

Java

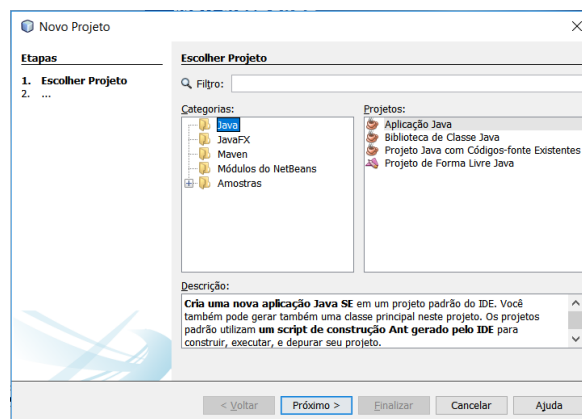
Neste exercício, faremos um cálculo da idade de uma pessoa, utilizando um ano estático, ou seja, digitado pelo programador.



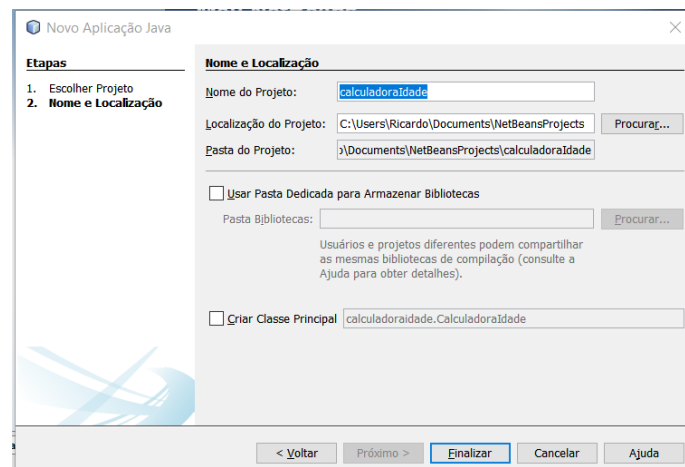
Abra o NetBeans.



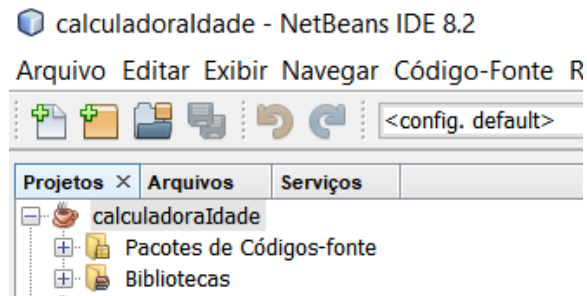
Clique no botão Novo Projeto e mantenha selecionado Categoria Java, Projeto aplicação Java e clique em Próximo.



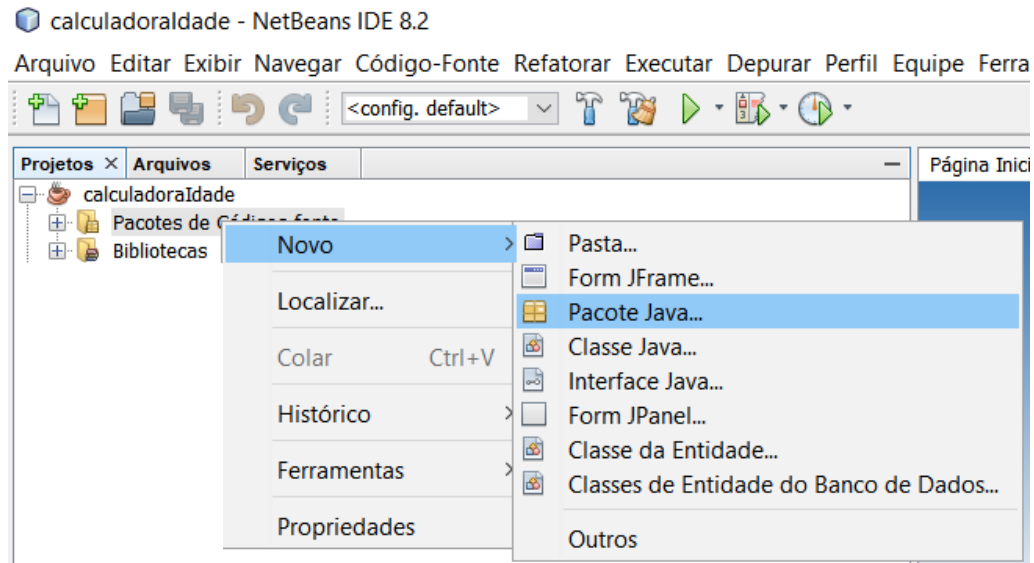
Em “Nome do Projeto”, digite “calculadoraIdade”, desmarque a caixa de seleção “Criar Classe Principal” e clique no botão Finalizar.



