



UNIVERSIDADE CRUZEIRO DO SUL
TECNOLOGIA DE SISTEMAS DE INFORMAÇÃO

Interfaces Gráficas em Java

Parte 2 - Eventos




Eventos

- Console x Janelas;
- Necessidade de capturar eventos na janela para que uma ação seja executada (clique, abrir, passar o mouse, fechar, etc.);
- Interação do usuário com a GUI.



Eventos

- Cada evento está associado a um objeto (componente da GUI);
- O evento gera outro objeto (ActionEvent);
- O evento é passado para um *listener* (escutador de eventos);
- Para criar um *listener*, devemos implementar uma interface *listener* (*java.awt.event.**)



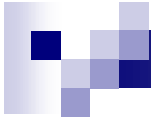
```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;
```

Implementa interface do *listener*
(escutador de eventos)

```
public class Eventos extends JFrame implements ActionListener {
```

```
    private JPanel p1, p2;  
    private JButton botSoma, botSubtracao, botMultiplicacao, botDivisao;  
    private JTextField txtValor1, txtValor2, txtResp;  
    private JLabel sinal;
```

```
    public Eventos() {  
        super("Trabalhando com Eventos");  
        this.getContentPane().setLayout(new BorderLayout());  
    }
```



```
public void init() {
```

Instancia os
objetos
(componentes) que
serão colocados na
janela.

```
p1 = new JPanel(new FlowLayout());
```

```
p2 = new JPanel(new FlowLayout());
```

```
botSoma      = new JButton("+");
```

```
botSubtracao = new JButton("-");
```

```
botMultiplicacao = new JButton("*");
```

```
botDivisao    = new JButton("/");
```

```
txtValor1 = new JTextField(5);
```

```
txtValor2 = new JTextField(5);
```

```
txtResp = new JTextField(5);
```

```
sinal = new JLabel(" = ");
```



```
botSoma.addActionListener(this);  
botSubtracao.addActionListener(this);  
botMultiplicacao.addActionListener(this);  
botDivisao.addActionListener(this);
```

Registra o objeto para o
escutador capturar seu
evento


```
p1.add(txtValor1);  
p1.add(txtValor2);  
p1.add(sinal);  
p1.add(txtResp);  
  
p2.add(botSoma);  
p2.add(botSubtracao);  
p2.add(botMultiplicacao);  
p2.add(botDivisao);
```

Monta os painéis para serem
colocados no Frame (Janela)

```
this.getContentPane().add(p1, BorderLayout.NORTH);  
this.getContentPane().add(p2, BorderLayout.CENTER);
```

Insere os painéis
na janela

```
setSize(500,300);  
setVisible(true);
```



```
private void calcular(char sinal)
{
```

```
    float v1 = Float.parseFloat(txtValor1.getText());
    float v2 = Float.parseFloat(txtValor2.getText());
    float resp = 0;
```

```
    switch(sinal)
    {
```

```
        case '+':
```

```
            resp = v1 + v2;
            break;
```

```
        case '-':
```

```
            resp = v1 - v2;
            break;
```

```
        case '*':
```

