

Desenvolvimento Desktop

Aula 05 - Bot o de Op o, Grupo de Bot es e Caixa de Combina o

Apresentação

Nesta aula, você irá conhecer mais três componentes do Java: o **Botão de opção (JRadioButton)**, que é utilizado para selecionar ou desselecionar uma opção; o **Grupo de botões (ButtonGroup)**, que é um objeto para se criar um grupo de botões de opção no qual um e somente um do grupo poderá ser selecionado; e a **Caixa de combinação (JComboBox)**, que é utilizada para criar uma lista de itens da qual o usuário poderá selecionar apenas uma opção.



Vídeo 01 - Apresentação

Objetivos

Ao final desta aula, você será capaz de:

- Descrever o funcionamento dos componentes Java: Botão de opção, Grupo de botões e Caixa de combinação;
- Criar aplicações robustas utilizando esses componentes.

O Componente Botão de Opção (JRadioButton)

O componente **Botão de opção**, também conhecido como **JRadioButton**, é utilizado para você selecionar uma e somente uma opção entre várias oferecidas em um determinado grupo, ou seja, um grupo desses botões não permite mais de uma seleção por vez. Esse componente deverá ser utilizado em situações nas quais existam pelo menos duas opções de escolha. Você deve ter observado o uso desse componente em formulários de muitos sites na internet para escolha de opções do tipo:

- Sim ou Não
- Masculino ou Feminino
- Faixa de salários
- Faixa de idades
- E assim por diante, quando uma única seleção é permitida

Selecionando-se qualquer um dos botões de opção em um grupo de opções através do mouse ou da barra de espaços, o que estiver selecionado anteriormente será desselecionado automaticamente.

Interação do Usuário com o Componente JRadioButton

Um **Botão de opção** pode estar habilitado ou desabilitado. Quando desabilitado, ele não recebe entrada nem do mouse nem do teclado. Quando o usuário clica ou usa a tecla **Tab** em um grupo de componentes **Botão de opção**, apenas o **Botão de opção** selecionado recebe o foco, ou seja, ele fica com uma borda em sua volta indicando que ele é o botão da vez. Quando um **Botão de opção** está com o foco, o usuário poderá utilizar as seguintes teclas para controlá-lo:

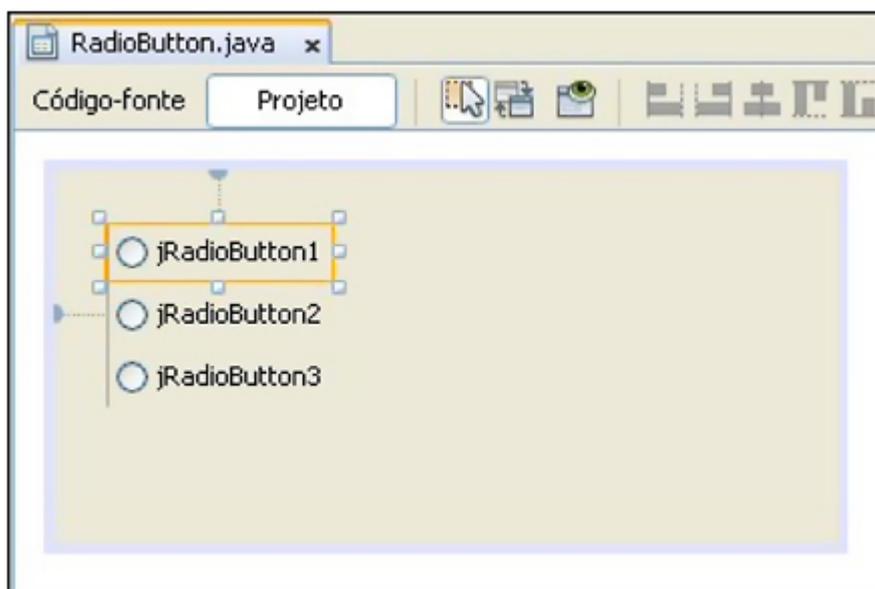
Tecla	Descrição
Barra de espaço	Seleciona e desseleciona o componente.
Shift + Tab	Move o foco para o Botão de opção anterior do mesmo grupo.
Tab	Move o foco para o próximo Botão de opção do mesmo grupo.

