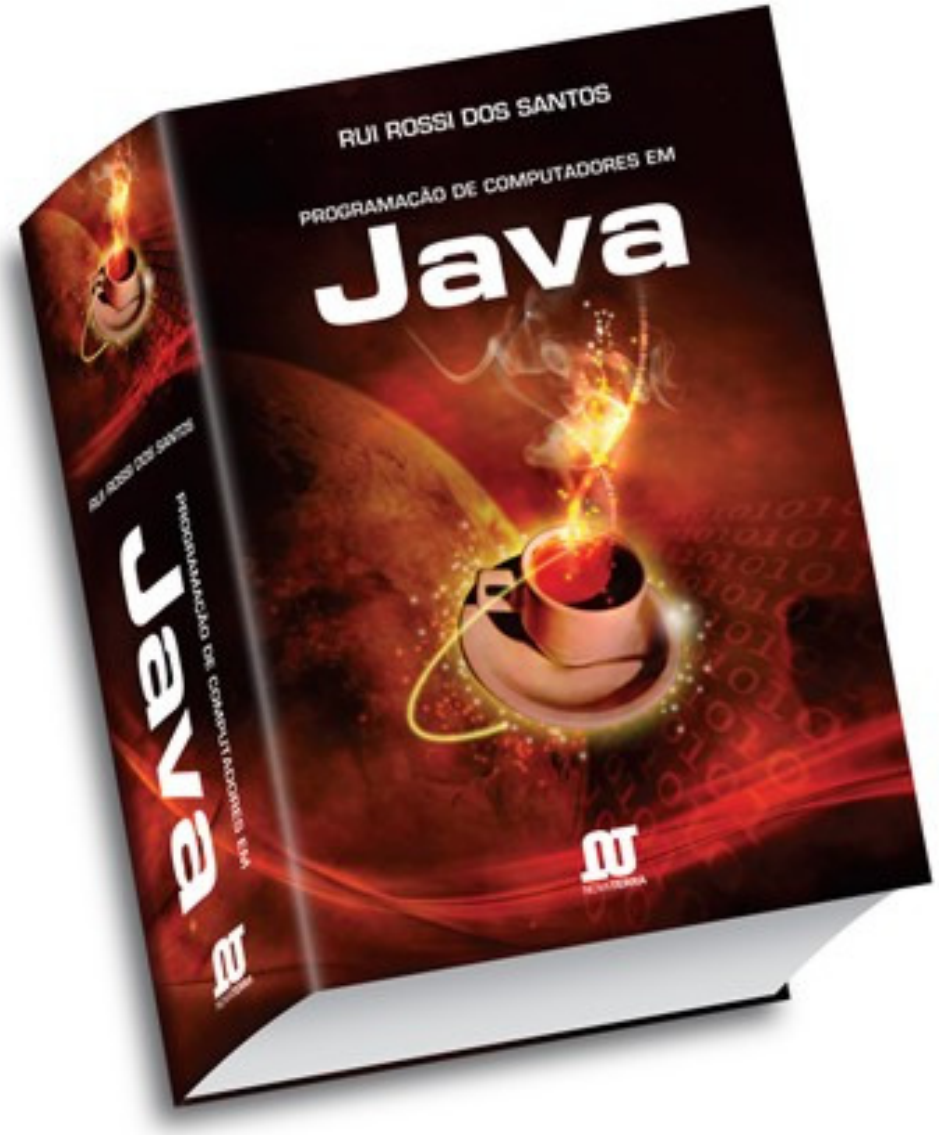


Capítulo 23

Introdução às Interfaces Gráficas

Objetivos do Capítulo

- Apresentar as duas APIs aplicáveis ao desenvolvimento de interfaces gráficas: o AWT e o Swing.
- Explorar algumas classes do AWT aplicáveis à realização de tarefas comuns.
- Analisar o uso da classe javax.swing.JFrame para construção de janelas.
- Explorar as classes do Swing que são empregadas para a criação de rótulos, botões e painéis.

AWT e Swing

- **GUI**
  - **Graphic User Interface**
  - **Baseada em componentes**
  - **Ambiente gráfico**

