

Unidade 11: Programando Swing com o NetBeans

Prof. Daniel Caetano

Objetivo: Construir uma aplicação baseada na classe JDialog do Java Swing.

INTRODUÇÃO

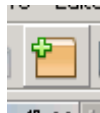
Na aula passada vimos como construir uma interface simples, do tipo JFrame, usando apenas instruções da linguagem Java. Este tipo de janela é ideal para a janela principal de aplicativos, mas existem muitas outras situações em que construir uma janela da maneira vista pode ser bastante entediante e ineficiente.

Assim, nesta aula veremos como utilizar o NetBeans para nos ajudar a construir uma aplicação Swing simples, utilizando a classe JDialog, ideal para janelas de configuração das diversas funções de um programa.

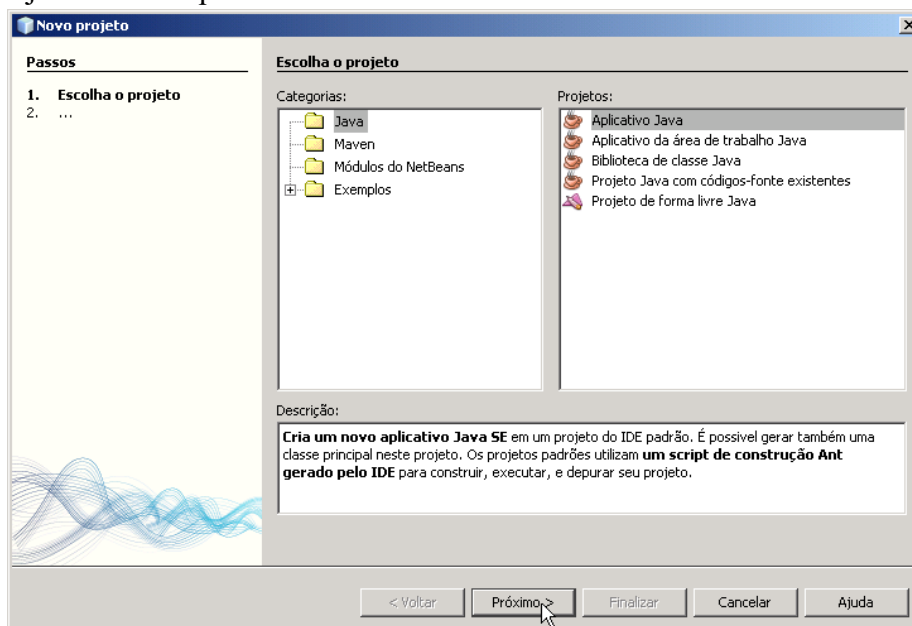
Neste exemplo, veremos como construir facilmente uma calculadora com funções bastante básicas. Fica como exercício para o aluno implementar funções mais complexas.

1. CALCULADORA USANDO SWING E NETBEANS

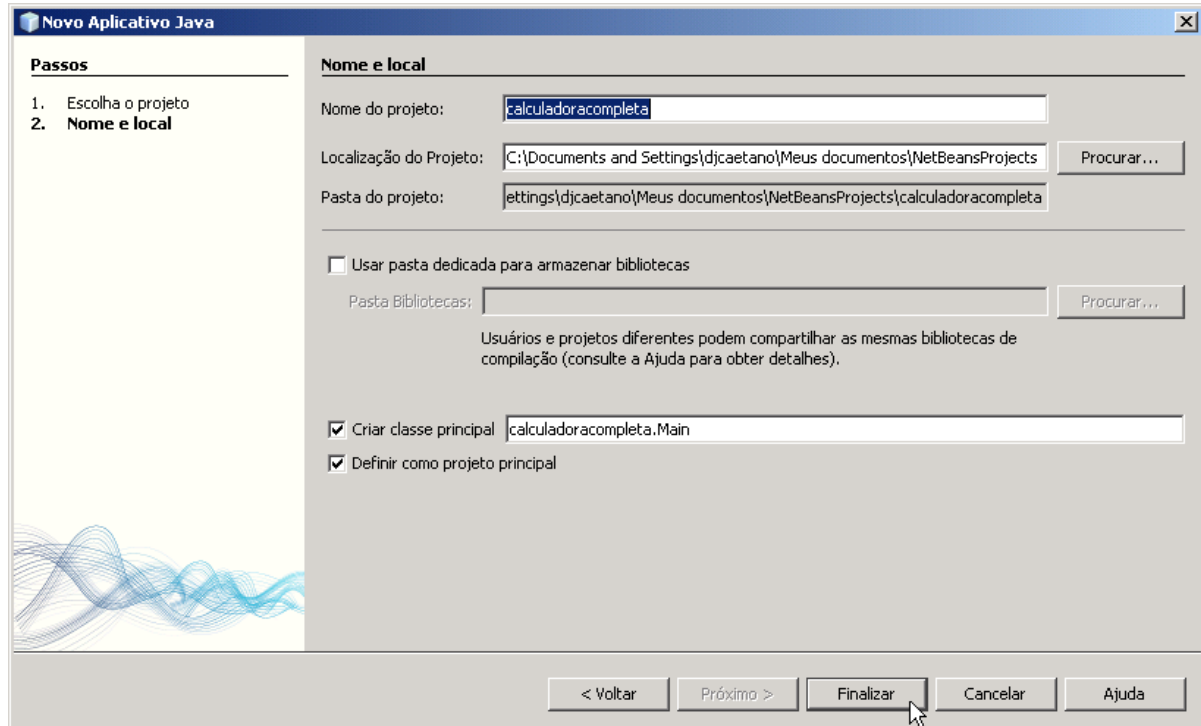
PASSO 1 - A primeira coisa a se fazer é criar um novo projeto. Você pode ir pelo menu (Arquivo > Novo > Projeto) ou clicar no ícone indicado abaixo:



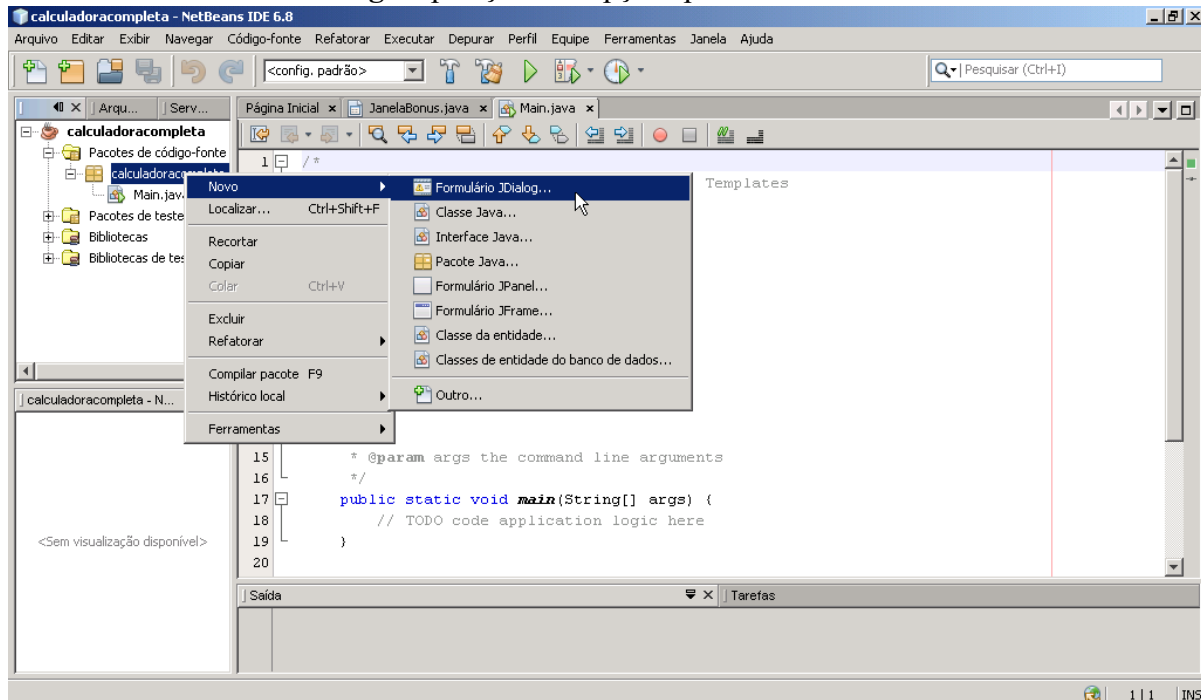
PASSO 2 - Agora é necessário escolher o tipo de projeto. Escolha a categoria "Java" e o tipo de projeto como "Aplicativo Java".