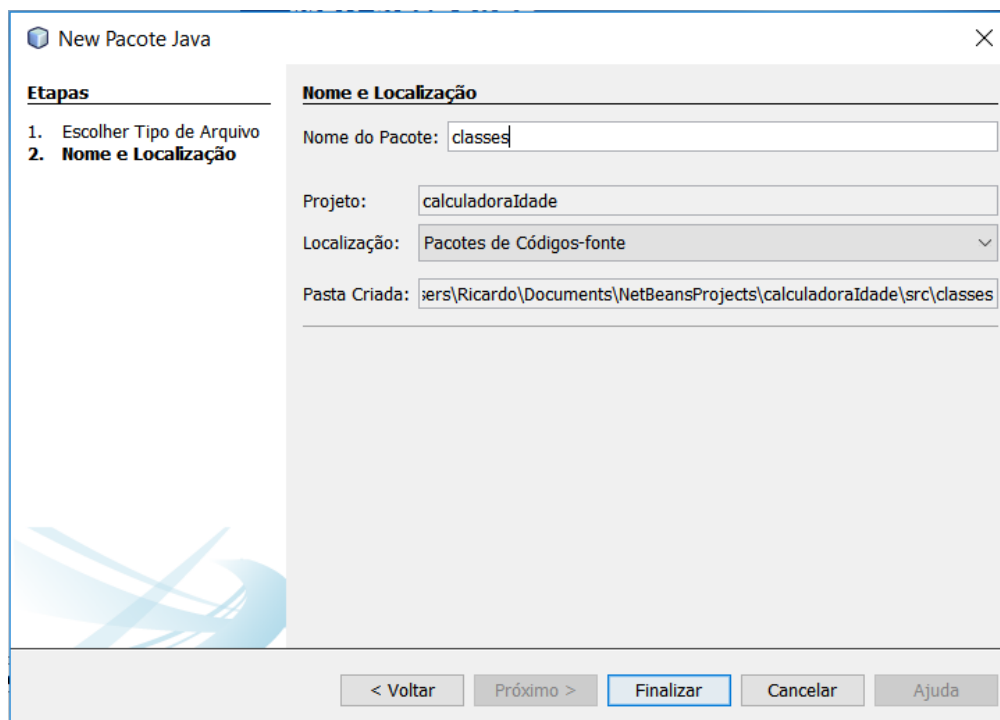
O projeto é criado vazio, sem nenhum pacote.



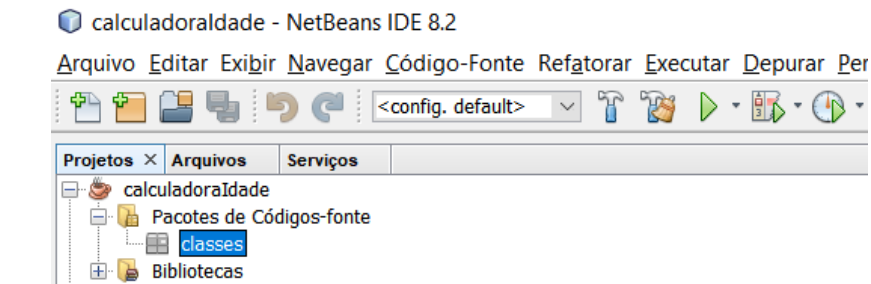
Clique com o botão direito do mouse para criar um **Novo => Pacote Java**.



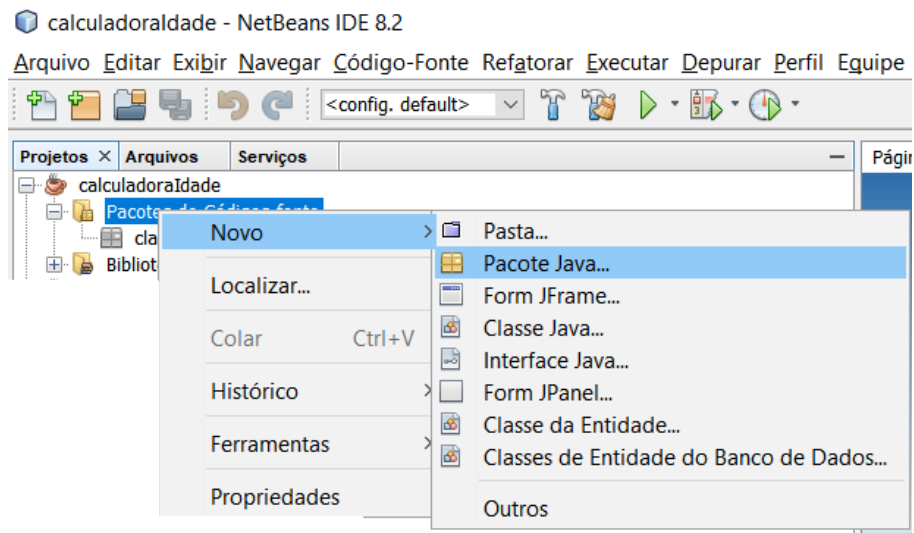
Nomeie o projeto como “classes” e clique no botão Finalizar.



