

Desenvolvimento Desktop

Aula 06 -  rea de Texto, Lista e Caixa de Sele o

Apresentação

Nesta aula, você conhecerá três novos componentes do Java: **área de texto (JTextArea)**, que é utilizado para criar textos de múltiplas linhas; **Lista (JList)**, que é utilizado para exibir uma série de itens como se fosse uma lista de compras; e a **Caixa de Seleção (JCheckBox)**, utilizado para marcar ou desmarcar uma opção.



Vídeo 01 - Apresentação

Objetivos

Ao final desta aula, você será capaz de:

- Descrever o que é uma área de texto, uma lista e uma caixa de seleção.
- Aplicar esses componentes em suas aplicações e personalizar suas propriedades.

O Componente Área de Texto (JTextArea)

O componente **Área de Texto** é nada mais nada menos do que uma caixa de texto com maiores recursos do que uma caixa de texto convencional (**JTextField**), já visto na **Aula 4**. Você poderá usá-lo para mostrar texto e também editar e receber entrada de dados, caso o checkbox da propriedade **editable** esteja marcado. Esse componente é utilizado com muita frequência em web sites, principalmente em formulários de cadastramento de currículos, em fóruns e opiniões sobre algum produto, entre outros usos, onde o usuário precisa escrever com mais liberdade e com um maior limite de caracteres, e praticamente sem regras pré-estabelecidas de preenchimento (dependendo do caso).

O texto tanto pode ser digitado, como importado de um arquivo externo. Ele poderá mostrar ou receber múltiplas linhas, como, também, quebrar longas linhas, caso a propriedade **lineWrap** esteja marcada. Se o texto exceder as bordas da caixa de texto, barras de rolamento horizontal ou vertical surgirão automaticamente, dependendo do caso.

Utilize esse componente somente quando necessitar de um campo de texto com mais de uma linha. Para textos com apenas uma linha, utilize o componente **caixa de texto (JTextField)**, que é mais apropriado.

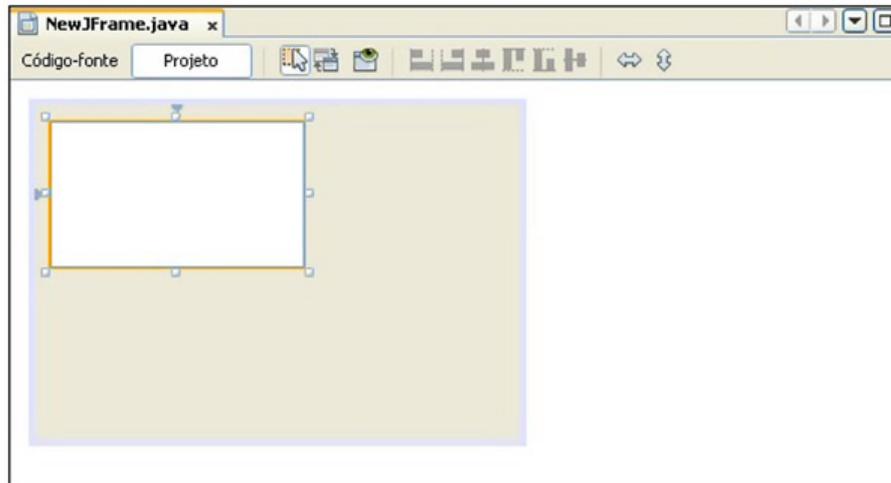
Como Utilizá-lo em uma Aplicação

Vejamos como utilizar esse componente em uma aplicação:

1. Execute o NetBeans e crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome que você considerar 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 **Área de texto** para a área do projeto.

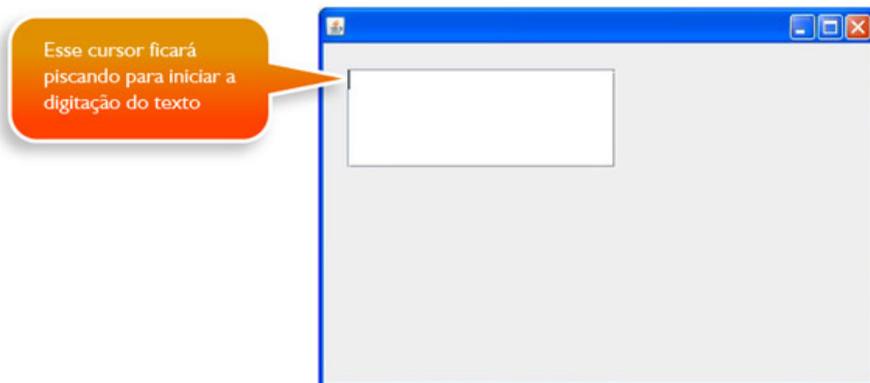
5. isualmente esse componente se apresenta como um retângulo branco com uma borda azul, conforme mostra a **Figura 1** abaixo:

Figura 01 - Componente Área de texto durante o projeto



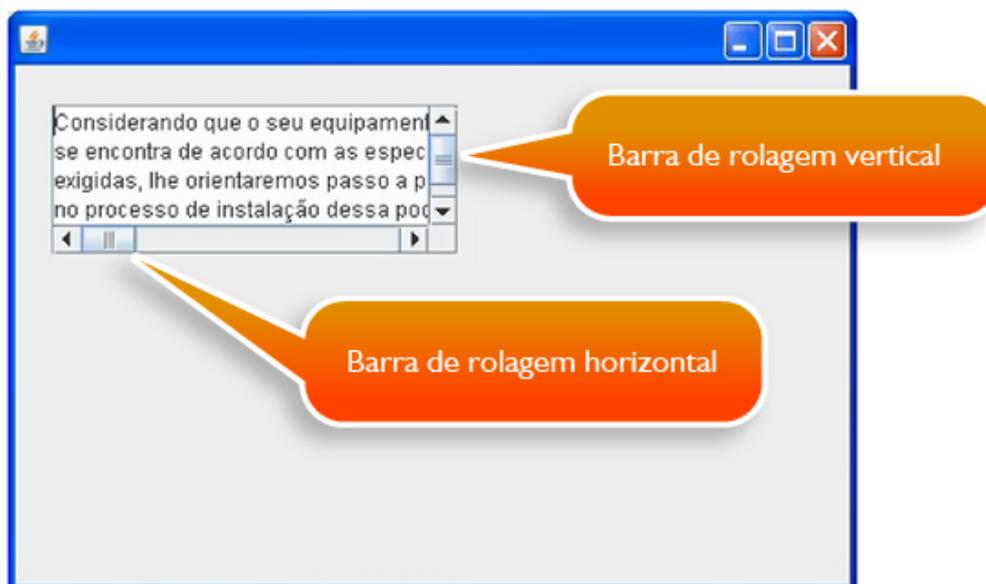
6. Para alterar as suas dimensões (largura e altura), você poderá utilizar as alças de dimensionamento mostradas na figura. Essas dimensões também poderão ser alteradas diretamente nas suas propriedades.
7. Se você executar essa aplicação agora (tecla **F6**), o componente será apresentado como um retângulo branco com o cursor piscando no início da caixa de texto, conforme mostra a **Figura 2** a seguir:

Figura 02 - Componente Área de texto após a execução da aplicação



8. Nessa oportunidade, você poderá digitar um texto qualquer ou copiá-lo de uma fonte externa, aparentemente de qualquer tamanho. Se, por acaso, o conteúdo da caixa de texto ultrapassar os limites da área estabelecida, serão criadas, automaticamente, duas barras de rolagem, uma horizontal e uma vertical, conforme mostra a **Figura 3** a seguir:

Figura 03 - Componente Área de texto com barras de rolagem criadas automaticamente



A seguir, discutiremos algumas propriedades desse componente de forma que possamos melhorar a sua aparência e alguns atributos.