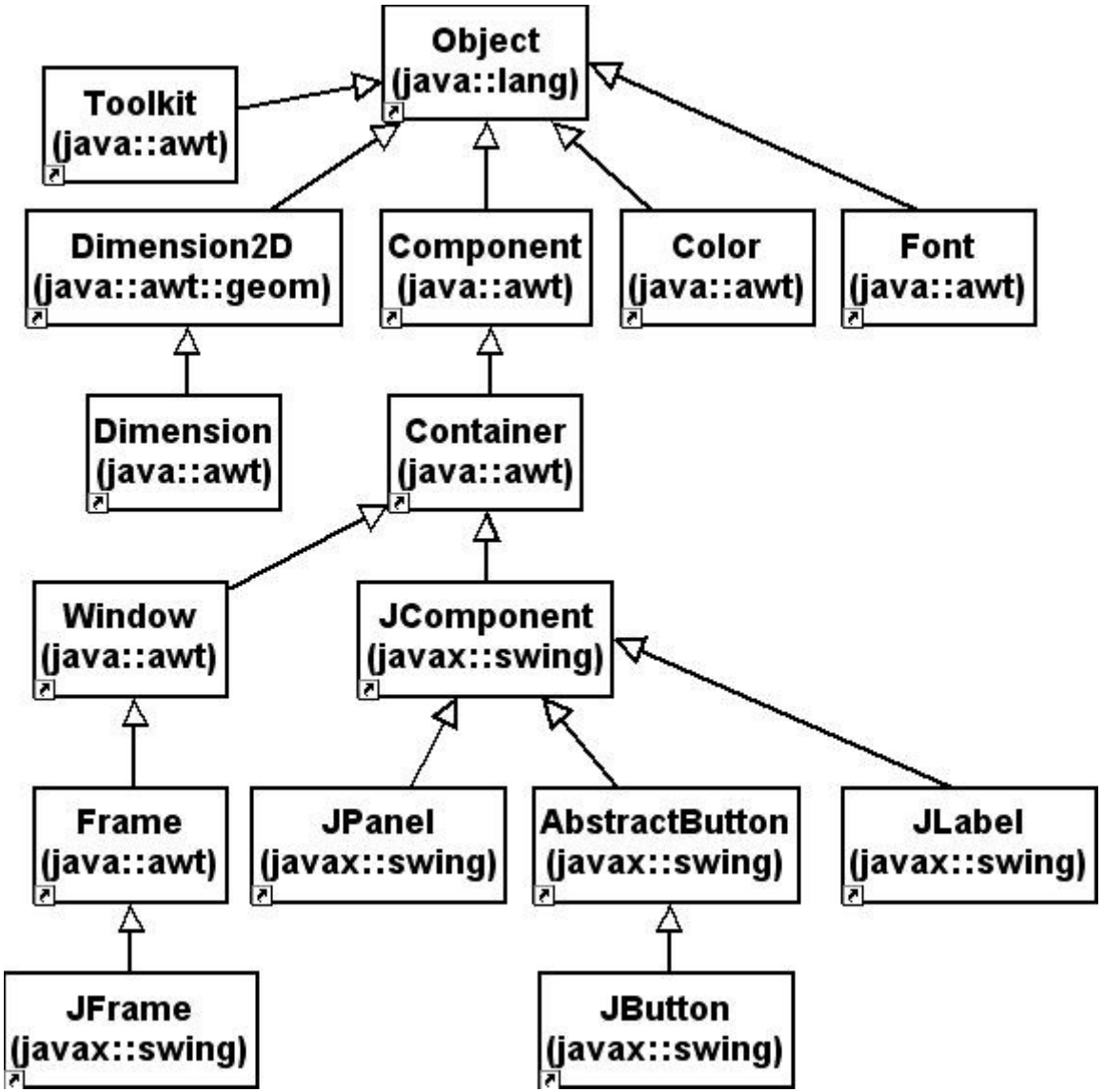
AWT e Swing

- **AWT**
  - **Abstract Window Toolkit**
  - **Desde: 1.0**
  - **Pacote: java.awt**
  - **Características:**
    - **Criação dos componentes delegada ao Sistema Operacional (SO)**
    - **Aparência e comportamento aderentes ao SO**
    - **Slogan: "escreva uma vez, execute em qualquer lugar"**
  - **Dificuldades:**
    - **Aparência e comportamento diferente em cada SO**
    - **Poucos componentes (mínimo denominador comum dos SOs)**
    - **Erros diferentes em cada SO**
      - **Ironia: "escreva uma vez, depure em todo lugar"**

AWT e Swing

- **Swing**
  - **Desde: 1.0**
  - **Pacote: javax.swing**
  - **Características:**
    - **Componentes escritos em Java**
      - **Indepententes de plataforma**
      - **Exceção: java.awt.Window e classes derivadas (javax.swing.JFrame)**
    - **Número maior de componentes**
    - **Aparência e comportamento plugáveis e dinânicos**
      - **Independentes ou aderentes aos SOs**
  - **Limitações:**
    - **O Swing não é um substituto completo do AWT**
    - **Exemplos de componentes substituídos**
      - **java.awt.Button -> javax.swing.JButton**
      - **java.awt.Label -> javax.swing.JLabel**
      - **java.awt.Panel -> javax.swing.JPanel**
    - **Exemplos de recursos não substituídos**
      - **Tratamento de eventos**
      - **Outros: Component, Container, Window, Font, Color, Toolkit, etc.**

AWT e Swing


Object
(java::lang)
Toolkit
(java::awt)
Dimension2D
(java::awt::geom)
Component
(java::awt)
Color
(java::awt)
Font
(java::awt)
Dimension
(java::awt)
Container
(java::awt)
Window
(java::awt)
JComponent
(javax::swing)
Frame
(java::awt)
JPanel
(javax::swing)
AbstractButton
(javax::swing)
JLabel
(javax::swing)
JFrame
(javax::swing)
JButton
(javax::swing)