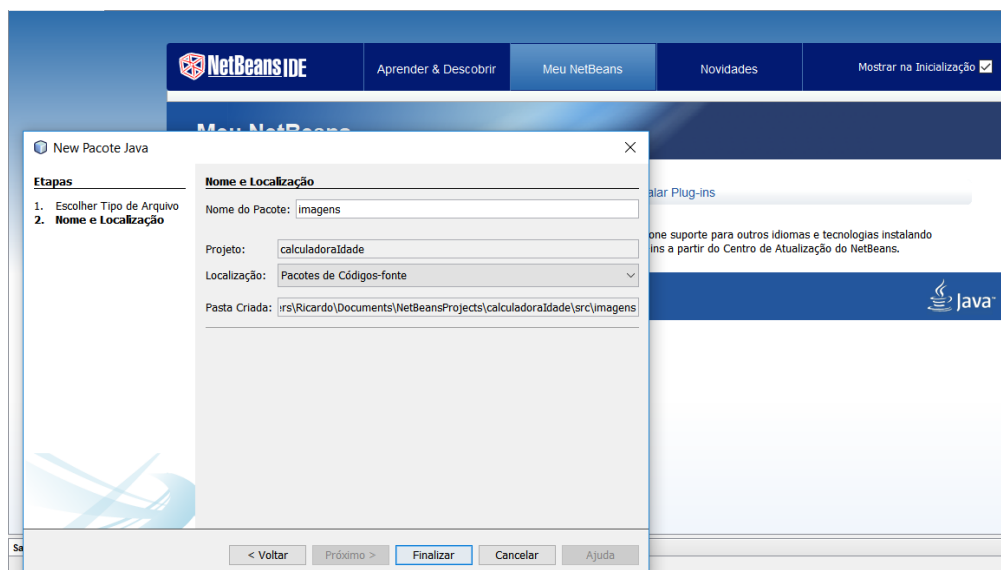
Após clicar em Finalizar o pacote “classes” é exibido no painel.



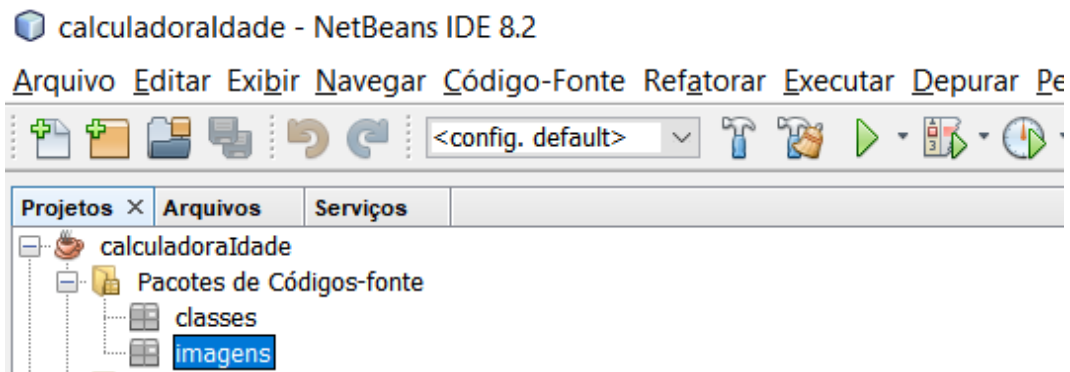
Crie o pacote de imagens clicando com o botão direito do mouse em **Pacotes de Códigos-fonte => Novo => Pacote Java**.



