

Java Standard Edition (JSE)

Capítulo 09. Swing básico



Esp. Márcio Palheta

MSN: marcio.palheta@hotmail.com



Novos recursos a aprender

- Interfaces gráficas;
- Portabilidade;
- O pacote do Swing;
- Primeiro projeto Swing;



Interfaces gráficas em JAVA

- Java suporta, oficialmente, as APIs AWT e Swing;
- AWT foi a primeira API a surgir e, a partir do java 1.2, foi superada pela Swing;
- A API gráfica é grande e bem documentada;
- AWT e Swing funcionam em qualquer JRE ou JDK;

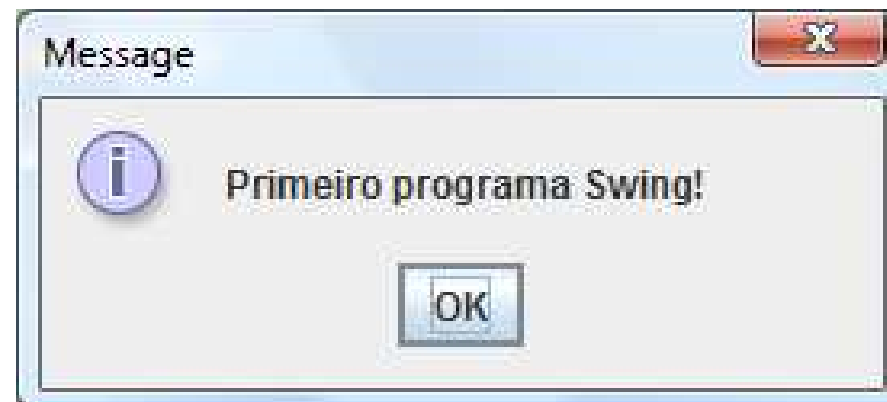


Portabilidade

- O look-and-feel é único para as plataformas em que roda: windows, linux, mac etc;
- Busca minimizar problemas de fontes e cores, que podem variar de acordo com o sistema operacional em que é executada;

Primeiro programa Swing

```
1 import javax.swing.JOptionPane;  
2 public class HelloWorld {  
3     public static void main(String[] args) {  
4         JOptionPane.showMessageDialog(null,  
5             "Primeiro programa Swing!");  
6     }  
7 }
```





Trabalhando com Formulários

- A API Swing é composta por um grande grupo de componentes que, em conjunto, servem de interface com o usuário;
- Na API Swing, precisamos criar um formulário para agruparmos nossos componentes;
- Vamos criar um formulário com um campo de texto e uma etiqueta;



Primeiros componentes

- // Label com título do campo:
 - JLabel label = new JLabel("Seu nome:");
- // Campo texto com tamanho 20
 - JTextField textField = new JTextField(20);
- //Todo componente deve pertencer a um container:
 - JPanel panel = new JPanel();
 - panel.add(label);
 - panel.add(textField);



Criação do formulário

- Agora, precisamos criar o formulário com título e atribuir o JPanel criado:
 - `JFrame frame = new JFrame("Meu primeiro formulário");`
- `// Adicionamos nosso JPanel`
 - `frame.add(panel);`
- `// Preparando o JFrame para exibição`
 - `frame.pack(); //redimensiona`
 - `frame.setVisible(true);`



Criação do formulário

- //Indica que a aplicação deve ser encerrada quando fecharmos a janela:
- `frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
- No exemplo a seguir, as variáveis foram definidas como atributos e o método `montaFormulario()` constrói nosso formulário:

```

1 import javax.swing.*;
2 public class Formulario {
3     private JLabel label;
4     private JTextField textField;
5     private JPanel panel;
6     private JFrame frame;
7
8     private void montaFormulario() {
9         label = new JLabel("Seu nome:");
10        textField = new JTextField(20);
11        panel = new JPanel();
12        panel.add(label);
13        panel.add(textField);
14        frame = new JFrame("Meu primeiro formulário");
15        frame.add(panel);
16        frame.pack();
17        frame.setVisible(true);
18        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19    }
20
21    public static void main(String[] args) {
22        new Formulario().montaFormulario();
23    }
24 }

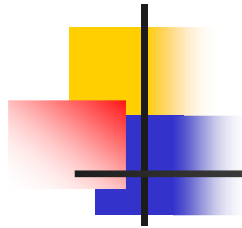
```

Resultado do código:

- A seguir, é exibido o formulário criado, bem como seus respectivos componentes:



A screenshot of a graphical user interface window. The window has a title bar with the text 'Meu primeiro formulário' and standard Windows window controls (minimize, maximize, close). Inside the window, there is a label 'Seu nome:' followed by a single-line text input field.