Recursos Essenciais do AWT

- **java.awt.Color: encapsula uma cor no padrão RGB**
  - **Atributos estáticos**
    - **Color BLACK**
    - **Color BLUE**
    - **Color CYAN**
    - **Color DARK_GRAY**
    - **Color GRAY**
    - **Color GREEN**
    - **Color LIGHT_GRAY**
    - **Color MAGENTA**
    - **Color ORANGE**
    - **Color PINK**
    - **Color RED**
    - **Color WHITE**
    - **Color YELLOW**
  - **Construtor**
    - **Color(int r, int g, int b)**
    - **Color(float r, float g, float b)**

Recursos Essenciais do AWT

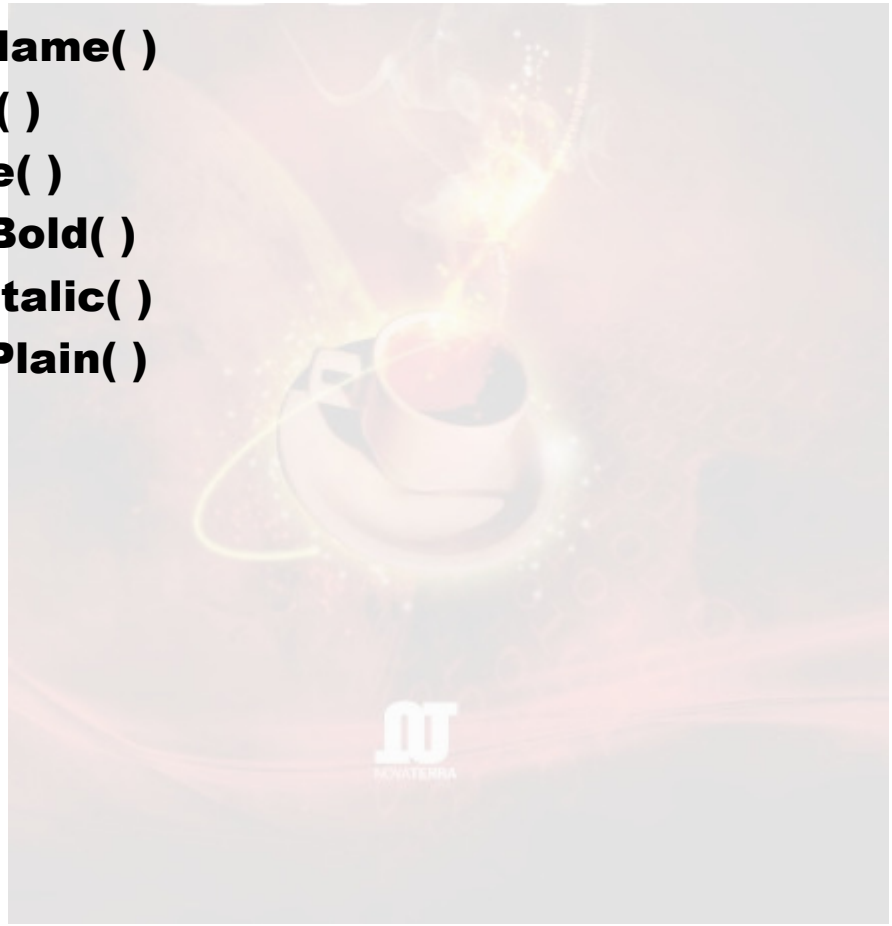
- **java.awt.Color**
  - **Métodos**
    - **int getBlue( )**
    - **int getGreen( )**
    - **int getRed( )**

Recursos Essenciais do AWT

- **java.awt.Font: representa as fontes aplicáveis aos textos dos componentes (nome, estilo e tamanho)**
  - **Atributos estáticos para o estilo**
    - **int BOLD**
    - **int ITALIC**
    - **int PLAIN**
  - **Atributos estáticos para nomes de famílias de fontes comuns**
    - **String DIALOG**
    - **String DIALOG_INPUT**
    - **String MONOSPACED**
    - **String SANS_SERIF**
    - **String SERIF**
  - **Construtor**
    - **Font(String name, int style, int size)**

Recursos Essenciais do AWT

- **java.awt.Font**
  - **Métodos**
    - **String getName( )**
    - **int getSize( )**
    - **int getStyle( )**
    - **boolean isBold( )**
    - **boolean isItalic( )**
    - **boolean isPlain( )**



Recursos Essenciais do AWT

- java.awt.Component: características comuns a todos os componentes
  - Métodos
    - Color getBackground( )
    - Font getFont( )
    - Color getForeground( )
    - void setBackground(Color c)
    - void setFont(Font f)
    - void setForeground(Color c)