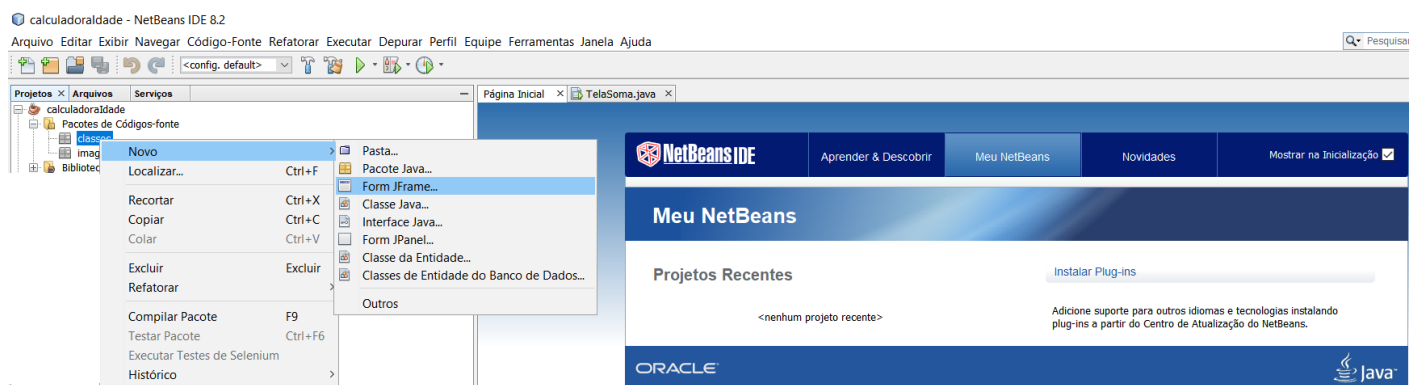
Nomeie o Pacote como “imagens”.



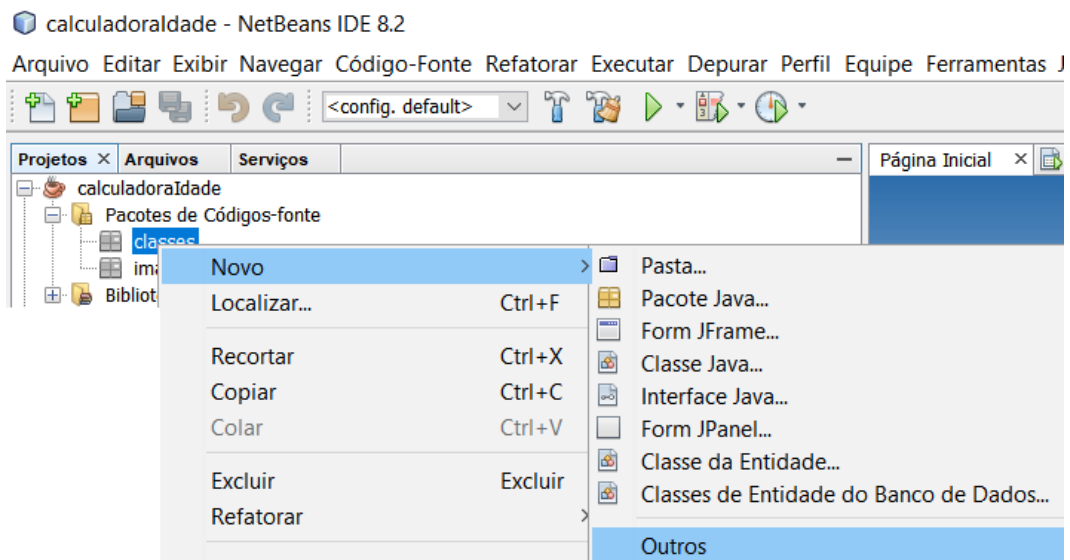
O pacote “imagens” será exibido no painel, juntamente com o pacote “classes” criado anteriormente.



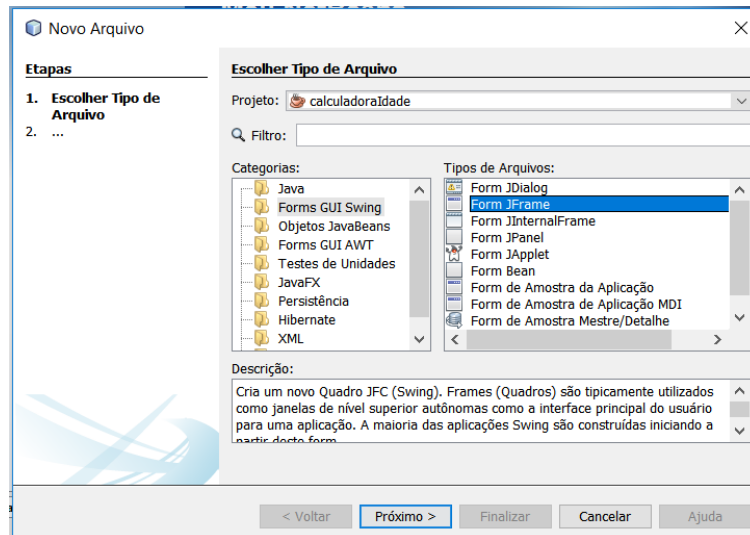
Clique com o botão direito em “imagens” => Novo => Form JFrame.



