



UNIVERSIDADE PAULISTA

Professor: Me. Pablo I. Gandulfo

Data: 01/08/2020

Versão: 1.1

Aplicações de Linguagem de Programação Orientada a Objetos

MÓDULO 2 a 4



Professor: Me. Pablo I. Gandulfo
Data: 01/08/2020
Versão: 1.1

SWING – Parte 01 a 03

- Criação de Objetos – via código
 - Utilização de Objetos Visualmente
 - Tratamento de Eventos
-

- Desenvolvendo uma Aplicação com GUI
- Swing
- Componentes Swing
- Eventos
- Eventos e Observadores de Eventos (*Listeners*)
- Primeiro Exemplo
- Outro Exemplo
- Exercícios

Desenvolvendo uma Aplicação com GUI - O Swing é similar ao AWT neste aspecto

- Processo básico: OO + Eventos
 1. Instanciar os componentes de interface
 - Janelas, campos de textos, listas, botões, etc.
 2. Adicionar os componentes em *containers*
 - *Containers* permitem agrupar, posicionar e alinhar os componentes
 3. Estabelecer o tratamento dos eventos de interface
 - Definir o comportamento a ser executado após preencher um campo (validar o seu preenchimento) ou a ação resultante do clique de um botão

Swing

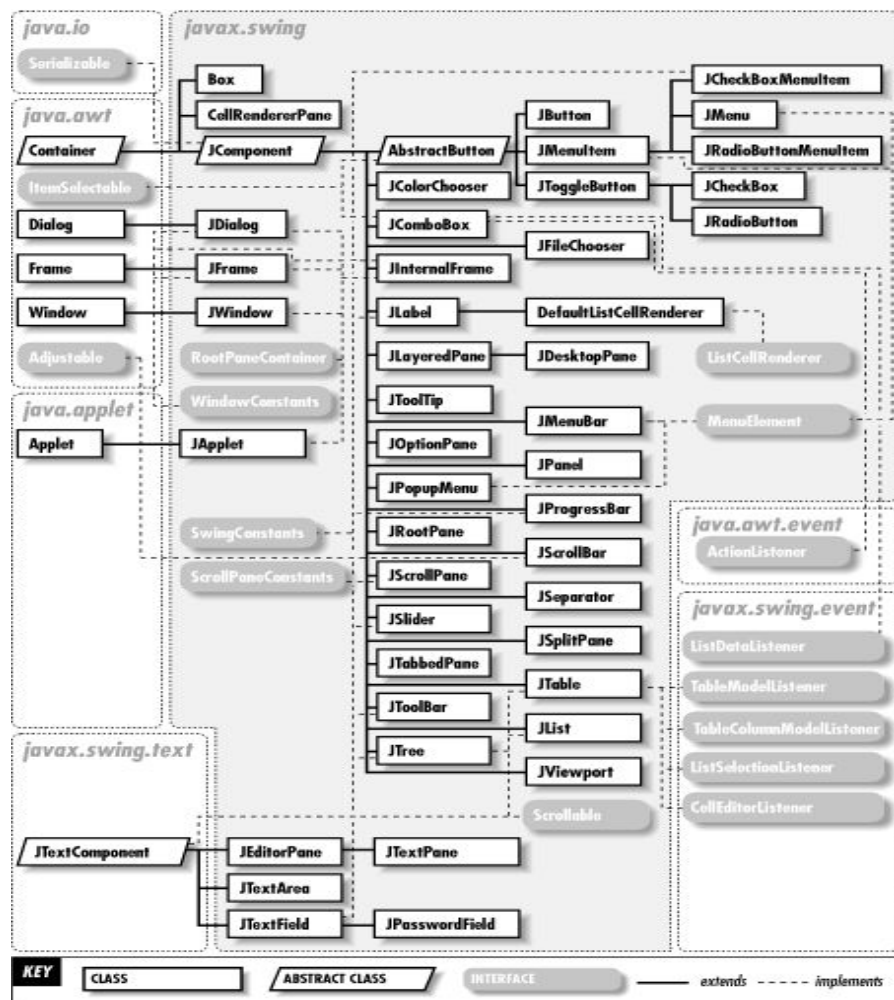
- É um *kit* de ferramentas Java GUI disponível desde o JDK 1.1.x
- Swing é relativamente mais complexo que o AWT e tem componentes de GUI alto nível, como estrutura em árvore (*tree view*), caixa de listagem (*list box*), painéis com abas, Java2D e *drag & drop*, gerenciadores de *layout* e *kit* de ferramentas de eventos
- O *kit* foi implementado inteiramente em Java, ou seja, mantém um padrão de código e interface que não depende do S.O. hospedeiro
- Dessa forma, os componentes visuais “pintam” pixels a partir de um container inicial, que é um componente do S.O. (daí a denominação de “*lightweight*”)

