

# Algoritmos III

## Swing: JButton e Events

Raphael de Souza, Karen Figueiredo

IC/UFMT

# Roteiro

- 1 JButton e Events
  - javax.swing.JButton
  - Events

# javax.swing.JButton

## Definição

Componente que serve como botão

# javax.swing.JButton

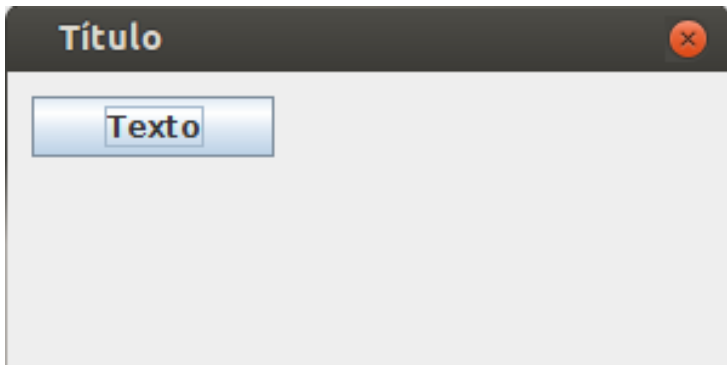
## Definição

Componente que serve como botão

## javax.swing.JButton

```
import javax.swing.JLabel;
import javax.swing.SwingConstants;
...
public class Tela extends javax.swing.JFrame {
    public Tela() {
        ...
        JButton botao = new JButton();
        botao.setText("Texto");
        botao.setLocation(10, 10);
        botao.setSize(100, 25);
        add(botao);
    }
}
```

# javax.swing.JButton



# Events

## Definição

É a interação que ocorre entre o usuário e o componente GUI e que indica ao programa que algo deve ser executado

# Events

## Definição

É a interação que ocorre entre o usuário e o componente GUI e que indica ao programa que algo deve ser executado



# Events

São exemplos de eventos:

- Algo ser clicado
- Algo ser digitado
- Algo a receber o foco
- Etc

# Events

São exemplos de eventos:

- Algo ser clicado
- Algo ser digitado
- Algo a receber o foco
- Etc

# Events

São exemplos de eventos:

- Algo ser clicado
- Algo ser digitado
- Algo a receber o foco
- Etc

# Events

São exemplos de eventos:

- Algo ser clicado
- Algo ser digitado
- Algo a receber o foco
- Etc

# Events

São exemplos de eventos:

- Algo ser clicado
- Algo ser digitado
- Algo a receber o foco
- Etc

# Events

- O código que realiza uma tarefa em resposta a um evento é conhecido como **handler de evento**
- O processo total de responder a eventos é conhecido como **tratamento de eventos**

# Events

- O código que realiza uma tarefa em resposta a um evento é conhecido como **handler de evento**
- O processo total de responder a eventos é conhecido como **tratamento de eventos**

# Events

- O código que realiza uma tarefa em resposta a um evento é conhecido como **handler de evento**
- O processo total de responder a eventos é conhecido como **tratamento de eventos**



# Tratamento de Evento

Para realizar o tratamento de evento:

- 1 - Cria uma classe que vai realizar a tarefa quando o evento ocorrer, ou seja, crie uma classe para representar o handler de evento
- 2 - Essa classe deve implementar a interface correspondente ao evento a ser "escutado"
- 3 - Registrar o handler de evento ao componente associado

# Tratamento de Evento

Para realizar o tratamento de evento:

- 1 - Cria uma classe que vai realizar a tarefa quando o evento ocorrer, ou seja, crie uma classe para representar o handler de evento
- 2 - Essa classe deve implementar a interface correspondente ao evento a ser "escutado"
- 3 - Registrar o handler de evento ao componente associado

# Tratamento de Evento

Para realizar o tratamento de evento:

- 1 - Cria uma classe que vai realizar a tarefa quando o evento ocorrer, ou seja, crie uma classe para representar o handler de evento
- 2 - Essa classe deve implementar a interface correspondente ao evento a ser "escutado"
- 3 - Registrar o handler de evento ao componente associado

