



**UNIVERSIDADE PAULISTA**

Professor: Me. Pablo I. Gandulfo

Data: 01/08/2020

Versão: 1.1

# Aplicações de Linguagem de Programação Orientada a Objetos

---

## MÓDULO 1



Professor: Me. Pablo I. Gandulfo

Data: 01/08/2020

Versão: 1.1

## AWT – Abstract Windowing Toolkit

---

- Java e GUI
- Desenvolvendo uma Aplicação com GUI
- AWT – Abstract Window ToolKit
- Componentes AWT
- Eventos
- Componentes e seus Eventos
- Observadores de Eventos (*Listeners*)
- Primeiro Exemplo
- Outros Exemplos
- Exemplos Mais Avançados
- Exercícios

## Java e GUI

---

- Ao desenvolver uma interface gráfica com o usuário (GUI), é importante estudar as opções disponíveis de componentes gráficos
- No Java existem dois conjuntos (*kits*) básicos de componentes gráficos:
  - Abstract Window Toolkit (AWT)
  - Swing
- Ambos fazem parte da Java Foundation Classes (JFC), um *kit* de ferramentas para construir interfaces GUI portáteis entre sistemas operacionais nativo da linguagem, ou seja, não precisam ser instalados

## Desenvolvendo uma Aplicação com GUI

---

- 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

## AWT – Abstract Window Toolkit

---

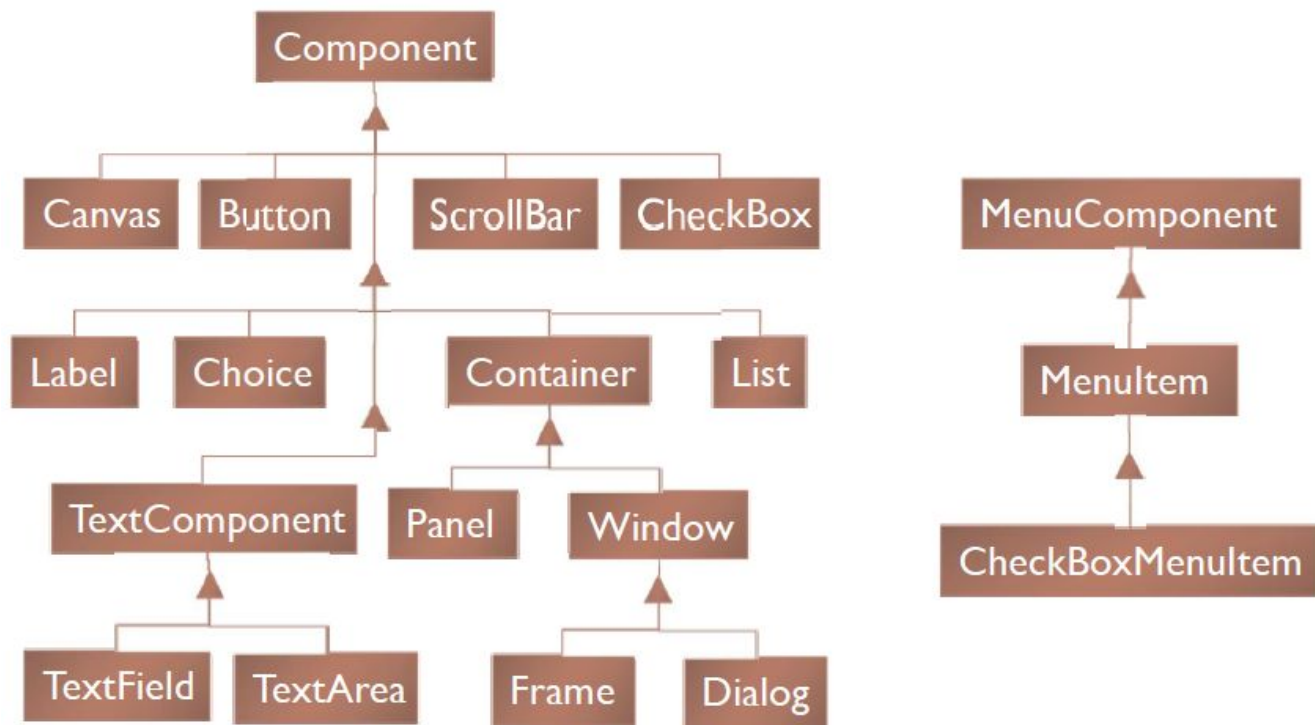
- É o *kit* de ferramentas Java GUI original, desde o JDK 1.0.x
- AWT é muito simples e tem componentes de GUI limitados, gerenciadores de *layout* e *kit* de ferramentas de eventos
- O *kit* utiliza interface gráfica nativa, ou seja, faz uso de códigos C nativos do sistema operacional para construir os componentes (diferente do Swing)
- Dessa forma, segue o padrão visual do S.O. em que é executado (daí a denominação de “*heavyweight*”)
- Os componentes possuem limitações, pois, dependem do denominador comum entre os S.O.’s, possuindo apenas componentes e opções de GUI existentes em todos os ambientes
- **import java.awt**