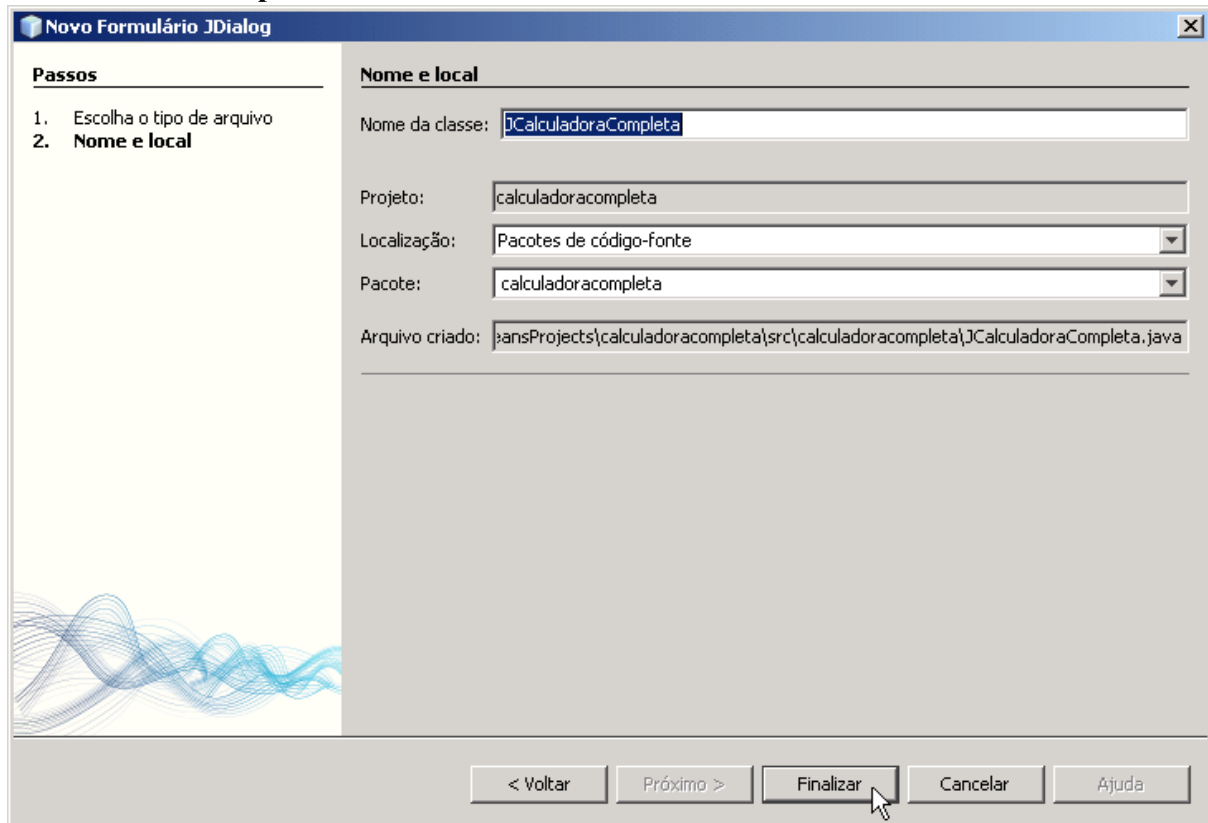
PASSO 3 - Agora é necessário dar um nome ao projeto. Lembre-se: NÃO use caracteres especiais, NÃO use letras maiúsculas, NÃO use espaços. No exemplo abaixo, usamos o nome de "calculadoracompleta".



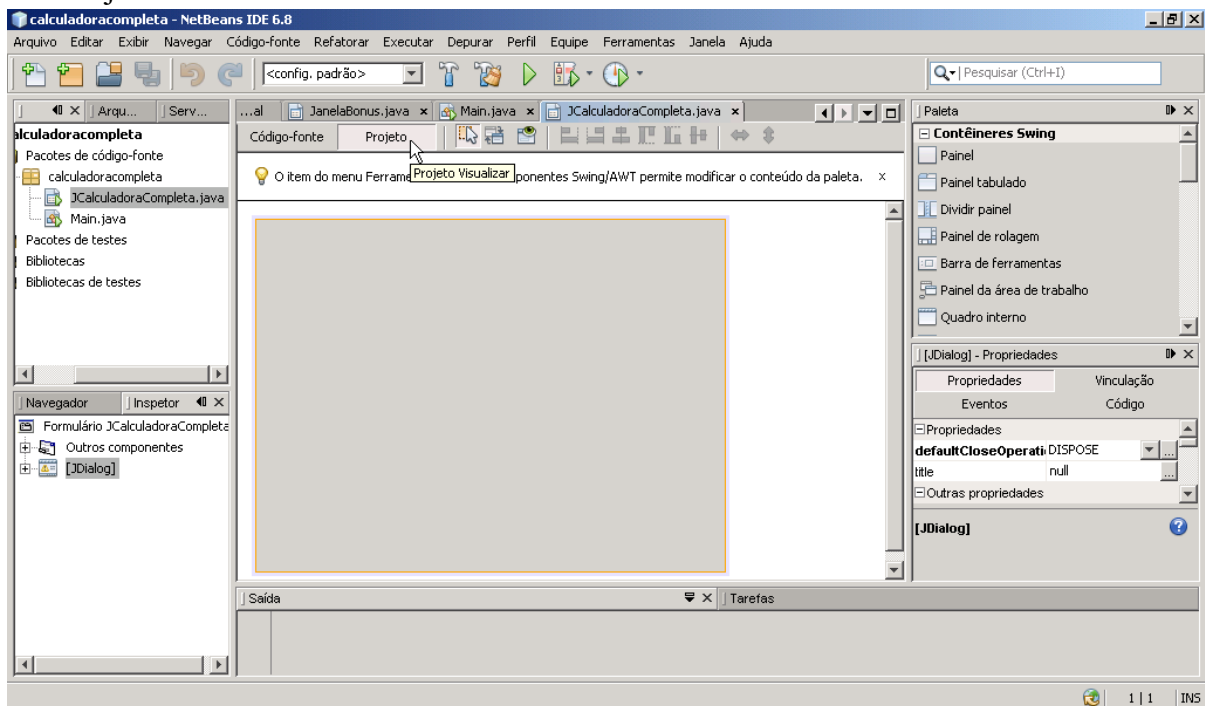
PASSO 4 - A próxima coisa a fazer é criar a classe da janela do tipo `JDialog`. Clique com o botão direito do mouse no ícone do PACOTE com o nome **calculadoracompleta** (é o que tem o pacote do lado, **não é** o que tem a xícara de café do lado). Selecione as opções **Novo > Formulário JDialog**. A posição das opções pode ser diferente no seu NetBeans.



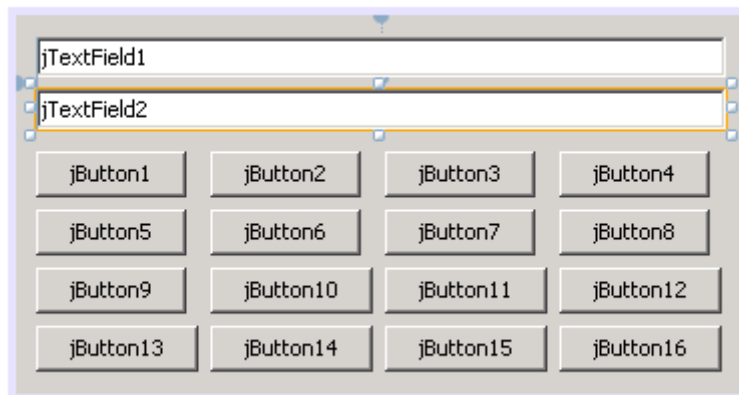
PASSO 5 - Precisamos dar um nome para a janela de nossa aplicação. Como é um nome de classe, lembre-se de começa-lo com letra maiúscula. No exemplo, o nome dado foi **JCalculadoraCompleta**.



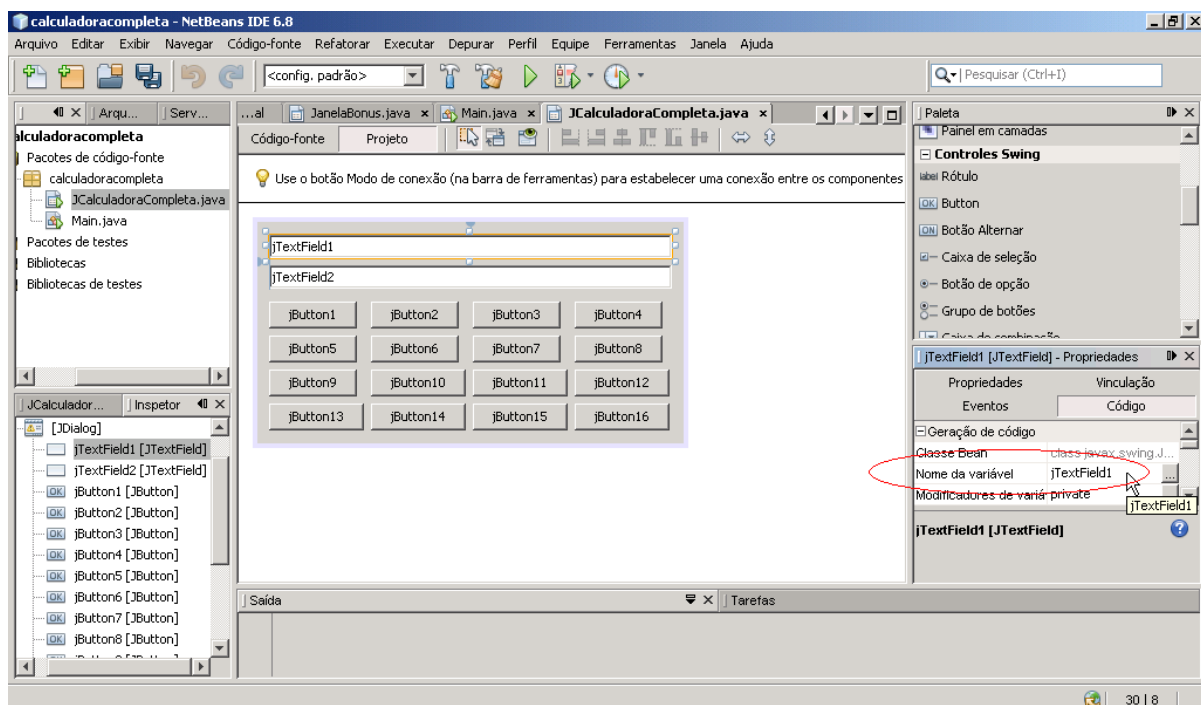
PASSO 6 - Selecionando o botão "Projeto" indicado na figura abaixo, podemos editar o layout da janela, arrastando os itens (botões, caixas de texto etc) da lista da direita para a área da janela.



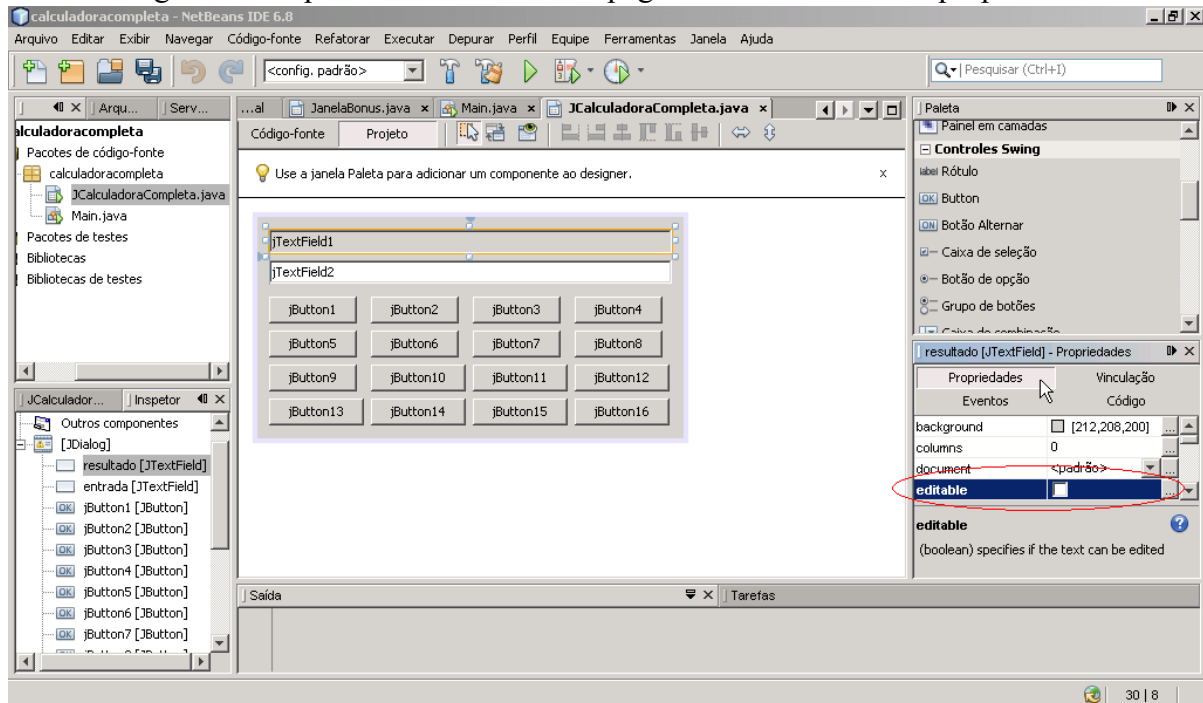
PASSO 7 - Arraste dois campos de texto (JTextField) e 16 botões (JButton) e organize-os conforme indicado abaixo.



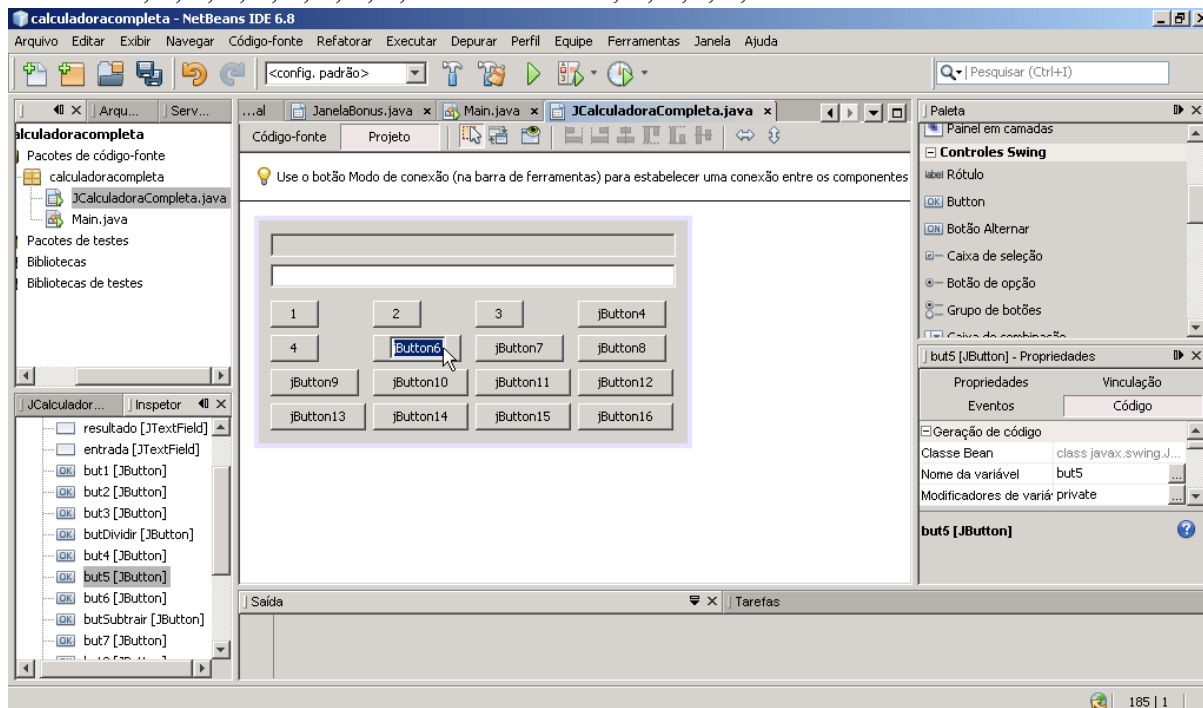
PASSO 8 - Selecione o campo de texto (JTextField) mais do alto (no exemplo, aquele que está com o text "jTextField1"). Clicando no botão "Código" no lado direito, encontre a opção **Nome da variável** e modifique o valor dela para **resultado**. Repita o processo para o jTextField2 e mude o **Nome da variável** para **entrada**.



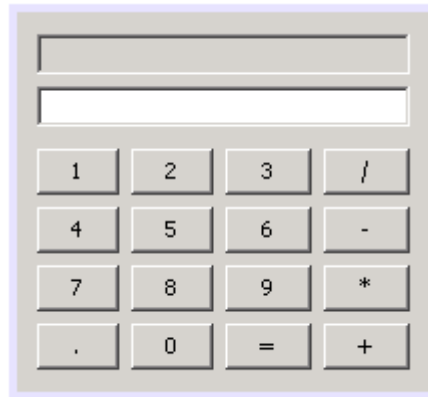
PASSO 9 - Clique agora no botão "Propriedades". Selecione o campo de texto de cima e, na opção **title**, **apague** o texto, deixando a caixa de texto vazia. Procure a opção **editable** e **desmarque-a**, tornando a caixa de texto desligada para edição do usuário. Selecione agora o campo de texto de baixo e apague também o valor da propriedade **title**.



PASSO 10 - Agora dê dois cliques bem lentamente (com uma pausa maior entre os dois cliques) em cada botão para editar seu texto. Modifique-os para que contenham os números 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 e os sinais ".", =, +, -, *, /.



O resultado deve ter a aparência indicada na figura a seguir. Agora, selecione o botão com o número "1" e, no lado direito da janela, clique em "Código" para visualizar as propriedades do botão. Mude o **nome da variável** para **but1**.