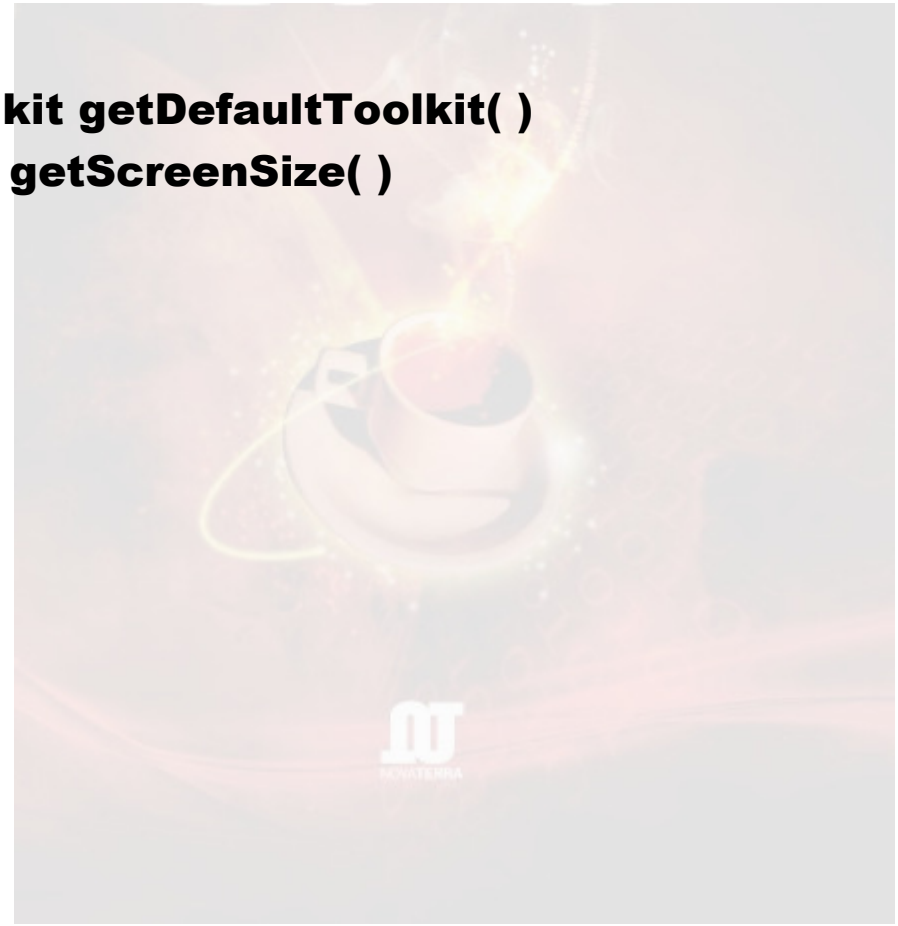
- java.awt.Container: representa um contêiner
  - Métodos
    - Component add(Component comp)
    - void setLayout(LayoutManager mgr)

Recursos Essenciais do AWT

- **java.awt.Dimension: encapsula uma dimensão de um componente (largura e altura)**
  - **Construtor**
    - **Dimension(int width, int height)**
  - **Métodos**
    - **double getHeight( )**
    - **double getWidth( )**

Recursos Essenciais do AWT

- **java.awt.Toolkit: representa o kit de ferramentas do SO subjacente e é empregada para criar os componentes AWT**
  - **Métodos:**
    - **static Toolkit getDefaultToolkit( )**
    - **Dimension getScreenSize( )**



Janelas

- javax.swing.JFrame
  - Painel de conteúdo: javax.swing.JRootPane
  - Instanciação:

    JFrame f = new JFrame( );
  - Adição de componentes:

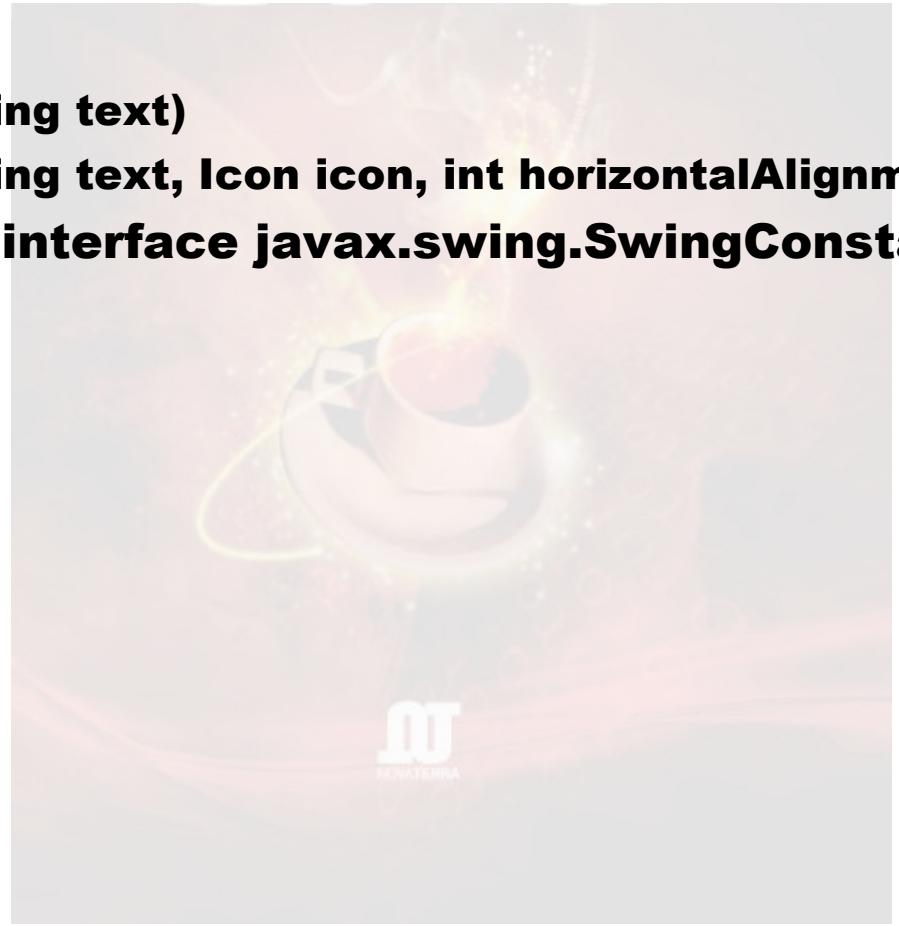
    f.getContentPane( ).add(<nome_do _componente>);
  - Métodos:
    - setTitle(String title)
    - setSize(int width, int height)
    - setIconImage(Image image)
    - setDefaultCloseOperation(int operation)
  - Atributos estáticos:
    - DO_NOTHING_ON_CLOSE
    - HIDE_ON_CLOSE
    - DISPOSE_ON_CLOSE
    - EXIT_ON_CLOSE

Janelas

- **Código 23.1 – Janela.java**

Rótulos

- javax.swing.JLabel
  - Construtores:
    - JLabel( )
    - JLabel(String text)
    - JLabel(String text, Icon icon, int horizontalAlignment)
  - Atributos da interface javax.swing.SwingConstants
    - LEFT
    - RIGHT
    - CENTER
    - TOP
    - BOTTOM



Rótulos

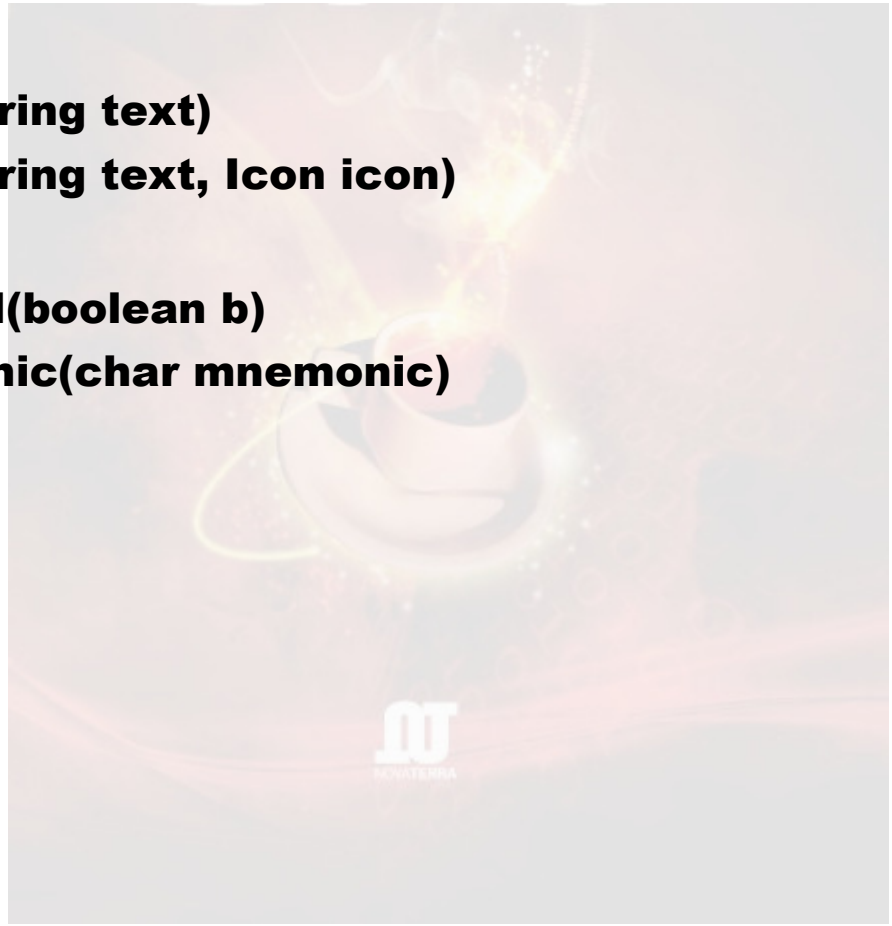
- javax.swing.JLabel
  - Métodos:
    - setText(String text)
    - setLocation(int x, int y)
    - setSize(int width, int height)
    - setBounds(int x, int y, int width, int height)
    - setOpaque(boolean isOpaque)
    - setBackground(Color bg)
    - setForeground(Color fg)
    - setFont(Font font)
    - setToolTipText(String text)
    - setIcon(Icon icon)
    - setHorizontalAlignment(int alignment)
    - setVerticalAlignment(int alignment)
    - setHorizontalTextPosition(int textPosition)
    - setVerticalTextPosition(int textPosition)

