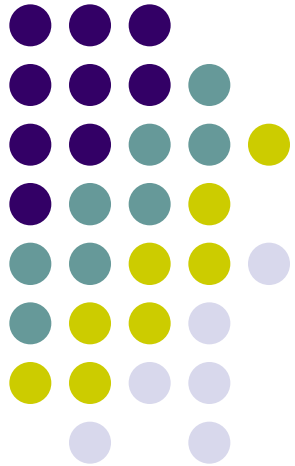
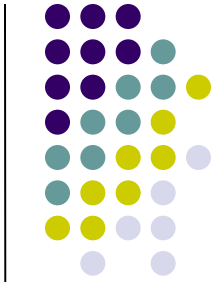
COM220

Aula 20: Gerenciadores de Layout Parte 1

Prof. Laércio Baldochi

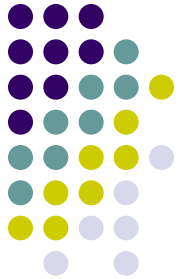


Gerenciadores de Layout



- Visão Geral
- Posicionamento Absoluto
- FlowLayout
- BorderLayout
- GridLayout
- CardLayout
- BoxLayout
- GridBagLayout

Gerenciadores de Layout
Visão Geral

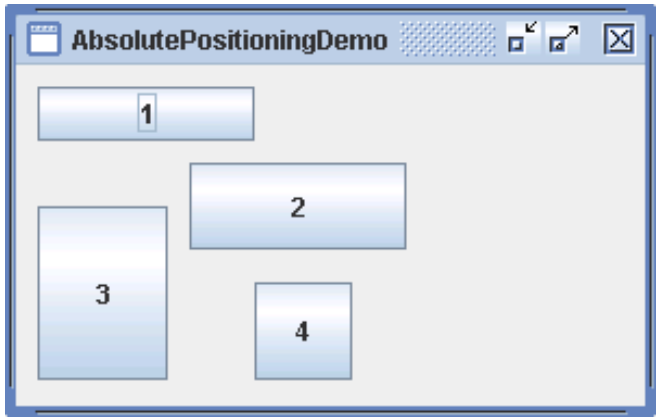


- São objetos responsáveis por controlar o tamanho e o posicionamento dos componentes dentro de um container
- Possibilitam que a interface seja apresentada de maneira consistente, independentemente da resolução da tela ou do tamanho da janela
- Gerenciadores
  - Implementam a interface LayoutManager, a qual é associada ao container
    - setLayout(layoutManager)

Posicionamento Absoluto

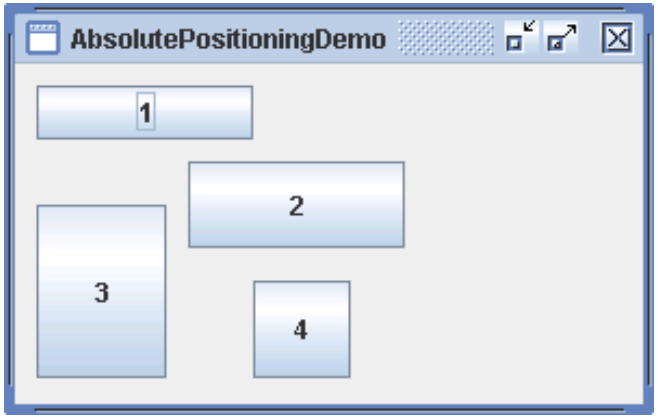
- Dispensa o uso de um gerenciador de layout
- Componentes são posicionados por meio de suas coordenadas no plano (x, y)
- Procedimento:
  - setLayout(null)
  - Métodos do componente
    - setLocation(x, y)
    - setSize(width, height)

    ou
    - setBounds(x, y, width, height)



