

Programação Estruturada

Aula 15 - Interfaces Gráficas

Apresentação

Até a última aula, você só escreveu programas que interagem com o usuário no modo texto, sem o uso de conceitos como o de janelas, botões, etc., características específicas das interfaces gráficas dos programas. Nesta aula, você irá aprender as ideias por trás da criação de interfaces gráficas.

Ilustraremos os conceitos através de uma aplicação simples, uma calculadora. Esse exemplo será suficiente para praticar um pouco e para deixar você confortável para pesquisar recursos para criação de interfaces gráficas para programas mais elaborados.

Faça uma boa leitura!



Vídeo 01 - Apresentação

Objetivos

Ao final desta aula, você será capaz de:

- Explicar a ideia geral do modelo de eventos das interfaces gráficas.
- Criar interfaces gráficas para programas simples.

1. Introdução

A criação de interfaces gráficas geralmente está relacionada ao que chamamos de programação visual. É uma programação que, em geral, você irá fazer uso de ferramentas de suporte, selecionando componentes visuais e arrastando-os para montar a interface gráfica (uma janela, por exemplo).

Nesta aula, estamos considerando que você está usando o ambiente de desenvolvimento NetBeans. Outros ambientes podem ser utilizados, mas nesse caso pode ser necessário realizar a instalação de *plug-ins* específicos na ferramenta para a criação de interfaces gráficas. De qualquer forma, cada ambiente de desenvolvimento diferente pode ter recursos adicionais e específicos para a criação de janelas. Cabe a você escolher qual ambiente melhor atende suas necessidades.

Nas próximas seções, você irá aprender o passo a passo para criação de uma interface gráfica para o programa Calculadora!

2. Criação do Projeto

O primeiro passo para construir o programa Calculadora utilizando interfaces gráficas é a criação do projeto. A criação de projetos que envolvem a construção de interfaces gráficas geralmente é feita da mesma forma como são criados os projetos que construímos até o momento, como mostra as Figuras de 1 a 3.

Note na Figura 3 que o nome do projeto que está sendo criado se chama Calculadora.

Figura 01 - Criação de um novo projeto

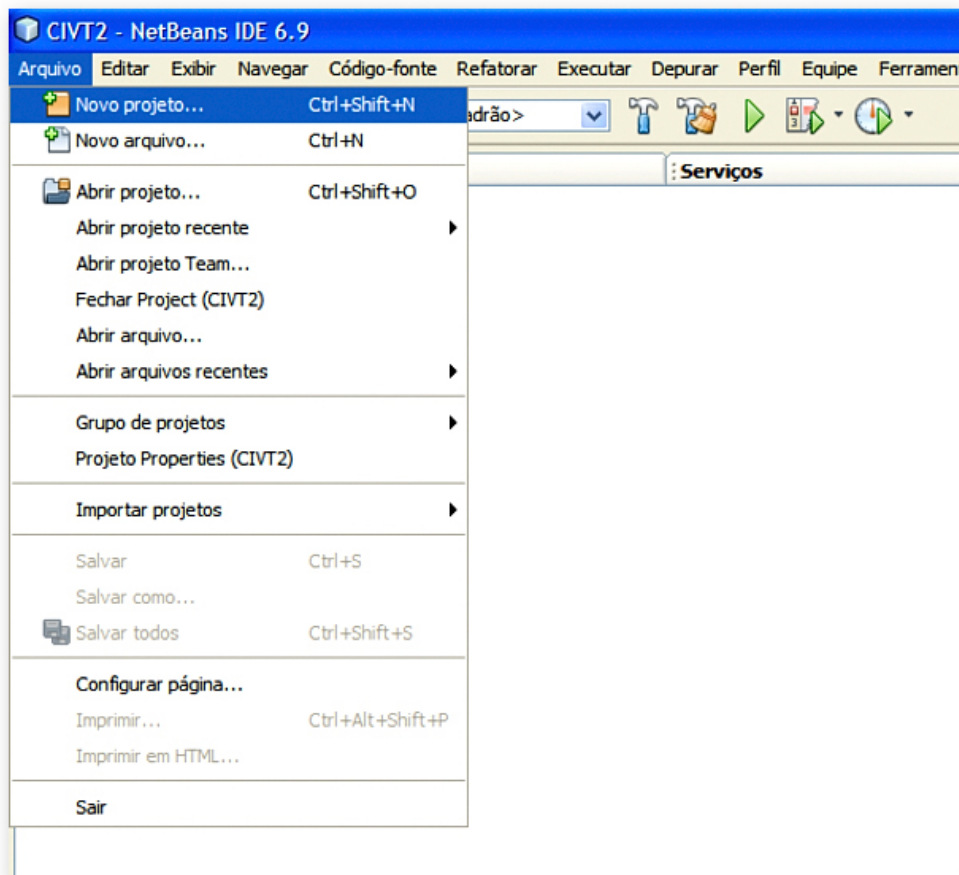


Figura 02 - Seleção do tipo de aplicativo: aplicativo Java

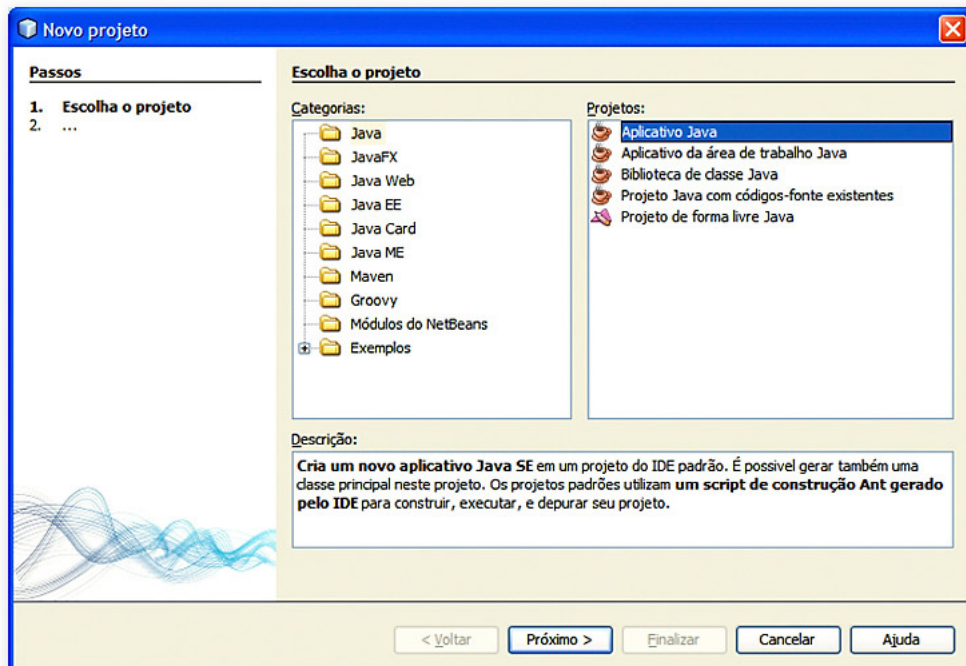
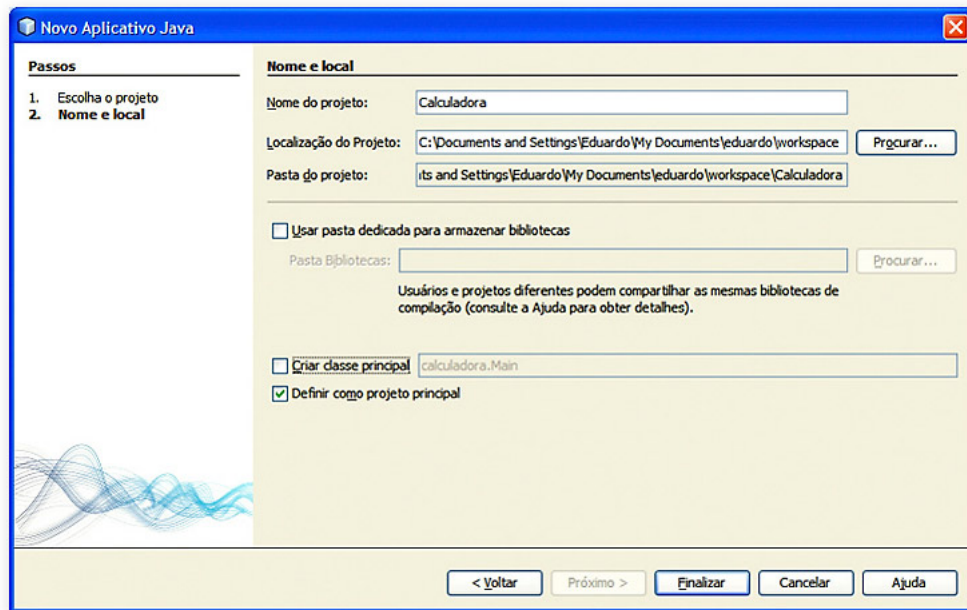


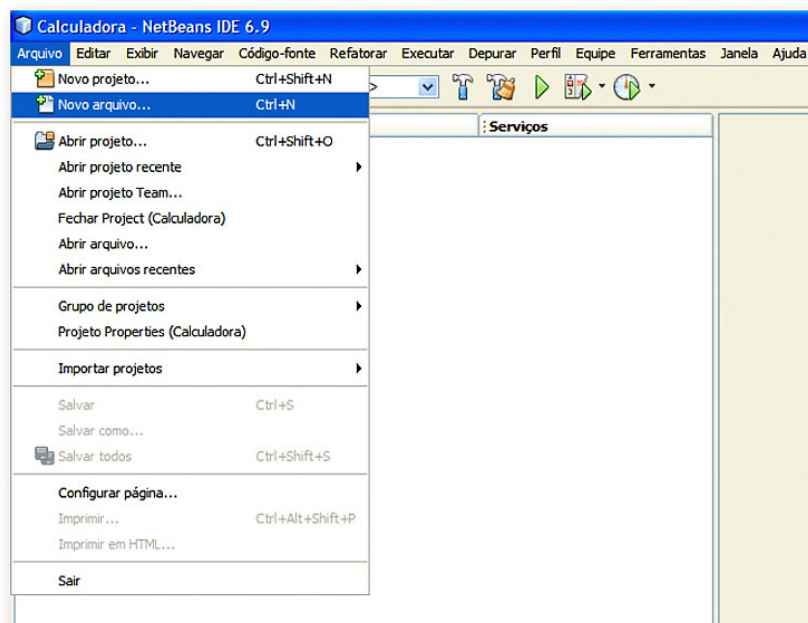
Figura 03 - Informações sobre o projeto a ser criado