As Propriedades do Componente Área de Texto

Algumas propriedades que podemos alterar para melhorar o desempenho desse componente são as seguintes:

- **columns** – essa propriedade especifica o número de colunas da caixa de texto. O valor padrão é 20, mas você poderá alterar esse valor para outro qualquer, desde que a largura do componente não ultrapasse os limites do seu formulário comprometendo o *layout*.
- **editable** – se esse checkbox estiver marcado, significa que o componente poderá ser preenchido e editado, caso contrário, ficará desabilitado para inclusão e edição de texto. Por padrão, esse checkbox já é marcado.
- **lineWrap** – se esse checkbox estiver marcado, significa que as linhas serão quebradas automaticamente quando o texto atingir a margem direita da caixa de texto, caso contrário, será considerado como uma única linha. O valor padrão é **false**, ou seja, desmarcado.
- **rows** – essa propriedade especifica o número de linhas visíveis da caixa de texto. O valor padrão é 5, mas você poderá alterar esse

valor para outro qualquer, desde que a altura do componente não ultrapasse os limites do seu formulário, comprometendo o *layout*.

- **tabSize** – essa propriedade especifica a quantidade de caracteres em branco quando um novo parágrafo é criado usando-se a tecla **TAB**. O valor padrão é 8, mas você poderá alterar esse valor para outro qualquer.
- **text** – essa propriedade é utilizada para você especificar um texto padrão para o componente quando a aplicação for executada, ao invés de ficar em branco. Por exemplo, você poderia informar ao usuário que nessa caixa de texto ele deverá inserir um resumo do seu currículo, adicionando, nessa propriedade, a mensagem: **“Informe aqui um resumo do seu currículo.”** Você poderá utilizar essa propriedade também para inserir um texto antes da execução da aplicação, ou até mesmo alterar o conteúdo do componente durante a execução através de código.

Na seção **Outras propriedades**, você poderá também configurar a propriedade:

- **enabled** – se esse checkbox estiver marcado, significa que o componente ficará habilitado para inclusão e edição de texto, dependendo da configuração da propriedade **editable**. Caso contrário, ficará desabilitado e não será permitida a inclusão de texto nem a edição. Essa propriedade anula por completo a propriedade **editable**. Por padrão, esse checkbox já é marcado.
- **opaque** – essa propriedade determina a opacidade do componente, ou seja, se esse checkbox estiver marcado, a cor de fundo do componente será opaco, cuja cor padrão é branco, caso contrário, o fundo do componente ficará transparente.

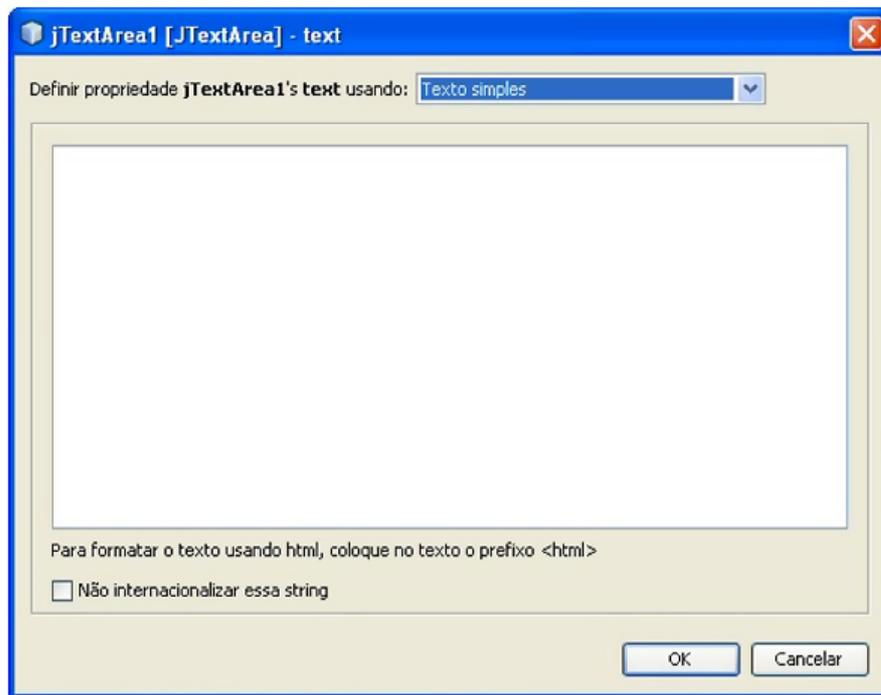
Vejamos agora uma aplicação onde você poderá inserir um texto antes de executá-la:

1. Crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome (sugestão): **ComponenteAreaDeTexto**, em uma pasta de sua escolha.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame** com o nome de: **AreaeTexto**.

4. A partir da paleta de componentes, na seção **Controles Swing**, clique e arraste um componente **Área de texto** para a área do projeto.
5. Com o componente já selecionado, marque a propriedade **lineWrap** para que o texto seja quebrado quando atingir a margem direita.

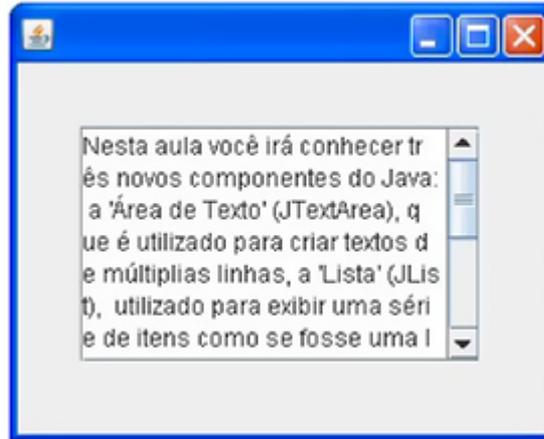
Clique no pequeno botão à direita da propriedade **text**. Será mostrada a seguinte janela:

Figura 04 - Janela para inserção do conteúdo de um componente Área de texto utilizando a sua propriedade text



6. Na área em branco, digite um texto qualquer ou copie de alguma fonte externa. Em seguida, clique em **OK** para concluir.
7. De volta ao projeto, execute a aplicação (**F6**) e confira o resultado com o apresentado na **Figura 5** a seguir:

Figura 05 - Componente Área de texto com conteúdo preenchido antecipadamente



Mesmo que o texto tenha sido inserido antecipadamente, isso não impede de você alterá-lo ou apagá-lo, desde que a propriedade **editable** esteja habilitada.



Vídeo 02 - JTextArea

Atividade 01

1. Que propriedade do componente **Área de texto** deverá ser utilizada para exibir uma mensagem padrão a título de informação para o usuário?
2. Qual a utilidade da propriedade **lineWrap**?
3. Se o checkbox da propriedade **editable** estiver desmarcado, o que acontece?
4. Que propriedade devemos utilizar para inserirmos um texto em um componente **Área de texto** antes de executá-lo?
5. O que acontece se um texto qualquer (digitado ou colado) ultrapassar a área disponível em um componente Área de texto?

O Componente Lista (JList)

Esse componente é encontrado, com muita frequência, em quase todas as aplicações que possuem algum tipo de formulário. Ele poderá ser utilizado para o preenchimento de faixas de datas, lista de produtos, lista dos meses do ano, lista de profissões, entre outras finalidades. Nesses casos, o usuário seleciona um ou mais itens da lista, alternados ou não, dependendo do caso, e a partir daí poderá ocorrer uma ação em resposta a essa seleção, ou quando outro componente for acionado, como, por exemplo, um botão.

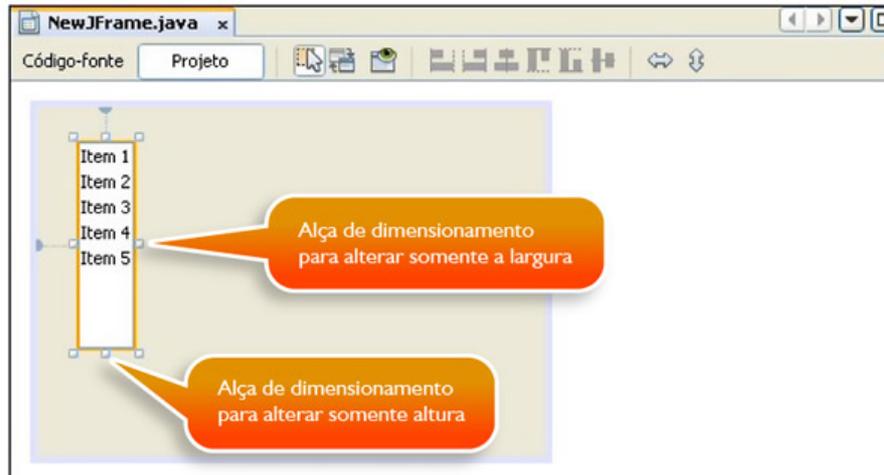
Cada item da lista possui um valor correspondente, chamado de index (ou índice), começando com zero (para o primeiro item da lista). Quando um item é adicionado ou excluído da lista, utilizando-se suas propriedades ou métodos, o índice de cada elemento da lista é automaticamente alterado.

Como Utilizá-lo em uma Aplicação

Vejamos um exemplo prático de como utilizar esse componente:

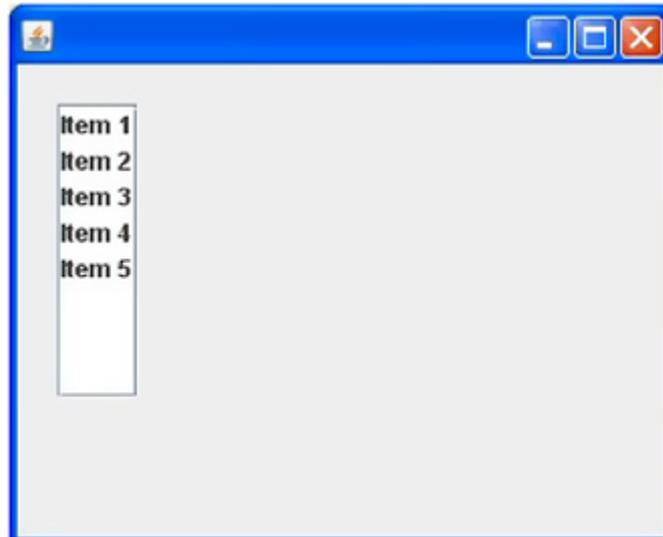
1. Execute o NetBeans e crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome que você considerar mais conveniente.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **Lista** para a área do projeto.
5. Visualmente, esse componente se apresenta como um retângulo branco na vertical preenchido com cinco itens como exemplo, conforme mostra a **Figura 6**:

Figura 06 - Componente Lista durante o projeto



6. Para alterar as suas dimensões (largura e altura), você poderá utilizar as alças de dimensionamento mostradas na figura ou, então, utilizar as suas propriedades.
7. Se você executar essa aplicação agora (tecla **F6**), o componente será apresentado como um retângulo branco na vertical com os cinco itens sugeridos pelo próprio NetBeans, conforme mostra a **Figura 7**:

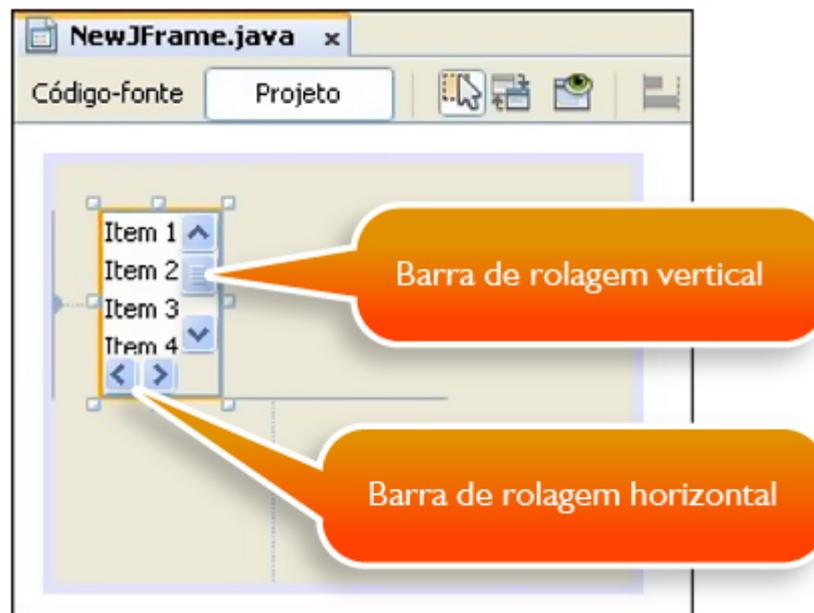
Figura 07 - Componente Lista após a execução da aplicação



8. Agora você poderá selecionar um ou mais itens, inclusive até mesmo todas as opções da lista com o auxílio da tecla Ctrl (**Ctrl + clique**) para seleções alternadas, ou **Shit + clique** para seleções em sequência. Da mesma forma que o componente anterior, se, por acaso, o conteúdo da caixa da lista ultrapassar os limites da área estabelecida, serão criadas,

automaticamente, barras de rolagem, uma horizontal e/ou uma vertical, conforme o caso. Veja a **Figura 8** a seguir:

Figura 08 - Componente Lista com barras de rolagem criadas automaticamente



A seguir, veremos como personalizar algumas propriedades desse componente de forma que possamos preenchê-lo corretamente.

