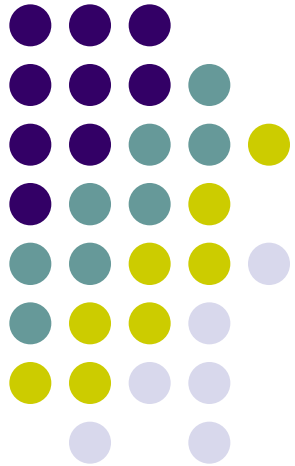
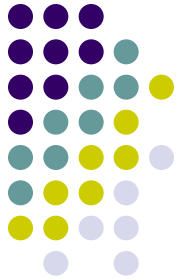
COM220

Aula: 19 Componentes Básicos de Controle – Parte 2

Prof. Laércio Baldochi

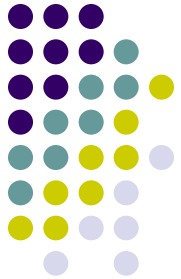


Componentes básicos de controle Parte 2



- Listas
  - JComboBox
  - JList
- Menus
  - JMenuBar
  - JPopupMenu
  - JSeparator
  - JMenuItem
  - JMenu
  - JCheckBoxMenuItem
  - JRadioButtonMenuItem
- Scrolling
  - JScrollPane

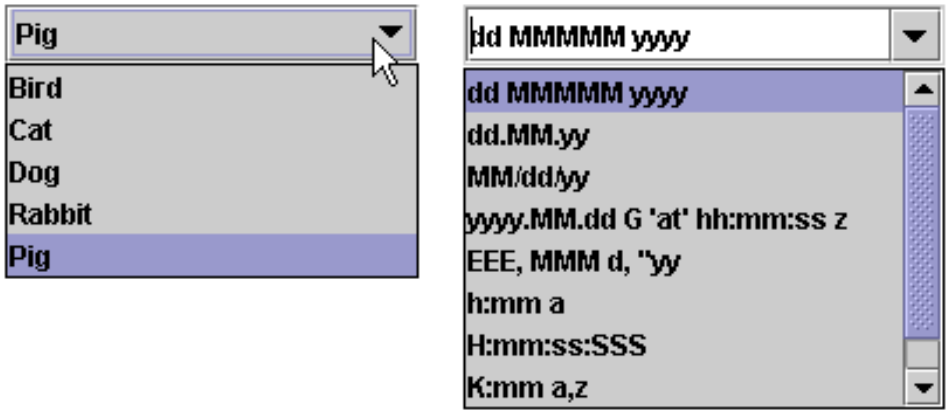
Listas



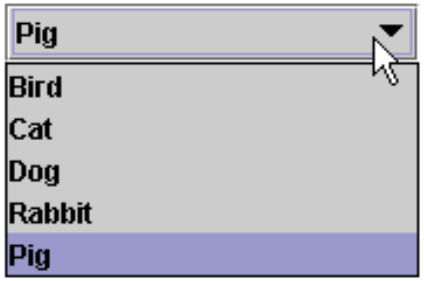
- JComboBox
  - Apenas 1 item aparece na interface
  - Demais elementos são apresentados em um menu flutuante
  - Lista opções permitindo que apenas uma seja escolhida
- JList
  - Diversos itens aparecem na interface
  - Permite a escolha de uma ou mais opções

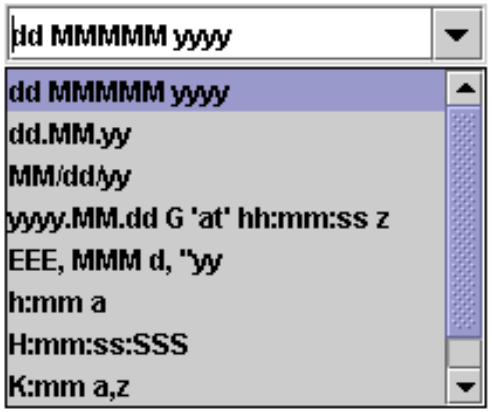
JComboBox

- Apresenta itens de 2 formas
  - Editável
  - Não editável (default)
- Eventos gerados podem ser manipulados por meio das interfaces ActionListener e ItemListener