Rótulos

- **Código 23.2 – Rotulo.java**

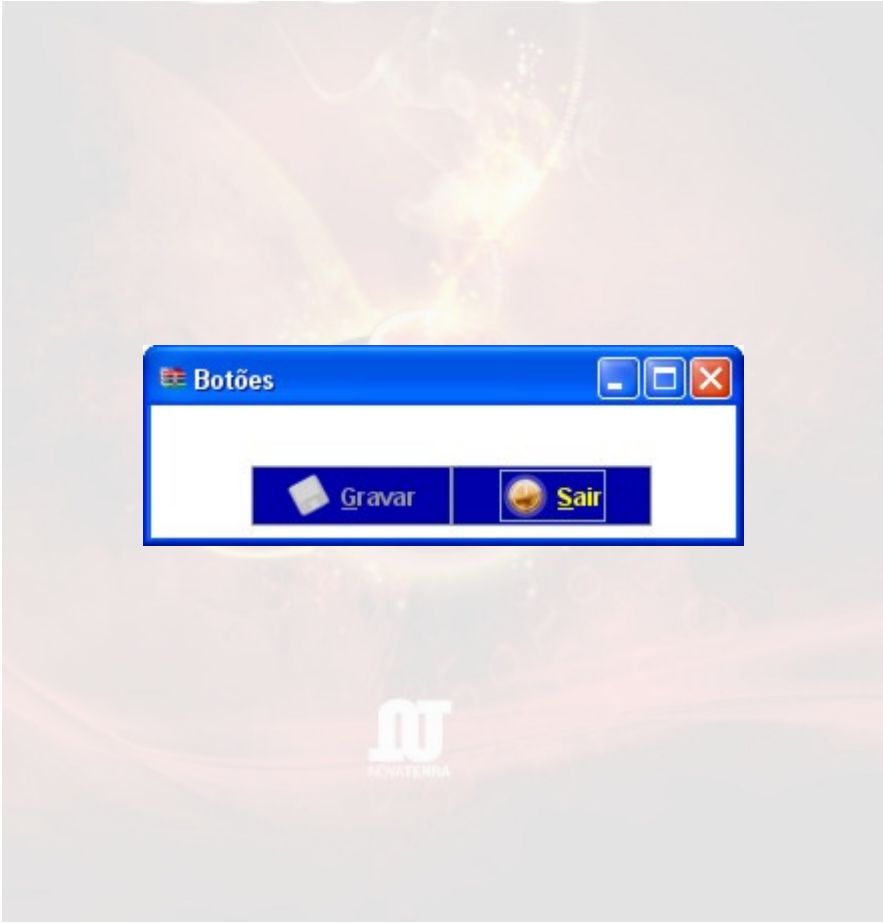
Botões

- **javax.swing.JButton**
  - **Construtores:**
    - **JButton( )**
    - **JButton(String text)**
    - **JButton(String text, Icon icon)**
  - **Métodos:**
    - **setEnabled(boolean b)**
    - **setMnemonic(char mnemonic)**