As Propriedades do Componente Lista

Você verá agora algumas propriedades desse componente que poderão ser alteradas para uso em uma aplicação:

- **model** – essa propriedade é utilizada para preencher os itens do componente **Lista**, sendo um item por cada linha. No próximo tópico, veremos como preencher manualmente uma lista na criação de uma aplicação prática.
- **selectionMode** – essa propriedade permite-lhe selecionar uma das seguintes situações para os itens da lista:
 - **SINGLE** – selecionando essa opção, o usuário só poderá selecionar apenas um item de cada vez.

- **SINGLE_INTERVAL** – essa opção permite ao usuário selecionar vários itens por vez, mas somente em sequência, com o auxílio da tecla **Shift** ou **Ctrl**. Não permite múltiplas seleções alternadas.
- **MULTIPLE_INTERVAL** – essa opção permite ao usuário selecionar vários itens por vez, em sequência ou alternados, com o auxílio da tecla **Shift** ou **Ctrl**.

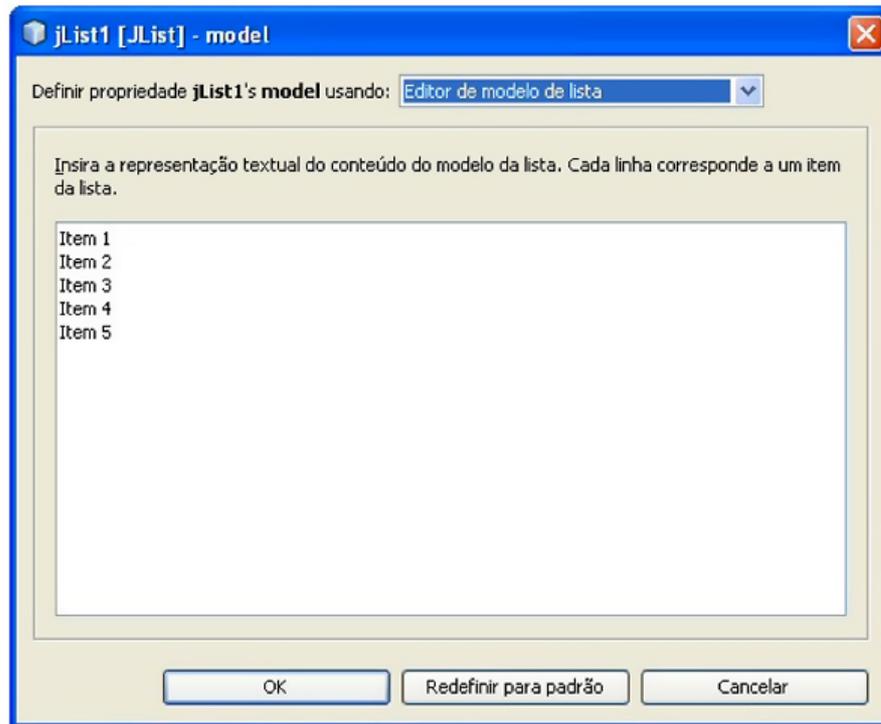
Na seção **Outras propriedades** você poderá também configurar a propriedade:

- **enabled** – se esse checkbox estiver marcado, significa que os itens da lista poderão ser selecionados, caso contrário, ficarão desabilitados.

Veamos agora uma aplicação prática usando esse componente e um componente **Label** para a saída das informações:

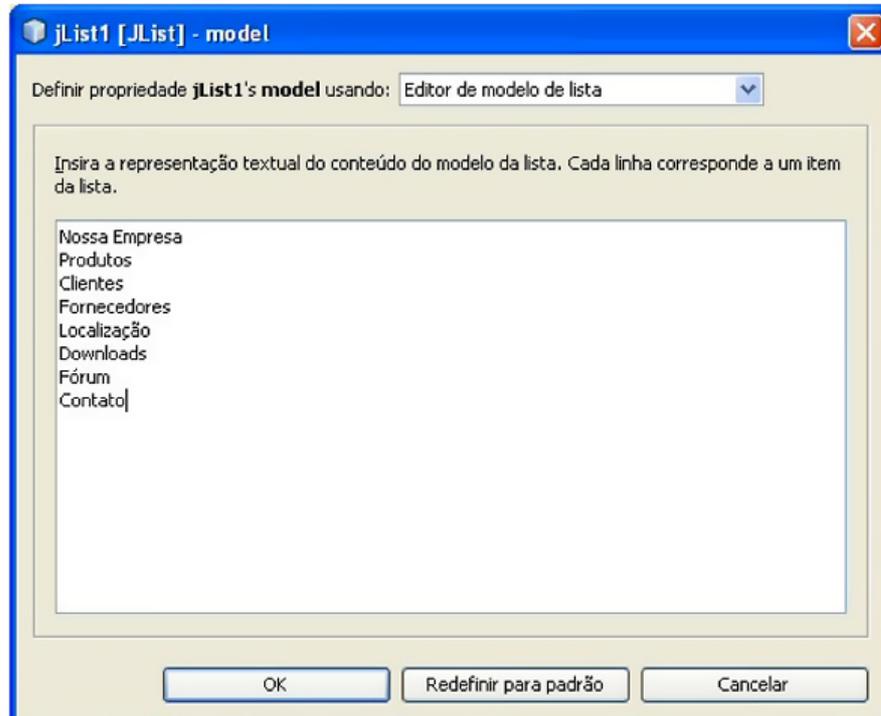
1. Crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome de: **ComponenteLista**, em uma pasta de sua escolha.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame** com o nome de: **Lista**.
4. Clique e arraste um componente **Lista** para a área do projeto.
5. Com o componente já selecionado, clique no pequeno botão à direita da propriedade **model**. Será mostrada a seguinte janela:

Figura 09 - Janela para inserção dos itens da Lista



6. Substitua os itens dessa lista pelos mostrados na **Figura 10**:

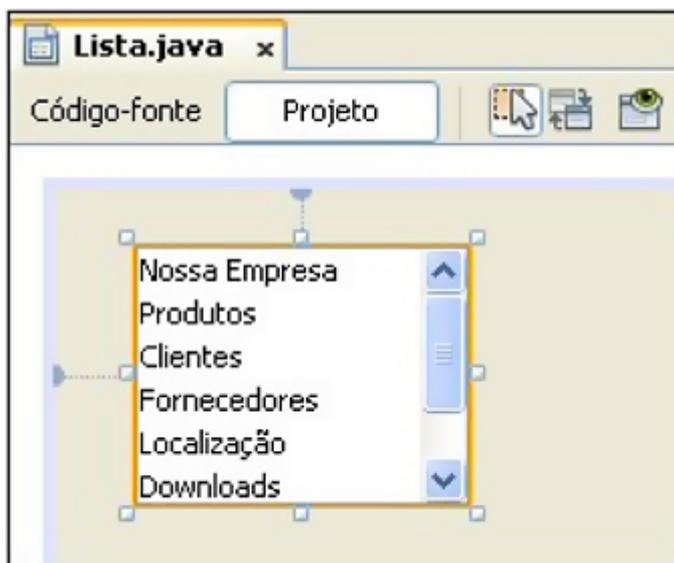
Figura 10 - Janela como os novos itens da Lista