3. Uso de Janelas

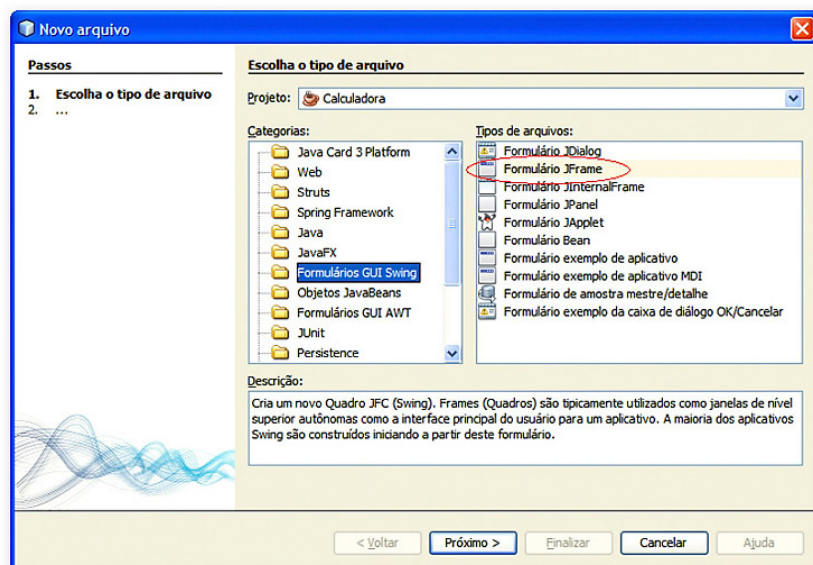
Muito bem, agora que você já criou um projeto, precisamos agora de uma classe especial que representa a interface gráfica (janela) do programa. Para isso, você pode selecionar a criação de um novo arquivo, como mostra a Figura 4.

Figura 04 - Criação de um novo arquivo para representar uma janela



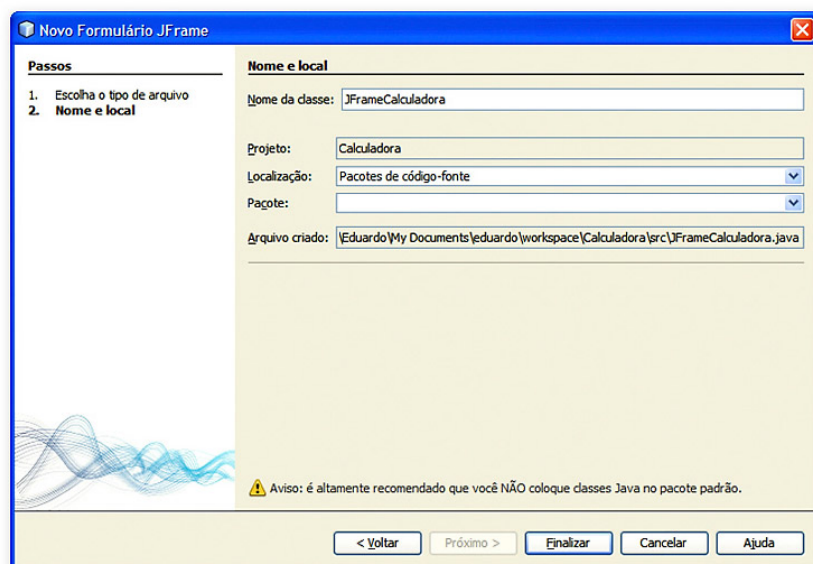
Após isso, será mostrada uma nova janela assistente de criação de arquivos. Nela, você deve selecionar a opção de criação de “formulários GUI swing”, como mostrado na Figura 5. Dentre as opções que serão mostradas, selecione a opção “formulário JFrame”, que é o tipo que representa janelas de aplicativos Java.

Figura 05 - Dentre as opções de GUI Swing disponíveis, selecionar o JFrame



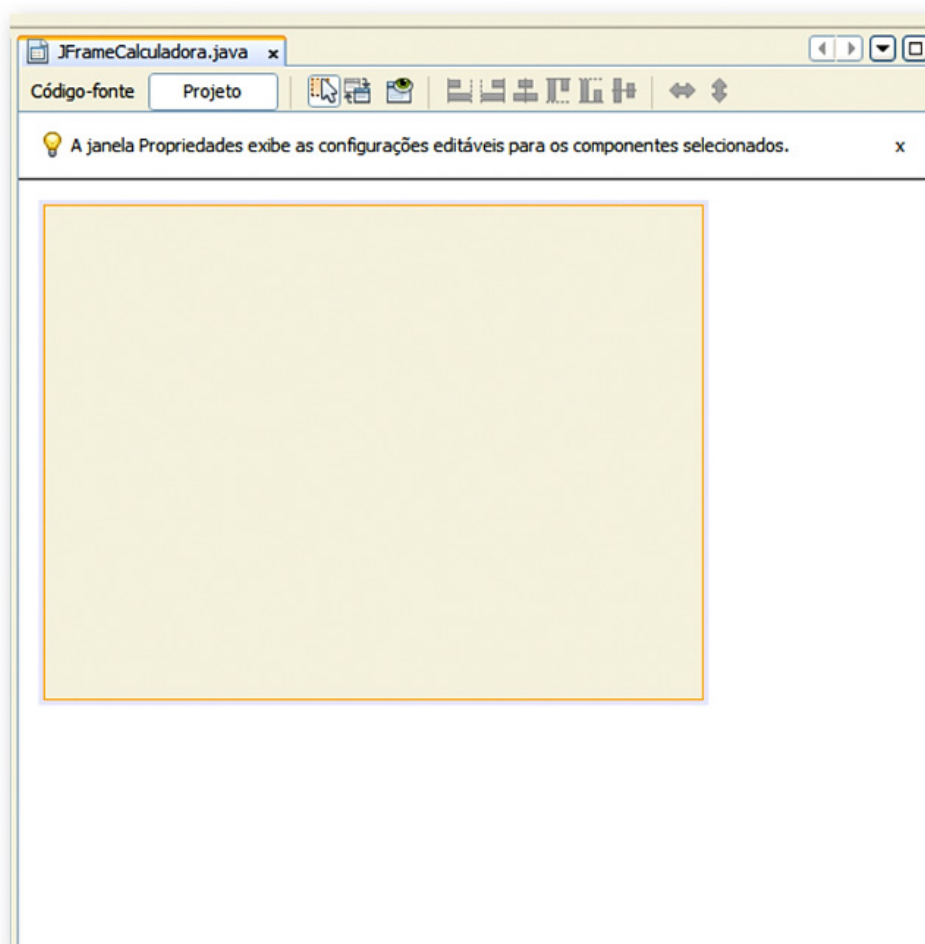
Por fim (Figura 6), devemos informar o nome do arquivo (classe) a ser criado que, no caso, é JFrameCalculadora.

Figura 06 - Indicar nome do arquivo (classe) a ser criado



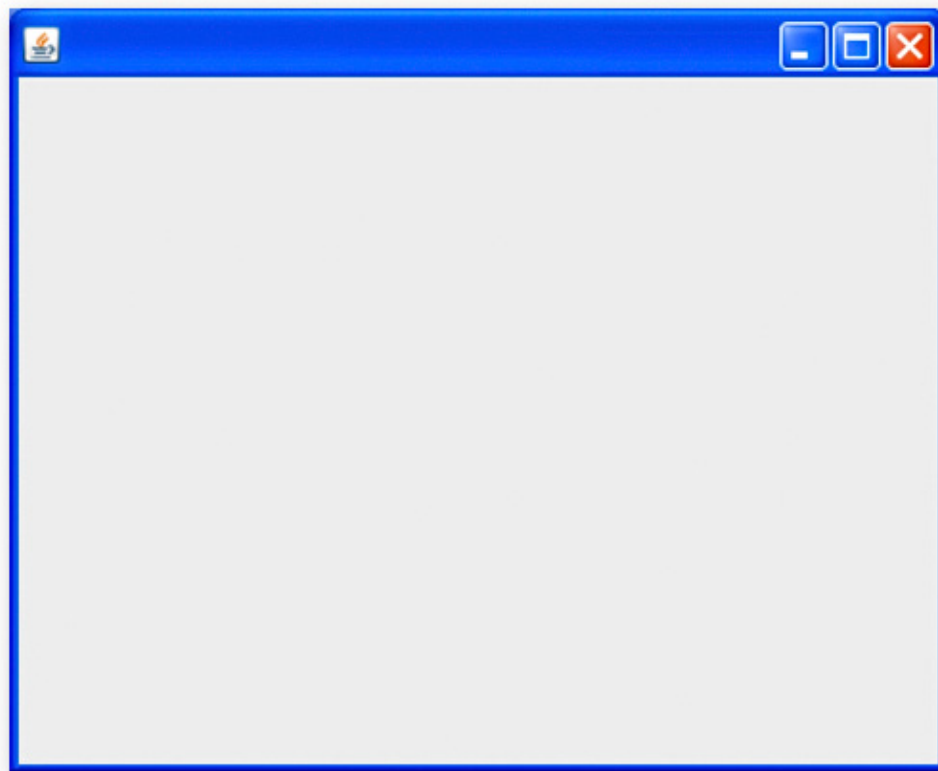
Após a criação do arquivo JFrameCalculadora, ele será aberto no modo de programação visual (projeto), como mostrado na Figura 7. Este modo é o mais utilizado para programação de interfaces gráficas. Note que existe também a opção “código-fonte”, a qual mostra o código-fonte Java da interface gráfica, caso você queira ver o código Java do programa.

Figura 07 - Modo visual de programação do JFrameCalculadora



Uma vez criado o JFrame, você até pode executar o projeto, pois uma rotina *main* é criada automaticamente. Essa rotina abre a janela JFrameCalculadora, como mostrado na Figura 8. Ao rodar o projeto, você pode fazer o JFrameCalculadora ser a classe principal do projeto. Para terminar a execução, basta clicar no botão ‘x’ da janela.

Figura 08 - Janela JFrameCalculadora quando executado sem nenhuma alteração



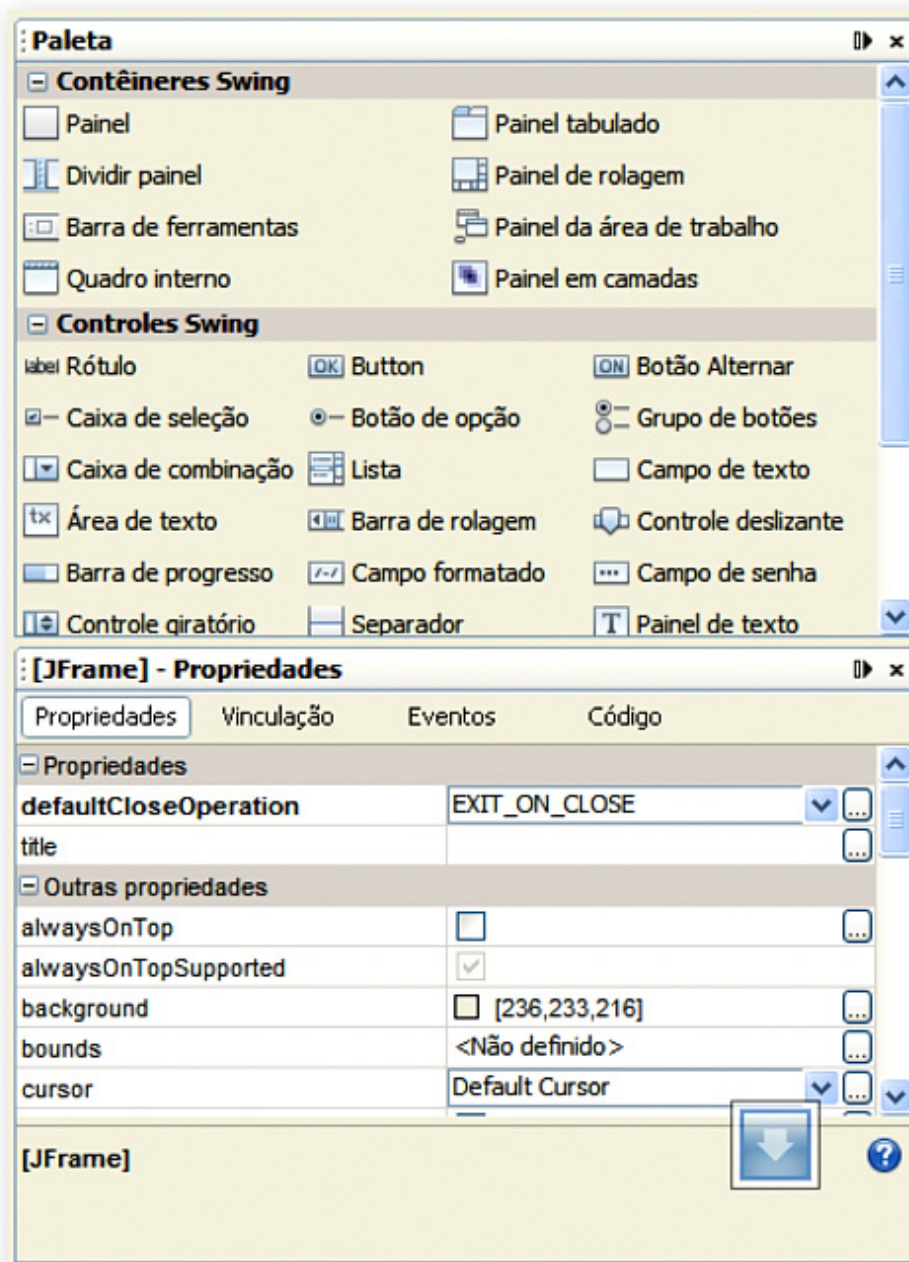
Atividade 01

1. Proceda no NetBeans os passos indicados para a criação do JFrameCalculadora. Execute o programa e verifique se ele está abrindo a janela.

4. Inclusão de Componentes Visuais

Nesse momento, podemos realizar a programação visual do JFrameCalculadora através da paleta de componentes mostrada na Figura 9. A área da Figura 9 pode na verdade ser dividida em duas partes. A primeira delas está relacionada aos componentes que podemos incluir no JFrame (paleta). A segunda parte se refere às propriedades, eventos, entre outras coisas, relacionados ao componente visual que está selecionado, que no caso é o próprio JFrame (janela).

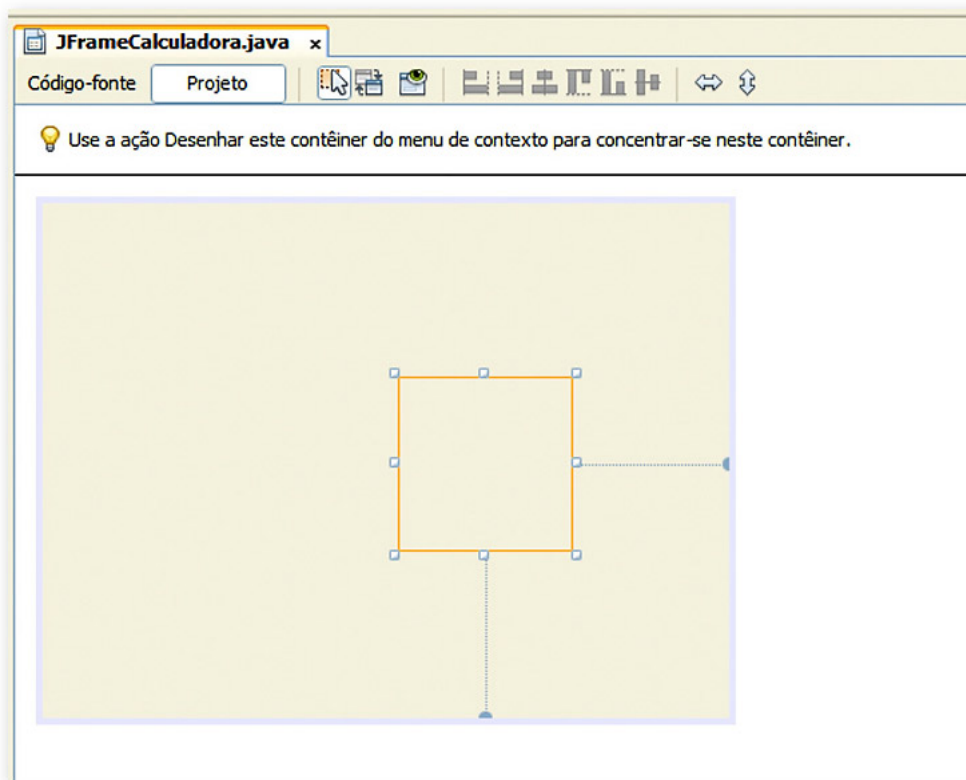
Figura 09 - Paleta de componentes e propriedades associada ao JFrame



O primeiro componente visual que devemos inserir em um JFrame é um **painel** (JPanel). Veja que esse componente é um contêiner, ou seja, é um elemento que pode conter outros componentes visuais. Para adicionar um painel ao JFrame, clique na opção "painel" (em "contêineres swing") mostrada na Figura 9 e depois dê um clique na área da janela do JFrame.