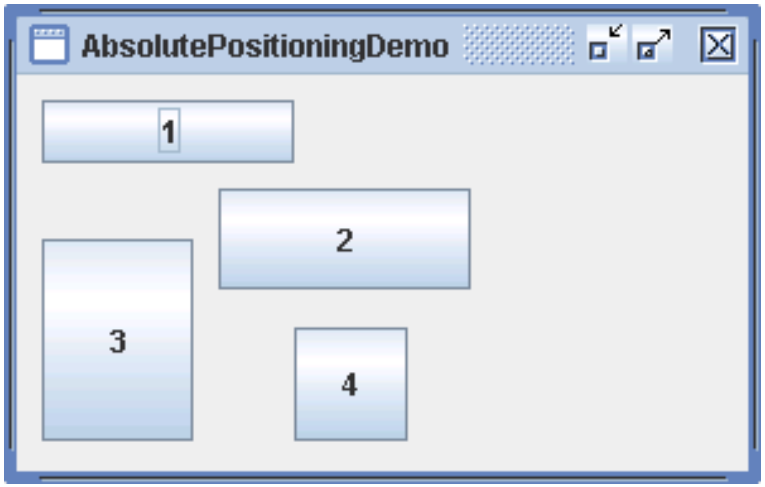
Posicionamento Absoluto Exemplo

```
public AbsolutePositioningDemo() {
    // Obtém o container e seta o layout para null
    Container contentPane = this.getContentPane();
    contentPane.setLayout(null);
    // Constrói os componentes e configura seus tamanhos e posições
    JButton b1 = new JButton("1");
    b1.setSize(100, 25);
    b1.setLocation(10, 10);
    JButton b2 = new JButton("2");
    b2.setBounds(80, 45, 100, 40);
    JButton b3 = new JButton("3");
    b3.setBounds(10, 65, 60, 80);
    JButton b4 = new JButton("4");
    b4.setBounds(110, 100, 45, 45);
    // Adiciona os componentes ao container
    contentPane.add(b1);
    contentPane.add(b2);
    contentPane.add(b3);
    contentPane.add(b4);
    // Ajusta o frame
    setSize(300, 190);
    setVisible(true);
    setTitle("AbsolutePositioningDemo");
}
```



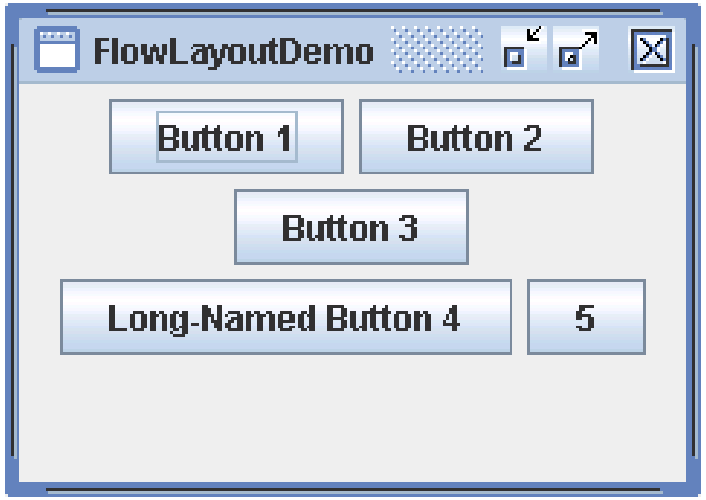
Posicionamento Absoluto

- Importante
  - Quando uma janela é redimensionada, componentes permanecem com o mesmo tamanho e na mesma posição

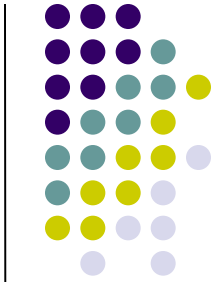


FlowLayout

- Arranja os componentes na ordem em que vão sendo adicionados ao container
  - Da esquerda para a direita, de cima para baixo
  - Utilizando o tamanho preferencial de cada componente



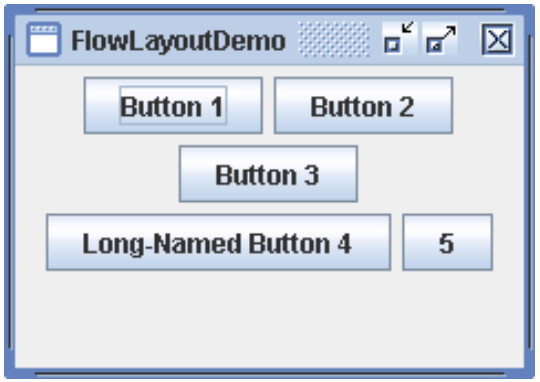
FlowLayout



- Disponibiliza construtor que permite o alinhamento dos componentes dentro do container e a distância mínima entre eles
  - FlowLayout(align, hgap, vgap)
- Valores para align
  - FlowLayout.LEFT
  - FlowLayout.RIGHT
  - FlowLayout.CENTER
  - FlowLayout.LEADING
    - Justifica os componentes de acordo com a 1ª borda
  - FlowLayout.TRAILING
    - Justifica os componentes de acordo com a última borda