Swing (Cont.)

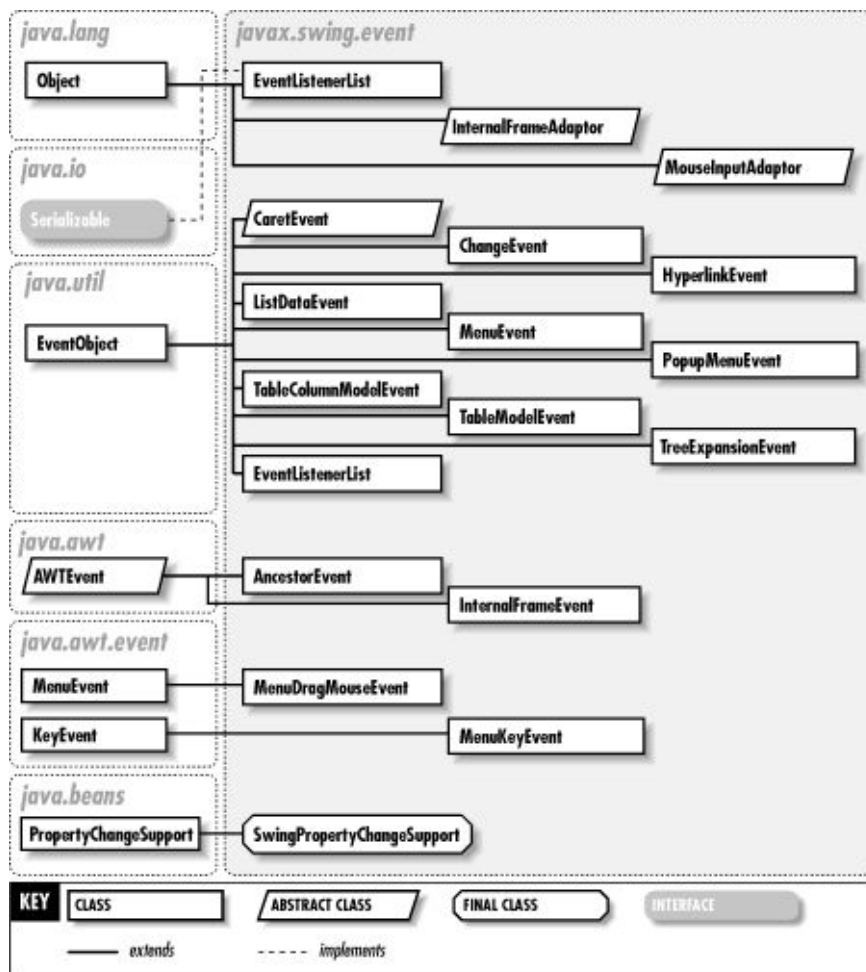
- Os componentes são mais flexíveis e completos, pois, não dependem de prover um denominador comum entre os S.O.'s
- Permitem utilizar um visual e comportamento plugável: ao selecionar um dos temas disponíveis, o estilo dos componentes é customizado para aderir a esse tema
- **import javax.swing**