# Tratamento de Evento

Para realizar o tratamento de evento:

- 1 - Cria uma classe que vai realizar a tarefa quando o evento ocorrer, ou seja, crie uma classe para representar o handler de evento
- 2 - Essa classe deve implementar a interface correspondente ao evento a ser "escutado"
- 3 - Registrar o handler de evento ao componente associado

## java.awt.event.ActionListener

```
import java.awt.event.ActionListener;
class Acao implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.out.println("Clicou no componente!");
    }
}

public class Tela extends javax.swing.JFrame {
    public Tela() {
        ...
        JButton botao = new JButton();
        botao.setText("Texto");
        botao.setLocation(10, 10);
        botao.setSize(100, 25);
        botao.addActionListener(new Acao());
        add(botao);
    }
}
```

## java.awt.event.MouseAdapter

```
import java.awt.event.MouseAdapter;
class Acao extends MouseAdapter {
    public void mouseClicked(MouseEvent e) {
        System.out.println("Clicou no componente!");
    }
}

public class Tela extends javax.swing.JFrame {
    public Tela() {
        ...
        JButton botao = new JButton();
        botao.setText("Texto");
        botao.setLocation(10, 10);
        botao.setSize(100, 25);
        botao.addMouseListener(new Acao());
        add(botao);
    }
}
```

## java.awt.event.FocusAdapter

```
import java.awt.event.FocusAdapter;
class Acao extends FocusAdapter {
    public void focusGained(FocusEvent e) {
        System.out.println("Ganhou Foco!");
    }
}

public class Tela extends javax.swing.JFrame {
    public Tela() {
        ...
        JButton botao = new JButton();
        botao.setText("Texto");
        botao.setLocation(10, 10);
        botao.setSize(100, 25);
        botao.addFocusListener(new Acao());
        add(botao);
    }
}
```

# Events

Algumas classes para eventos:

- MouseAdapter
- ActionListener
- FocusAdapter
- KeyAdapter
- WindowAdapter
- Etc...



# Events

Algumas classes para eventos:

- **MouseAdapter**
- ActionListener
- FocusAdapter
- KeyAdapter
- WindowAdapter
- Etc...

# Events

Algumas classes para eventos:

- MouseAdapter
- ActionListener
- FocusAdapter
- KeyAdapter
- WindowAdapter
- Etc...

# Events

Algumas classes para eventos:

- MouseAdapter
- ActionListener
- FocusAdapter
- KeyAdapter
- WindowAdapter
- Etc...

# Events

Algumas classes para eventos:

- MouseAdapter
- ActionListener
- FocusAdapter
- KeyAdapter
- WindowAdapter
- Etc...

# Events

Algumas classes para eventos:

- MouseAdapter
- ActionListener
- FocusAdapter
- KeyAdapter
- WindowAdapter
- Etc...

# Events

Algumas classes para eventos:

- MouseAdapter
- ActionListener
- FocusAdapter
- KeyAdapter
- WindowAdapter
- Etc...

## Exercício

- Construa uma tela em Java, utilizando o pacote Swing, que corresponda a uma calculadora, que realize as quatro operações básicas



## Exercício

- Implemente e organize uma interface gráfica para exibir uma janela como um cartão de visita seu





# Referências Bibliográficas

Aula baseada em slides preparados pelos seguintes professores:

- Prof. Bruno B. Boniati - [www.cafw.ufsm.br/bruno](http://www.cafw.ufsm.br/bruno) - Universidade Federal de Santa Maria
- Prof. José Gustavo de Souza Paiva - <http://www.facom.ufu.br/jgustavo/disc/poo.html> - Faculdade de Computação da Universidade Federal de Uberlândia
- Prof. Vítor E. Silva Souza - <http://www.inf.ufes.br/vitorsouza/pt/> - Departamento de Informática da Universidade Federal do Espírito Santo

Livro:

- Guia de Estudo: Certificação Sun para Programador Java 6. Kathy Sierra e Bert Bates