JComboBox Exemplo

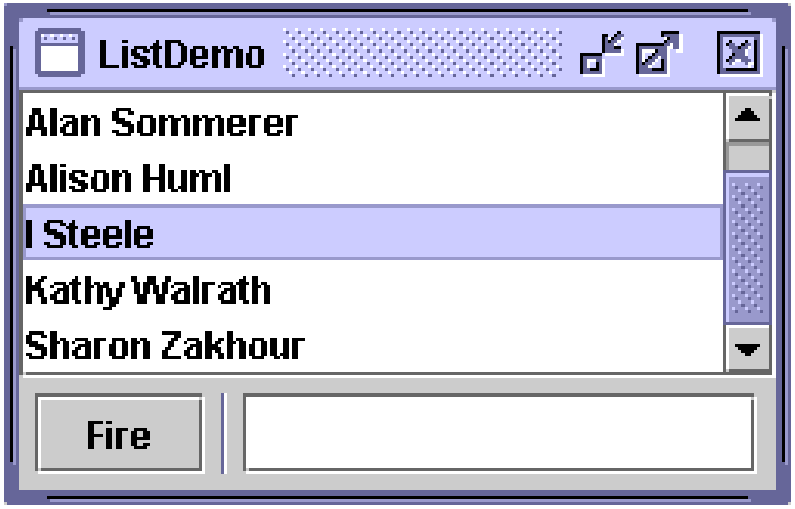





```
// Define a lista de itens a serem apresentados
String[] str1 = { "Bird", "Cat", "Dog", "Rabbit", "Pig" };
// Cria o combo box, seleciona o índice e adiciona o listener
JComboBox cb1 = new JComboBox(str);
cb1.setSelectedIndex(4);
cb1.addActionListener(this);
// Define a lista de itens
String[] str2 = { "dd MMMMM yyyy", dd.MM.yy", MM/dd/yy", ... };
// Cria o combo, torna-o editável e adiciona o listener
JComboBox cb2 = new JComboBox(str2);
cb2.setEditable(true);
cb2.addActionListener(this);
// Implementa a interface ActionListener
public void actionPerformed(ActionEvent e) {
    JComboBox c = (JComboBox) e.getSource();
    String item = c.getSelectedItem();
    System.out.println("User chose " + item);
}
```

JList

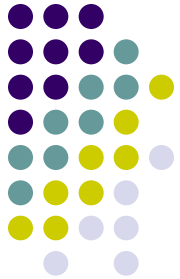
- Apresenta uma lista de itens selecionáveis que podem ser compostos de texto, imagem ou ambos
- Ocupa tamanho fixo na tela
  - Pode utilizar barra de rolagens para visualização de todos os itens



JList

- Layout de apresentação pode ser configurado por meio do método setLayoutManager(orientation)
  - JList. VERTICAL_WRAP
  - JList. HORIZONTAL_WRAP

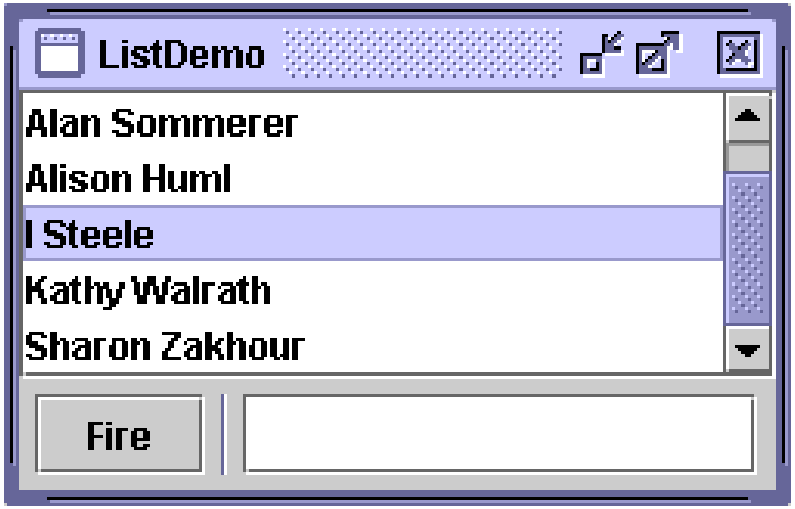
JList



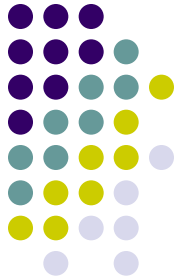
- Modo de seleção dos itens da lista também pode ser configurado
  - Método setSelectionMode(mode)
    - SINGLE_SELECTION
    - SINGLE_INTERVAL
    - MULTIPLE_INTERVAL_SELECTION
- Lista dispara eventos quando a seleção é modificada
  - ListSelectionListener
  - valueChanged()

