

Lição 7



Abstract Window Toolkit e Swing

Objetivos

Ao final desta lição, o estudante será capaz de:

- Explicar similaridades e diferenças entre *AWT* e *Swing*
- Diferenciar entre os *components* e *containers*
- Criar aplicações *GUI* utilizando *AWT*
- Criar aplicações *GUI* utilizando *Swing*
- Descrever como os gerenciadores de *layout* tais como, *FlowLayout*, *BorderLayout* e *GridLayout* posicionam os componentes *GUI*
- Criar *layouts* complexos ao elaborar aplicações *GUI*



AWT versus Swing

- *Java Foundation Classe (JFC)*
- Parte importante do Java *SDK*
- Coleção de *APIs* que simplifica o desenvolvimento
- Ferramentas fornecidas por Java para desenvolver aplicações *GUI* interativas
- Fornece componentes *GUI* que podem ser usados na criação de aplicações Java e *Applet*



AWT versus Swing

- *AWT*
 - Alguns componentes *AWT* usam código nativo
 - Dependente de plataforma
 - Assegura que a aparência e o sentir de uma aplicação executada em diferentes máquinas seja comparável
- *Swing*
 - Escrita inteiramente usando a linguagem de programação Java
 - Independente de plataforma
 - Assegura que aplicações desenvolvidas em plataformas diferentes tenham a mesma aparência
 - Construída sobre um número de *APIs* que implementa várias partes do *AWT*



Componentes AWT: *Fundamental Window Classes*

- *Component*
- *Container*
- *Panel*
- *Window*
- *Dialog*
- *Frame*



Componentes *AWT*: Métodos da classe *Window*

- Tamanho da janela:

```
void setSize(int width, int height)
```

```
void setSize(Dimension d)
```

- Visibilidade da janela:

```
void setVisible(boolean b)
```



Componentes AWT: *Fundamental Window Classes*

- Passaremos agora para o NetBeans



Componentes *AWT: Graphics*

- Classe *Graphics* métodos abstratos:

<code>drawLine()</code>	<code>drawPolyline()</code>	<code>setColor()</code>
<code>fillRect()</code>	<code>drawPolygon()</code>	<code>getFont()</code>
<code>drawRect()</code>	<code>fillPolygon()</code>	<code>setFont()</code>
<code>clearRect()</code>	<code>getColor()</code>	<code>drawString()</code>

- Construtores da classe *Color*:

`Color(int r, int g, int b)`

`Color(float r, float g, float b)`

`Color(int rgbValue)`



Componentes *AWT*: Exemplo de *Graphics*

- Passaremos agora para o NetBeans



Mais componentes *AWT*

Label	Button	Choice
TextField	Checkbox	List
TextArea	CheckboxGroup	Scrollbar

Mais componentes *AWT*: Exemplo

- Passaremos agora para o NetBeans



Gerenciadores de *layout*

- Determina a posição e o tamanho dos componentes no container
- Gerencia a disposição dos componentes no container
- Alguns dos gerenciadores de *layout* no Java
 - *FlowLayout*
 - *BorderLayout*
 - *GridLayout*
 - *GridBagLayout*
 - *CardLayout*
 - *BoxLayout*



Gerenciadores de *layout*: Métodos

- Configurando o gerenciador de *layout*:

```
void setLayout(LayoutManager mgr)
```

- Para posicionar os elementos manualmente:

```
public void setBounds(int x, int y, int width,  
    int height)
```



Gerenciadores de *layout*: *FlowLayout*

- O gerenciador padrão para a classe *Panel* e suas subclasses
- Posiciona os componentes da esquerda para a direita e de cima para baixo, começando do canto superior esquerdo

Gerenciadores de *layout*: Construtor de *FlowLayout*

```
FlowLayout()
```

```
FlowLayout(int align)
```

```
FlowLayout(int align, int hgap, int vgap)
```



Gerenciadores de *layout*: *FlowLayout*

- Intervalo (gap)
- Possíveis valores de alinhamento:

`FlowLayout.LEFT`

`FlowLayout.CENTER`

`FlowLayout.RIGHT`



Gerenciadores de layout: Exemplo de *FlowLayout*

- Passaremos agora para o NetBeans



Gerenciadores de *layout*: *BorderLayout*

- *Layout* padrão para objetos *Window* e suas subclasses
- Divide o objeto *Container* em cinco partes onde objetos *Component* são adicionados:
 - *North* - espalha-se horizontalmente
 - *South* - espalha-se horizontalmente
 - *East* - ajusta-se verticalmente
 - *West* - ajusta-se verticalmente
 - *Center* - ajusta-se em ambas as direções



Gerenciadores de layout: Construtor de *BorderLayout*

`BorderLayout()`

`BorderLayout(int hgap, int vgap)`

Gerenciadores de layout:

BorderLayout

- Adicionando um componente a uma região específica:
 - Use o método *add* e passe dois argumentos
 - Só um componente pode ser colocado em uma região
- Regiões válidas:
 - BorderLayout.NORTH
 - BorderLayout.SOUTH
 - BorderLayout.EAST
 - BorderLayout.WEST
 - BorderLayout.CENTER



Gerenciadores de layout: Exemplo de *BorderLayout*

- Passaremos agora para o NetBeans



Gerenciadores de layout:

GridLayout

- Posiciona os componentes da esquerda para direita e de cima para baixo
- Divide o *container* em um número de linhas e colunas



Gerenciadores de layout: Construtor de *GridLayout*

- Construtores:

```
GridLayout()
```

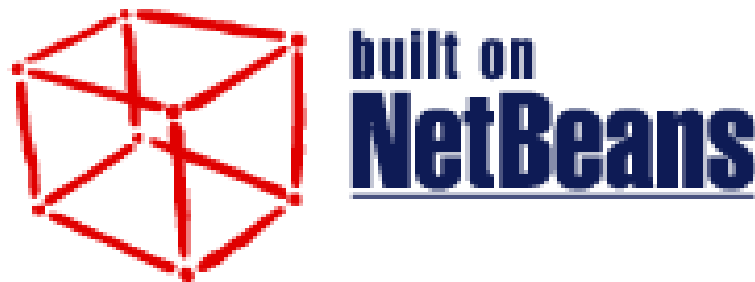
```
GridLayout(int rows, int cols)
```

```
GridLayout(int rows, int cols, int hgap, int vgap)
```



Gerenciadores de layout: Exemplo de *GridLayout*

- Passaremos agora para o NetBeans



Painéis e *layouts* complexos

- Pode-se combinar os diferentes gerenciadores de *layout*
- Utilizar vários painéis ao mesmo tempo
- Um *Panel* é um *Container* e um *Component*
- Pode-se inserir *Components* em um *Panel*
- Pode-se adicionar *Panel* a um *Container*



Painéis e *layouts* complexos

- Passaremos agora para o NetBeans



Componentes *Swing*

- Pacote encontrado em *javax.swing*
- Escrito inteiramente em Java
- Fornece componentes mais interessantes



Componentes *Swing*

- Os nomes dos componentes *GUI Swing* são quase similares aos do *AWT*
- Nome dos componentes *AWT*, mas prefixados com J

Componentes *Swing*

- *JComponent*
- *JButton*
- *JCheckBox*
- *JFileChooser*
- *JTextField*
- *JFrame*
- *JPanel*
- *JApplet*
- *JOptionPane*
- *JDialog*
- *JColorChooser*



Swing: Configurando os containers JFrame e JApplet

- *Containers* top-level no *Swing* são ligeiramente incompatíveis com *AWT*
- Adicionando um componente ao *container*:
 - Obter o conteúdo da *pane* do *container*
 - Adicione componentes ao conteúdo do *pane*

Swing: Exemplo de JFrame

- Passaremos agora para o NetBeans



Sumário

- *Abstract Windowing Toolkit (AWT) versus Swing*
- Componentes *GUI AWT*
 - *Fundamental Window Classes*
 - Gráficos
 - Mais Componentes *AWT*
- Gerenciadores de *Layout*
- Componentes *GUI Swing*

Parceiros

- Os seguintes parceiros tornaram JEDITM possível em Língua Portuguesa:

