Orientação a Objetos

Prof. Me. Fabiano Fernandes

Conteúdo da aula

- AWT e Swing

Conteúdo da aula

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

AWT e Swing


Sistema para escolas
Cadastro Sistema
Cadastro de usuários
Seleção Manutenção
Código Nome
1 Carlos Vieira
2 Patrícia Pires
3 Homero Junqueira
4 Sandro Lima
5 Luciana Gomes
Sistema para escolas
Cadastro Sistema
Cadastro de usuários
Seleção Manutenção
Codigo 2
Nome Patrícia Pires
Login ppires
Senha @@@@@@@@
Incluir Alterar Excluir

AWT

Abstract Window Toolkit

Pacote: java.awt

Características:

- Criação dos componentes delegada ao Sistema Operacional (SO)
- Aparência e comportamento aderentes ao SO

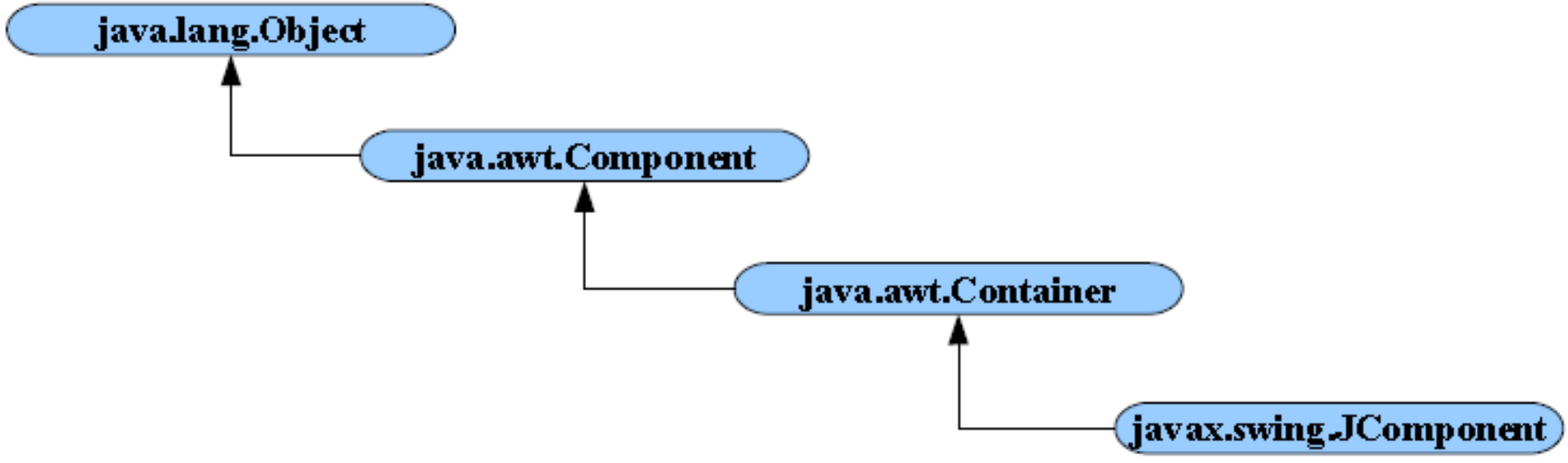
Swing

Pacote: javax.swing

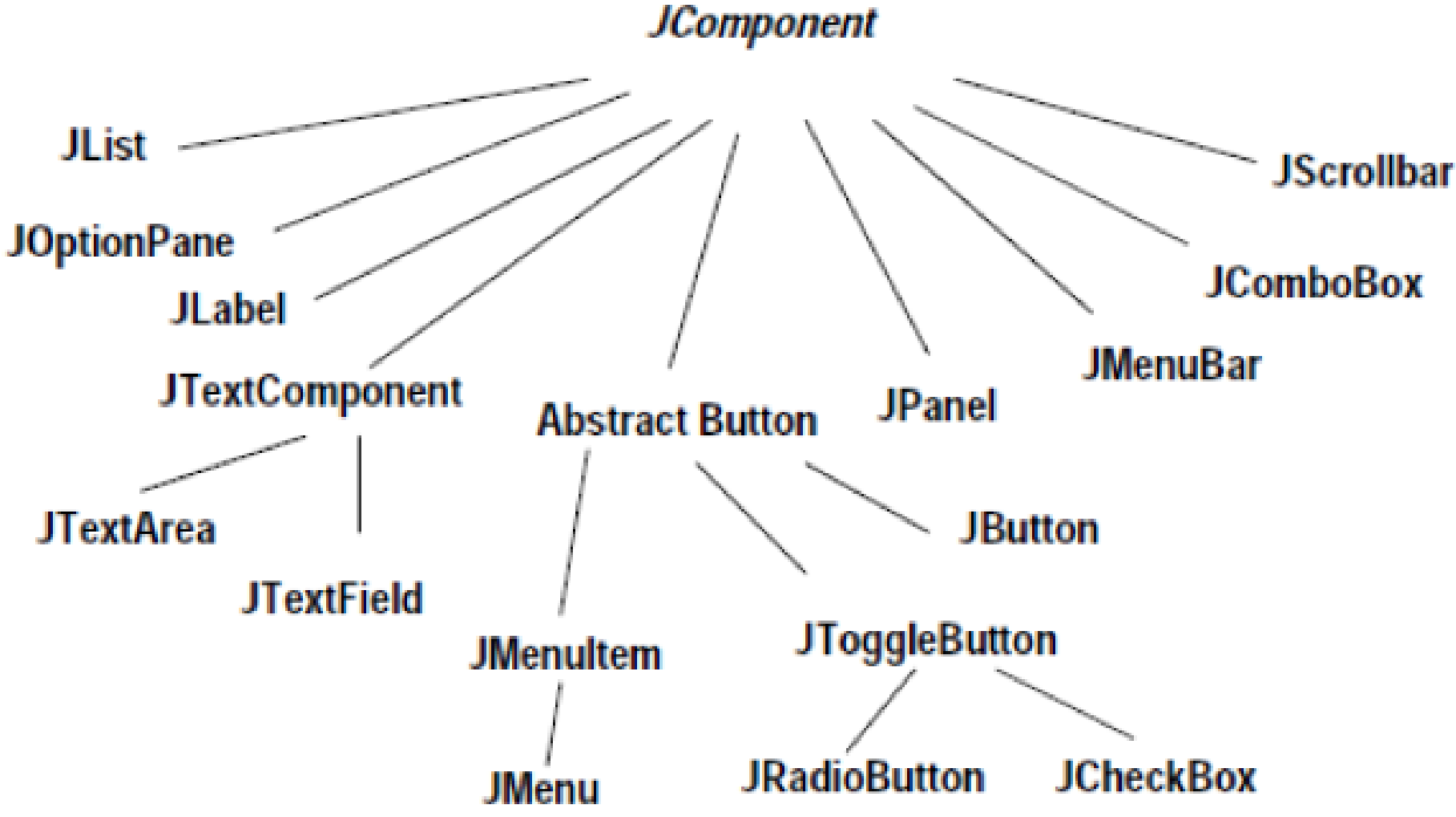
Características:

- Componentes escritos em Java
- Número maior de componentes

Hierarquia dos componentes



Hierarquia dos componentes



Hierarquia dos componentes

- ***top-level container:*** fornece a área base para que os outros componentes swing possam ser inseridos. (ex: JFrame)
- ***intermediate container****:* fornece suporte de posicionamento e acessórios para que outros componentes swing possam ser inseridos. (ex: JPanel, JscrollPane, ...)
- ***atomic components:*** entidades auto- suficientes que representam informações para o usuário. (ex: JTextField)

Componentes Básicos

- **JFrame** - É um *container* (formulário) para outros componentes GUI.
- **JPanel** - *Container* em que os componentes podem ser colocados.
- **JLabel** - Área em que podem ser exibidos texto não-editável ou ícones.
- **JTextField** - Área em que o usuário insere dados pelo teclado.
- **JButton** - Área que aciona um evento quando o usuário clica.
- **JCheckBox** - Possui dois estados: selecionado ou não-selecionado.
- **JComboBox** - Lista de itens que o usuário pode fazer uma seleção clicando em um item na lista ou digitando na caixa.
- **JList** - Área em que uma lista é exibida, possibilitando a seleção clicando em qualquer item da lista.