## Componentes AWT

---

- *Components, Containers, MenuComponents*
  - *Todo Container é um Component*
  - *Nem todo Component é um Container*
  - *MenuComponentes formam uma hierarquia à parte*
- *Containers*
  - *Frame, Panel*
- *Components atômicos (que não são Containers)*
  - *Label, TextField, TextArea, Choice, Checkbox, List, Button*
- *MenuComponents*
  - *MenuBar, PopupMenu*

## Componentes AWT



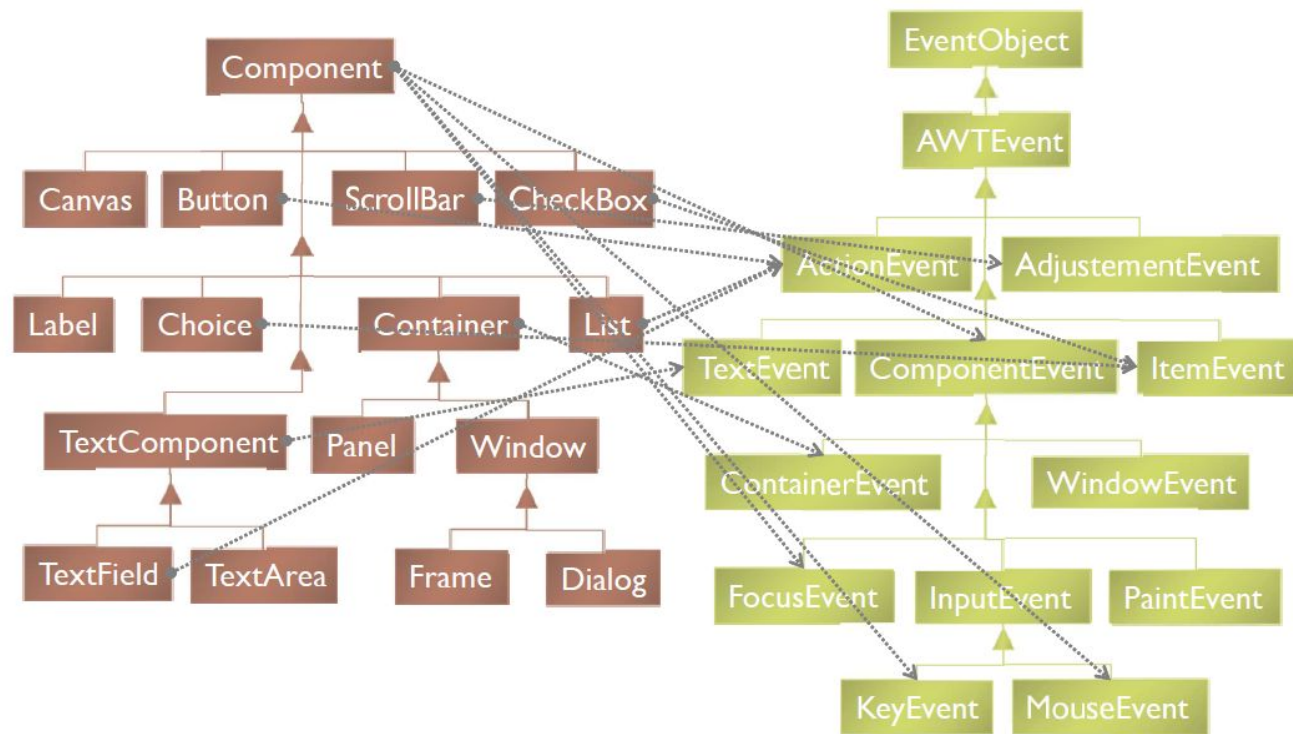
## Eventos

---

- **ActionEvent**
  - Usuário clica em um botão, pressiona <ENTER> dentro de um textfield ou escolhe um item de menu
- **WindowEvent**
  - Usuário fecha, minimiza, maximiza, etc., uma janela
- **MouseEvent**
  - Usuário pressiona entra ou sai com o mouse de dentro um componente
- **MouseMotionEvent**
  - Usuário move o mouse sobre um componente
- **ComponentEvent**
  - Componente fica visível, é adicionado a um container
- **FocusEvent**
  - Componente recebe o foco do teclado
- **ItemEvent**
  - Seleção de uma tabela ou lista é modificada