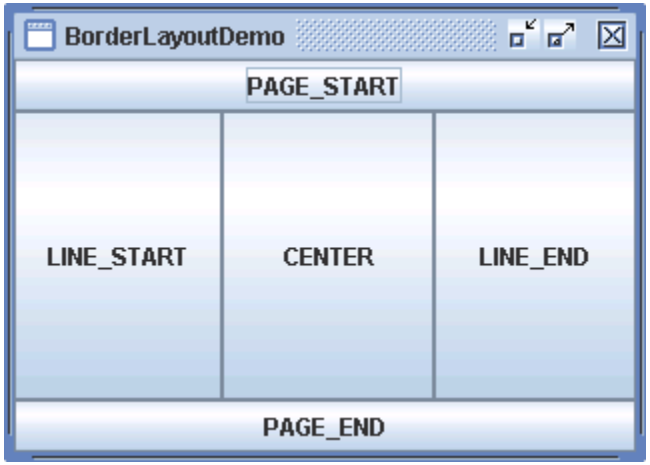
FlowLayout Exemplo

```
public FlowLayoutDemo() {

    // Obtém o container e seta o layout
    Container contentPane = this.getContentPane();
    contentPane.setLayout(new FlowLayout());

    // Adiciona componentes de diferentes tamanhos
    contentPane.add(new JButton("Button 1"));
    contentPane.add(new JButton("Button 2"));
    contentPane.add(new JButton("Button 3"));
    contentPane.add(new JButton("Long-Named Button 4"));
    contentPane.add(new JButton("5"));

    // Inicializa o frame
    setSize(240, 170);
    setTitle("FlowLayoutDemo");
    setVisible(true);
}
```

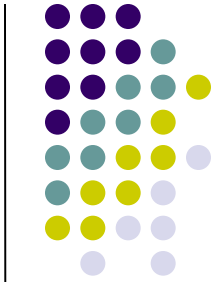


BorderLayout

- Divide o container em 5 regiões
  - PAGE_START
  - LINE_START
  - CENTER
  - LINE_END
  - PAGE_END
- Componentes são adicionados preenchendo todo espaço de cada região



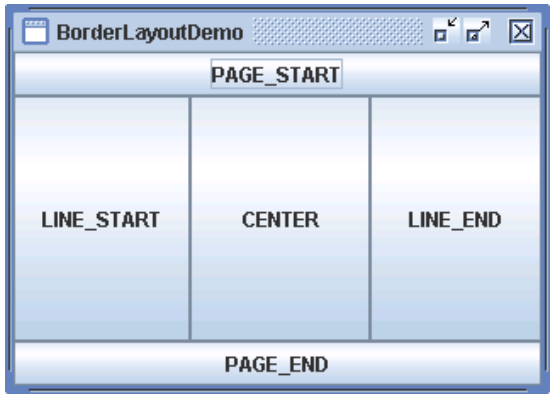
BorderLayout



- É possível ajustar a distância horizontal e vertical entre as cinco regiões
  - Construtor
    - BorderLayout(hgap, vgap)
  - Métodos
    - setHgap(hgap)
    - setVgap(vgap)
- Posição do componente no container
  - Método -> add(component, position)
  - position
    - BorderLayout.PAGE_START
    - ...

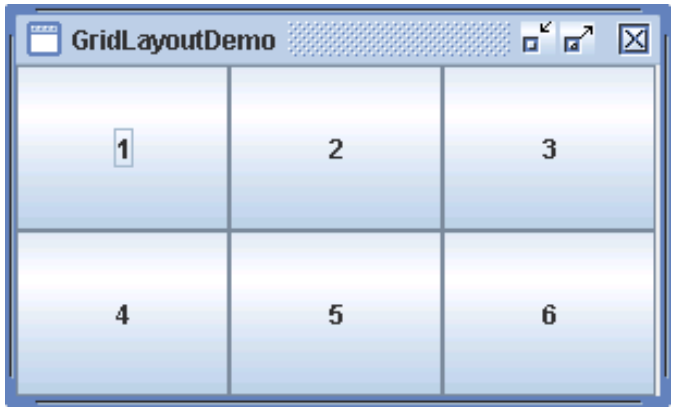
BorderLayout Exemplo

```
public BorderLayoutDemo() {
   // Obtém o container e seta o layout
   Container contentPane = this.getContentPane();
   contentPane.setLayout(new BorderLayout());
   // Cria os componentes
   JButton b1 = new JButton("PAGE_START");
   JButton b2 = new JButton("LINE_START");
   JButton b3 = new JButton("CENTER");
   JButton b4 = new JButton("LINE_END");
   JButton b5 = new JButton("PAGE_END");
   // Adiciona os componentes e define suas localizações no layout
   contentPane.add(b1, BorderLayout.PAGE_START);
   contentPane.add(b2, BorderLayout.LINE_START);
   contentPane.add(b3, BorderLayout.CENTER);
   contentPane.add(b4, BorderLayout.LINE_END);
   contentPane.add(b5, BorderLayout.PAGE_END);
   // Inicializa o frame
   setSize(240, 170);
   setVisible(true);
   setTitle("BorderLayoutDemo");
}
```