7. Clique em **OK** para concluir.

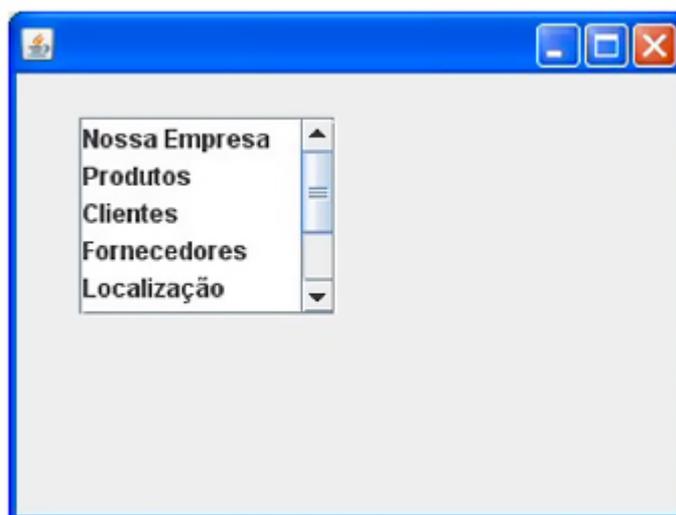
- De volta ao projeto, utilize as alças do componente para redimensionar a largura e a altura, de forma que o componente **Lista** fique semelhante ao apresentado na **Figura 11**:

Figura 11 - Componente Lista com os novos itens durante o projeto



- Altere a propriedade **selectionMode** para **SINGLE**, para que seja permitida apenas uma seleção por vez.
- Se você executar **F6** na sua aplicação agora, ela será apresentada da seguinte forma:

Figura 12 - Componente Lista em execução com os novos itens



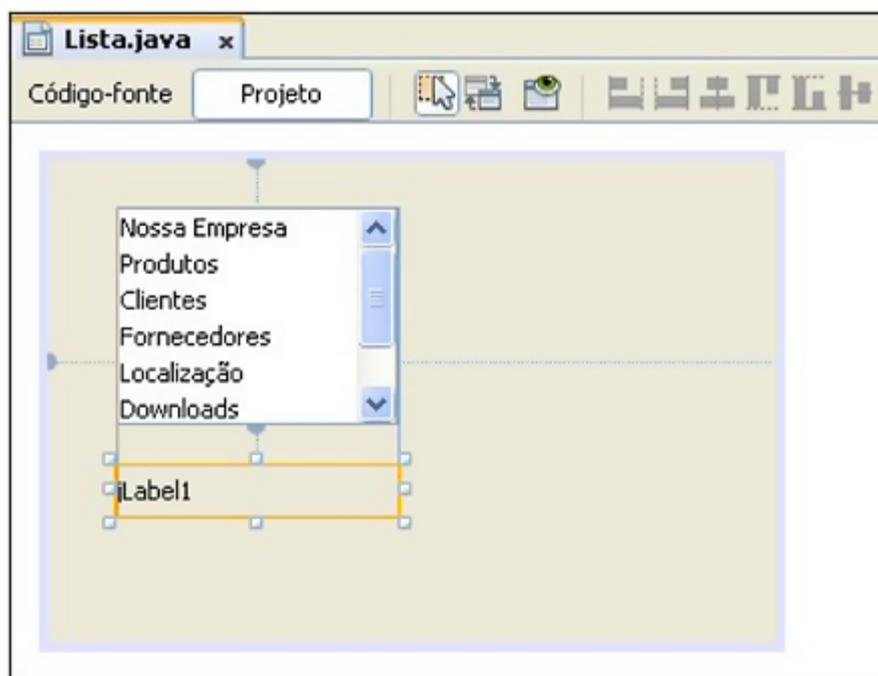
Observe que foi criada uma barra de rolagem vertical no componente, tendo em vista que o conteúdo da lista ultrapassa a sua altura. Note também que só é possível selecionar apenas um item da Lista, visto que configuramos a propriedade

selectionMode para **SINGLE**.

Somente para você ter uma ideia de como funciona uma **Lista** na prática, adicionaremos um componente **Rótulo (JLabel)** na aplicação para mostrar o item que for selecionado da lista. Para isso:

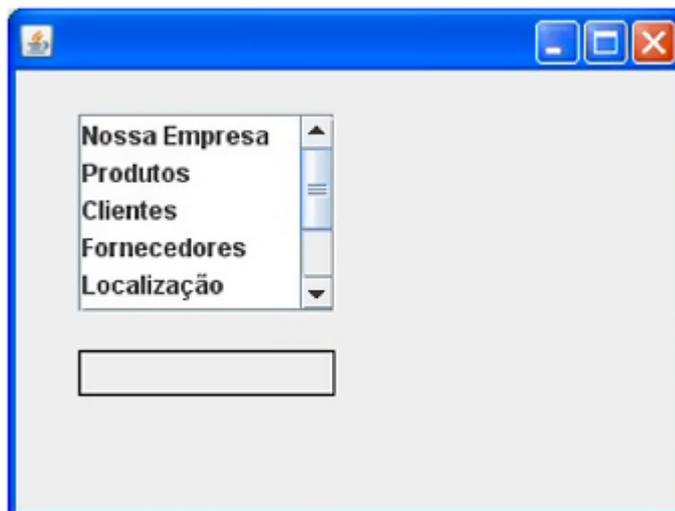
11. Arraste um componente **Rótulo** para uma posição qualquer da aplicação. Veja uma sugestão na **Figura 13** abaixo:

Figura 13 - Componente Rótulo (JLabel) adicionado à aplicação



12. Selecione a propriedade **text** do componente **Rótulo** e apague-a.
13. Para que o componente possa ser visualizado quando a aplicação for executada, utilize a propriedade **border** para criar uma borda.
14. Feito isso, sua aplicação deverá ficar semelhante à da **Figura 14** apresentada a seguir:

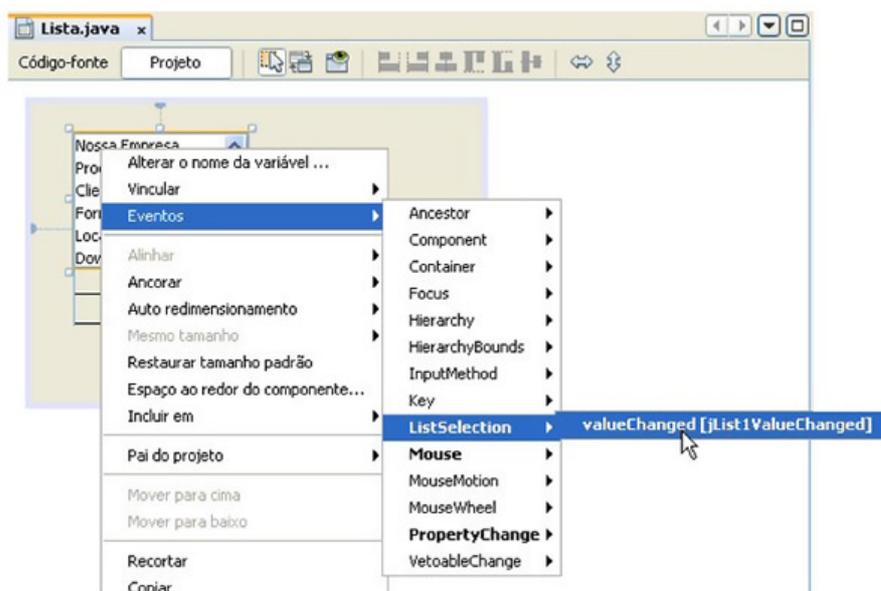
Figura 14 - Aplicação em execução com o componente Rótulo visível