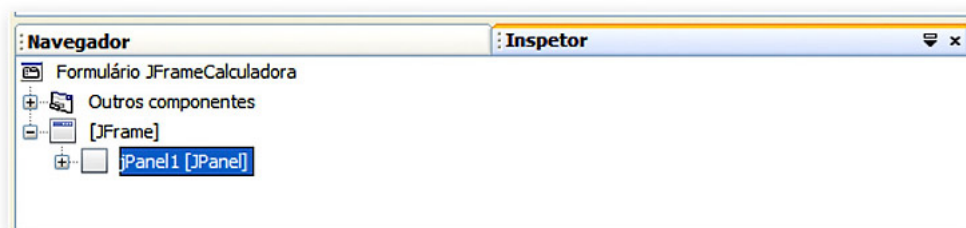
Há agora duas coisas para observar. A primeira é que o painel será colocado dentro do JFrame, como mostrado pela Figura 10. Amplie as dimensões do painel para que ele ocupe toda a área da janela. Você pode fazer isso clicando e arrastando as bordas do painel.

Figura 10 - Painel colocado dentro do JFrame



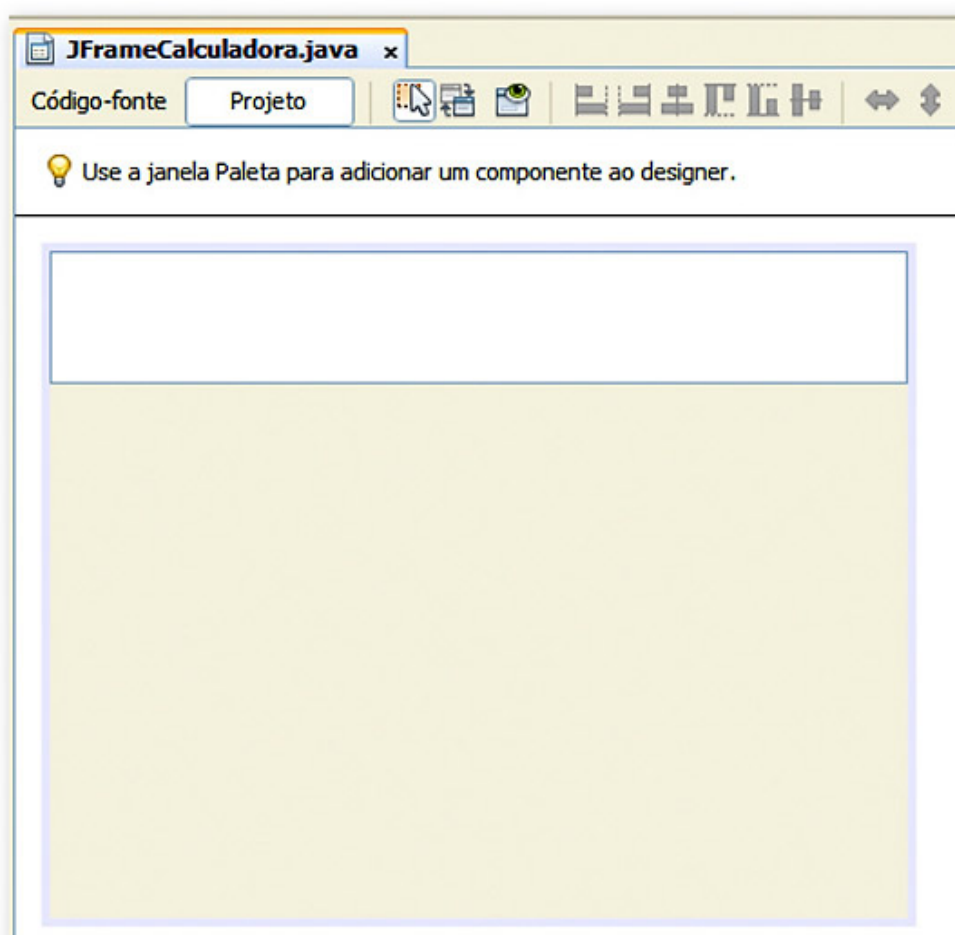
A segunda coisa a se observar é que a seção “inspetor” do NetBeans irá mostrar uma hierarquia de componentes, como podemos ver na Figura 11. Esta hierarquia mostra que dentro do JFrame agora existe um painel (JPanel). E aqui vai uma dica: se você quiser ver as propriedades de um componente visual, pode selecioná-lo na janela do JFrame ou selecioná-lo através do “inspetor”.

Figura 11 - Hierarquia de componentes mostrando que dentro do JFrame existe um JPanel



Pois bem, adicione agora ao painel um componente visual chamado **campo de texto**, localizado na seção de controles de *swing* da paleta. Posicione-o para que fique como mostrado na Figura 12. Para deixá-lo em branco, selecione o componente e veja a parte de “propriedades”, procure pelo campo *text* e apague seu conteúdo.