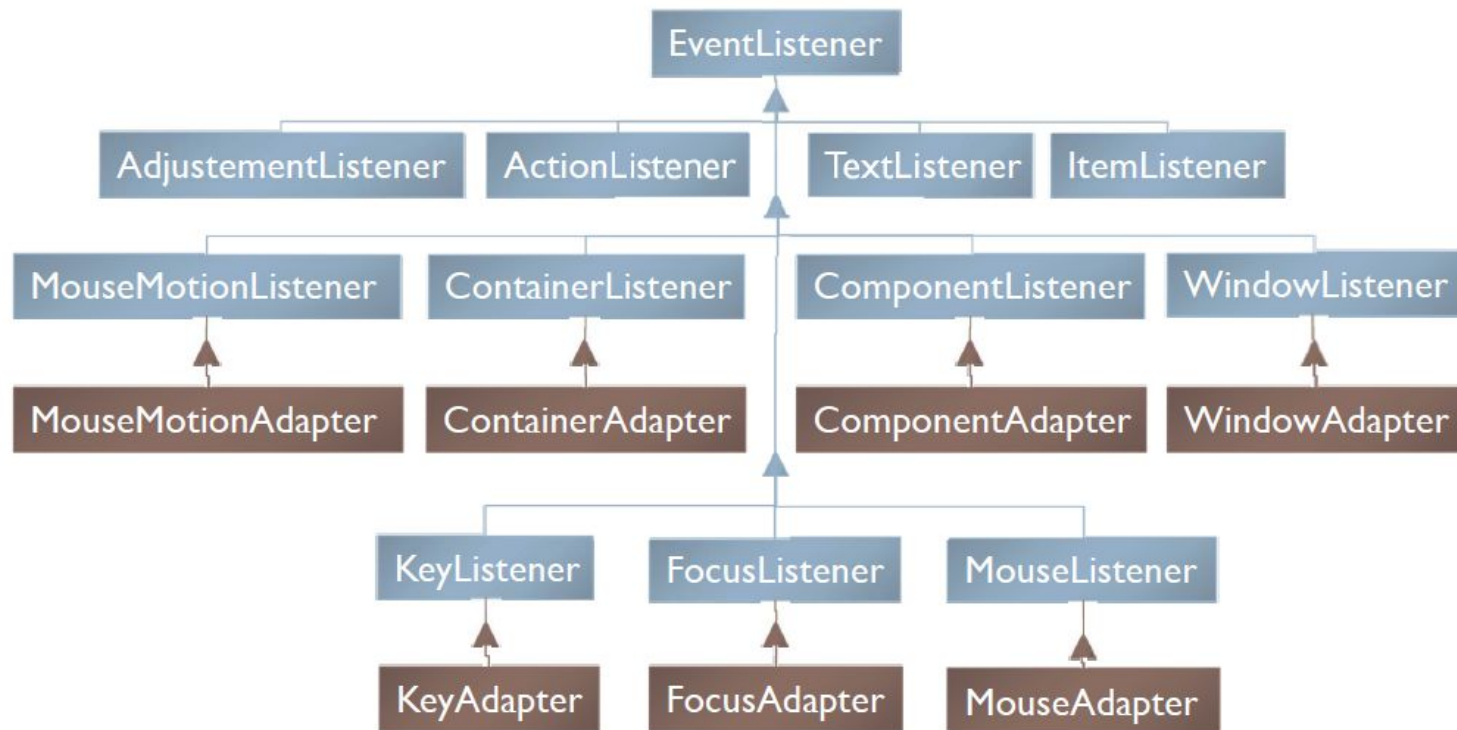


## Componentes e seus Eventos



## Observadores de Eventos (*Listeners*)

---



## Observadores de Eventos (*Listeners*)

---

- **AdjustementListener**
  - `adjustementValueChanged(AdjustementEvent)`
- **ActionListener**
  - `actionPerformed(ActionEvent)`
- **ItemListener**
  - `itemStateChanged(ItemEvent)`
- **TextListener**
  - `textValueChanged(TextEvent)`
- **KeyListener**
  - `keyPressed(KeyEvent)`
  - `keyReleased(KeyEvent)`
  - `keyTyped(KeyEvent)`
- **ContainerListener**
  - `componentAdded(ContainerEvent)`
  - `componentRemoved(ContainerEvent)`
- **ComponetListener**
  - `componentHidden(ComponentEvent)`
  - `componentMoved(ComponentEvent)`
  - `componentResized(ComponentEvent)`
  - `componentShown(ComponentEvent)`
- **WindowListener**
  - `windowActivated(WindowEvent)`
  - `windowClosed(WindowEvent)`
  - `windowClosing(WindowEvent)`
  - `windowDeactivated(WindowEvent)`
  - `windowDeiconified(WindowEvent)`
  - `windowIconified(WindowEvent)`
  - `windowOpened(WindowEvent)`
- **MouseMotionListener**
  - `mouseDragged(MouseEvent)`
  - `mouseMoved(MouseEvent)`
- **MouseListener**
  - `mouseClicked(MouseEvent)`
  - `mouseEntered(MouseEvent)`
  - `mouseExited(MouseEvent)`
  - `mousePressed(MouseEvent)`
  - `mouseReleased(MouseEvent)`
- **FocusListener**
  - `focusGained(FocusEvent)`
  - `focusLost(FocusEvent)`

## Primeiro Exemplo

---

```
public class PrimeiroExemplo extends Frame {  
    public static void main(String args[]) {  
        new PrimeiroExemplo();  
    }  
  
    public PrimeiroExemplo() {  