Como Utilizá-lo em uma Aplicação

Para utilizar um componente **Botão de opção** na sua aplicação, faça o seguinte:

1. Execute o NetBeans e crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome que você achar mais conveniente.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame**.
4. A partir da **Paleta de componentes**, na seção **Controles Swing**, clique e arraste três componentes **Botão de opção** para a área do projeto.
5. A aparência desse componente durante o projeto é mostrado na **Figura 1**:

Figura 01 - Componente Botão de opção durante o projeto



6. Se você executar essa aplicação agora (tecla **F6**), o componente será apresentado conforme mostra a **Figura 2**:

Figura 02 - Componentes Botão de opção em tempo de execução



Se você clicar em cada um desses botões, deverá notar que todos os três poderão ser marcados ou selecionados, o que contradiz o que foi dito anteriormente, ou seja, esse componente só deveria permitir selecionar apenas um e somente um do grupo. Além disso, ainda é permitido desmarcar se clicarmos no mesmo botão novamente. E você poderia se perguntar: “E por que está sendo permitido selecionar todos os três botões?” Você está certo, mas vamos com calma.

Isso acontece porque cada um desses botões está isolado, eles não fazem parte do mesmo grupo. Para que um deles seja selecionado e outro desselecionado automaticamente, teremos que agrupá-los, como se fosse uma equipe. E então, como fazer isso? Bom, mais adiante você saberá como fazer isso, mas antes veremos algumas propriedades desse componente.

As Propriedades do Componente Botão de Opção

A seguir, algumas propriedades que podemos alterar para melhorar o desempenho desse componente.

- **buttonGroup** – essa propriedade é utilizada para agrupar vários **Botões de opção** e garantir que apenas um componente do grupo poderá ser selecionado. Quando um componente **Grupo de botões** é inserido em uma aplicação, um novo item chamado **buttonGroup1** é acrescentado automaticamente ao combo box dessa propriedade. Se um segundo componente **Grupo de botões**

for inserido, o combo box será acrescido de um novo item **buttonGroup2**, e assim por diante.

- **selected** – se o check box dessa propriedade for selecionado, significa que o componente **Botão de opção** que estiver sendo configurado será mostrado já marcado quando a aplicação for executada. Por padrão essa propriedade é desmarcada.
- **text** – essa propriedade é utilizada para identificar o componente com um rótulo.

Na seção **Outras propriedades** você poderá também configurar as seguintes propriedades:

- **border** – utilize essa propriedade para criar uma borda em volta do componente. Não se esqueça de marcar o check box da propriedade **border Painted** para que a borda escolhida apareça no componente.
- **enabled** – se esse check box estiver marcado significa que o componente ficará habilitado para seleção, caso contrário, ficará desabilitado. Por padrão esse check box já é marcado.
- **horizontalTextPosition** – selecione no combo box a posição do rótulo do componente em relação ao círculo de marcação (esquerda, direita, centro etc.). A posição padrão é à direita.



Vídeo 02 - JRadioButton

O Componente Grupo de Botões (ButtonGroup)

Para resolver o problema dos **Botões de opção** discutidos no início desta aula, primeiramente devemos acomodar os botões (no mínimo dois) dentro de um painel, um recipiente (contêiner) que tem por finalidade separar esses botões de opção dos outros grupos de botões de opção (se houver na mesma aplicação).

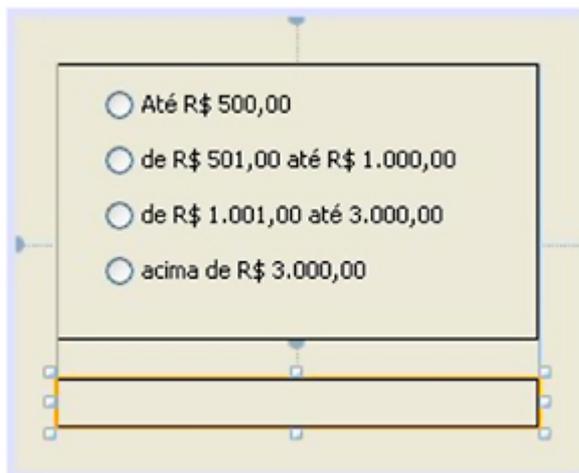
Depois disso, entra em campo um outro componente chamado **Grupo de botões (ButtonGroup)**, cuja finalidade é exatamente agrupar e dar um nome a esse grupo de botões de opção e gerenciar o estado de cada um deles que fizerem parte do mesmo grupo, garantido que um único botão do grupo possa ser selecionado de cada vez. Geralmente esses componentes dependem uns dos outros.