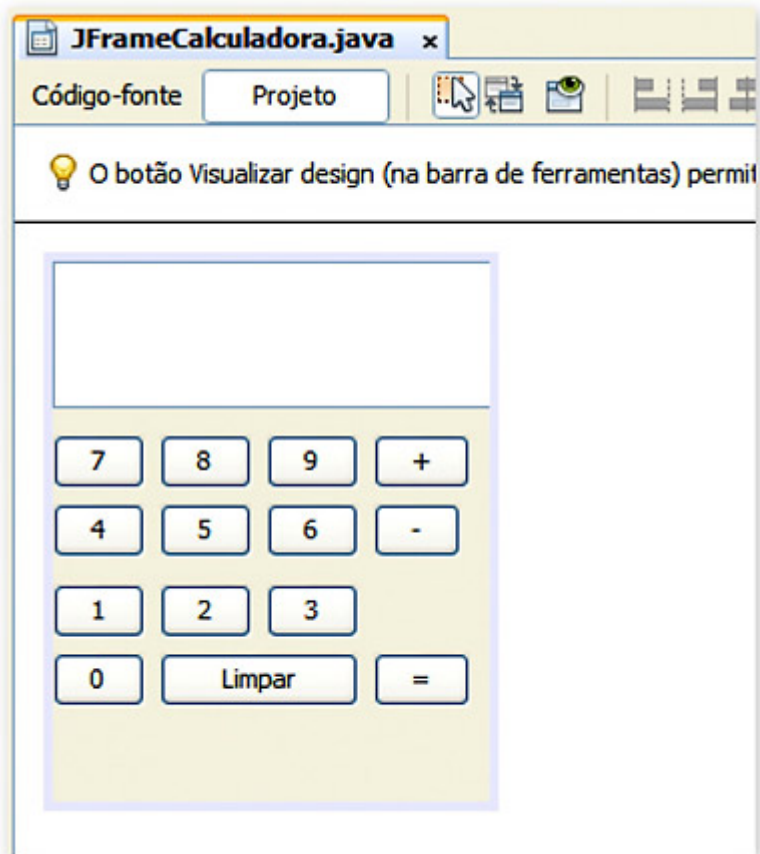
Figura 12 - Campo texto posicionado no início da janela



Para que esse campo não fique modificável direto pelo teclado (ex: clicar no campo texto ao executar o programa e digitar letras e caracteres), você pode desabilitar a propriedade *editable* do componente.

Agora vamos adicionar os botões da calculadora? Clique no componente visual *button* existente nos “controles *swing*”. Adicione 10 botões, um para cada número de 0 a 9. Além dos botões para as operações de soma e de subtração, um botão de limpar e outro de igual (=). Você pode fazer o arranjo visual que deseja, como por exemplo, o mostrado na Figura 13. O texto que é mostrado nos botões pode ser alterado através da propriedade *text* dos mesmos.

Figura 13 - Botões adicionados à janela da calculadora



Vídeo 02 - Componentes Básicos

Atividade 02

1. Proceda no NetBeans os passos indicados para a evolução do JFrameCalculadora. Execute o programa e verifique se ele está abrindo a janela e mostrando os componentes visuais de acordo com o projetado.
2. Execute o programa da calculadora e clique nos botões. Relate se há algum comportamento, como a apresentação dos números clicados no campo texto.
3. Adicione botões para as operações de divisão e de multiplicação.

5. Adicionando Comportamento

Para que nosso programa de calculadora possa funcionar é necessário que você codifique o comportamento resultante das operações de clique nos botões. Essa parte da programação não é feita visualmente. Para começar, vamos criar uma enumeração, como mostrado a seguir:

```
1 public enum Operacao {  
2     NENHUMA, SOMA, SUBTRACAO  
3 }
```

Essas são as operações da nossa calculadora. A opção “nenhuma” é a opção padrão (inicialmente selecionada) ao se executar a calculadora. Além disso, precisamos adicionar variáveis no JFrame. Acesse o arquivo do JFrame e clique na opção “código-fonte”. Esta opção irá abrir o código-fonte da janela. Adicione as seguinte variáveis logo após a declaração da classe JFrameCalculadora:

Figura 14

```
16 public class JFrameCalculadora extends javax.swing.JFrame {  
17  
18     private double valor1;  
19     private double valor2;  
20     private Operacao operacao = Operacao.NENHUMA;  
21     private boolean limpar;
```

As variáveis “valor1” e “valor2” são utilizadas para armazenar os valores a serem operados pela calculadora. Já a variável “operacao” indica a operação a ser realizada, que no início é “nenhuma”. Além disso, há uma variável booleana “limpar”, que será explicada mais adiante.

Note que a declaração não utiliza a palavra chave *static*. Você entenderá melhor quando usá-la ou não assim que esse módulo acabar e começar o de orientação a objetos.

Vamos aproveitar e adicionar dois procedimentos nesse arquivo, como mostrado a seguir:

Figura 15

```
23 public void operar() {
24     switch (operacao) {
25         case NENHUMA:
26             valor1 = valor2;
27             break;
28         case SOMA:
29             valor1 = valor1 + valor2;
30             break;
31         case SUBTRACAO:
32             valor1 = valor1 - valor2;
33             break;
34     }
35     jTextField1.setText(String.valueOf(valor1));
36     limpar = true;
37 }
38
39 public void limpar() {
40     if (limpar) {
41         jTextField1.setText("");
42     }
43     limpar = false;
44 }
```

O primeiro procedimento é o “operar”, que tem como objetivo realizar a operação correta. Note o uso da variável “operação” para decidir sobre qual operação deve ser realizada. Se for “nenhuma”, isso quer dizer que o valor atualmente digitado (considere que é o “valor2”) é na verdade o primeiro número a ser operado.