JList
Tratando eventos

```
public void valueChanged(ListSelectionEvent e) {
    // Se o usuário não está manipulando a seleção
    if (e.getValueIsAdjusting() == false) {
        // Se não há objeto selecionado, desabilita o botão
        if (list.getSelectedIndex() == -1)
            fireButton.setEnabled(false);
        // Senão, habilita o botão
        else
            fireButton.setEnabled(true);
    }
}
```

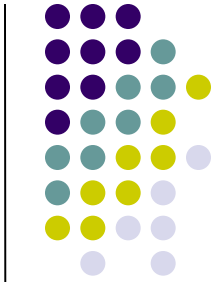


JList



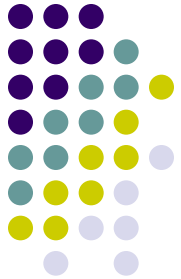
- Conteúdo de uma lista pode ser modificado dinamicamente
  - Objeto DefaultListModel
- Outra opção é criar um JList a partir de um array ou Vector
  - Lista estática

JList
Exemplo



```
// Exemplo de lista estática
String[] data = { "Arlo", "Cosmo", "Elmo", "Hugo", "Jethro", ... };
JList staticList = new JList(data);
// Exemplo de lista dinâmica
listModel = new DefaultListModel();
listModel.addElement("Alison Huml");
listModel.addElement("Kathy Walrath");
listModel.addElement("Lisa Friendly");
// Cria a lista passando o modelo como parametro
JList dinamicList = new JList(listModel);
dinamicList.addListSelectionListener(this);
...
// Implementa a interface ListSelectionListener
 public void valueChanged(ListSelectionEvent e) {
       if (e.getValueIsAdjusting() == false) {
          if (dinamicList.getSelectedIndex() != -1) {
             int index = dinamicList.getSelectedIndex();
             System.out.println(index);
             listModel.remove(index);
  } } }
```

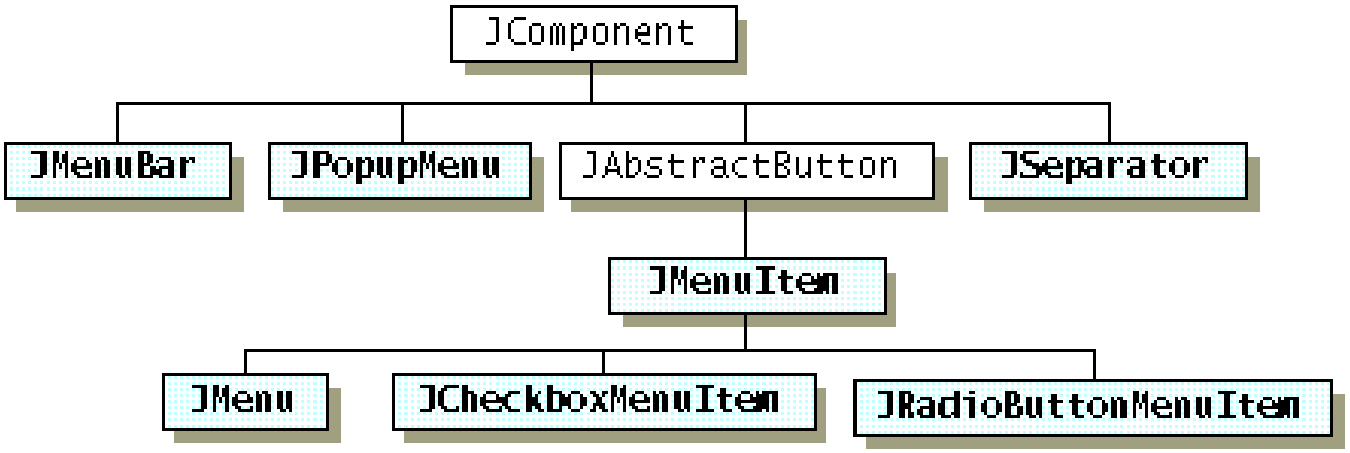
Exercício 1



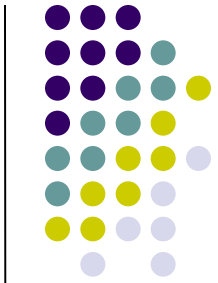
- Implemente o exemplo do slide anterior
  - JList deve ser criado contendo 10 itens
  - Toda vez que um dos itens for selecionado, ele deve ser removido do JList