Componentes Swing

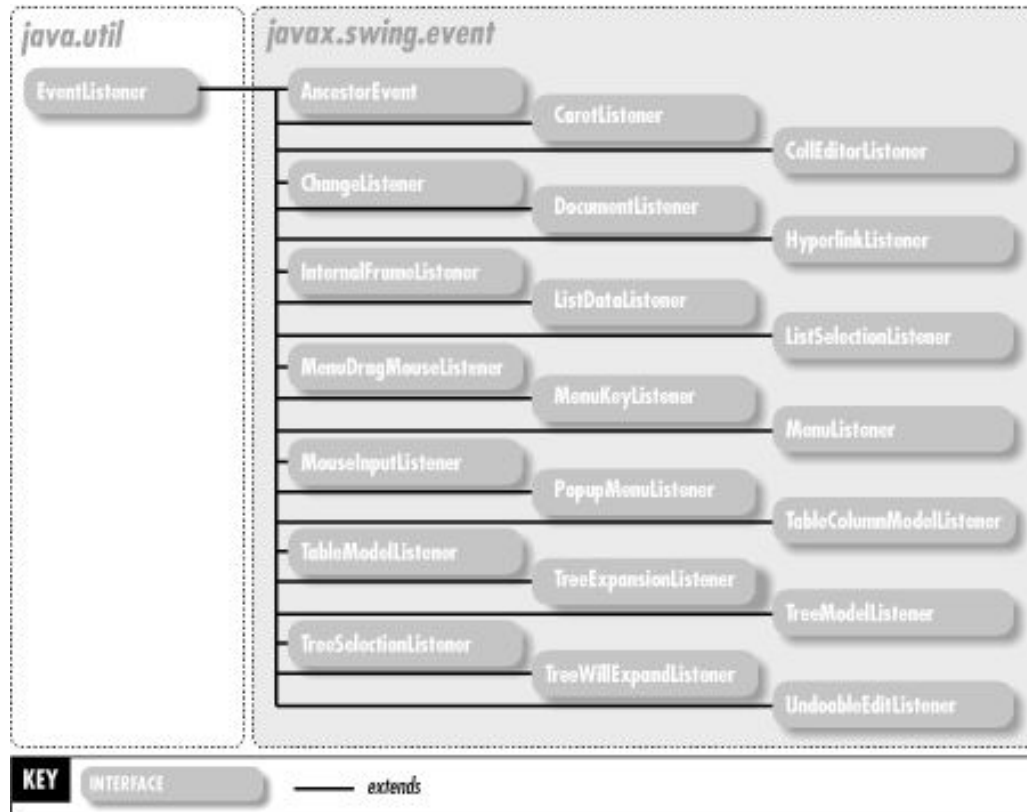
- *Components e Containers*
 - *Containers e Components* são componentes visuais independentes
 - *Containers*
 - JFrame, JDialog, JPanel, JXxxPane, JToolBar, JFrame e JXxxFrame
 - Controles Básicos
 - JTextField, JList, JButton, JComboBox, JMenu e JSlider
 - *Displays de Informação*
 - JLabel, JProgressBar e JToolTip
 - *Displays Editáveis*
 - JColorChooser, JFileChooser, JTable, JText e JTree



Eventos



Eventos e Observadores de Eventos (*Listeners*)



Primeiro Exemplo

```
public class PrimeiroExemplo extends JFrame {  
    public static void main(String args[]) {  
        new PrimeiroExemplo();  
    }  
  
    public PrimeiroExemplo() {  
        // Inibe mensagens de erro por causa da troca de Visual (Look & Feel)  
        Thread.setDefaultUncaughtExceptionHandler(new ExceptionHandler());  
        System.setProperty("sun.awt.exception.handler", ExceptionHandler.class.getName());  
  
        JFrame f = this;  
        setTitle("AWT - Primeiro Exemplo");  
  
        JLabel lbVisual = new JLabel("Visual: "); lbVisual.setBounds(50, 20, 200, 30); add(lbVisual);  
        JComboBox c = new JComboBox();  
  
        ...  
    }  
}
```