Já as outras opções realizam o devido cálculo, armazenando o resultado na variável “valor1”. Por fim, esse valor é transferido para o campo texto e o valor da variável limpar é colocado para *true*. Note o uso da função “jTextField1.setText()” para substituir o texto atual do campo “texto”. Seu parâmetro é o valor da variável “valor1” convertido para *String* através da função de conversão “String.valueOf()”.

Já a operação “limpar()” tem como objetivo limpar o campo “texto” se o valor da variável for *true*, passando seu valor para *false* no final. O objetivo desse procedimento será explicado mais adiante.

Pois bem, voltemos agora para a programação visual, clicando no botão “projeto”. Para fazermos com que o clique no botão ‘1’ escreva esse valor no campo “texto”, você precisa dar clique duplo em cima desse botão. Ao fazer isso, um espaço

de código será apresentado para você preencher. Esse espaço representa o código a ser executado ao acontecer o evento de clique no botão. Por isso dizemos que as interfaces gráficas seguem um modelo baseado em eventos.

Preencha de acordo com o mostrado a seguir:

Figura 16



```
211  
212 private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {  
213     limpar();  
214     jTextField1.setText(jTextField1.getText() + "1");  
215 }  
216
```

Observe que apenas a parte em branco é que pode ser alterada. Isso é uma garantia do NetBeans de que você só vai alterar o que é necessário. O comando escrito chama a operação "limpar()". Essa operação é para limpar o campo "texto" caso o último botão a ser pressionado seja o de uma operação. Isso porque, depois de clicar em uma operação, devemos digitar um segundo número, certo? Para isso, precisamos apagar o número anterior.

Depois, usamos a variável "jTextField1", que representa o campo "texto" da janela. Veja esses nomes através do "inspetor" do NetBeans. Como você já sabe, o comando "jTextField1.setText()" substitui o texto a ser mostrado pelo componente. No caso, o novo texto é igual a "jTextField1.getText()", função que retorna o texto atual do componente, concatenado com o número '1'. Isso vai adicionar o número 1 à direita do número já digitado.

Esse procedimento deve ser repetido para todos os demais botões de números. E lembre-se, para voltar ao modo de programação visual, é só clicar no botão "projeto" logo acima da área do JFrame. Outra dica, para que os números fiquem alinhados à direita, você pode alterar a propriedade "horizontalAlignment" para *right* do "jTextField1".

Agora vamos implementar os botões das operações. Começando pela adição, efetue o duplo clique em cima desse botão para escrever seu comportamento:

Figura 17

```
217 private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {  
218     valor2 = Double.parseDouble(jTextField1.getText());  
219     operacao = Operacao.SOMA;  
220     operar();  
221 }
```

Observe na primeira linha que a variável “valor2”, do tipo *double*, receberá o valor que está no campo “texto”. Para pegar esse valor, você já sabe que deve usar a função “jTextField1.getText()”, porém, o retorno dessa função é uma *String*. Para converter uma *String* para *double*, usamos a função “Double.parseDouble”. Após isso, é indicada a operação a ser realizada (variável “operação”) e chamamos a rotina “operar()”.

Esse mesmo procedimento (duplo clique no botão e escrita do código) deve ser escrito para as operações de subtração e de igualdade. No caso da operação de subtração, o código será “Operacao.SUBTRACAO”. Já para a igualdade, temos o seguinte:

Figura 18

```
223 private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {  
224     valor2 = Double.parseDouble(jTextField1.getText());  
225     operar();  
226     operacao = Operacao.NENHUMA;  
227 }
```

Veja que chamamos diretamente o “operar()”, já que espera-se que a operação a ser realizada já tenha sido indicada. Por fim, indica-se que não há mais nenhuma operação a ser feita, pois ela já foi finalizada.



Vídeo 03 - Interface



Vídeo 04 - Programa Completo

Atividade 03

1. Proceda no NetBeans os passos indicados para a evolução do JFrameCalculadora. Execute o programa e verifique se a calculadora está funcionando de acordo com o projetado.
2. Adicione botões para as operações de divisão e de multiplicação. Altere o código para que eles funcionem.

6. Leitura Complementar

Para explorar mais o tema de interfaces gráficas, explore o conteúdo do seguinte site:

- <<http://download.oracle.com/javase/tutorial/uiswing/>>

7. Resumo

Nesta aula, você aprendeu um pequeno pedaço das interfaces gráficas no Java. Você viu o uso dos componentes JFrame para representar uma janela, JPanel para agrupar outros componentes, JTextField como campo “texto” e JButton como botões. Esse é um tema bastante extenso para ser explorado. Por exemplo, existem diversos componentes visuais que podem ser utilizados. Para aprender mais sobre esse assunto, consulte a documentação contida na Leitura Complementar, bem como tutorias e vídeos na internet. Aqueles que forem da ênfase de Informática para Internet também verão esse assunto mais a fundo em Desenvolvimento Desktop.

Esse foi o nosso último encontro! Espero que vocês tenham gostado da disciplina. Um forte abraço!

8. Autoavaliação

1. Explique qual a ideia da programação visual de interfaces gráficas, incluindo a questão de ser baseado em eventos.
2. Identifique componentes visuais que podemos utilizar para construir as interfaces gráficas. Explique suas funções.

9. Referências

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores:** algoritmos, Pascal, C/C++ e Java. São Paulo: Editora Pearson, 2008.

THE JAVA tutorials. Disponível em:
<http://download.oracle.com/javase/tutorial/>. Acesso em: 6 dez. 2011.