Menus

- API Swing provê suporte à construção de interfaces baseadas em menus
- Todos os componentes de menu descendem da classe JAbstractButton

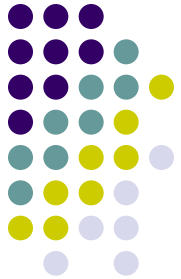


Menus



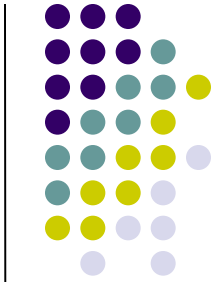
- JMenuBar
  - Representa o conceito de uma barra de menu acoplada à janela
    - Adicionado a um JFrame ou JApplet
      - setJMenuBar(menuBar)
    - Não é possível adicionar uma barra de menus a um JPanel

Menus



- JPopupMenu
  - Representa um menu que é exibido flutuando sobre a tela
  - Não é ligado a nenhuma barra de menus
  - Pode conter componentes herdados da classe JMenuItem
  - Método show(base, x, y) é usado para exibir o popup na posição (x, y)
- JSeparator
  - Implementa uma linha divisora usada para separar grupos de menus ou itens logicamente relacionados

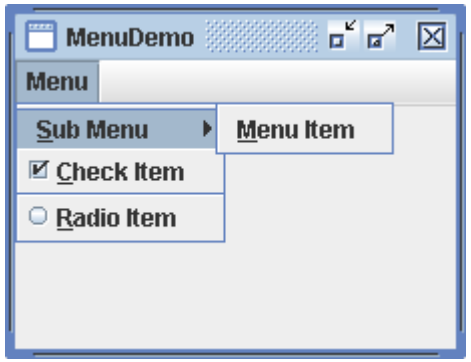
Menus



- JMenuItem
  - Representa os itens de um menu
  - Herda diretamente da classe JAbstractButton
  - Possui todas as propriedades comuns de um botão
    - Pode conter ícones
    - Responde a cliques disparando eventos
- JMenu
  - Representa menus que podem conter JMenuItems dentro
    - Possibilita a criação de sub-menus
    - Usa o método add(component) para adicionar items ao menu

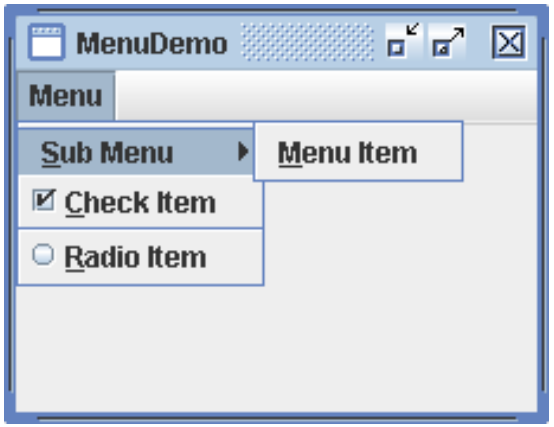
Menus

- JCheckBoxMenuItem
  - Corresponde a um item de menu que implementa a funcionalidade de um checkBox
    - Praticamente idêntico ao checkBox
- JRadioButtonMenuItem
  - Corresponde a um item de menu que implementa a funcionalidade de um radioButton
    - Utiliza o esquema de grupos para permitir que apenas um botão seja selecionado por vez
      - Praticamente idêntico ao radioButton

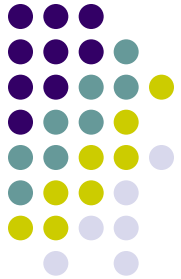


Menus
Exemplo

```
// Constrói a barra de menus
JMenuBar mb = new JMenuBar();
setJMenuBar(mb);
// Constrói um menu e o adiciona a barra
JMenu m = new JMenu("Menu", true);
mb.add(m);
// Cria um submenu e o adiciona
JMenu sm = new JMenu("Sub Menu");
sm.setMnemonic(KeyEvent.VK_S);
m.add(sm);
// Cria um item de menu e o adiciona ao submenu
JMenuItem mi = new JMenuItem("Menu Item");
mi.setMnemonic(KeyEvent.VK_M);
sm.add(mi);
// Cria um checkbox e o adiciona ao menu
JCheckBoxMenuItem cbmi = new JCheckBoxMenuItem("Check Item");
cbmi.setMnemonic(KeyEvent.VK_C);
m.add(cbmi);
// Cria um separador entre o checkbox e o radiobutton
JSeparator s = new JSeparator();
m.add(s);
// Cria um radiobutton e o adiciona ao menu
JRadioButtonMenuItem rbmi = new JRadioButtonMenuItem("Radio Item");
rbmi.setMnemonic(KeyEvent.VK_R);
m.add(rbmi);
```

