

Objetivos: Utilizar botões rádio e suas funções para realizar seleções;
Agrupar botões;

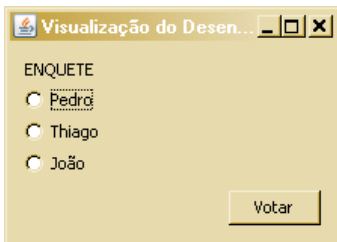
Recursos: Aula em laboratório com uso do computador; Quadro e pincel; Roteiro de Aula;

BOTÕES DE SELEÇÃO

Em muitos sites e mesmo programas desktop encontramos formulários onde precisamos preencher algumas opções pré-definidas. Em Java podemos usar diversos componentes, entre eles o **JRadioButton**.



Para inserir no formulário basta arrastar o componente para dentro da janela.



No código fonte, para verificar se o botão está clicado ou não, use:

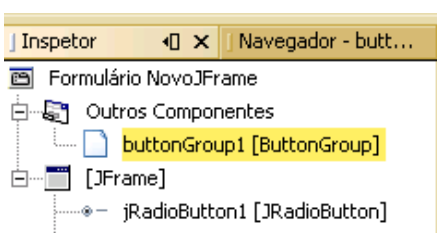
Arquivo: Radio1.java

```
1. if (jRadioButton1.isSelected()) {  
2.     // código a ser executado  
3. }
```

Onde `jRadioButton1` é o nome do componente e `isSelected()` o método que testa se ele está ativado (retorna TRUE) ou não (retorna FALSE). O código dentro do `if` será executado apenas quando ele estiver selecionado.

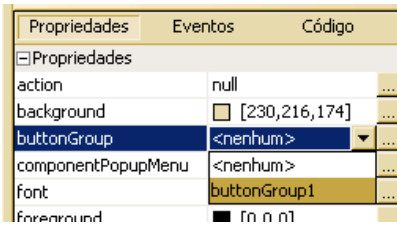
GRUPOS DE BOTÕES

Entretanto, quando colocamos vários botões no formulário, qualquer um deles pode ficar clicado. Na maioria das vezes esse tipo de botão é usado para selecionar UMA opção entre várias. Para isso, devemos usar o componente **ButtonGroup**. Para inseri-lo no programa basta arrastar pra dentro da janela. Este componente não aparece no formulário, apenas no inspetor de objetos.



Depois de inserir um **ButtonGroup**, clique sobre um dos **JRadioButton** no formulário e veja as propriedades deste. Observe que a terceira opção se chama **buttonGroup** e permite selecionar a qual grupo este botão

pertence. Clique no combo e selecione o nome do grupo, nesse caso **buttonGroup1**. Repita a mesma operação para todos os botões deste grupo.



Se existir mais de uma enquete/resposta, será necessário criar um grupo para cada. Adicione mais ButtonGroups e modifique as propriedades dos botões para apontar para o grupo específico.



OUTROS MÉTODOS

Os objetos `JRadioButton` possuem diversos outros métodos úteis, entre eles:

- `getText()` – retorna o texto que está escrito ao lado no botão.
- `setSelected()` – marca o botão através de código (use `true` ou `false` dentro do parêntese).

Métodos do `ButtonGroup`:

- `clearSelection()` – limpa a seleção de todos os botões do grupo.

CAIXAS DE SELEÇÃO E BOTÕES QUE RETEM O CLICK

Outro elemento bem parecido com o `JRadioButton` é o `JCheckBox` que possui as mesmas propriedades e métodos já vistos, apenas é exibido de forma diferente (caixa quadrada com um V dentro).



O **JToggleButton** é um tipo de botão que permite clicar sobre ele e este fica pressionado automaticamente, é encontrado por exemplo nas barras de ferramentas que exibem opções de alinhamento do texto (que só permite um alinhamento por linha). Use em conjunto com o **ButtonGroup** para obter esse efeito. Possui os métodos `isSelected()` e `setSelected()` para testar/ativar o botão.

EXERCÍCIOS

1. O proprietário de uma típica lanchonete da cidade, oferece a seus clientes a opção de criar seu próprio sanduíche, ou seja, através de um formulário o usuário seleciona os ingredientes. Como o valor do sanduíche não é no peso, o proprietário deseja um sistema que **calcule o valor total** a ser cobrado do cliente. Desenvolva este programa observando as seguintes exigências:

- | | |
|--|----------|
| a) Tipos de pão: Hamburger, Hotdog, Francês, Pão de Queijo (apenas 1 tipo por sanduíche) | R\$ 0,30 |
| b) Tipos de recheio: Queijo, Presunto, Hambúrguer, Salsicha, Ovo, Frango | R\$ 0,70 |
| c) Verduras: Tomate, Cebola, Alface, Milho Verde, Ervilha | R\$ 0,15 |
| d) Molho: Sem Molho, Branco, Rosê, Mostarda (apenas 1 tipo por sanduíche) | R\$ 0,30 |

O cliente pode escolher acrescentar mais de um recheio de cada tipo (ex. 2 hambúrgers), então deve-se colocar um campo onde seja possível digitar a quantidade de cada recheio. Os valores acima são para cada item selecionado.

A lanchonete também trabalha com sistema de entregas em domicílio, e cobra R\$ 2,00 a mais sobre o valor do pedido quando o cliente solicita essa opção.

Boa Sorte!