Para você entender direitinho como esses componentes funcionam juntos, criaremos uma pequena aplicação:

1. Crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome de (sugestão) **ComponenteBotaoDeOpcao**, em uma pasta de sua escolha.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame** com o nome de **BotaoDeOpcao**.
4. A partir da **Paleta de componentes**, na seção **Contêineres Swing**, clique e arraste um componente **Painel (JPanel)** para a área do projeto.
5. Em seguida, arraste quatro componentes **Botão de opção** para dentro do painel.
6. Altere os rótulos (propriedade **text**) desses componentes para:
 - Até R\$ 500,00
 - de R\$ 501,00 até 1.000,00
 - de R\$ 1.001,00 até 3.000,00
 - acima de R\$ 3.000,00
7. Altere (usando a alça da direita) também a largura de cada componente de forma que o conteúdo de cada um apareça completamente.
8. Agora arraste e alinhe todos os componentes à esquerda.
9. Altere a largura e a altura do **Painel** de forma que comporte todos os componentes. Crie também uma borda para que ele seja visualizado.
10. Arraste um componente **Rótulo (JLabel)** e insira-o logo abaixo do **Painel**. Crie uma borda para que ele seja visualizado e apague o seu conteúdo (propriedade **text**).

11. Se você tiver seguido todos os passos direitinho, seu projeto deverá estar agora semelhante ao mostrado na **Figura 3**:

Figura 03 - Componentes Botão de opção inseridos em um contêiner Painel



Mesmo que você executasse essa aplicação agora, todos os componentes **Botão de opção** ainda poderiam ser selecionados e não apenas um. Para que apenas um botão de opção seja selecionado, precisamos dar um nome ao grupo, como se fosse uma seleção de futebol, na qual existe apenas um goleiro ou um capitão. Para nomear esse grupo, utilizamos o componente **Grupo de botões (ButtonGroup)**. Para isso:

12. Arraste um componente **Grupo de botões** para dentro do painel na área de projeto.
13. Ao fazer isso, no combo box da propriedade **buttonGroup** dos botões de opção será acrescentado um novo item chamado **buttonGroup1**. Veja na **Figura 4**.