Janela - JFrame

Métodos:

- setTitle(String title)
- setSize(int width, int height)
- setDefaultCloseOperation(int operation)

Atributos estáticos:

- DO_NOTHING_ON_CLOSE
- HIDE_ON_CLOSE
- DISPOSE_ON_CLOSE
- EXIT_ON_CLOSE

Janela - JFrame

- javax.swing.JFrame Utilizado geralmente através de herança

  public class MinhaJanela **extends JFrame** {...}

- **JFrame**() e **JFrame**(String title): Construtores

- void **setDefaultCloseOperation**(int operation)

  Altera a operação padrão a ser executada quando o usuário fechar o Jframe
- void **setVisible**(): exibe o janela do Jframe
- void **setIconImage**(Image image): altera o ícone do canto superior esquerdo
- void **setResizable**(boolean resizable): define se a janela poderá ser redimensionada
- void **setTitle**(String title): altera o título da janela

Painel - JPanel

- **JPanel**() e **JPanel**(LayoutManager layout): Construtores

- setLayout(LayoutManager layout)
- void add(Component c)
- boolean contains(int x, int y)
- Color getBackground() e void setBackground(Color c)
- Rectangle getBounds() e void setBounds(int x, int y, int width, int height)
- Font getFont() e void setFont(Font f)
- Color getForeground() e void setForeground(Color c)
- Dimension getPreferredSize()
- Dimension getSize() e void setSize(int width, int height)
- int getWidth() e int getHeight()
- void setEnabled(boolean b)
- void setLocation(int x, int y)

javax.swing.JLabel - Métodos

- JLabel(String text) e JLabel(String text, int horizontalAlignment)

  Construtores mais utilizados
- JLabel(Icon image) e JLabel(String text, Icon icon, int horizontalAlignment)

  Construtores com imagem (ícones)
- String getText() e void setText(String text)

  Recupera ou altera o texto
- void setHorizontalTextPosition(int textPosition)

  Altera o alinhamento horizontal do texto
- void setVerticalTextPosition(int textPosition)

  Altera o alinhamento vertical do texto

javax.swing.text.JTextComponent - Métodos

- Superclasse direta dos componentes de edição de texto: JTextField, JTextArea e JEditorPane
- void copy(): copia para a área de transferência o texto selecionado
- void cut(): move para a área de transferência o texto selecionado
- void paste(): transfere o conteúdo da área de transferência para o componente
- String getSelectedText(): retorna o texto selecionado
- int getSelectionStart() e int getSelectionEnd(): retorna o início e o final da seleção
- String getText() e String getText(int offs, int len): retorna o texto
- void select(int selectionStart, int selectionEnd) e void selectAll(): seleciona o texto
- void setEditable(boolean b): permite editar ou não o conteúdo do componentevoid
- void setText(String t): altera o texto

- **javax.swing.JButton - Métodos**
- JButton(Icon icon); JButton(String text) e JButton(String text, Icon icon): construtores
- public void setMnemonic(char mnemonic): destaca uma letra para servir como atalho
- void setText(String text): altera o texto do botão

- **javax.swing.JCheckBox - Métodos**
- JCheckBox(String text);JCheckBox(String text, boolean selected); JCheckBox(Icon icon)
- Principais construtores
- public void setSelected(boolean b): habilita ou desabilita a seleção
- public void doClick(): executa o evento referente a um clique no componente

- **javax.swing.JRadioButton - Métodos**
- Todos os métodos e construtores são idênticos ao JCheckBox

- **javax.swing.ButtonGroup – Métodos**
- É utilizado com um container para botões, onde somente um botão pode estar ativo em
- um dado instante
- ButtonGroup(): construtor
- void add(AbstractButton b): adiciona um botão ao grupo

- **javax.swing.JComboBox – Métodos**
- JComboBox(); JComboBox(Object[] items) e JComboBox(Vector items)
- void addItem(Object anObject): adiciona um item a lista do combo
- Object getItemAt(int index): retorna o item pelo seu índice
- int getItemCount(): retorna o número de itens
- Object getSelectedItem() : retorna o item selecionado
- int getSelectedIndex(): retorna o índice do item selecionado
- void insertItemAt(Object anObject, int index): insere um item em uma posição
  void removeItem(Object anObject): remove o item especificado

