

Orientação a Objetos

Prof. Me. Fabiano Fernandes

Conteúdo da aula

- Gerenciamento de leiaute.

Conteúdo da aula

- Apresentar o conceito de gerenciamento de leiaute.
- Indicar as vantagens advindas do uso de gerenciadores de leiaute.
- Demonstrar o uso dos gerenciadores de leiaute mais comuns.

Gerenciadores de leiaute

- **Posicionamento absoluto**
- **Gerenciadores de leiaute**
- **Programação visual em uma IDE**

Gerenciadores de leiautes

Posicionamento e dimensionamento com valores fixos apresenta problemas

- Portabilidade
- Manutenibilidade

Solução

- Posicionamento e dimensionamento auto-adaptável

Gerenciadores de leiautes

Cada componente Java tem um tamanho preferencial:

Normalmente é o menor tamanho necessário para apresentar o componente de maneira visualmente significativa

Exemplo de um botão é o tamanho de sua etiqueta de texto

O Gerenciador de Layout equilibra duas considerações: a política de layout e o tamanho preferencial de cada componente. A prioridade é para a política de layout

Gerenciadores de leiautes

O Gerenciador de Layout equilibra duas considerações:

- a política de layout e o tamanho preferencial de cada componente.
- A prioridade é para a política de layout

Gerenciadores de leiautes

Apresentar os gerenciadores de layout:

- FlowLayout
- BorderLayout
- GridLayout
- BoxLayout
- GridBagLayout

Gerenciadores de leiautes

Principais métodos da classe `java.awt.Container`

- `setLayout(LayoutManager m)`
- `void add(Component c)` – adiciona um componente ao container sob o controle do gerenciador de layout utilizado

FlowLayout

- É o gerenciador de layout mais simples
- Dispõe os objetos sequencialmente da esquerda para a direita na ordem em que foram adicionados

```
container.setLayout(new FlowLayout());
```

BorderLayout

- Organiza os componentes em cinco áreas
 - Norte, Sul, Leste, Oeste e Centro
- Os componentes devem ser adicionadas a regiões nomeadas no gerenciador de layout, caso contrário não ficarão visíveis: NORTH, SOUTH, EAST, WEST, CENTER

```
container.setLayout(new BorderLayout());
```

GridLayout

- Divide o container em uma grade de modo que os componentes podem ser colocados em linhas e colunas
- Os componentes são adicionados iniciando na célula na parte superior esquerda da grade e prosseguindo da esquerda para a direita até a linha estar cheia. Então, o processo continua na próxima linha

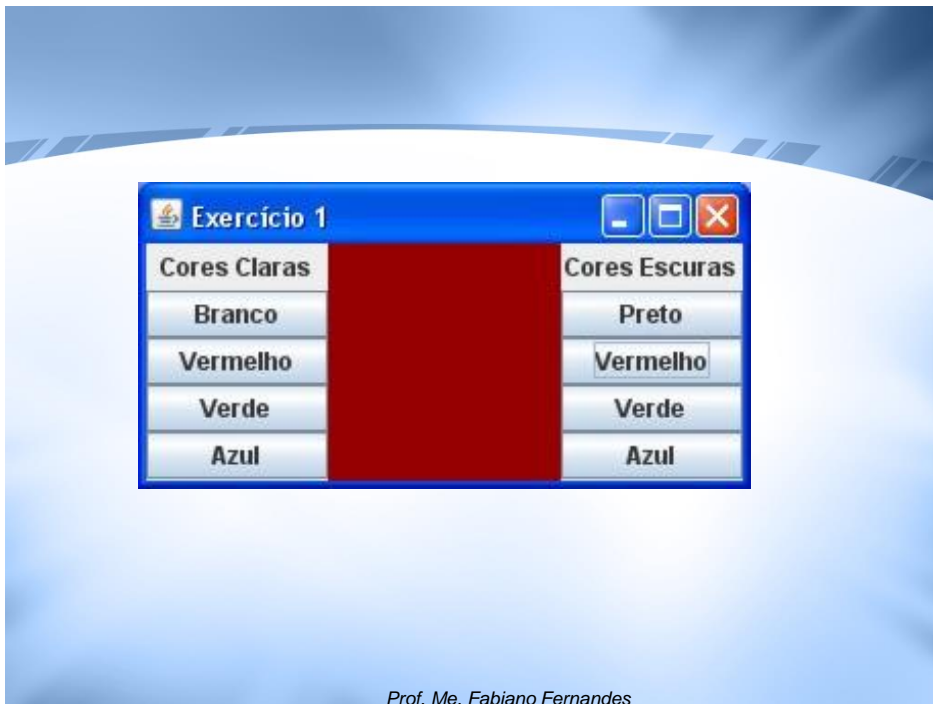
```
container.setLayout(new GridLayout());
```