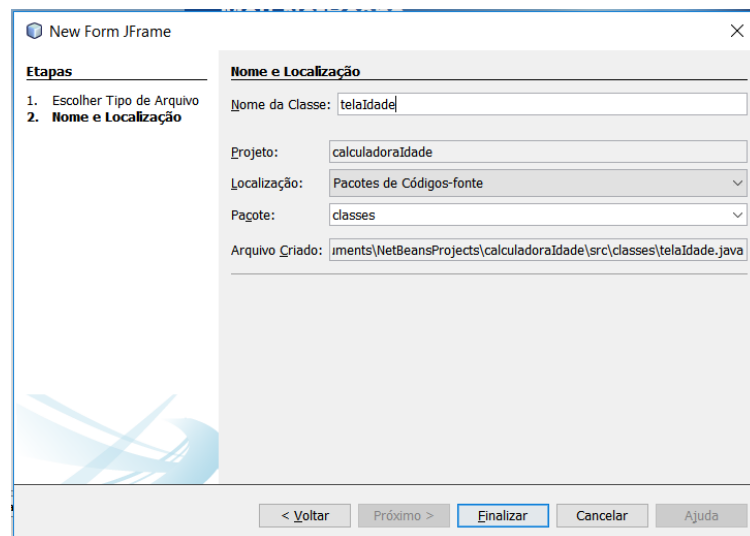
Caso não haja esta opção, clique com o botão direito, **Novo => Outros**.



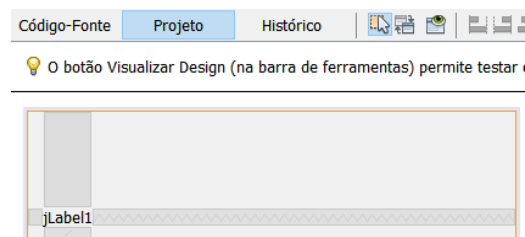
Selecione a categoria **Forms GUI Swing**. Em Tipo de Arquivos, selecione **Form JFrame** e após clique em Próximo.



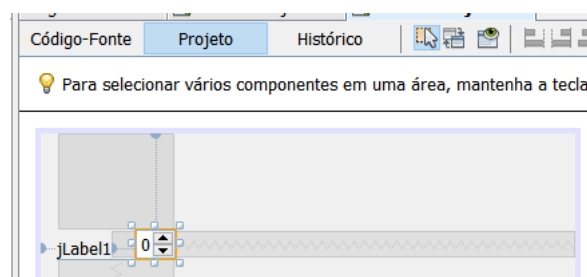
Nomeie como “telaldade” e clique no botão Finalizar.



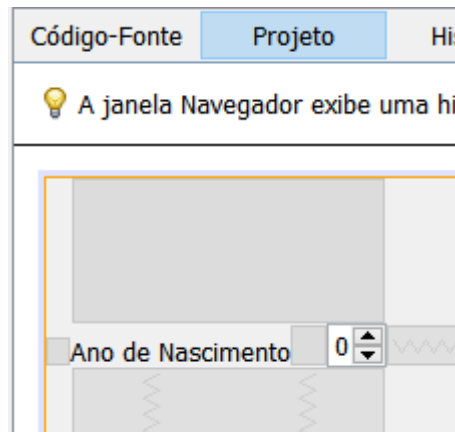
Adicione um Label.



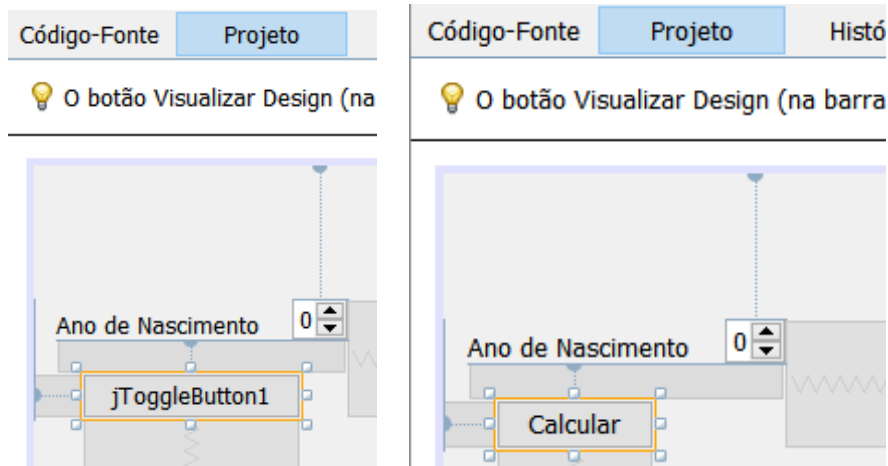
Adicione um Controle Giratório. Com o Controle Giratório somente será possível inserir valores numéricos e estes, não precisam ser digitados; bastando clicar nas setas para aumentar ou diminuir o valor.