Primeiro Exemplo

```
...  
    c.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e1) {  
            for (LookAndFeelInfo lf : UIManager.getInstalledLookAndFeels()) {  
                if (lf.getName().equals(c.getSelectedItem())) {  
                    try {  
                        UIManager.setLookAndFeel(lf.getClassName());  
                        JFrame.setDefaultLookAndFeelDecorated(true);  
                        SwingUtilities.updateComponentTreeUI(f);  
                    } catch (Exception e) { }  
                    break;  
                }  
            }  
        }  
    });  
    c.setBounds(50, 50, 200, 30); add(c);  
    JLabel lbNome = new JLabel("Nome: "); lbNome.setBounds(50, 80, 200, 30); add(lbNome);  
...
```

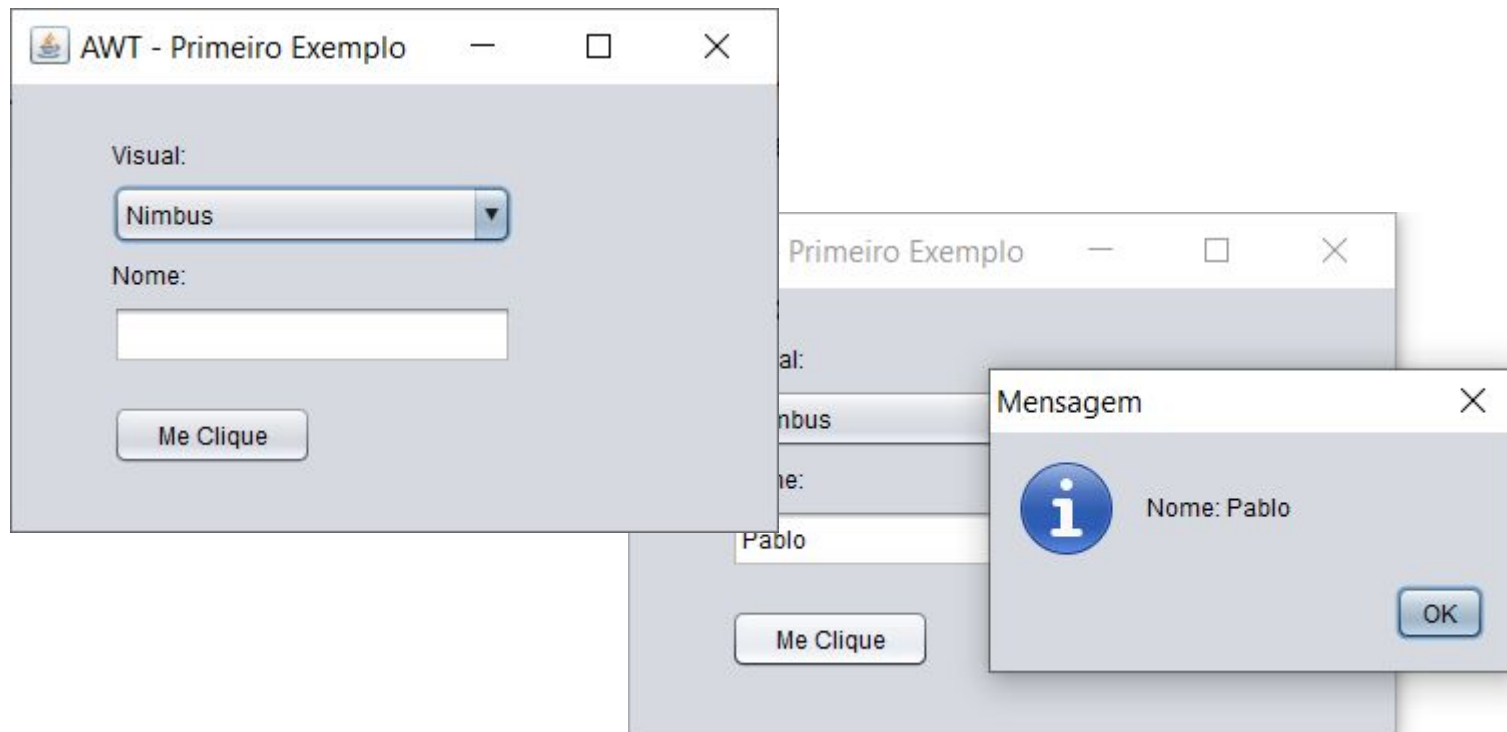
Primeiro Exemplo

```
...  
  
    JTextField t = new JTextField(); t.setBounds(50, 110, 200, 30); add(t);  
    JButton b = new JButton("Me Clique"); b.setBounds(50, 160, 100, 30); add(b);  
    b.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e2) {  
            JOptionPane.showMessageDialog(null, "Nome: " + t.getText());  
        }  
    });  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    Dimension d = getToolkit().getScreenSize();  
    setLocation(d.width / 3, d.height / 3);  
    setSize(400, 270);  
    setLayout(null);  
    setVisible(true);  
}  
...
```