Layouts complexos

- Interfaces complexas exigem que cada componente
- Para tanto, geralmente usam-se múltiplos painéis, cada um com seu layout específico

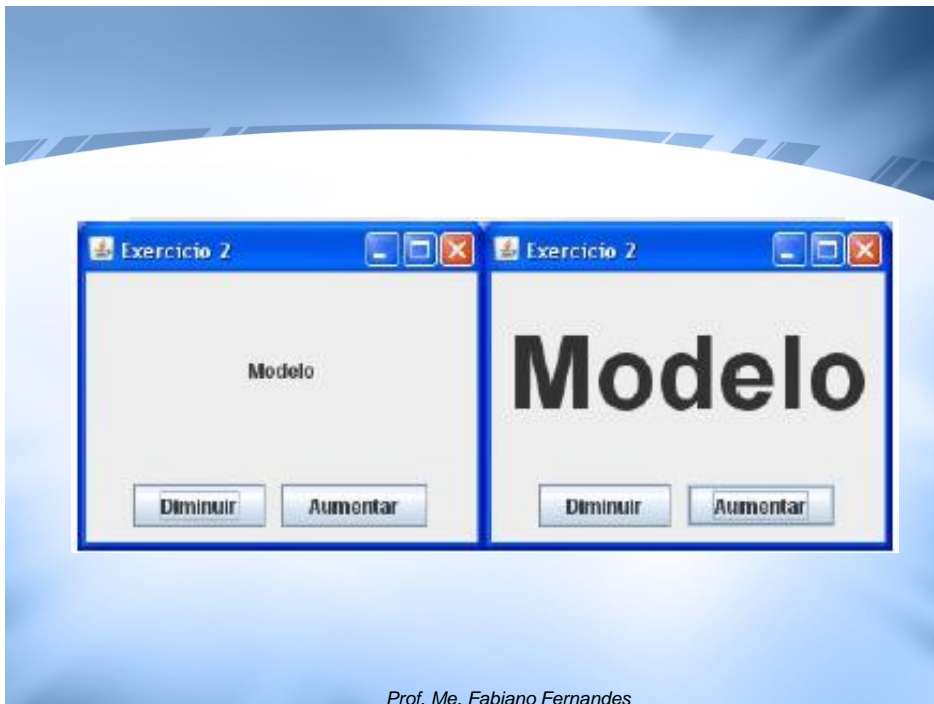
Exercício 1

- Crie uma nova janela, chamada `SelecaoCor`, de acordo com o modelo apresentado na figura abaixo.
 - Esta janela deve apresentar oito botões, quatro deles dispostos no lado esquerdo e quatro no lado direito.
 - Sempre que um destes botões for pressionado, a cor correspondente deve ser aplicada ao painel de conteúdo desta janela.



Exercício 2

- Crie uma nova janela, chamada TamanhoFonte, de acordo com o modelo apresentado na figura.
 - Esta janela deve conter apenas um rótulo e dois botões.
 - Os botões devem ser posicionados na região inferior da janela e o rótulo deve permanecer no centro da área restante.
 - Sempre que botão rotulado como “Diminuir” for pressionado, o tamanho do texto do rótulo deve ser reduzido em uma unidade.
 - Sempre que botão rotulado como “Aumentar” for pressionado, o tamanho do texto do rótulo deve ser aumentado em uma unidade.



Prof. Me. Fabiano Fernandes