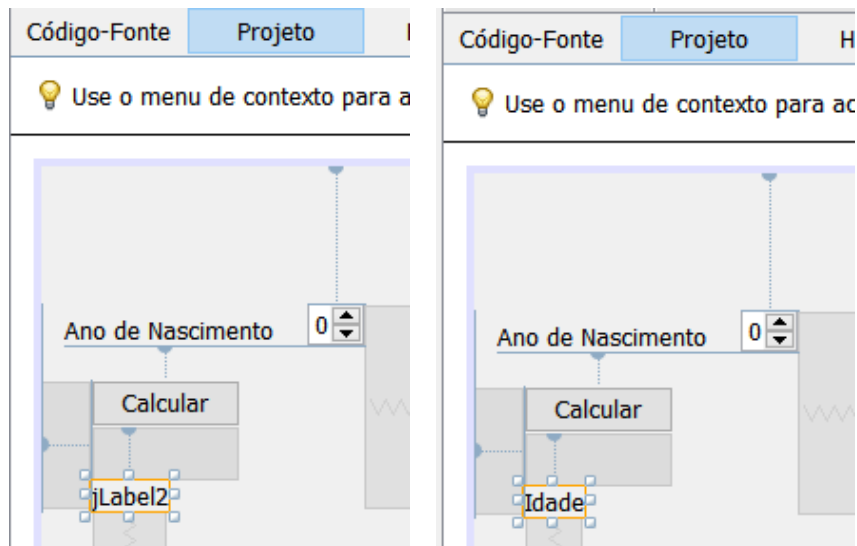
Altere o nome do Label de “jLabel1” para “Ano de Nascimento”.



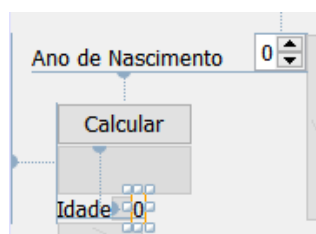
Crie um Botão e altere o texto de “jToggleButton1” para “Calcular”.



Adicione um novo Label e renomeie de “jLabel2” para “Idade”.

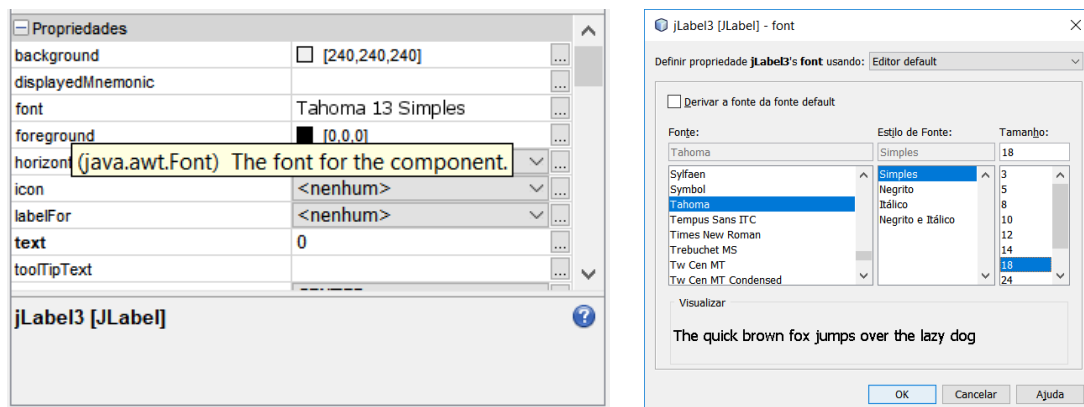


Adicione mais um Label, desta vez renomeando para o valor “0”.

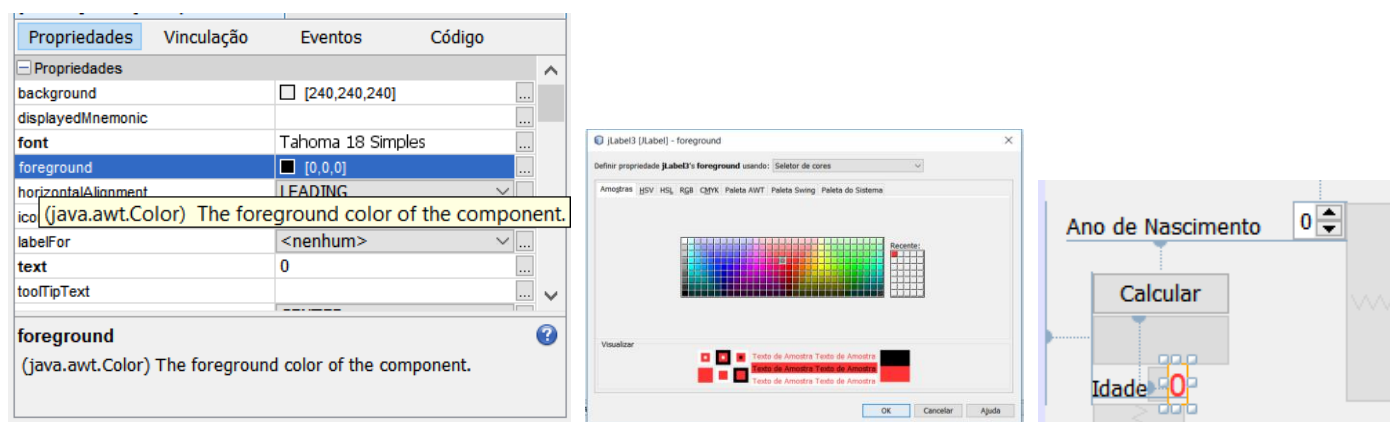


Lembrando...