15. Agora criaremos uma ação para o componente **Lista**, de forma que quando um item for selecionado ele seja mostrado no componente **Rótulo**, da seguinte forma:
16. Clique no componente **Lista** com o botão direito e selecione o seguinte no menu apresentado:

Eventos -> ListSelection -> valueChanged [jList1ValueChanged]. Veja **Figura 15**:

Figura 15 - Criando um evento para o componente Lista



17. Feito isso, será gerado automaticamente o código apresentado na **Figura 16**, que poderá ser visto na seção **Código-fonte** do projeto:

Figura 16 - Código gerado para o evento do componente Lista

```
94  
95 private void jList1ValueChanged(javax.swing.event.ListSelectionEvent evt) {  
96     // TODO add your handling code here:  
97 }  
98
```

Por enquanto, esse código não tem nenhuma funcionalidade. Precisamos adicionar algumas linhas de código para que ele possa executar o que queremos. Para isso, faça o seguinte:

1. Posicione o cursor no final da linha 96 e tecle **Enter** para criar uma linha em branco.
2. A seguir digite as seguintes linhas:

```
String i = (String)jList1.getSelectedValue();  
jLabel1.setText(i);
```

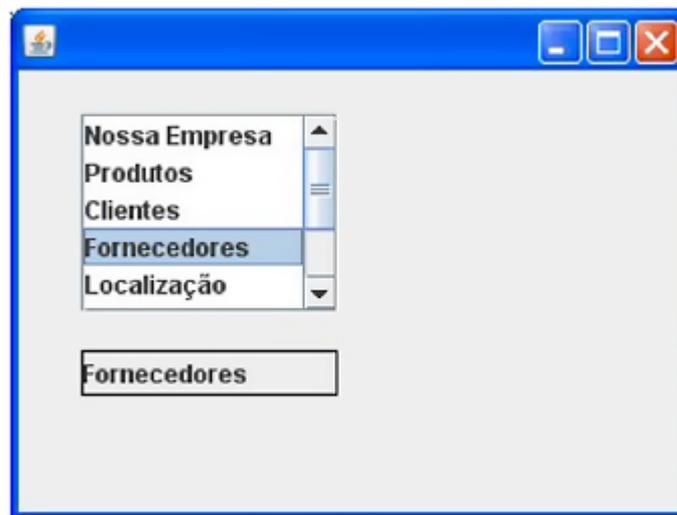
3. Seu código completo deverá ficar de acordo com o mostrado na **Figura 17**:

Figura 17 - Código completo para o evento jList1ValueChanged

```
94  
95 private void jList1ValueChanged(javax.swing.event.ListSelectionEvent evt) {  
96     // TODO add your handling code here:  
97     String i = (String)jList1.getSelectedValue();  
98     jLabel1.setText(i);  
99 }  
100
```

No momento, você não precisa se preocupar com o que está escrito nesse código. Os eventos e métodos dos componentes serão estudados em detalhes em aulas posteriores aos componentes. A finalidade desse evento é mostrar, no componente **Rótulo (jLabel1)**, o nome do item clicado no componente **Lista** após a aplicação ser executada. Para verificar se o evento está funcionando corretamente, execute a aplicação **F6** e confira o resultado, como o mostrado na **Figura 18**:

Figura 18 - Componente Lista em execução juntamente com o componente Rótulo para exibição do resultado



Vídeo 03 - JList

Atividade 02

1. Que propriedade(s) devemos utilizar para preencher os itens de um componente **Lista**?
2. escreva, sucintamente, os modos de seleção do componente **Lista**.
3. Se o checkbox da propriedade **enabled** estiver desmarcado, o que acontece?
4. Que tecla devemos utilizar, em conjunto com o clique do mouse, para selecionarmos os itens de uma lista de forma alternada, ou seja, não sequencial?

O Componente Caixa de Seleção (JCheckBox)

O componente **Caixa de seleção**, ou **CheckBox**, é um pequeno quadrado que pode ser marcado ou desmarcado. Quando marcado, um sinal de checagem aparece na caixinha. Um texto poderá ser adicionado a esse componente, e poderá ser colocado à sua direita, esquerda ou no centro. O mais comum é à direita.

Um **CheckBox** pode estar habilitado ou desabilitado em uma aplicação, dependendo da situação. Quando um **CheckBox** está habilitado e o usuário clica em seu rótulo, o **CheckBox** recebe o foco de entrada e mostra sua aparência pressionada. A seleção de um **CheckBox** também pode ser feita utilizando-se a barra de espaço, para o botão que estiver com o foco no momento.

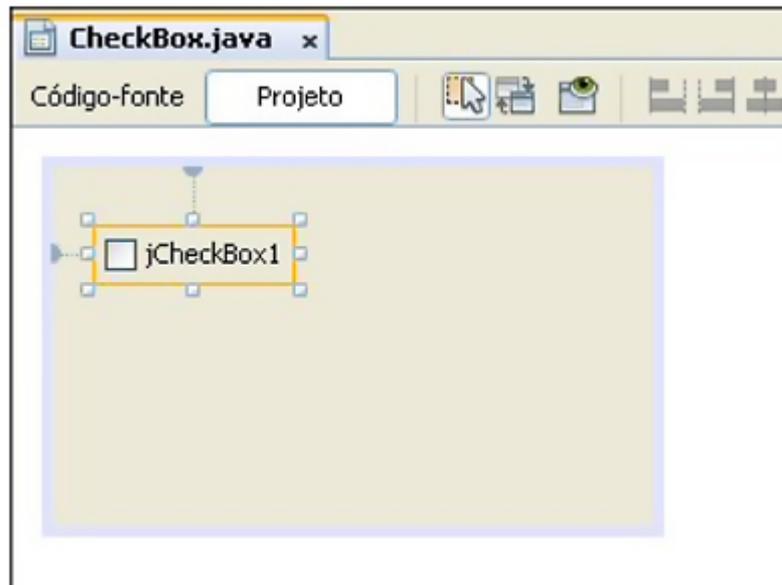
Esse componente é exatamente o oposto do **RadioButton**, ou seja, é permitido selecionar tantos componentes **CheckBox** quantos forem necessários, e não apenas uma opção. Ele é muito utilizado onde mais de uma situação é satisfeita à questão em pauta.