Figura 04 - Propriedade buttonGroup com o novo item adicionado

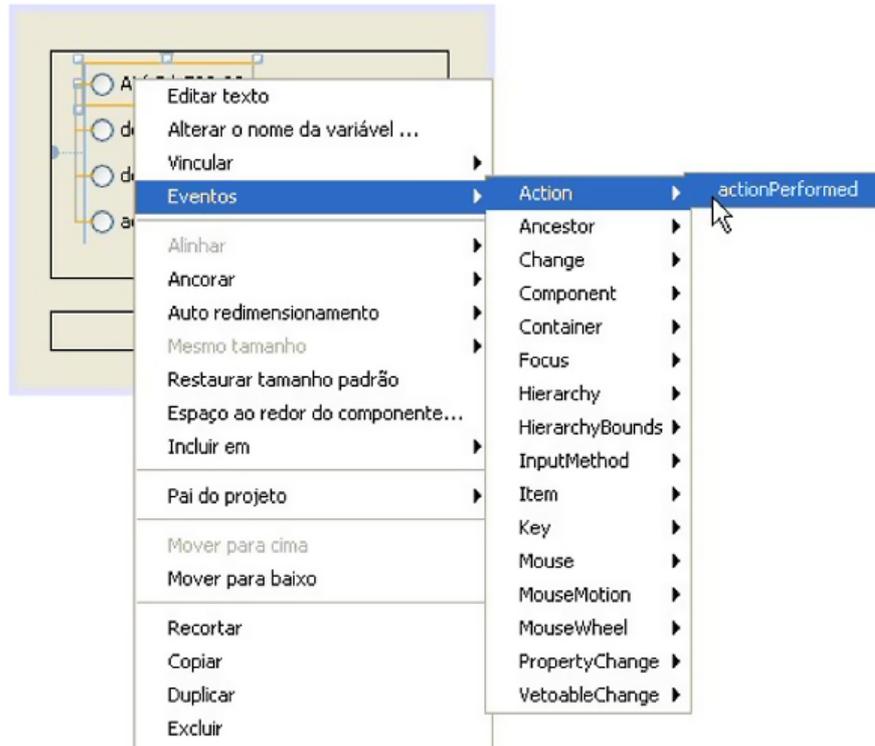


14. O que precisamos fazer agora é associar cada um dos **Botões de opção** à propriedade **buttonGroup1**. Para isso, selecione o primeiro botão de opção e clique no item chamado **buttonGroup1** da propriedade **buttonGroup**. Observe que os outros botões ficarão automaticamente associados à mesma propriedade. Portanto, não será necessário você fazer esse processo para os outros três botões.
15. Feito isso, execute a aplicação (**F6**) e verifique se os botões de opção estão funcionando de acordo com o previsto, ou seja, cada um de cada vez. Com certeza deverão estar.

Mas nossa aplicação ainda não está completa. Para finalizar, precisamos dar uma funcionalidade a ela, ou seja, criar alguma ação para os botões. No nosso caso, queremos que quando cada um dos botões de opção for clicado, apareça uma mensagem no componente **Rótulo**. Para isso:

1. Clique com o botão direito no primeiro componente **Botão de opção**.
2. No menu apresentado, selecione: **Eventos -> Action -> actionPerformed**.
Veja a **Figura 5**.

Figura 05 - Criando um evento para o Botão de opção



3. Ao fazer isso, será criado automaticamente o código mostrado na **Figura 6**, para o primeiro botão.

Figura 06 - Código gerado automaticamente para o evento do primeiro Botão de opção

```
117  
118 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
119     // TODO add your handling code here:  
120 }  
121
```

4. Posicione o cursor no final da linha 119 e tecle **ENTER** para criar uma nova linha.
5. Digite o seguinte código nessa linha:

jLabel1.setText("Você ganha muito pouco.");

OBS: O nome **jLabel1** é o padrão estabelecido pelo NetBeans para a criação de cada JLabel da aplicação, em ordem crescente, como JLabel1, JLabel2 e assim por diante. Portanto, verifique se o JLabel1 está realmente associado ao rótulo que você criou. Caso contrário, indique o nome correto.

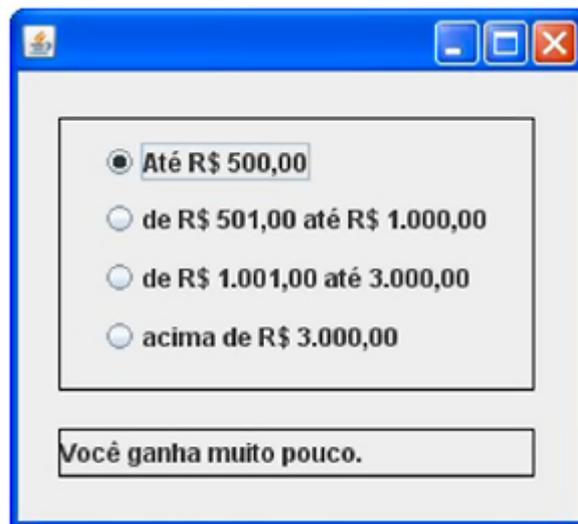
6. Agora seu código deverá ficar assim:

Figura 07 - A mensagem entre parênteses desse código será mostrada no componente Rótulo (jLabel1) quando o primeiro Botão de opção for clicado

```
117  
118 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
119     // TODO add your handling code here:  
120     jLabel1.setText("Você ganha muito pouco.");  
121 }  
122
```

7. Execute a aplicação e confira o resultado como mostrado na **Figura 8**:

Figura 08 - Aplicação sendo executada apenas com o evento do primeiro 'Botão de opção'



Se sua aplicação estiver funcionando de acordo com o que foi planejado, e esperamos que esteja, agora você terá que repetir esse mesmo processo para cada um dos três botões restantes, alterando apenas o código da mensagem para o seguinte:

Botão 2:

```
jLabel1.setText("Você está precisando melhorar.");
```

Botão 3:

```
jLabel1.setText("Você já ganha razoável.");
```

Botão 4:

```
jLabel1.setText("Está melhorando.");
```



Vídeo 03 - ButtonGroup

Atividade 01

1. Qual a propriedade de um componente **Botão de opção** utilizada para associar vários desses botões em um grupo?
2. Qual o número mínimo de botões de opção que poderá ser utilizado em uma situação de escolha?
3. O que acontece se o check box da propriedade **selected** de um botão de opção estiver marcado?
4. Cite três exemplos utilizando apenas dois botões de opção para cada situação em uma mesma aplicação.
5. O que acontece se três botões de opção não forem associados a um **ButtonGroup** quando a aplicação for executada?

O Componente Caixa de Combinação (JComboBox)

Esse componente é semelhante ao componente **Lista** (que será discutido posteriormente em outra aula). A diferença é que a lista de itens de um componente **Caixa de combinação** fica escondida, restrita apenas a uma linha, ocupando dessa forma menos espaço do que outros componentes, como o **Lista (JList)** e o **Área de texto (JTextArea)**, que você irá conhecer no decorrer deste curso. Esse componente é utilizado para armazenar informações em forma de uma lista, sendo cada item em uma linha, e apenas uma seleção será permitida.

Você poderá utilizar um componente **Caixa de combinação** em qualquer formulário ou aplicação, desde que a escolha seja de um único item da lista. Veja algumas sugestões:

- os estados de um país;
- faixas de idades;
- faixas de salários;
- gêneros de filmes;
- categorias de livros, entre outras.

Essas são situações em que apenas uma seleção é válida.

Um componente **Caixa de combinação** pode ser estático ou editável. Um componente **Caixa de combinação** estático permite ao usuário fazer uma seleção de uma lista apresentada, e uma **Caixa de combinação** editável é mais flexível, permite ao usuário digitar um texto diretamente em um campo de texto no topo da lista. Se a lista alcançar o rodapé do documento, ela será aberta para cima e não para baixo. Esse componente é composto de três subcomponentes:

- um componente Botão (**JButton**);
- um componente Campo de texto (**JTextField**);
- um componente Lista (**JList**).

Interação do Usuário com o Componente Caixa de Combinação

Quando uma seleção é feita na lista, o item selecionado é copiado para a camada de texto do topo do componente. Não importa se a seleção foi feita com o mouse ou com o teclado.

Um componente **Caixa de combinação** recebe foco se você clicar na caixa de texto ou no botão. Quando ele estiver com o foco e editável, você poderá utilizar as seguintes teclas para manipular o texto:

Tecla	Descrição
Tecla de seta para baixo	A seleção é movida um item para baixo.
End	O ponto de inserção é movido para o final da caixa de texto.
Enter	Se a lista estiver aberta, o foco retorna para a caixa de texto e a lista fecha.
Escape	Se a lista estiver aberta, o foco retorna para a caixa de texto e a lista fecha.
Home	O ponto de inserção é movido para o início da caixa de texto.
Page Down	A seleção é movida uma página para baixo.
Page Up	A seleção é movida uma página para cima.
Tab	Move o foco para o próximo objeto (se houver).
Shift + End	Seleciona o texto a partir do ponto de inserção até o final.
Shift + Home	Seleciona o texto a partir do ponto de inserção até o início.
Shift + Tab	Move o foco para o objeto anterior (se houver).
Tecla de seta para cima	A seleção é movida um item para cima.

Quando um componente **Caixa de combinação** possui o foco e é estático, as teclas alfanuméricas movem a seleção para cima e para baixo da lista para o próximo item que possuir o primeiro caractere digitado. Você também poderá utilizar as seguintes teclas para controlar um componente **Caixa de combinação** estático:

Tecla	Descrição
Tecla de seta para baixo	Abre a lista e permite selecionar um item.
Tecla de seta para cima	Permite selecionar um item para cima se a lista estiver aberta.
End	A seleção é movida para o último item da lista.
Escape	Fecha a lista.
Enter	Fecha a lista com o último item selecionado, se ela estiver aberta.
Home	A seleção é movida para o primeiro item da lista.
Page Down	A seleção é movida uma página para baixo.
Page Up	A seleção é movida uma página para cima.
Shift + Tab	Move o foco para o objeto anterior (se houver).
Tab	Move o foco para o próximo objeto (se houver).