Primeiro Exemplo

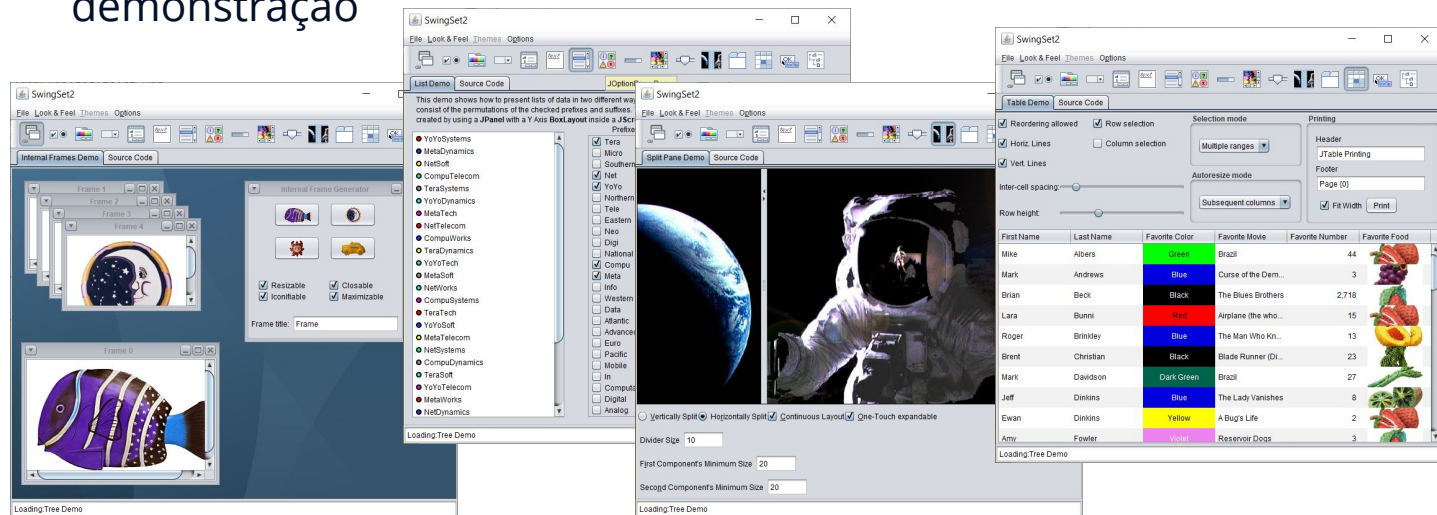
```
...  
  
class ExceptionHandler implements Thread.UncaughtExceptionHandler {  
    public void handle(Throwable t) {}  
    public void uncaughtException(Thread t, Throwable e) {}  
}  
}
```

Primeiro Exemplo



Outro Exemplo

- swingset2/SwingSet2.java
swingset2/**
 - SwingSet2 é uma aplicação de exemplo desenvolvida pela Oracle que demonstra algumas das habilidades do Toolkit de UI Swing, utilizando muitos dos componentes visuais existentes num único aplicativo de demonstração



Exercício 1

- Desenvolva um programa com interface gráfica via Swing com as seguintes características:
 -

DEFINIR DURANTE A AULA