Como Utilizá-lo em uma Aplicação

Vejamos como utilizar esse componente em uma aplicação:

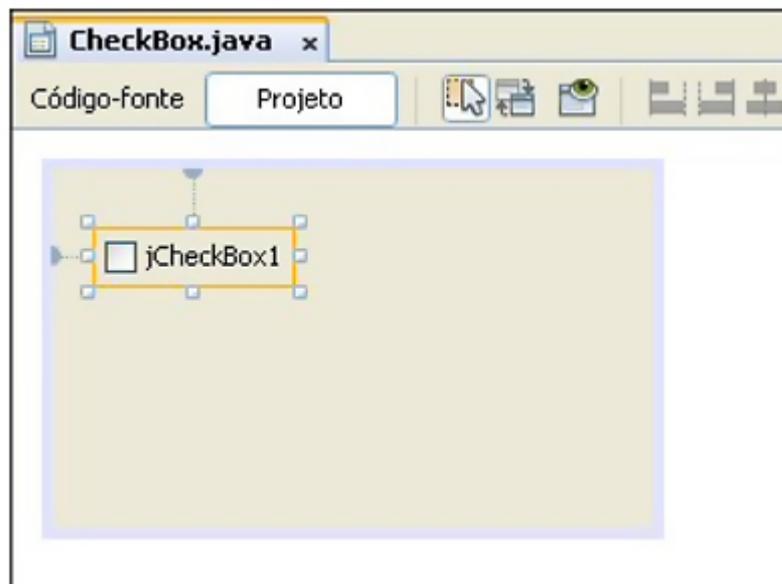
1. Execute o NetBeans e crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome que você achar mais conveniente.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **Caixa de Seleção** para a área do projeto.
5. Visualmente, esse componente se apresenta como um retângulo branco e um rótulo à sua direita, conforme mostra a **Figura 19**:

Figura 19 - Componente Caixa de seleção durante o projeto



6. Para alterar as suas dimensões (largura e altura), você poderá utilizar as alças de dimensionamento mostradas na figura ou, então, utilizar as suas propriedades.
7. Se você executar essa aplicação agora (tecla **F6**), o componente será apresentado com um quadrado e seu respectivo rótulo à direita, conforme mostra a **Figura 20**:

Figura 20 - Componente Caixa de seleção após a execução da aplicação



8. Se você clicar no pequeno quadrado ou pressionar a barra de espaço, aparecerá um pequeno sinal de checagem informando que a opção está

selecionada. Se você repetir os mesmos comandos, o quadrado ficará em branco, indicando que a opção está desselecionada.

Veremos a seguir algumas propriedades desse componente, de forma que possamos configurá-lo corretamente.

As Propriedades do Componente Caixa de Seleção

Você verá agora algumas propriedades que poderão ser alteradas para uso em uma aplicação:

- **selected** – se você quiser que alguns dos seus checkboxes apareçam já marcados em sua aplicação, utilize essa propriedade para fazer isso. Clique em cada componente e marque essa opção.
- **text** – essa propriedade é utilizada para alterar o rótulo do checkbox. Essa propriedade também pode ser alterada diretamente no componente. Basta clicar no componente, pressionar a tecla **F2** e alterar o rótulo.

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

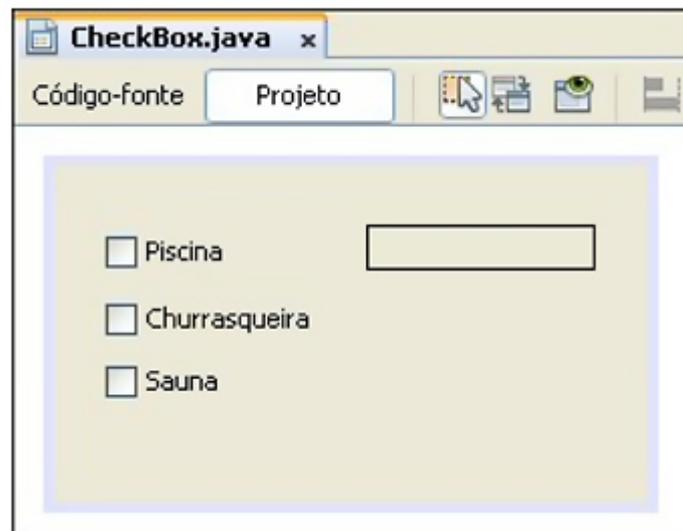
- **borderPainted** – se essa propriedade estiver marcada, será criada uma borda em alto relevo em volta do checkbox. Propriedade utilizada somente para efeito decorativo.
- **focusPainted** – se essa propriedade estiver marcada, significa que o checkbox mostrará o foco quando estiver ativo (uma borda em volta apenas do rótulo). Por padrão, essa propriedade já é marcada.

Vejamos agora uma aplicação prática usando esse componente e um componente **Label** para a saída das informações:

1. Crie um novo projeto (**Arquivo -> Novo projeto**).
2. Salve seu projeto com o nome de: **ComponenteCheckBox**, em uma pasta de sua escolha.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame** com o nome de: **CheckBox**.

4. Clique e arraste três componentes **Caixa de seleção** e um componente **Rótulo** para a área do projeto.
5. Utilize a propriedade **text** e altere os rótulos desses componentes para: **Piscina, Churrasqueira e Sauna.**
6. Clique no componente **Rótulo** e apague o conteúdo da propriedade **text** e crie uma borda em sua volta, conforme já vimos no componente **Lista**. Confira o seu projeto com o mostrado na **Figura 21**:

Figura 21 - Componentes Caixa de seleção e Rótulo durante o projeto

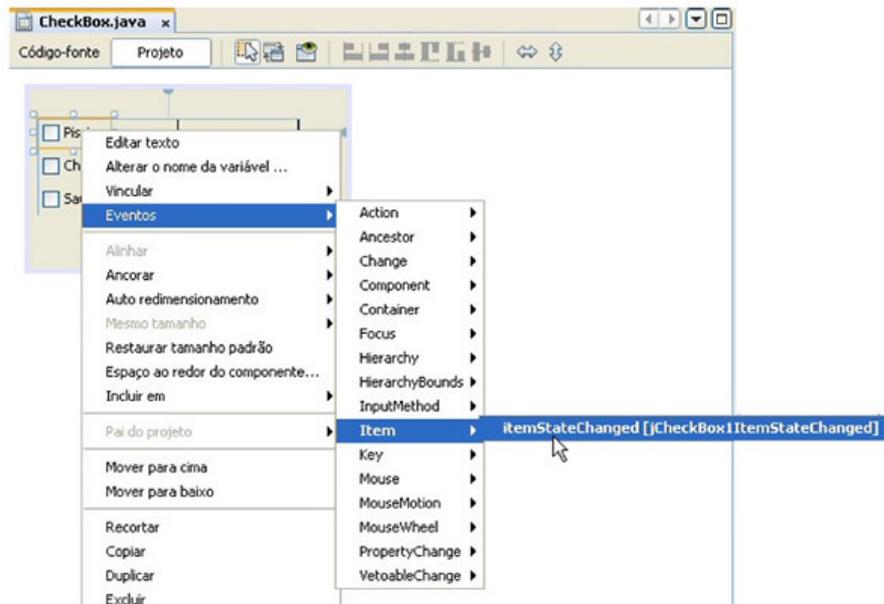


Se você executar a aplicação **F6**, nada acontecerá ainda. Precisamos inserir o código necessário para que a aplicação se torne funcional. Para isso:

Criaremos uma ação para cada um dos componentes **Caixa de seleção**, de forma que, quando um item for selecionado, o seu rótulo seja mostrado no componente **Rótulo**, e a caixa de seleção correspondente fique desabilitada. Para isso:

1. Clique no primeiro componente Caixa de seleção com o botão direito e selecione o seguinte no menu apresentado: **Eventos -> Item -> itemChanged [jCheckBox1ItemStateChanged]**. Veja **Figura 22**:

Figura 22 - Criando um evento para o componente Caixa de seleção



2. Feito isso, será gerado automaticamente o código apresentado na **Figura 23**, que poderá ser visto na seção **Código-fonte** do projeto:

Figura 23 - Código gerado para o evento do componente Caixa de seleção

```
84  
85 private void jCheckBox1ItemStateChanged(java.awt.event.ItemEvent evt) {  
86     // TODO add your handling code here:  
87 }  
88
```

3. Repita esse mesmo processo para cada componente **Caixa de seleção**.

Por enquanto, esse código ainda não tem nenhuma funcionalidade. Precisamos adicionar mais algumas linhas para que ele possa executar o que queremos. Para isso, faça o seguinte:

4. Posicione o cursor no final da linha 86 e tecla **Enter** para criar uma linha em branco.

5. A seguir digite as seguintes linhas:

```
String a = (String)jCheckBox1.getText();  
jLabel1.setText(a);  
jCheckBox1.setEnabled(false);
```

6. Copie esse mesmo código para os outros componentes **Caixa de seleção**. Altere apenas o número do checkbox para 2 e 3, respectivamente.

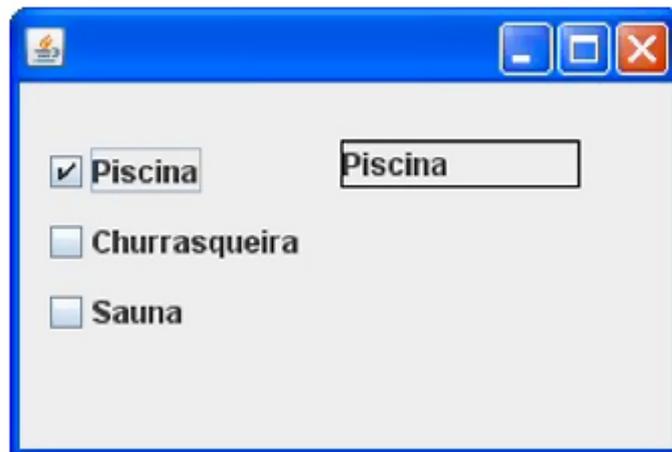
7. Seu código completo referente ao primeiro componente deverá ficar de acordo com o mostrado na **Figura 24**:

Figura 24 - Código completo para o evento 'jCheckBox1ItemStateChanged' do componente Caixa de seleção

```
85 private void jCheckBox1ItemStateChanged(java.awt.event.ItemEvent evt) {  
86     // TODO add your handling code here:  
87     String a = (String)jCheckBox2.getText();  
88     jLabel1.setText(a);  
89     jCheckBox3.setEnabled(false);  
90 }  
91
```

Como já dissemos no início desta aula, você não precisa se preocupar com o que está escrito nesse código. Os eventos e métodos dos componentes serão abordados em aulas futuras. A finalidade desse evento é mostrar, no componente **Rótulo (jLabel1)**, o nome do item clicado em cada caixa de seleção após a aplicação ser executada. Para verificar se o evento está funcionando corretamente, execute a aplicação **F6** e confira o resultado com o mostrado na **Figura 25** a seguir:

Figura 25 - Componentes Caixa de seleção em execução juntamente com o componente Rótulo para exibição do resultado



Vídeo 04 - JCheckBox

Atividade 03

1. Que tecla devemos utilizar para alterarmos o rótulo de um componente **Caixa de seleção** diretamente no componente?
2. Que propriedade deve ser utilizada para que um componente **Caixa de seleção** já apareça marcado quando a aplicação for executada?
3. Se o checkbox da propriedade **enabled** estiver desmarcado, o que acontece?
4. "Se a quantidade de itens de um componente **Lista** ultrapassar o seu tamanho (altura), o componente será redimensionado, automaticamente, de forma que todos os itens possam ser visualizados". Responda se essa afirmativa é verdadeira ou falsa e justifique sua resposta.
5. "O rótulo de uma Caixa de seleção (**JCheckBox**) só poderá ficar à direita da caixa". Essa afirmativa é verdadeira ou falsa?

Conclusão

Bem, chegamos ao final de nossa aula referente a alguns componentes do Java. Procure criar pequenas aplicações utilizando as propriedades abordadas, tendo em vista que grande parte delas servirá também para outros componentes que você conhecerá nas próximas aulas. Procure analisar os códigos vistos e tente criar outras situações com eles.

Resumo

Nessa aula, você conheceu o componente Área de texto, muito utilizado em formulários permitindo ao usuário escrever com mais liberdade e, praticamente, sem regras preestabelecidas, tendo em vista que o componente permite escrever textos bem mais extensos do que os componentes da mesma categoria.

Conheceu também o componente Lista, muito utilizado na criação de listas de itens onde o usuário poderá selecionar um ou mais itens de seu interesse, e, ainda, o componente Caixa de seleção, que são pequenas caixas que podem ser utilizadas para marcar ou desmarcar uma opção, do tipo liga/desliga, também muito utilizadas em formulários para seleção de áreas de interesses.

Autoavaliação

1. Crie uma aplicação utilizando um componente **Área de texto**, de forma que o seu conteúdo seja inserido durante o projeto, com fundo transparente, cor de fundo amarelo, cor da fonte azul e que não seja permitida a edição do texto.
2. Crie um novo projeto utilizando um componente **Caixa de seleção (JCheckbox)** com o nome **Azul**, e um componente **Rótulo (JLabel)**, de forma que, quando a **Caixa de seleção** for marcada, apareça, no componente **Rótulo**, o nome do rótulo do checkbox, e quando for desmarcada, o componente **Rótulo** apareça vazio (em branco).
3. A afirmação: "Se a propriedade **lineWrap** de um componente **Área de texto** estiver marcada, significa que o texto não poderá ser editado", é verdadeira ou falsa?

Referências

HUBBARD, John R. **Programação com Java**. 2. ed. Porto Alegre: Artmed Editora, 2010. (Coleção Schaum).

RODRIGUES FILHO, Renato. **Desenvolva Aplicativos com Java 6**. São Paulo: Editor Érica, 2010.

THE JAVA TUTORIALS. How to Use Lists. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/list.html>>. Acesso em: 5 abr. 2012a.

_____. How to Use Buttons, Check Boxes, and Radio Buttons. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/button.html>>. Acesso em: 5 abr. 2012b.

_____. How to Use Text Areas. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/textarea.html>>. Acesso em: 5 abr. 2012c.