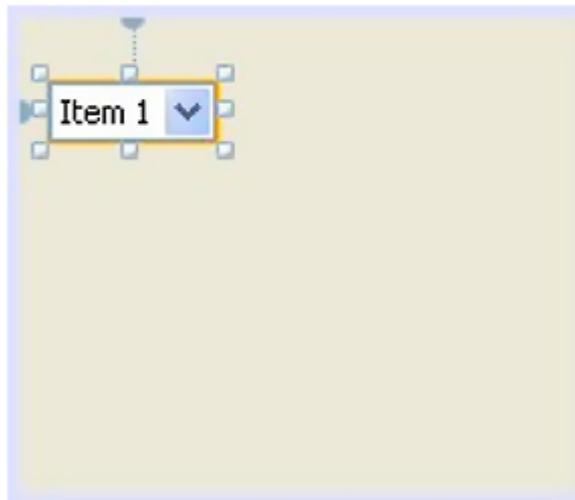
Quando a lista de uma **Caixa de combinação** possui o foco, as teclas alfanuméricas movem a seleção para cima ou para baixo da lista para o próximo item que possuir o primeiro caractere. Por exemplo, para selecionar o item **Cadastro** basta pressionar a tecla **C**, sem necessidade de percorrer toda a lista.

Como Utilizá-lo em uma Aplicação

Para utilizar um componente **Caixa de combinação** na sua aplicação, faça o seguinte:

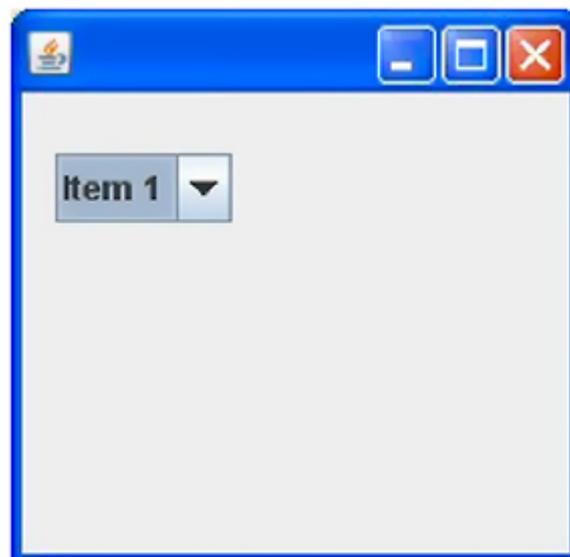
1. Execute o NetBeans e crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome que você achar mais conveniente.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame**.
4. A partir da **Paleta de componentes**, na seção **Controles Swing**, clique e arraste um componente **Caixa de combinação** para a área do projeto.
5. Veja na **Figura 9** a aparência desse componente durante o projeto.

Figura 09 - Componente Caixa de combinação durante o projeto



6. Se você executar essa aplicação agora (**F6**), o componente será apresentado conforme mostra a **Figura 10**:

Figura 10 - Componente Caixa de combinação em tempo de execução



Se você clicar na seta à sua direita ou pressionar a **seta para baixo** você notará que ele já está preenchido com quatro itens. É apenas uma sugestão do NetBeans para mostrar como o componente funciona. Antes de criarmos alguma aplicação prática utilizando esse componente, vejamos algumas de suas propriedades.

As Propriedades do Componente Caixa de Combinação

A seguir, você conhecerá algumas propriedades que podemos alterar para melhorar o desempenho desse componente.

- **editable** – se o check box dessa propriedade for selecionado, significa que os itens do componente **Caixa de combinação** poderão ser editados, caso contrário, só poderão ser visualizados e selecionados. O valor desse check box é desmarcado por padrão.
- **maximumRowCount** – essa propriedade informa o número máximo de itens que poderão ser visualizados quando a lista do componente for aberta. Se o número de itens da lista ultrapassar esse valor, uma barra de rolagem vertical será criada automaticamente. O valor padrão é 8.
- **model** – essa propriedade é utilizada para preencher o componente com os itens necessários. Veremos como fazer isso no exercício prático mais adiante.