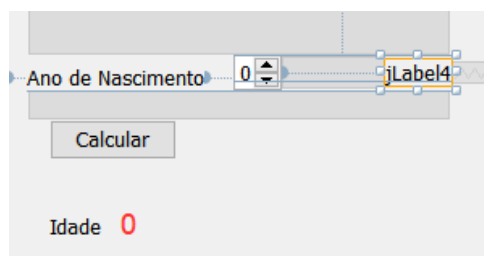
Para alterar o Tamanho da Fonte, clique na reticências (...) da Propriedade **“font”**.



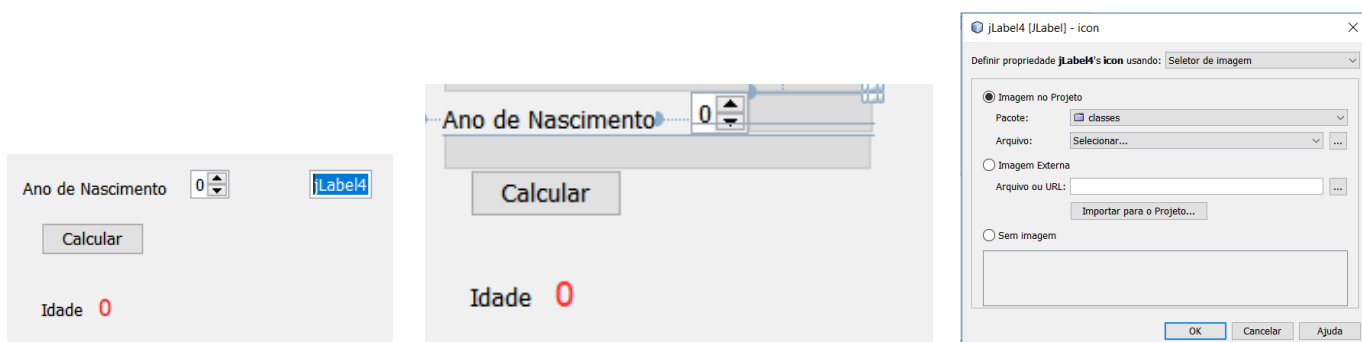
Para alterar a cor, clique na reticências (...) de **“foreground”**.



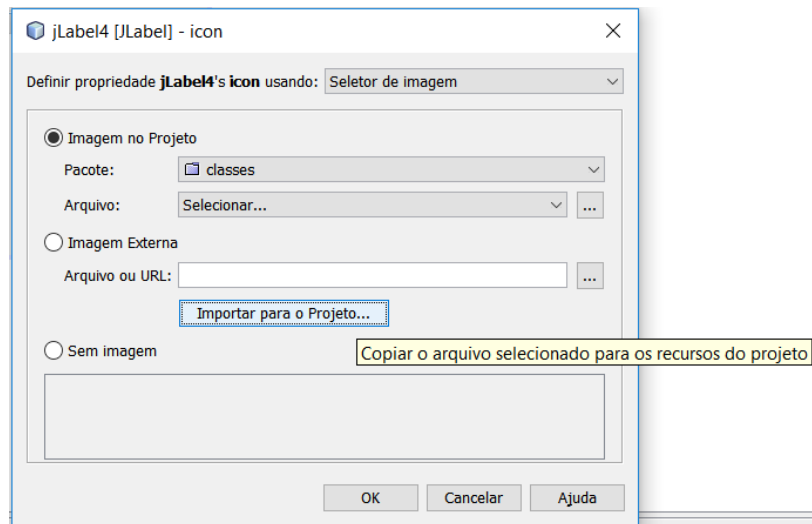
Adicione o Label que irá conter a imagem.