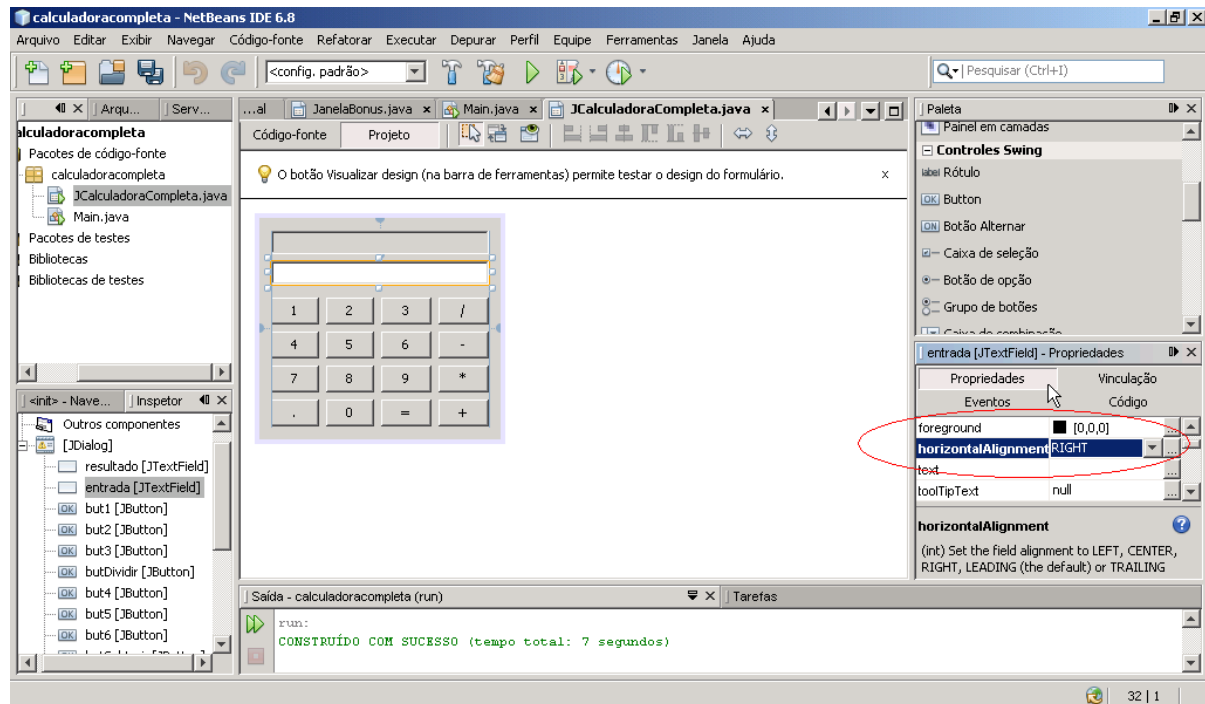


Selecione agora o botão 2 e mude o **nome da variável** para **but2**. Repita esse procedimento para todos os botões. A tabela abaixo indica o nome que deve ser dado a cada botão da janela:

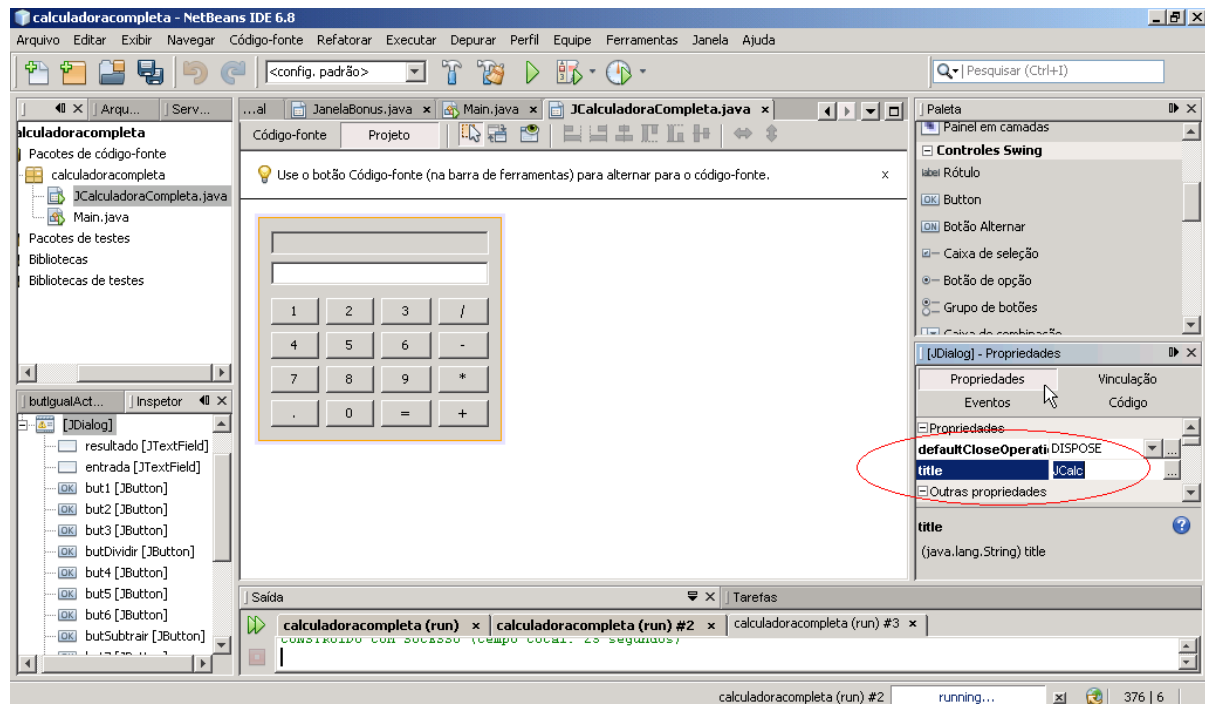
Botão	Nome da Variável
1	but1
2	but2
3	but3
4	but4
5	but5
6	but6
7	but7
8	but8
9	but9
0	but0
.	butPonto
=	butIgual
/	butDividir
-	butSubtrair
*	butMultiplicar
+	butSomar

Estes nomes são importantes para que possamos acessar estes elementos a partir de nosso código.

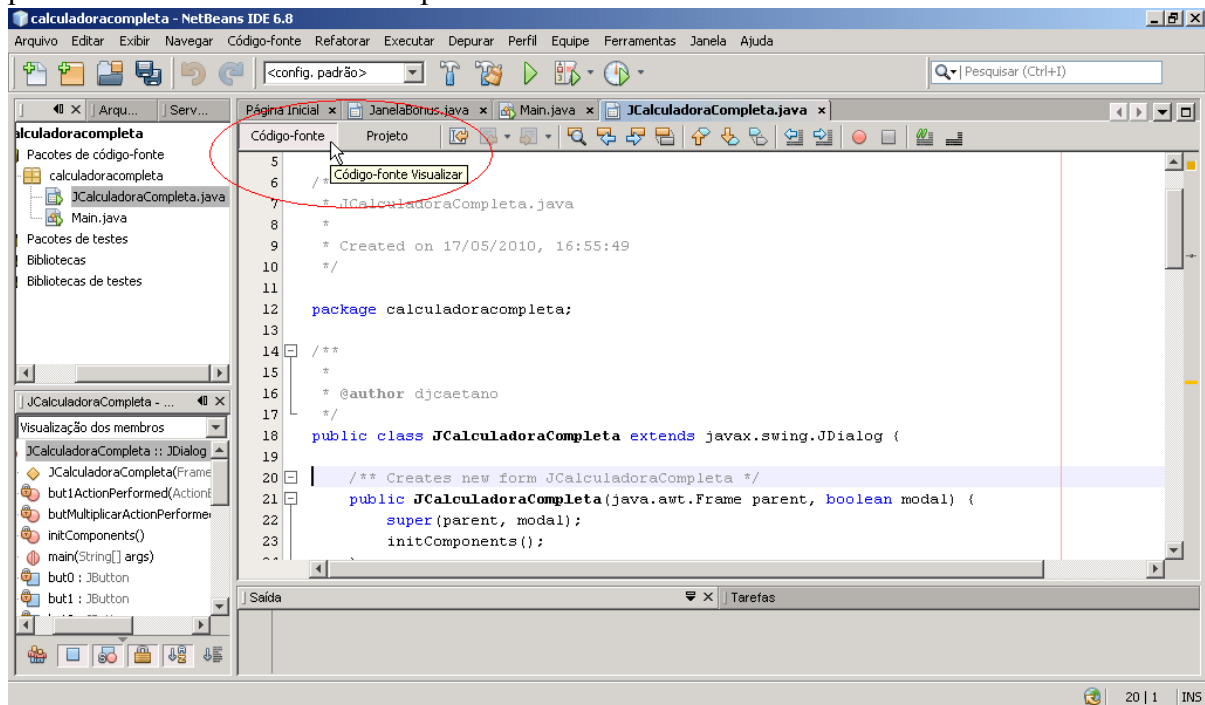
PASSO 11 - Vamos indicar alinhamento à esquerda para os dois campos texto. Selecione o campo texto de cima e, em suas propriedades, procure por **horizontalAlignment** e altere seu valor para "RIGHT", como indicado abaixo. Repita o procedimento para o campo de texto de baixo.