Na seção **Outras propriedades** você poderá também configurar as propriedades:

- **border** – essa propriedade permite criar uma borda em volta do componente.
- **enabled** – se esse check box estiver marcado significa que o componente ficará habilitado para seleção, caso contrário, ficará desabilitado. Por padrão, esse check box já é marcado.

Agora veremos como utilizar esse componente em uma aplicação prática juntamente com um componente **Rótulo** para mostrar o resultado quando um item da lista for selecionado:

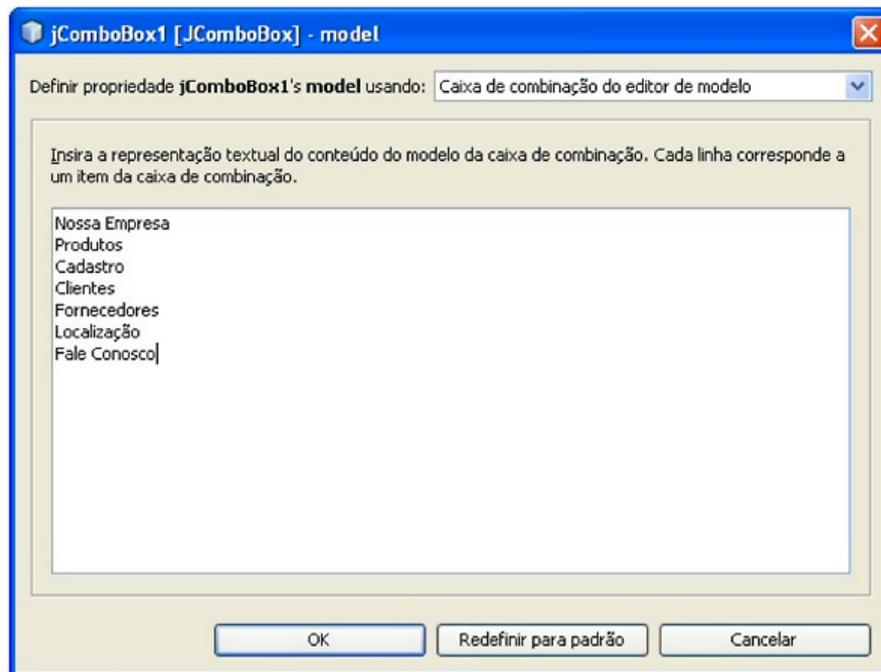
1. Crie um novo projeto (**Arquivo -> Novo projeto**).

2. Salve seu projeto com o nome de: '**ComponenteCaixaDeCombinacao**' em uma pasta de sua escolha.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame** com o nome de **CaixaDeCombinacao**.
4. A partir da **Paleta de componentes**, na seção **Controles Swing**, clique e arraste um componente **Caixa de combinação (JComboBox)** para a área do projeto.
5. Altere (usando a alça da direita) a largura desse componente para um tamanho um pouco maior do que o padrão.
6. Altere sua propriedade **maximumRowCount** para 4.
7. Para preencher a lista da caixa de combinação, clique na sua propriedade **model**. Na janela apresentada, apague os itens existentes e crie a seguinte lista de itens:

Nossa Empresa
Produtos
Cadastro
Clientes
Fornecedores
Localização
Fale Conosco

8. Confira na **Figura 11** como ficará a sua lista:

Figura 11 - Preenchimento dos itens da lista de um componente Caixa de combinação