        Frame f = this;  
        setTitle("AWT - Primeiro Exemplo");  
  
        Label lb = new Label("Nome: "); lb.setBounds(50, 50, 200, 30); add(lb);  
        TextField t = new TextField(); t.setBounds(50, 100, 200, 30); add(t);  
        Button b = new Button("Me Clique"); b.setBounds(50, 150, 80, 30); add(b);  
        b.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
                new MessageBox(f, "Nome: " + t.getText());  
            }  
        });  
    }  
    ...  
}
```

## Primeiro Exemplo

---

```
...  
    addWindowListener(new WindowAdapter() {  
        public void windowClosing(WindowEvent windowEvent) {  
            System.exit(0);  
        }  
    });  
  
    Dimension d = getToolkit().getScreenSize();  
    setLocation(d.width / 3, d.height / 3);  
    setSize(400, 250);  
    setLayout(null);  
    setVisible(true);  
}  
  
class MessageBox extends Dialog {  
    public MessageBox(Frame parent, String mensagem) {  
        super(parent, "Mensagem", true);  
    }  
}
```

## Primeiro Exemplo

---

```
...  
  
        Label lb = new Label(" " + mensagem); add("Center", lb);  
        Panel p = new Panel(); p.setLayout(new FlowLayout()); add("South", p);  
        Button ok = new Button("OK"); p.add(ok);  
        ok.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
                setVisible(false);  
            }  
        });  
  
        addWindowListener(new WindowAdapter() {  
            public void windowClosing(WindowEvent windowEvent) {  
                dispose();  
            }  
        });  
  
...
```