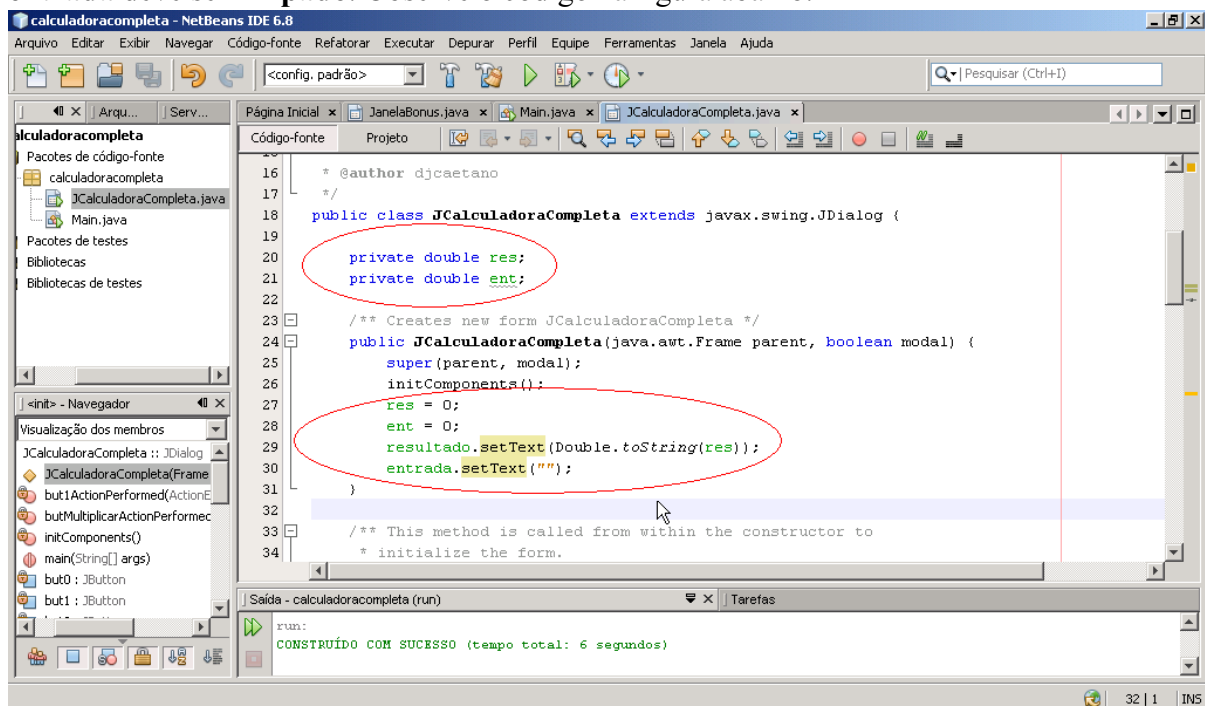
PASSO 12 - Vamos dar um nome para nossa janela agora. Clique na parte do fundo da área cinza da janela. Isso fará aparecer um contorno em toda a janela. Nas propriedades dela, localize **title** e mude seu valor para o nome da aplicação. No exemplo, usamos o nome **JCalc**.



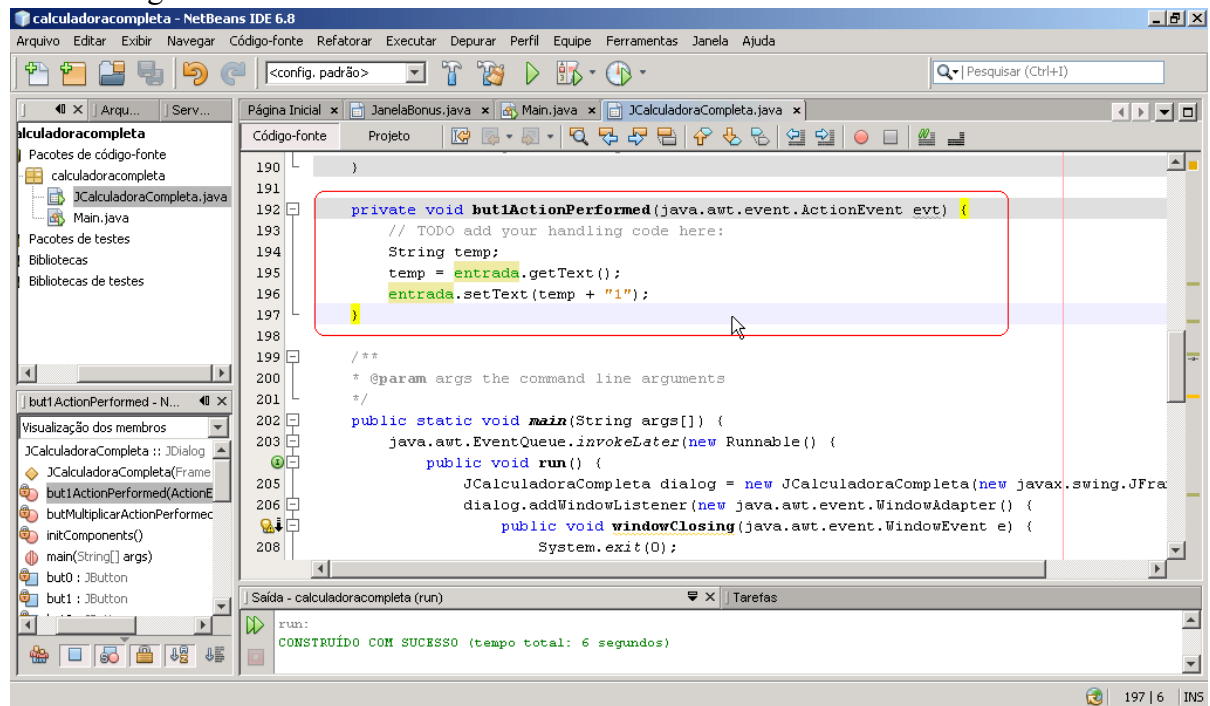
PASSO 13 - Se selecionarmos o botão "Código-fonte" como indicado abaixo, veremos o código de nossa classe **JanelaCalculadoraCompleta**. Observe que o NetBeans já preencheu um bocado de coisas para nós.



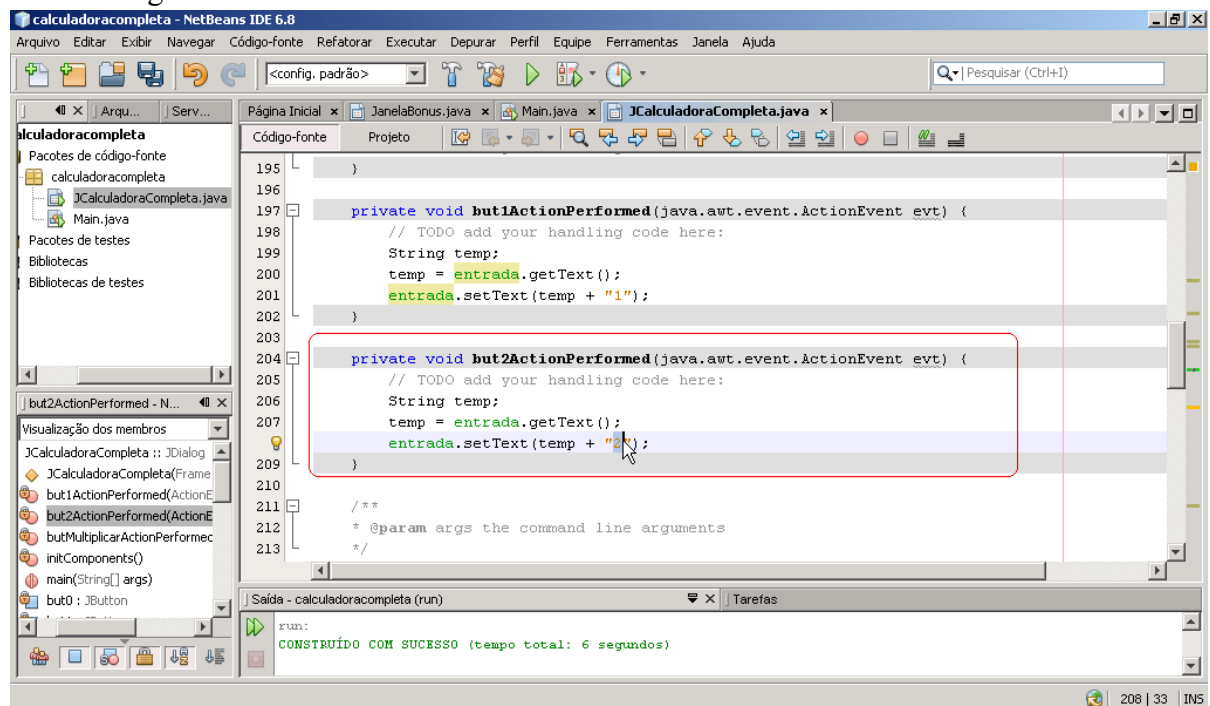
PASSO 14 - Vamos agora criar as variáveis de trabalho de nossa classe. Precisaremos de duas variáveis double: um para guardar os valores digitados pelo usuário (**ent**) e outra para guardar o resultado anterior (**res**). Declare-as como indicado na figura abaixo. Dentro do método construtor, **JanelaCalculadoraCompleta()**, inicialize as variáveis de trabalho com zero e preencha os textos iniciais dos campos: **resultado** deve receber o valor de **res** e **entrada** deve ser **limpado**. Observe o código na figura abaixo.