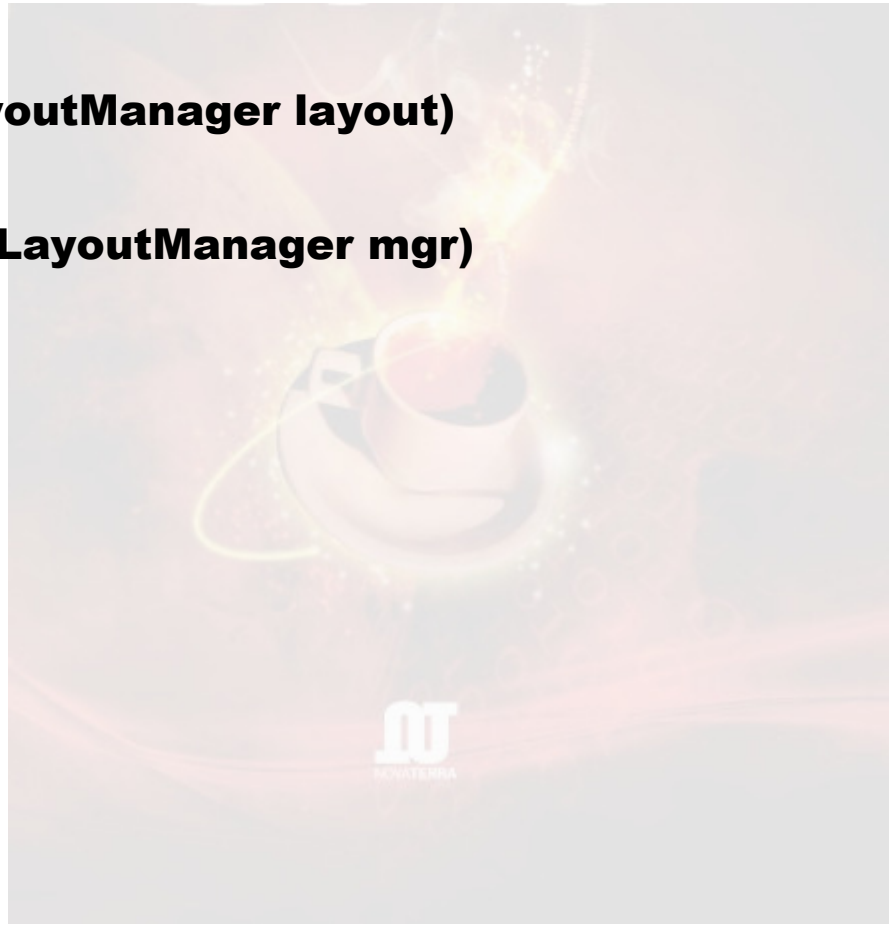
Botões

- **Código 23.3 – Botao.java**



Painéis

- **javax.swing.JPanel**
  - **Construtores:**
    - **JPanel( )**
    - **JPanel(LayoutManager layout)**
  - **Métodos:**
    - **setLayout(LayoutManager mgr)**

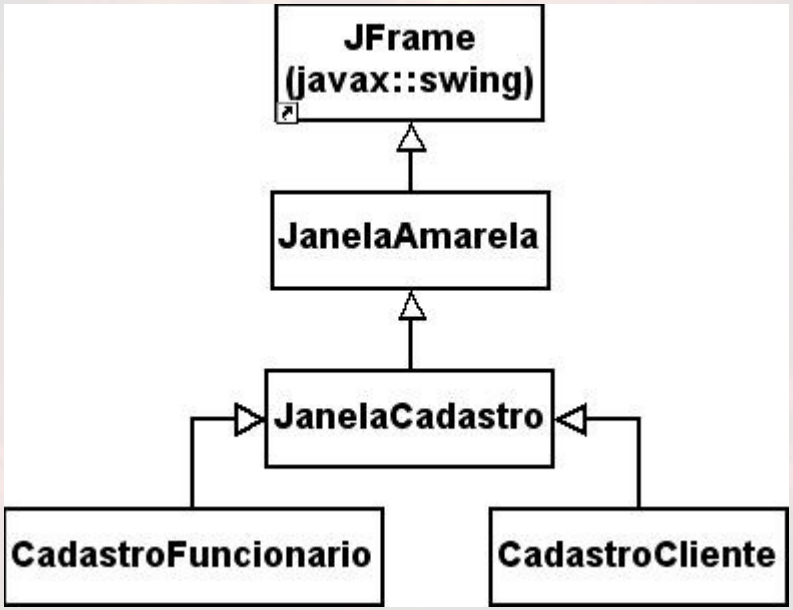


Painéis

- **Código 23.4 – Painel.java**

Exercícios

- Os exercícios propostos a seguir estão todos interligados. Todas as classes que você deverá construir estarão ligadas entre si pelo mecanismo da herança. A figura abaixo apresenta a hierarquia de classes que deverá ser construída.

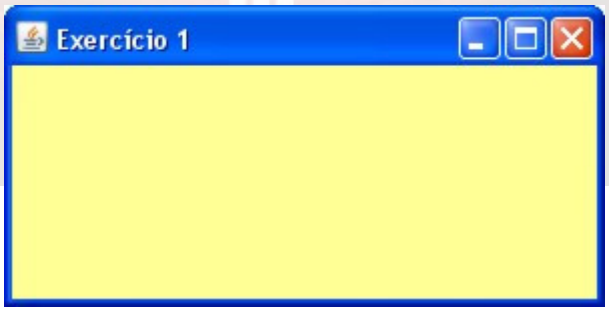


Exercícios

- A primeira classe que você irá construir é a classe chamada JanelaAmarela.
  - Note que ela derivará diretamente da classe javax.swing.JFrame.
  - O primeiro exercício especifica como esta classe deve ser construída.
- A segunda classe que você deve construir é a classe JanelaCadastro.
  - Esta classe derivará da classe JanelaAmarela e representará um modelo geral de janela para cadastros.
  - O segundo exercício especifica como construir esta classe.
- As duas últimas classes que deverão ser construídas são as classes CadastroCliente e CadastroFuncionario.
  - Elas derivarão da classe JanelaCadastro e representarão janelas de cadastro de clientes e de funcionários, respectivamente.
  - O terceiro exercício especifica com construir estas janelas.