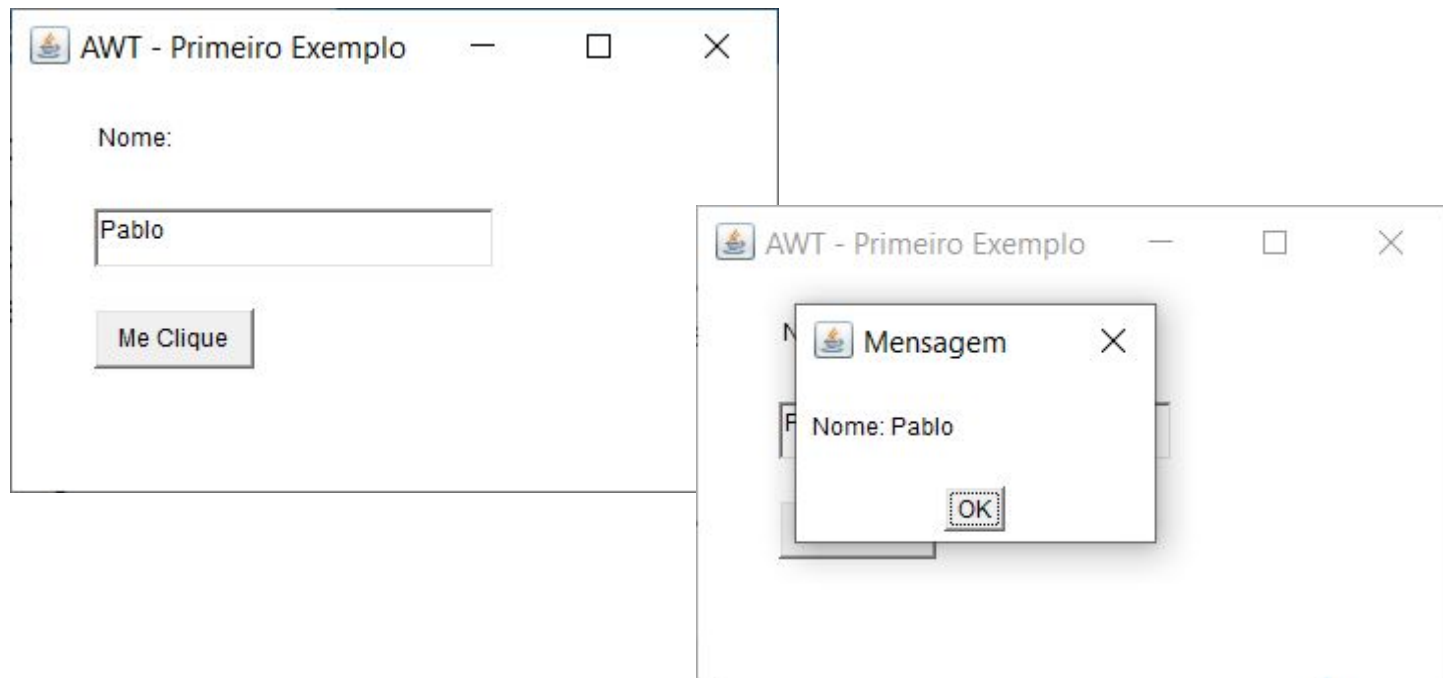
## Primeiro Exemplo

---

```
...  
    setBackground(Color.white);  
    Dimension d = getToolkit().getScreenSize();  
    setLocation(d.width / 3 + 50, d.height / 3 + 50);  
    setSize(200, 130);  
    setVisible(true);  
}  
}
```

## Primeiro Exemplo

---





## Outros Exemplos

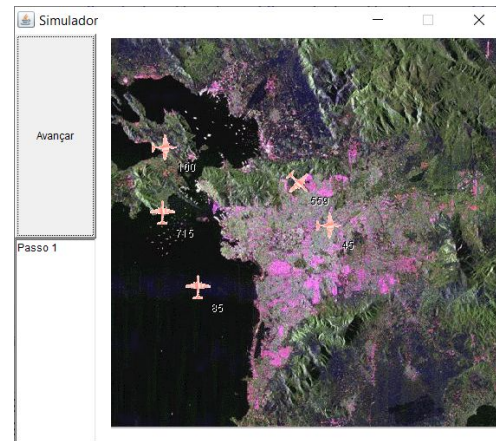
- `EventoAlteracaoTexto.java`
  - Faz uso de um evento de alteração do valor de campo
- `VariosCampos.java`
  - Possui caixa texto de uma linha (text field), de várias linhas (text area), opções de seleção (radio buttons), opções de marcação (check buttons), caixa de opções de uma linha (combo box), de várias linhas (list box), área de desenho (*canvas*) e botão
- `Calculadora.java`
  - Realiza operações aritméticas simples

The screenshot shows a window titled "AWT - Vários Campos". Inside, there is a form titled "PRODUTO". The form contains the following elements: a green oval and a blue rectangle at the top right; a "Nome:" label followed by a single-line text field with the placeholder "Entre com o nome completo"; a "Descrição:" label followed by a multi-line text area with the placeholder "Entre com a descrição"; a "Perecível:" section with two radio buttons, "Sim" and "Não", where "Não" is selected; a "Tipo:" section with two checkboxes, "Essencial" and "Crítico", both of which are unchecked; a "Tamanho:" label followed by a dropdown menu currently showing "Médio"; a "Peso:" label followed by a list box containing three options: "> 20 KG", "10 a 20 KG" (which is highlighted in blue), and "< 10 KG"; and a "Gravar" button at the bottom.

The screenshot shows a window titled "Calculadora". It features a large text field at the top for the display. Below the display is a grid of buttons for digits 0 through 9, as well as arithmetic operators: addition (+), subtraction (-), multiplication (\*), division (/), and an equals (=) button.

## Exemplos Mais Avançados

- Simulador de Controle de Tráfego Aéreo
  - Ilustra aviões em rotas fixas que podem ou não colidir
  - 95% AWT, 5% Swing



- Simulador de Elevador
  - Ilustra usuários e Ações realizadas por um elevador
  - 50% AWT, 50% Swing



## Exercício 1

---

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

DEFINIR DURANTE A AULA