PASSO 15 - Agora vamos adicionar os códigos dos botões. Para isso, precisamos entrar no modo projeto (botão acima da área de código), onde aparece o layout da nossa janela. Agora, dê um **duplo-clique no botão com o número 1**. Isso vai mudar para o modo de "Código-fonte" já com um método para o evento clique no botão 1, chamado **but1ActionPerformed()**. Insira o código abaixo, que simplesmente pega o texto do campo entrada e "gruda" o número 1 nele.

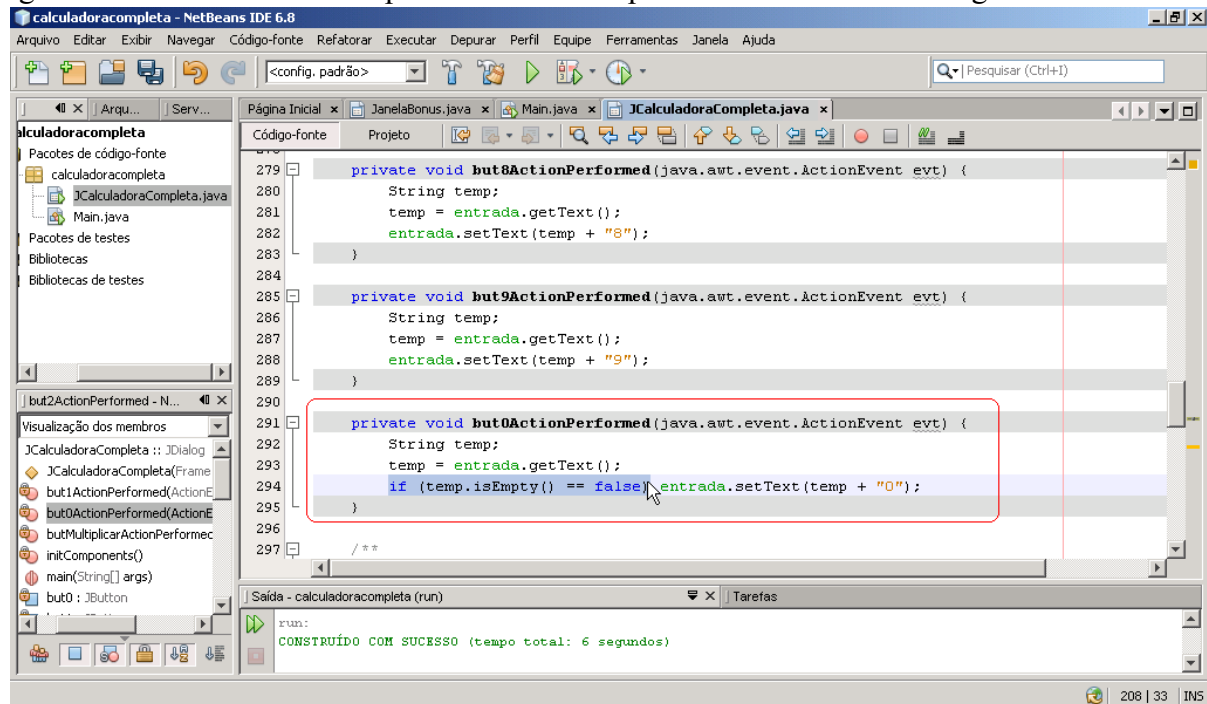


PASSO 16 - Volte para o modo **Projeto**, e dê um **duplo-clique no botão com o número 2**. Isso vai mudar para o modo de "Código-fonte", criando o método **but2ActionPerformed()**. Insira o código abaixo, que simplesmente pega o texto do campo entrada e "gruda" o número 2 nele.

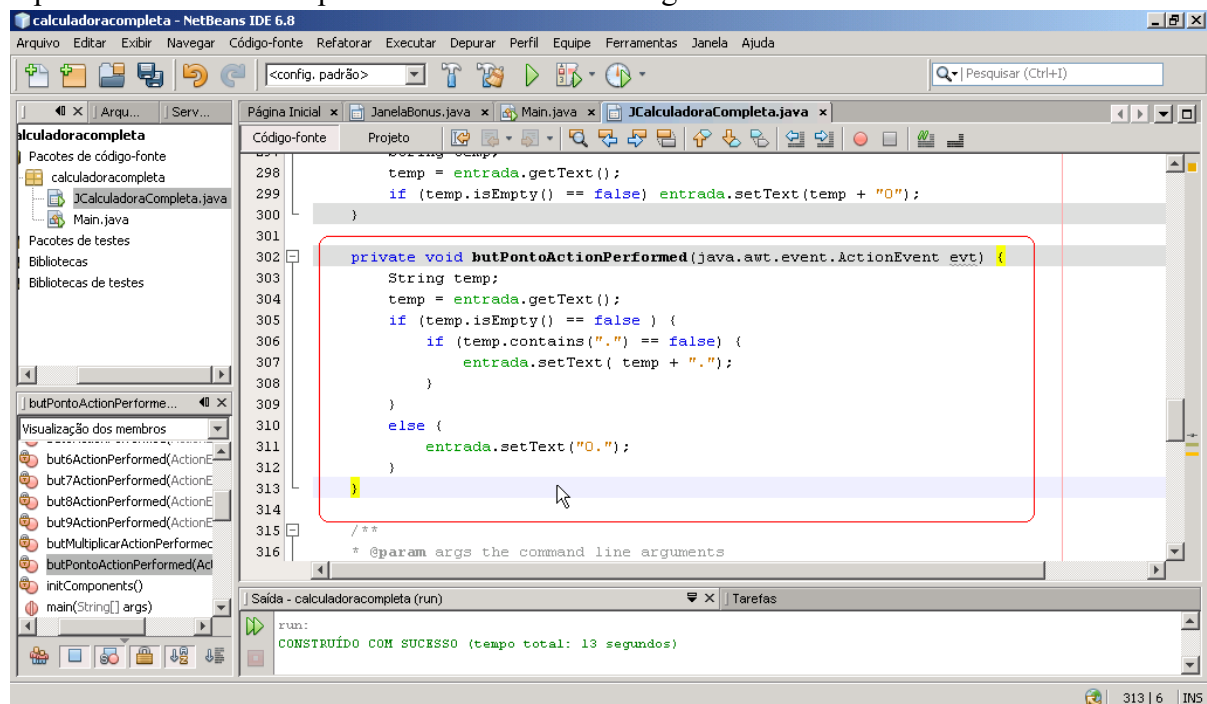


PASSO 17 - Repita o procedimento para os botões de 3 a 9.

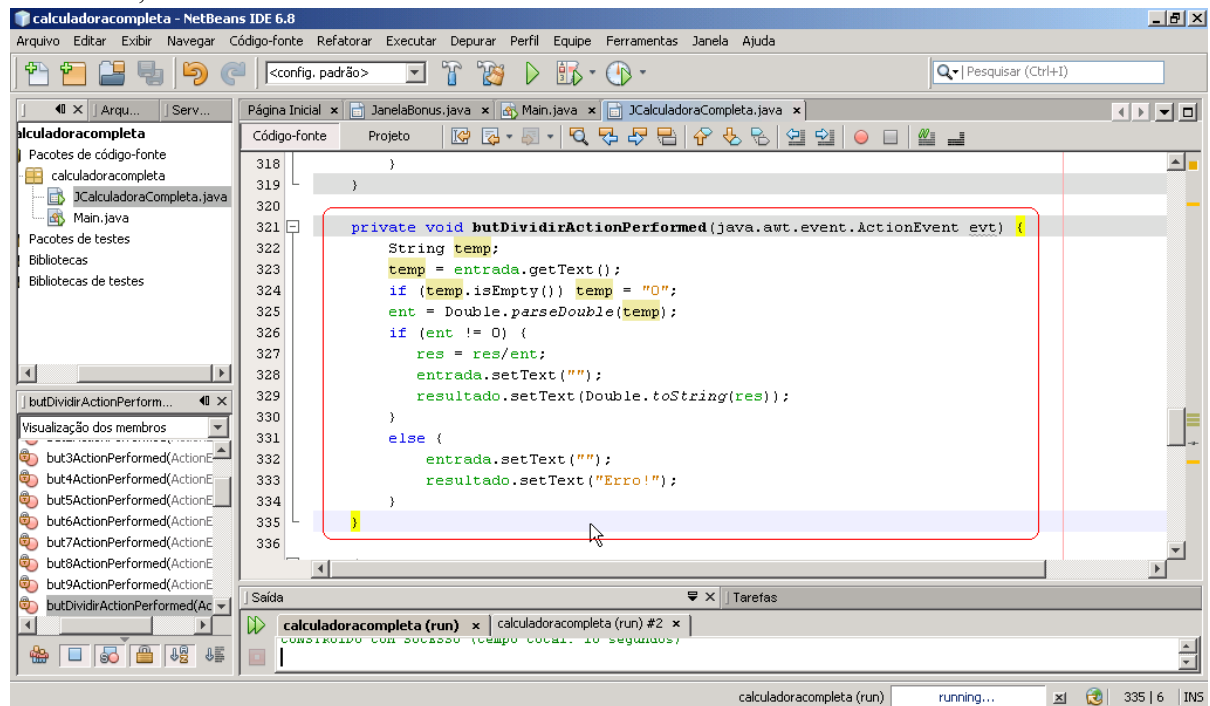
PASSO 18 - Para o botão 0, o código é um pouco diferente: o clique deve ser ignorado se não houver nada preenchido no campo entrada. Observe o código abaixo:



PASSO 19 - No caso do botão "." (ponto), temos de verificar o conteúdo de entrada. Se estiver vazio, acrescentamos "0." ao campo entrada. Se o campo não estiver vazio, precisamos verificar se o texto já contém ponto. Se contiver, devemos ignorar o clique, porque um número não pode ter dois pontos. Caso não exista um ponto ainda, acrescentamos o ponto ao texto do campo entrada. Observe o código abaixo:

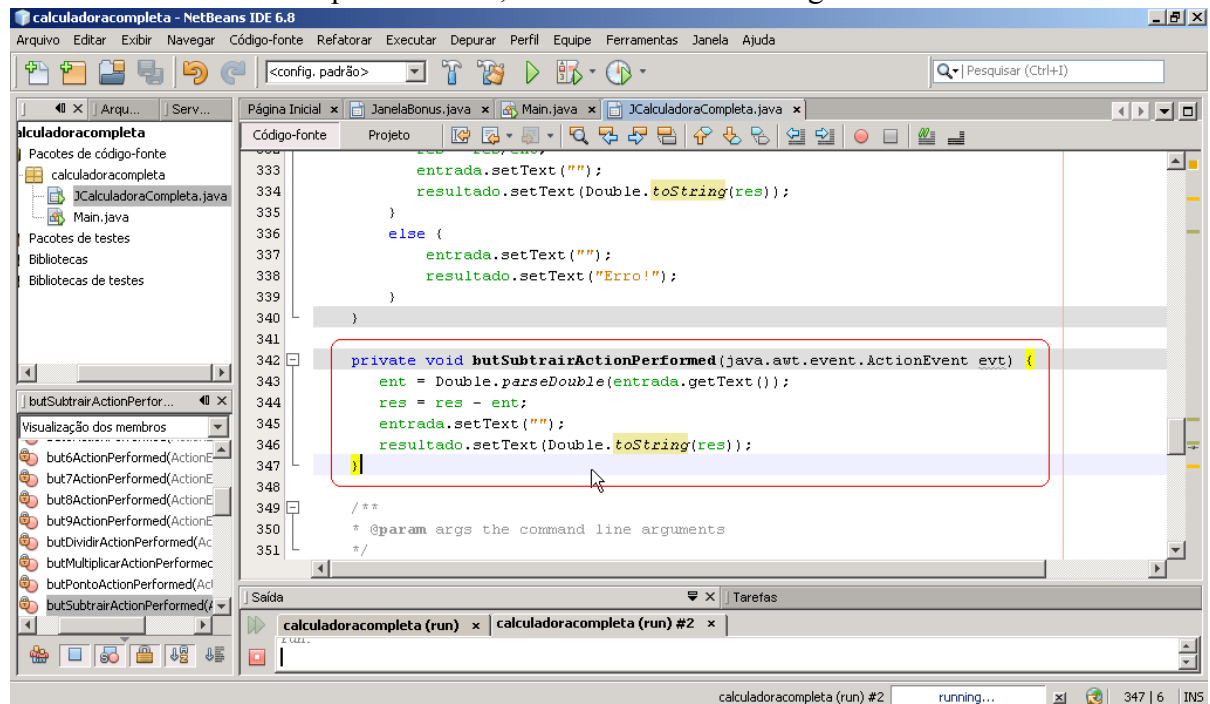


PASSO 20 - No caso do botão / (divisão), é preciso primeiramente verificar se o campo **entrada** está vazio. Se estiver, ele deve ser considerado zero. Em seguida, verifica-se se o campo de **entrada** tinha valor zero pois, nesta situação, não é possível realizar a divisão entre o valor do campo **resultado** e o valor do campo **entrada** e, portanto, deve-se indicar "Erro!" no campo **resultado**. Caso o valor do campo **entrada** seja aceitável, a divisão será executada, como indicado abaixo.



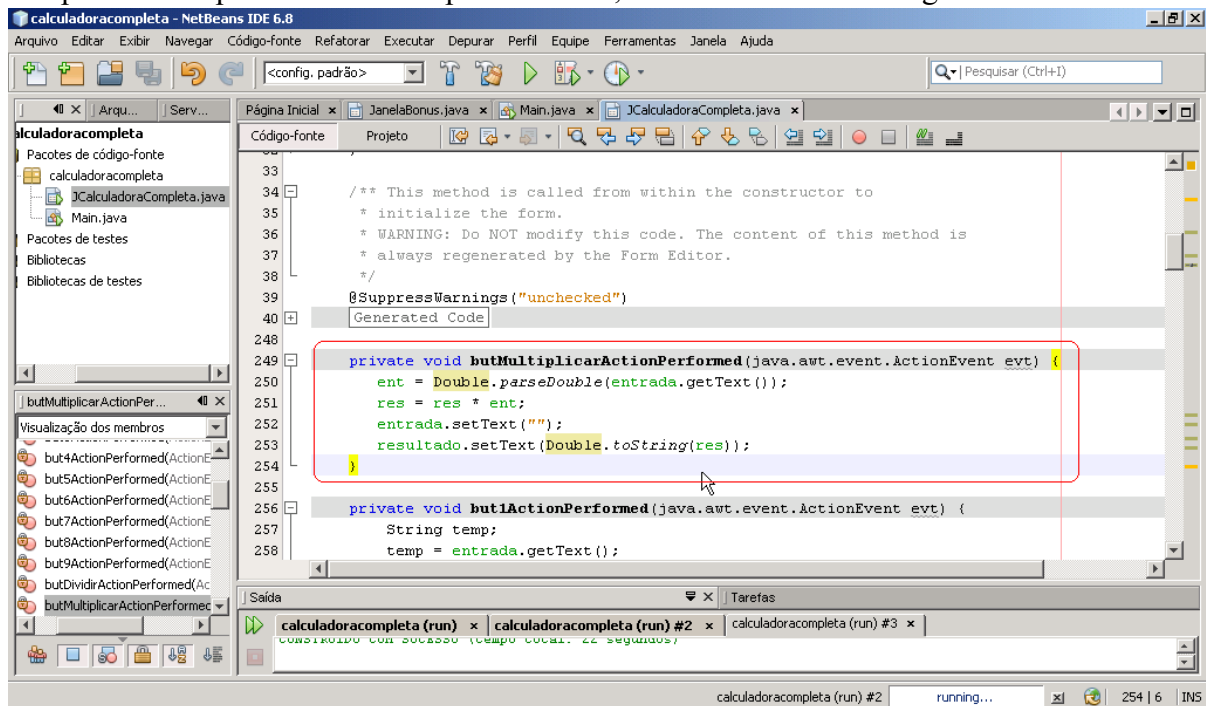
```
318     }
319 }
320
321 private void butDividirActionPerformed(java.awt.event.ActionEvent evt) {
322     String temp;
323     temp = entrada.getText();
324     if (temp.isEmpty()) temp = "0";
325     ent = Double.parseDouble(temp);
326     if (ent != 0) {
327         res = res/ent;
328         entrada.setText("");
329         resultado.setText(Double.toString(res));
330     }
331     else {
332         entrada.setText("");
333         resultado.setText("Erro!");
334     }
335 }
336
```

PASSO 21 - No caso do botão - (subtração), deve-se subtrair o valor do campo **entrada** do valor do campo **resultado**, como indicado no código abaixo.



