROCHA

AULA 29: INTERFACES GRÁFICAS DO USUÁRIO

MODELO DE EVENTOS

1º.SEMESTRE 2023

Modelo de Eventos

Modelo de Eventos

3

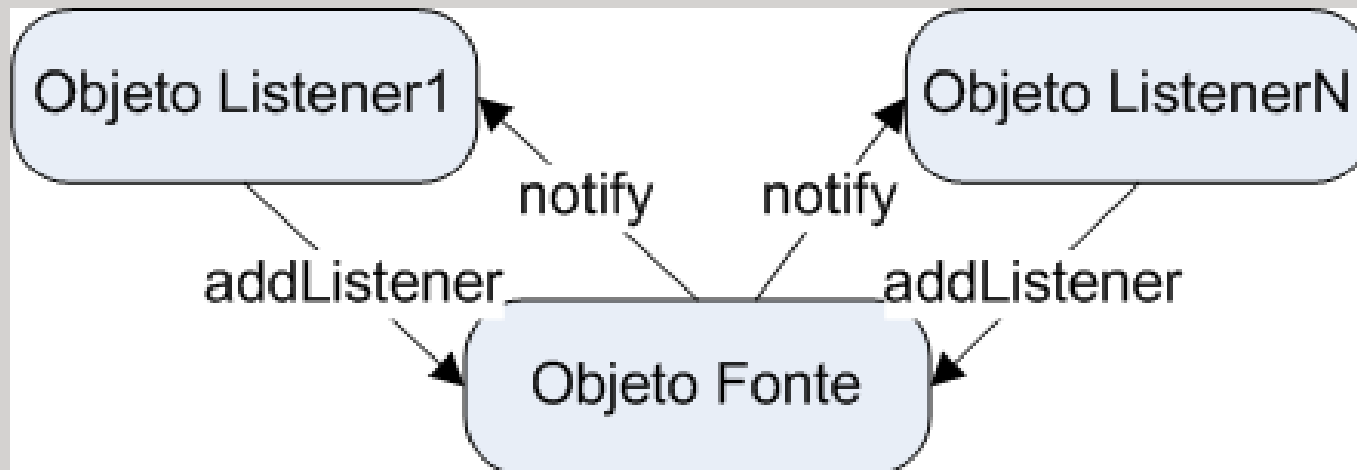


Figura: Modelo de Eventos.

Modelo de Eventos

4

-Objeto Fonte:

- JComponent: JLabel, JButton, JTextField, etc.

-Listener:

- Um objeto que “escuta” interações do objeto fonte.

-Mensagem de Notificação:

- Uma mensagem enviada do objeto fonte para o Listener, indicando que um evento aconteceu.

-Evento:

- É uma ação realizada sobre o objeto fonte.

- Cada objeto fonte pode adicionar um ou mais objetos listener próprios para os seus eventos.

Modelo de Eventos

5

- O objeto fonte pode estar associado a vários eventos:
 - Deve Implementar múltiplas interfaces Listener.
- O objeto listener deve implementar a interface que “escuta” o evento.
- O objeto listener, que implementa o Listener, é adicionado ao objeto fonte:

objetoFonte.addListener(Listener listener)