Exercício 1

- Crie uma nova classe, chamada JanelaAmarela, que derive da classe javax.swing.JFrame.
  - Esta classe deve representar uma janela genérica do sistema que possa ser utilizada como superclasse para a criação de janelas mais específicas.
  - Ela deve definir apenas algumas das características que deverão ser assumidas por todas as demais janelas que venham a ser construídas, quais sejam:
    - A cor de fundo deve ser uma tonalidade clara de amarelo.
    - Deve ser exibida no centro da tela.
    - Deve anular o gerenciador de leiaute do painel de conteúdo.
    - Deve liberar a memória utilizada quando for fechada.
- A figura abaixo ilustra como deve ser a aparência desta janela.



Exercício 1

- **Para construir a classe JanelaAmarela, procure seguir todas as especificações que estão contidas na sua representação gráfica, apresentada na figura abaixo.**
  - **Note que esta classe deve ter um construtor com dois parâmetros.**
  - **Estes parâmetros permitirão que sejam criadas instâncias desta classe que representem janelas com diferentes títulos e tamanhos.**
  - **Eles também permitirão que as suas subclasses definam um título e um tamanho padrão para determinado tipo de janela.**
  - **O método centralizar( ) deve ser invocado pelo próprio construtor desta classe para posicioná-la no centro da tela.**
  - **No método main( ), crie uma instância da própria classe JanelaAmarela e invoque o seu método setVisible( ) para exibi-la.**

Exercício 2

- **Crie uma nova classe, chamada JanelaCadastro, que derive da classe JanelaAmarela.**
  - **Ela deve ser uma janela genérica que possa ser utilizada como superclasse para a criação de quaisquer janelas de cadastros.**
  - **Ela deve conter três painéis com fundo em uma tonalidade clara de verde.**
  - **A figura abaixo ilustra como deve ser a aparência desta janela.**

Exercício 2

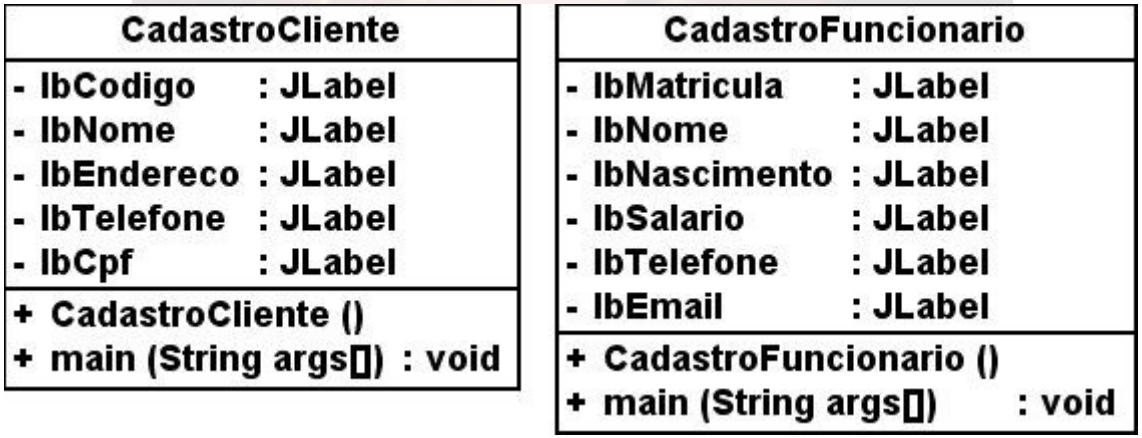
- O tamanho desta janela deve ser fixado em 400 pixels de largura e 300 pixels de altura.
  - Procure posicionar os painéis e os botões de acordo com o modelo apresentado pela figura do slide anterior.
  - Defina ícones, dicas e teclas de atalho para os botões e utilize tonalidades mais escuras de verde para o fundo e para a fonte dos mesmos.
- Para construir a classe JanelaCadastro, procure seguir todas as especificações que estão contidas na sua representação gráfica, apresentada na figura abaixo.
  - O método main( ) deve apenas criar uma instância desta classe e invocar o seu método setVisible( ) para exibi-la.

Exercício 3

- Crie duas novas classes, chamadas CadastroCliente e CadastroFuncionario, que derivem da classe JanelaCadastro.
  - Estas classes devem representar janelas de cadastros de clientes e de funcionários, respectivamente.
  - Elas devem apenas inserir os rótulos relativos aos dados cadastrais de clientes e de funcionários no painel que se encontra no lado esquerdo da janela.
  - A figura abaixo ilustra a aparência destas duas janelas.

Exercício 3

- **Para construir as classes CadastroCliente e CadastroFuncionario, procure seguir as especificações apresentadas na figura abaixo.**
  - **Note que todos os componentes destas janelas devem ser declarados como atributos privados.**
  - **Os seus construtores devem definir um título para a janela, criar os rótulos, configurá-los e adicioná-los ao painel correto.**
  - **O método main( ) deve instanciar a classe e invocar o seu setVisible( ) da instância criada para exibi-la.**



Contato

Com o autor:

Rui Rossi dos Santos
E-mail: livros@ruirossi.pro.br
Web Site: http://www.ruirossi.pro.br

Com a editora:

Editora NovaTerra
Telefone: (21) 2218-5314
Web Site: http://www.editoranovaterra.com.br