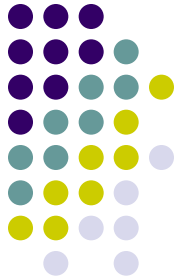


GridLayout

- Posiciona os componentes em uma grade
- Cada componente ocupa o espaço de uma célula
- Todas as células possuem o mesmo tamanho
- Se o tamanho da janela for alterado, o GridLayout modifica as células de modo que elas sempre preencham todo o container uniformemente



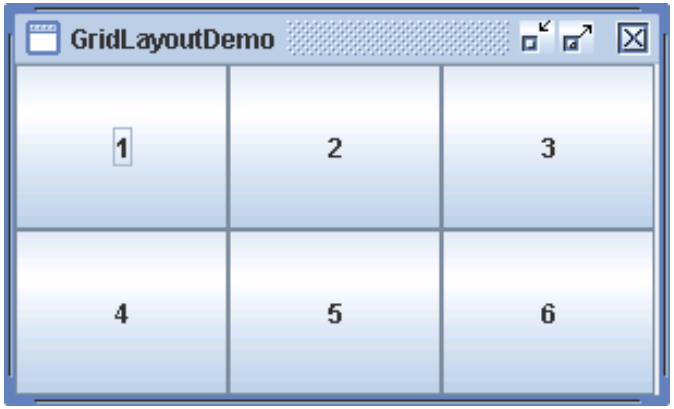
GridLayout



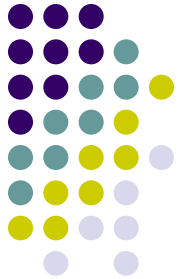
- Número de linhas e colunas é especificado no construtor
  - GridLayout(rows, cols)
- Pode-se também definir a distância horizontal e vertical entre as células
  - GridLayout(rows, cols, hgap, vgap)
- Componentes são posicionados na grade
  - Da esquerda para a direita
  - De cima para baixo
  - Na ordem em que são adicionados ao container

GridLayout Exemplo

```
public GridLayoutDemo() {
    // Obtém o container e seta o layout
    Container contentPane = this.getContentPane();
    // Duas linhas e três colunas na grade
    contentPane.setLayout(new GridLayout(2, 3));
    // Cria e adiciona os componentes
    contentPane.add(new JButton("1"));
    contentPane.add(new JButton("2"));
    contentPane.add(new JButton("3"));
    contentPane.add(new JButton("4"));
    contentPane.add(new JButton("5"));
    contentPane.add(new JButton("6"));
    // Seta o frame
    setSize(240, 170);
    setVisible(true);
    setTitle("GridLayoutDemo");
}
```



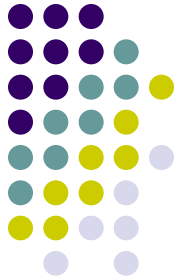
GridLayout



- Crescimento dinâmico da grade
- Se um dos parâmetros (linha ou coluna) é especificado com valor 0, grade pode crescer dinamicamente, com a adição de novos objetos ao container
- Exemplo
  - GridLayout(5, 0)
  - 5 linhas e n colunas

Exercício 1

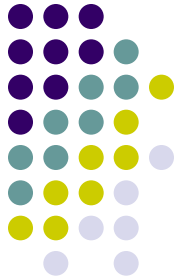
Entrega: 18/06



Crie uma calculadora com as quatro operações básicas (+, -, x, /), contendo os botões de números e operações arranjados sobre um GridLayout. Um JLabel deve ser usado para imprimir os números e o resultado. Use o espaçamento entre células.

Exercício 2

Entrega: 18/06



Crie uma aplicação utilizando o gerenciador BorderLayout contendo: um painel com um JTextField e um botão em PAGE_START, um JEditorPane em CENTER, e um JLabel em PAGE_END. Quando o usuário clicar no botão, o JEditorPane deve exibir a página HTML indicada pelo endereço no JTextField. O status do carregamento da página e mensagens de erro devem ser exibidos no JLabel.