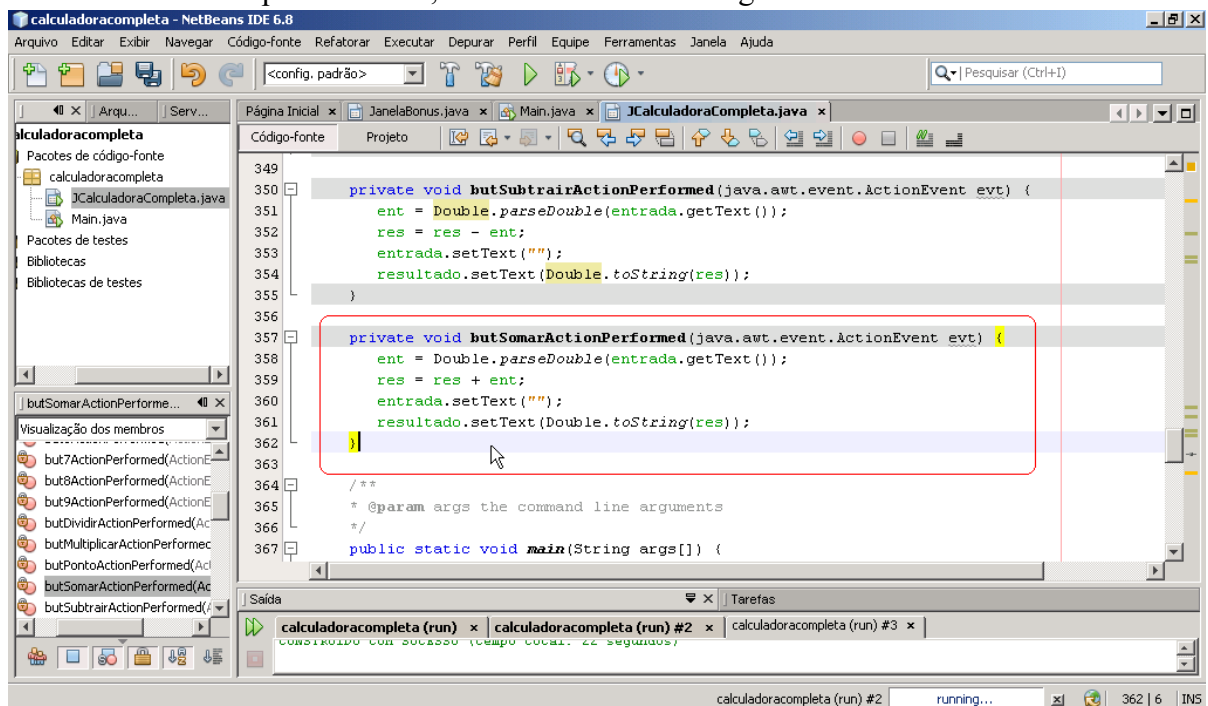
```
333     entrada.setText("");
334     resultado.setText(Double.toString(res));
335 }
336 else {
337     entrada.setText("");
338     resultado.setText("Erro!");
339 }
340 }
341
342 private void butSubtrairActionPerformed(java.awt.event.ActionEvent evt) {
343     ent = Double.parseDouble(entrada.getText());
344     res = res - ent;
345     entrada.setText("");
346     resultado.setText(Double.toString(res));
347 }
348
349 /**
350  * @param args the command line arguments
351  */
```

PASSO 22 - No caso do botão * (multiplicação), deve-se multiplicar o valor do campo **entrada** pelo valor do campo **resultado**, como indicado no código abaixo.



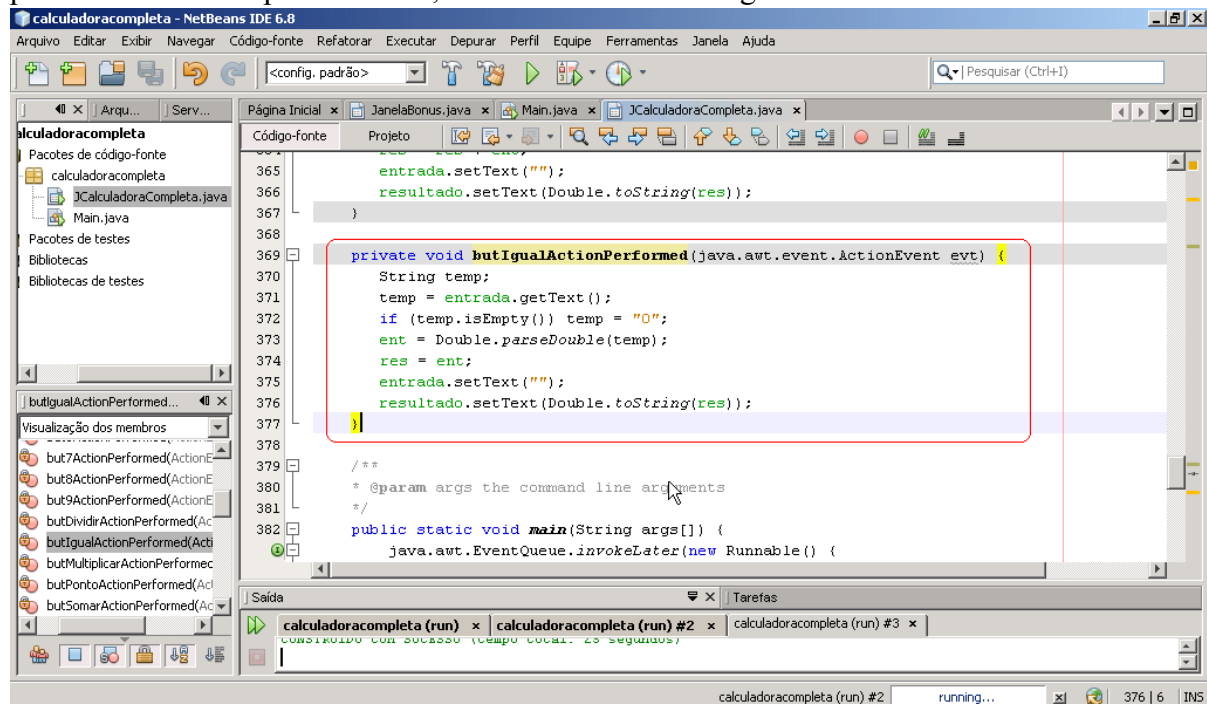
```
33
34  /** This method is called from within the constructor to
35   * initialize the form.
36   * WARNING: Do NOT modify this code. The content of this method is
37   * always regenerated by the Form Editor.
38   */
39  @SuppressWarnings("unchecked")
40  Generated Code
248
249
250  private void butMultiplicacaoActionPerformed(java.awt.event.ActionEvent evt) {
251      ent = Double.parseDouble(entrada.getText());
252      res = res * ent;
253      entrada.setText("");
254      resultado.setText(Double.toString(res));
255  }
256
257  private void but1ActionPerformed(java.awt.event.ActionEvent evt) {
258      String temp;
259      temp = entrada.getText();
```

PASSO 23 - No caso do botão + (soma), deve-se somar o valor do campo **entrada** com o valor do campo **resultado**, como indicado no código abaixo.

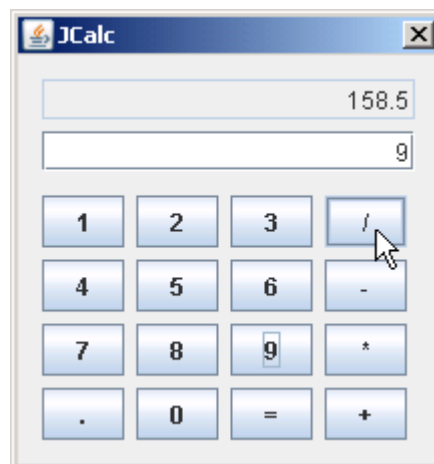


```
349
350  private void butSubtrairActionPerformed(java.awt.event.ActionEvent evt) {
351      ent = Double.parseDouble(entrada.getText());
352      res = res - ent;
353      entrada.setText("");
354      resultado.setText(Double.toString(res));
355  }
356
357  private void butSomarActionPerformed(java.awt.event.ActionEvent evt) {
358      ent = Double.parseDouble(entrada.getText());
359      res = res + ent;
360      entrada.setText("");
361      resultado.setText(Double.toString(res));
362  }
363
364  /**
365   * @param args the command line arguments
366   */
367  public static void main(String args[]) {
```

PASSO 24 - No caso do botão = (igual), deve-se copiar o valor do campo **entrada** para o valor do campo **resultado**, como indicado no código abaixo.



PASSO 25 - Se tudo foi feito como indicado, o resultado deve ser uma calculadora operante, como a indicada na figura abaixo.



EXERCÍCIOS (OPCIONAIS!)

PASSO 26 - Há várias funções faltantes nesta calculadora, como um botão "+/-", para inverter o sinal do resultado, uma tecla "C" para limpar o resultado, uma tecla "x²" para calcular o quadrado do resultado, e uma tecla "x^{1/2}", para calcular a raiz quadrada do resultado. Será que você consegue implementá-los?

PASSO 27 - Esta calculadora funciona no formato das calculadoras Hewlett Packard, isto é, em notação reversa (digita-se os números e, depois, a operação desejada). Você conseguiria escrever uma que funcione como as calculadoras usuais?