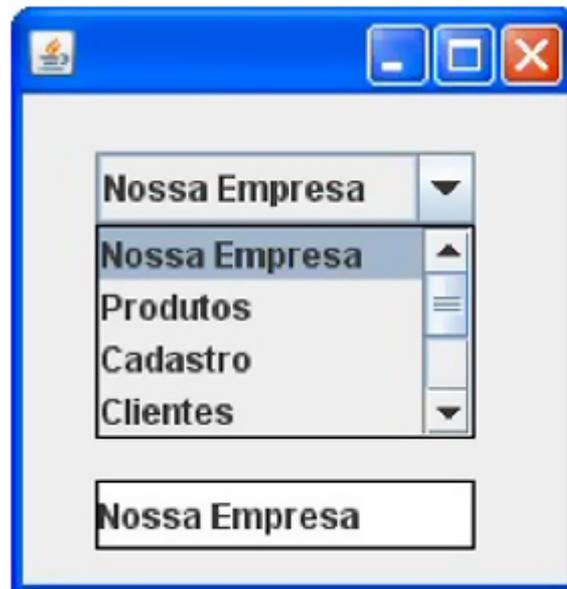
9. Clique no botão **OK** para confirmar.
10. Agora arraste um componente **Rótulo** e insira-o bem abaixo da caixa de combinação. Crie uma borda para que ele possa ser visualizado e apague o seu conteúdo (propriedade **text**). Altere também o seu tamanho, de forma que ele fique com a mesma largura da **Caixa de combinação**, e a cor de fundo para branco.
11. Marque o check box da propriedade **opaque** para que a cor de fundo possa ser mostrada.
12. Se você tiver seguido todas as instruções até aqui, seu projeto deverá estar agora semelhante ao mostrado na **Figura 12**:

4. ... crie uma linha em branco após a linha 84 e insira as seguintes linhas de código:

```
String a = (String)jComboBox1.getSelectedItem();  
jLabel1.setText(a);
```

5. Execute a aplicação e confira o resultado com o mostrado na **Figura 15**:

Figura 15 - Componente Caixa de combinação em execução



Vídeo 04 - JComboBox

Atividade 02

1. Cite três situações em que um componente **Caixa de combinação** possa ser utilizado em uma aplicação.
2. Qual a tecla utilizada para fechar um componente **Caixa de combinação**, caso ele esteja com a lista aberta?
3. Quais as teclas utilizadas para posicionar o cursor no início e no final da caixa de texto de um componente **Caixa de combinação** quando ele estiver com o foco e editável?
4. Quais são os subcomponentes que compõem uma **Caixa de combinação**?
5. O que acontece quando os itens de uma lista de uma **Caixa de combinação** ultrapassa o limite de itens visíveis estabelecido na propriedade **maximumRowCount**?

Resumo

Infelizmente, nossa aula termina por aqui. Nela você conheceu mais alguns componentes do Java e algumas de suas propriedades. Auxiliados por outros componentes já abordados em aulas anteriores, criamos algumas aplicações simples com eles. Procure criar aplicações utilizando todos os componentes já estudados até agora em conjunto, criando situações diferentes.

Autoavaliação

1. Crie uma aplicação utilizando dois grupos de **Botões de opção** com três botões cada um, de forma que os rótulos dos botões do segundo grupo sejam preenchidos com os mesmos rótulos dos botões do primeiro grupo quando eles forem clicados.
2. Desenvolva uma aplicação com dois grupos de **Botões de opção** com dois botões cada. Nos rótulos dos botões do primeiro grupo um nome de homem e um de mulher. No segundo grupo os rótulos: **Masculino e Feminino**. A finalidade dessa aplicação é a seguinte: quando um botão do primeiro grupo for clicado, o botão do segundo grupo referente ao sexo correspondente seja automaticamente marcado e vice-versa. Como sugestão utilize a linha de código a seguir para selecionar cada um dos botões:

```
jRadioButton3.setSelected(true);
```

Não se esqueça de alterar o número dos respectivos botões.

3. Desenvolva uma aplicação utilizando um componente Caixa de combinação e um componente Rótulo. A caixa de combinação deverá ter seis itens com nomes de carros em ordem alfabética, de forma que quando um item for selecionado, apareça no rótulo o preço de cada carro.

Referências

DEITEL, Harvey M.; DEITEL, Paul J. **Java: como programar**. 6. ed. São Paulo: Prentice-Hall, 2005.

JAVA FREE ORG. Disponível em: <<http://javafree.uol.com.br/artigos/>>. Acesso em: 6 abr. 2012.

THE JAVA TUTORIALS. **How to Use Buttons, Check Boxes, and Radio Buttons**. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/button.html#radiobutton>>. Acesso em: 6 abr. 2012.

_____. **How to Use the ButtonGroup Component**. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/buttongroup.html>>. Acesso em: 6 abr. 2012.

_____. **How to Use Combo Boxes**. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/combobox.html>>. Acesso em: 6 abr. 2012.

SIERRA, Kathy; BATES, Bert. **Use a Cabeça! Java**. 2. ed. São Paulo: Alta Books, 2007.