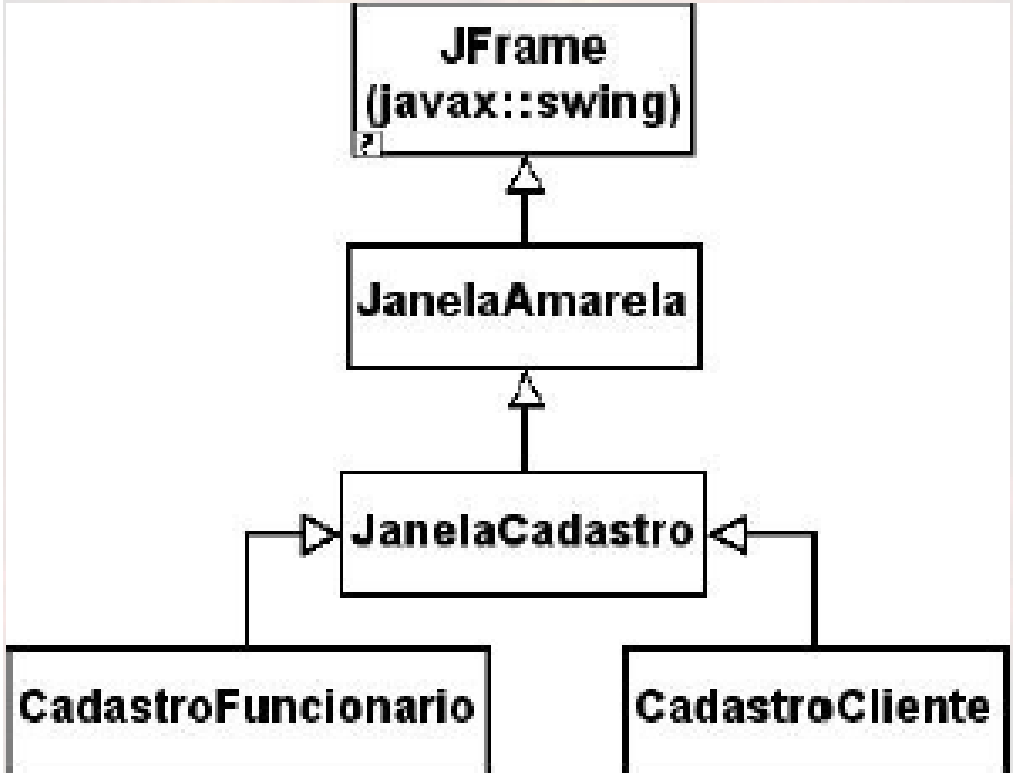
- **javax.swing.JList – Métodos**
- JList(); JList(Object[] listData) e JList(ListModel dataModel): principais construtores
- void clearSelection(): limpa a marca de seleção
- int getMinSelectionIndex() e int getMaxSelectionIndex(): retorna o mínimo e o máximo índice em uma seleção
- int getSelectedIndex() e int[] getSelectedIndices(): retorna o índice dos itens selecionados
- Object getSelectedValue() e Object[] getSelectedValues(): retorna os itens selecionados
- boolean isSelectionEmpty(): retorna true se não há nada selecionado
- void setListData(Object[] listData): constrói a lista com os valores do parâmetro

- **javax.swing.JScrollPane – Métodos**
- Fornece um container com barras de rolagens e cabeçalho vertical e horizontal
- JScrollPane(Component view): principal construtor
- void setViewportView(Component view): altera o componente visualizado
- void setViewportBorder(Border viewportBorder): adiciona uma borda

- **javax.swing.JTextArea – Métodos**
- JTextArea() e JTextArea(int rows, int columns): principais construtores
- void append(String str): adiciona texto após o final do conteúdo
- void insert(String str, int pos): insere o texto em uma posição específica
- void setLineWrap(boolean wrap): altera a quebra de linha
- void setText(String str): substitui todo o conteúdo pelo valor do parâmetro

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.

Exercícios

- 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.

Exercícios 01

- 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 layout do painel de conteúdo.
  - Deve liberar a memória utilizada quando for fechada.

Exercícios 02

- 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.