Modelo de Eventos

6

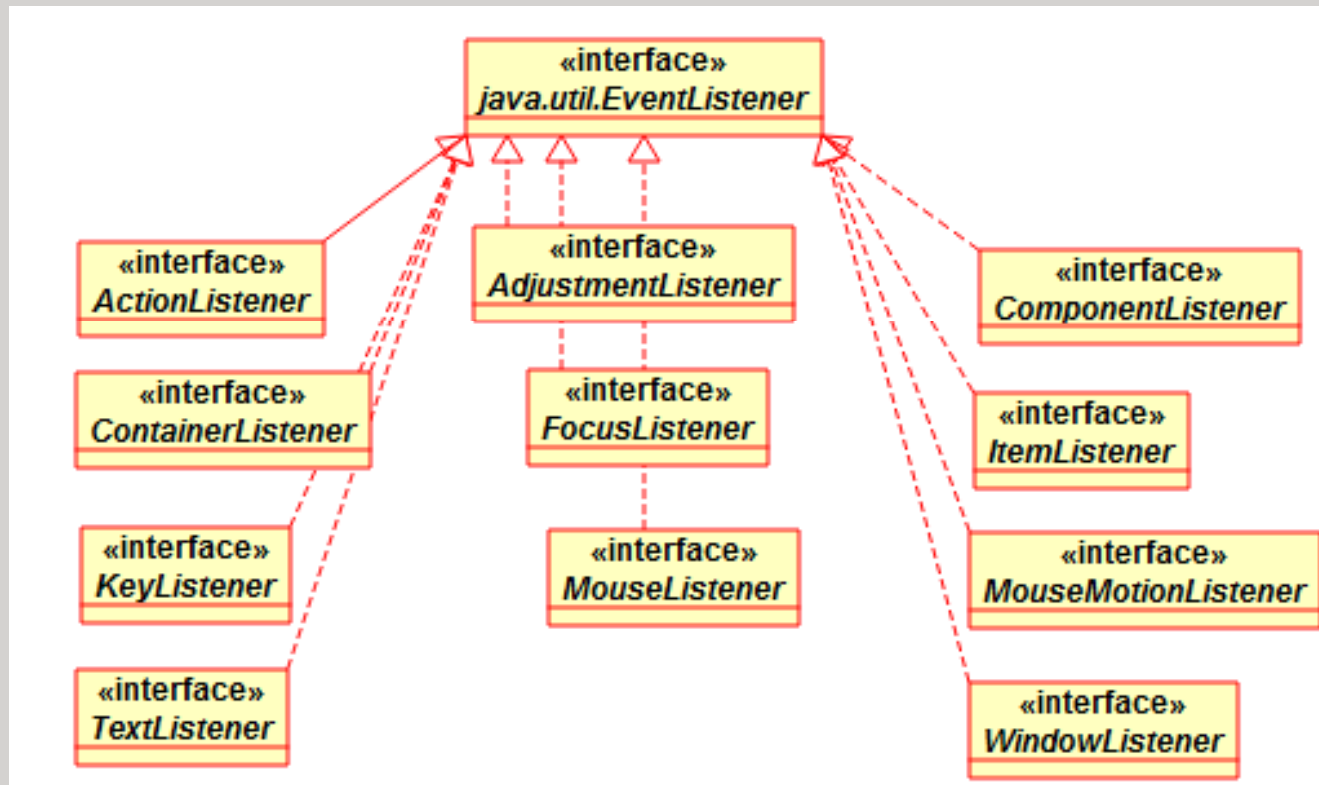


Figura: Interfaces Listeners.

Modelo de Eventos

7

-O objeto listener que deseja “escutar” um JButton deve implementar uma interface ActionListener que deriva de EventListener:

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

- O objeto 'ActionEvent e' possui:

- e.getSource(): um ponteiro para o objeto fonte.
- e.getWhen(): o timestamp de quando o evento aconteceu.
- e.getModifiers(): teclas pressionadas juntas (ex.: ALT, CTRL, etc.)
- e.getActionCommand(): a string do objeto que disparou a ação.

Modelo de Eventos

8

```
public class MeuListener implements ActionListener {  
  
    //JButton  notifica o objeto Listener  
    public void actionPerformed(ActionEvent evento){  
  
        Toolkit.getDefaultToolkit().beep();  
  
    }  
}  
...  
JButton  beep = new JButton (“Beep”);  
beep.addActionListener( new MeuListener() );
```


Modelo de Eventos

9

-Evento:

Botão é clicado.

- O objeto fonte envia para o objeto listener a notificação que o evento aconteceu.

-No caso, o objeto fonte JButton utiliza o método

`actionPerformed(ActionEvent e)`

para enviar a notificação para o objeto listener.

Modelo de Eventos

10

- Há diferentes formas de adicionar um Listener para um JComponent.
- Método 1:
 - Diretamente na Classe.
 - Utiliza o objeto 'this' como Listener.

```
public class BotaoPersonalizado extends JComponent
    implements ActionListener {

    public void actionPerformed(ActionEvent e){
        Toolkit.getDefaultToolkit().beep();
    }

    public BotaoPersonalizado(){
        JButton b = new JButton();
        b.addActionListener(this);
    }
}
```

Modelo de Eventos

11

- Método 2: - Criar uma Classe Interna que implementa o Listener

```
public class BotaoPersonalizado {  
    public BotaoPersonalizado(){  
        JButton b = new JButton();  
        ActionListener listener = new ClasseInterna();  
        b.addActionListener(listener);  
        //OU  
        //b.addActionListener(new ClasseInterna());  
    }  
    public class ClasseInterna implements ActionListener{  
        public void actionPerformed(ActionEvent e){  
            Toolkit.getDefaultToolkit().beep();  
        }  
    }  
}
```

Modelo de Eventos

12

- Método 3:
- Com uma Classe Interna Anônima.
 - No caso, a classe interna anônima implementa a interface ActionListener.

```
public class BotaoPersonalizado {  
    public BotaoPersonalizado(){  
        JButton b = new JButton();  
        b.addActionListener( new ActionListener(){  
            public void actionPerformed(ActionEvent e){  
                Toolkit.getDefaultToolkit().beep();  
            }  
        });  
    }  
}
```



Revisão

Modelo de Eventos

14

- Objeto Fonte:

- JComponent: JLabel, JButton, JTextField, etc.

- Listener:

- Um objeto que “escuta” interações do objeto fonte.

- Mensagem de Notificação:

- Uma mensagem enviada do objeto fonte para o Listener, indicando que um evento aconteceu.

- Evento:

- É uma ação realizada sobre o objeto fonte.

- Cada objeto fonte pode adicionar um ou mais objetos listener próprios para os seus eventos.

Modelo de Eventos

15

- O objeto fonte pode estar associado a vários eventos:
 - Deve Implementar múltiplas interfaces Listener.
- O objeto listener deve implementar a interface que “escuta” o evento.
- O objeto listener, que implementa o Listener, é adicionado ao objeto fonte:

objetoFonte.addListener(Listener listener)

Exercícios

16

<Ver conteúdo na plataforma de ensino>



Referências

17

- Referências bibliográficas da disciplina.