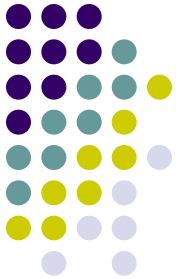


Painel de rolagem JScrollPane



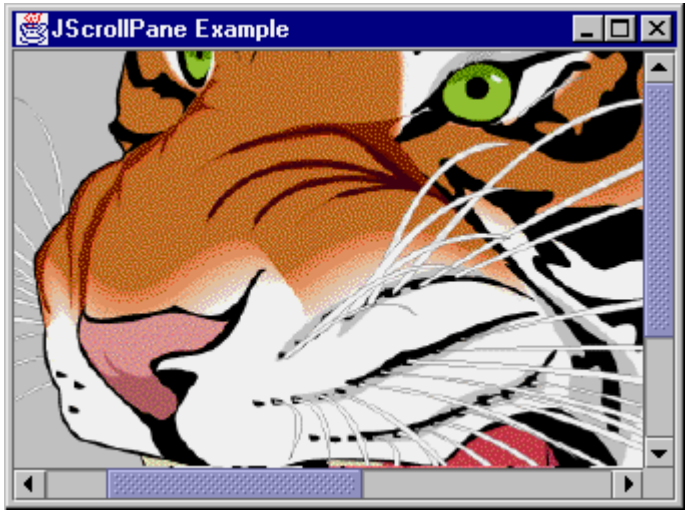
- Provê um tipo de painel que possui barras de rolagem embutidas
- Barras de rolagem podem ser usadas quando o componente é maior que a área disponível
  - Imagem ou tabela
- Ou para componentes que podem crescer dinamicamente
  - Listas e TextAreas

JScrollPane



- Consiste de barras de rolagem, representadas pela classe JScrollBar e de um viewPort, que corresponde à janela sobre a qual é visualizado o componente interno
- Componente pode ser adicionado ao JScrollPane diretamente através do construtor
- Ou ao viewPort
  - getViewPort().add(component)

JScrollPane Exemplo

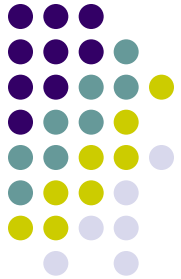



```
// Carrega a imagem e cria o label
Icon bigTiger = new ImageIcon("BigTiger.gif");
JLabel tigerLabel = new JLabel(bigTiger)

// Cria o scrollpane com o label
JScrollPane scrollPane = new JScrollPane(tigerLabel,   // componente
     JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,    // scroll vertical
    JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS); // scrollhorizontal

// Adiciona o painel
getContentPane().add(scrollPane, BorderLayout.CENTER);
```

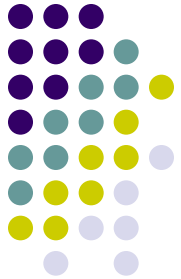
ScrollPane
Políticas de rolagem



- Barras de rolagem podem aparecer automatica-mente quando o componente se torna maior que o viewport, ou seguir políticas determinadas pelas seguintes constantes:
  - VERTICAL_SCROLLBAR_AS_NEEDED
  - VERTICAL_SCROLLBAR_ALWAYS
  - VERTICAL_SCROLLBAR_NEVER
  - HORIZONTAL_SCROLLBAR_AS_NEEDED
  - HORIZONTAL_SCROLLBAR_ALWAYS
  - HORIZONTAL_SCROLLBAR_NEVER

Exercícios

Entrega: 18/06



2) Crie um JComboBox que adiciona à lista de opções o texto entrado pelo usuário em um campo JTextField, quando este teclar ENTER. Dica: tente utilizar a interface ActionListener.
3) Implemente duas listas com a classe JList, cada uma contendo cinco itens. Sempre que um item for selecionado, ele deve ser movido para a outra lista. As listas devem exibir barras de rolagem quando necessário.
4) Crie um menu *pop-up* contendo JMenuItems com ícones e texto. Esse menu deve ser exibido sempre que o usuário clicar com o botão direito do *mouse* sobre uma área da janela.
5) Crie um painel contendo dois objetos: um JSlider e um JProgressBar. Quando o usuário mover o *slider*, o componente JProgressBar deve ser preenchido ou esvaziado na mesma proporção.