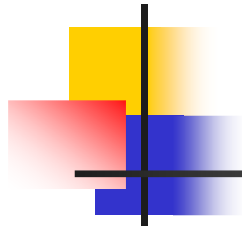
Trabalhando com eventos

- Em geral, as aplicações necessitam recuperar as informações digitadas por usuários, a fim de, por exemplo, salvar em banco de dados;
- Para recebermos os dados digitados, precisamos capturar os eventos disparados pelo usuário, como:
- Digitar, clicar ou passar o mouse, dentre outros;



Criando um botão

- Usaremos o componente botão (JButton) que, quando clicado pelo usuário (evento), dispara o método (tratador/handler);
- No handler, vamos recuperar o texto digitado e realizar uma tarefa;
- // cria o JButton passando o texto do botao
 - JButton button = new JButton("Exibir");
- // adiciona o botao ao JPanel
 - panel.add(button);



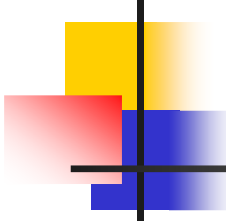
Usando Listeners

- Como disparar um método quando o botão for clicado?
- No Swing existem os **Listeners**, que são Interfaces implementadas com métodos a serem disparados por eventos;
- A interface **ActionListener**, através do método **actionPerformed()**, permite o disparo de métodos ao clique do botão:

Atualização da classe Formulario.java

- Vamos fazer nossa classe **Formulario** implementar a interface **ActionListener** e seu método **actionPerformed()**;
- Precisamos exibir(showMessage) o texto digitado(textField)

```
public void actionPerformed(ActionEvent e) {  
    //Captura o valor digitado no campo texto  
    String texto = textField.getText();  
    //Exibe o valor digitado;  
    JOptionPane.showMessageDialog(frame, texto);  
}
```



```

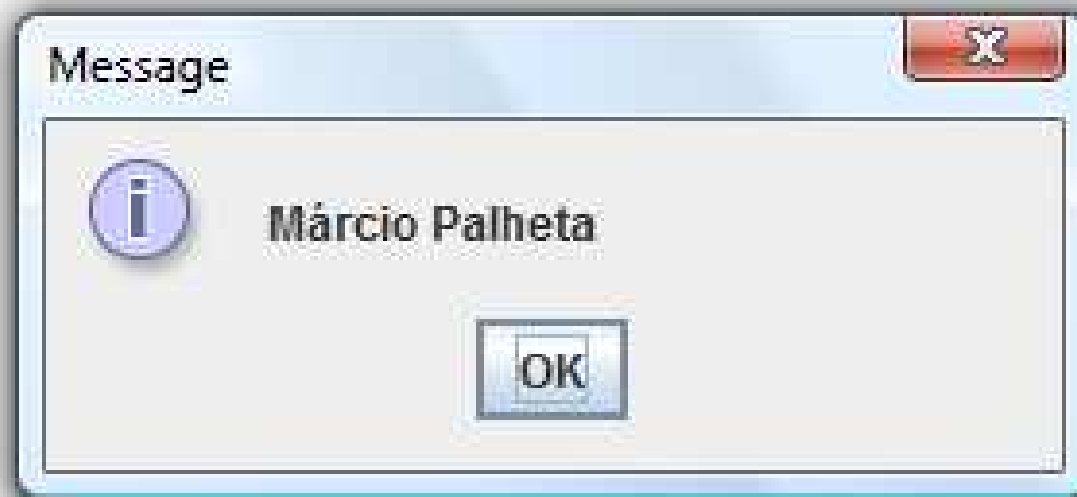
1⊕ import java.awt.event.ActionEvent;
10 public class Formulario2 implements ActionListener {
11     private JLabel label;
12     private JTextField textField;
13     private JPanel panel;
14     private JFrame frame;
15     2 private JButton button;
16⊖ private void montaFormulario() {
17         label = new JLabel("Seu nome:");
18         textField = new JTextField(20);
19         3 button = new JButton("Exibir");
20         button.addActionListener(this);
21         panel = new JPanel();
22         panel.add(label);
23         panel.add(textField);
24         4 panel.add(button);
25         frame = new JFrame("Meu primeiro formulário");
26         frame.add(panel);
27         frame.pack();
28         frame.setVisible(true);
29         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30     }
31⊖ public void actionPerformed(ActionEvent e) {
32     5 String texto = textField.getText();
33         JOptionPane.showMessageDialog(frame, texto);
34     }
35⊕ public static void main(String[] args) {
38 }

```


Resultado do código



A screenshot of a web browser window titled "Meu primeiro formulario". The window contains a text input field with the label "Seu nome:" and the text "Márcio Palheta" entered. To the right of the input field is a button labeled "Exibir". The window has standard minimize, maximize, and close buttons in the top right corner.





O que mais?

- Sun disponibiliza um ótimo tutorial online sobre Swing em seu Site:
 - <http://java.sun.com/docs/books/tutorial/uiswing/>
- Principais editores Swing:
 - - Matisse, que vem embutido no Netbeans e é considerado, hoje, o melhor editor
 - - VEP (Visual Editor Plugin), um plugin que pode ser instalado no Eclipse
- Aplicações grandes com Swing podem ganhar uma complexidade enorme e ficar difíceis de manter.
- No projeto Thinlet, você pode utilizar Swing escrevendo suas interfaces gráficas em XML.



Referências WEB

- SUN: www.java.sun.com

Fóruns e listas:

- Javaranch: www.javaranch.com
- GUJ: www.guj.com.br

Apostilas:

- Argonavis: www.argonavis.com.br
- Caelum: www.caelum.com.br

Java Standard Edition (JSE)

Capítulo 09. Swing básico



Esp. Márcio Palheta

MSN: marcio.palheta@hotmail.com