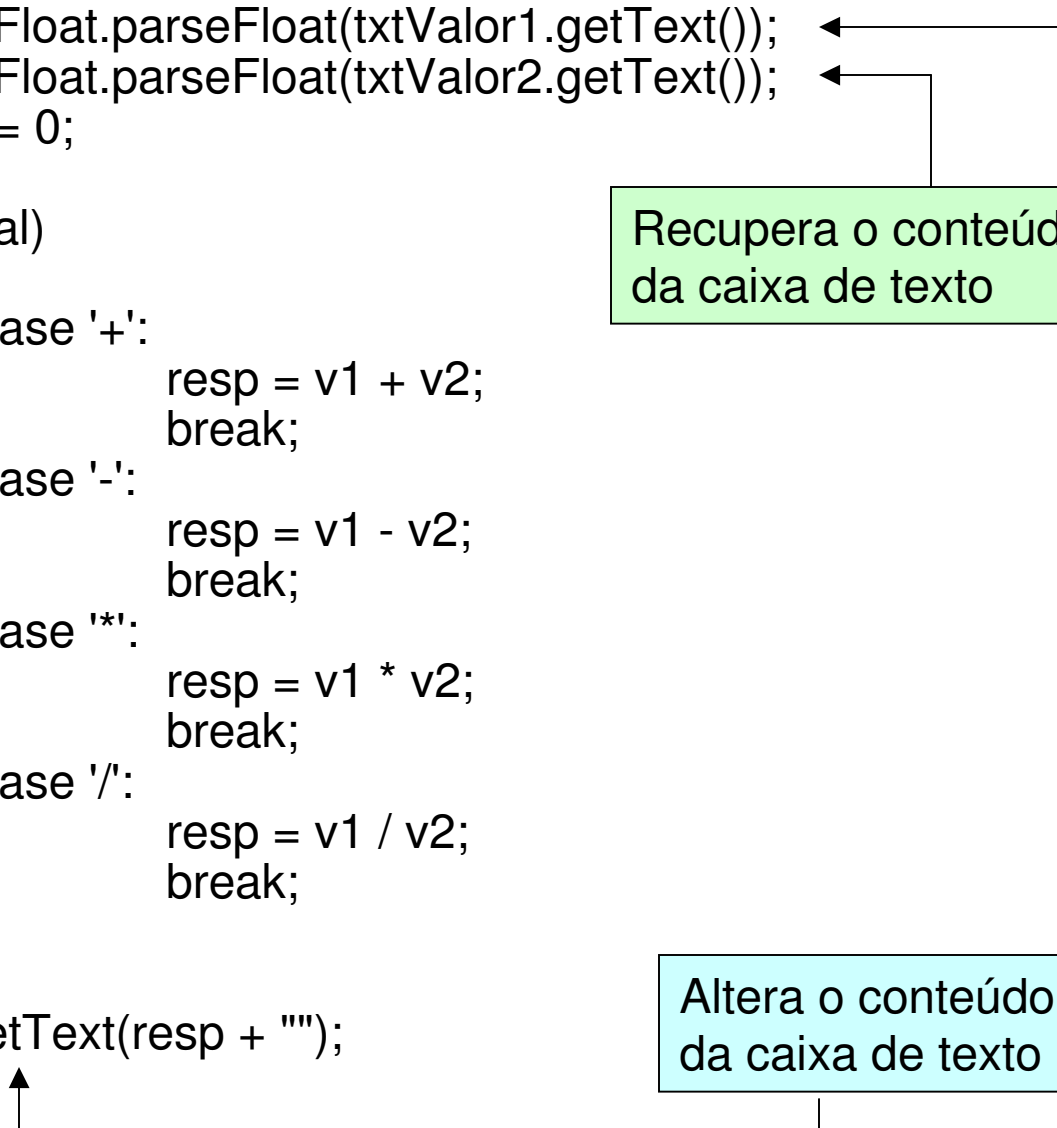
```
            resp = v1 * v2;
            break;
```

```
        case '/':
```

```
            resp = v1 / v2;
            break;
```

```
    }
```

```
    txtResp.setText(resp + "");
}
```



Recupera o conteúdo
da caixa de texto

Altera o conteúdo
da caixa de texto



```
public void actionPerformed(ActionEvent e) {
```

Método da interface
ActionListener que
escuta os eventos que
foram registrados

Captura o
evento que
foi gerado
e verifica
qual objeto
que gerou

```
    if (e.getSource()==botSoma) {  
        calcular('+');  
    }
```

```
    if (e.getSource()==botSubtracao) {  
        calcular('-');  
    }
```

```
    if (e.getSource()==botMultiplicacao) {  
        calcular('*');  
    }
```

```
    if (e.getSource()==botDivisao) {  
        calcular('/');  
    }
```

```
}
```



```
public static void main(String args[])
{
    Eventos a = new Eventos();
    a.init();
}
```





Exercício

- Elaborar um programa que permite o usuário digitar:
 - a) Nome do aluno
 - b) Prova Parcial (PP)
 - c) Prova Regimental (PR)
 - d) Projeto (PJ)
 - e) Trabalhos (TR)
- Após a digitação da nota, o programa exibe a média e se o aluno está aprovado ($média \geq 6$), DP($média < 3$) ou de Exame
- $média = 0,2 * PP + 0,4 * PR + 0,2 * PJ + 0,2 * TR$