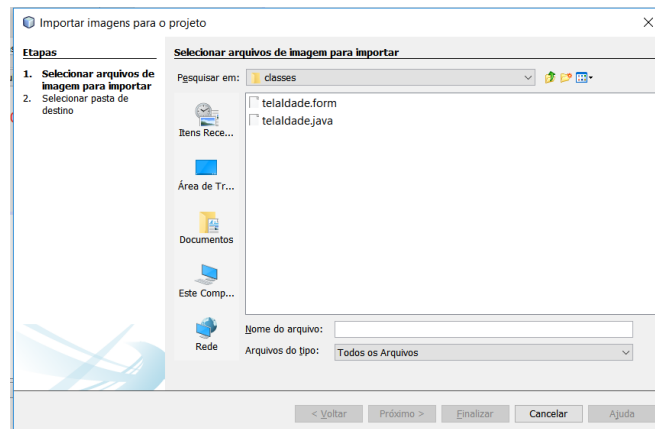
Para inserir a imagem, apague o texto, de maneira que fique vazio e clique na reticências (...) da Propriedade **“icon”**.



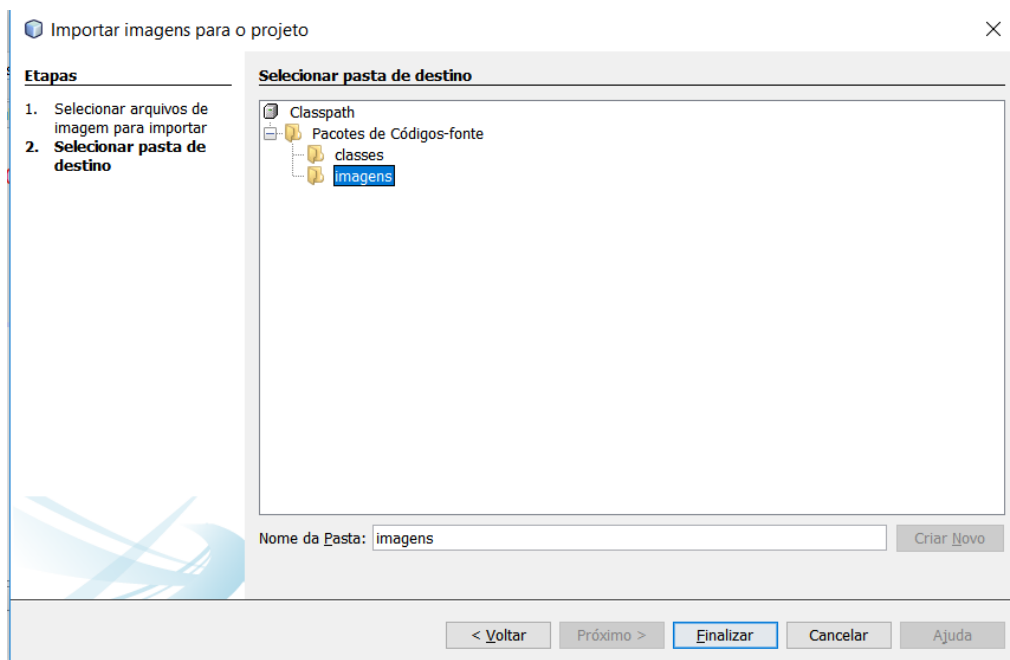
Clique em “Importar para o Projeto...”.



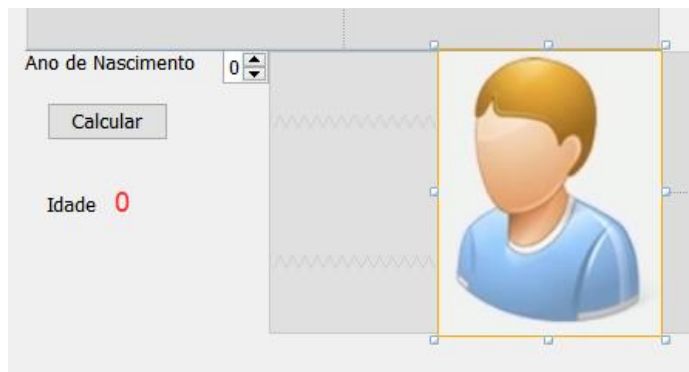
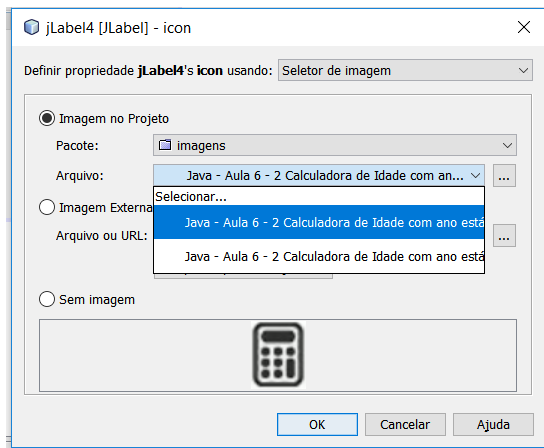
Selecione as imagens e clique em Próximo.



Carregue os arquivos em “imagens”, através da seleção da “pasta de destino” e clique em Finalizar.

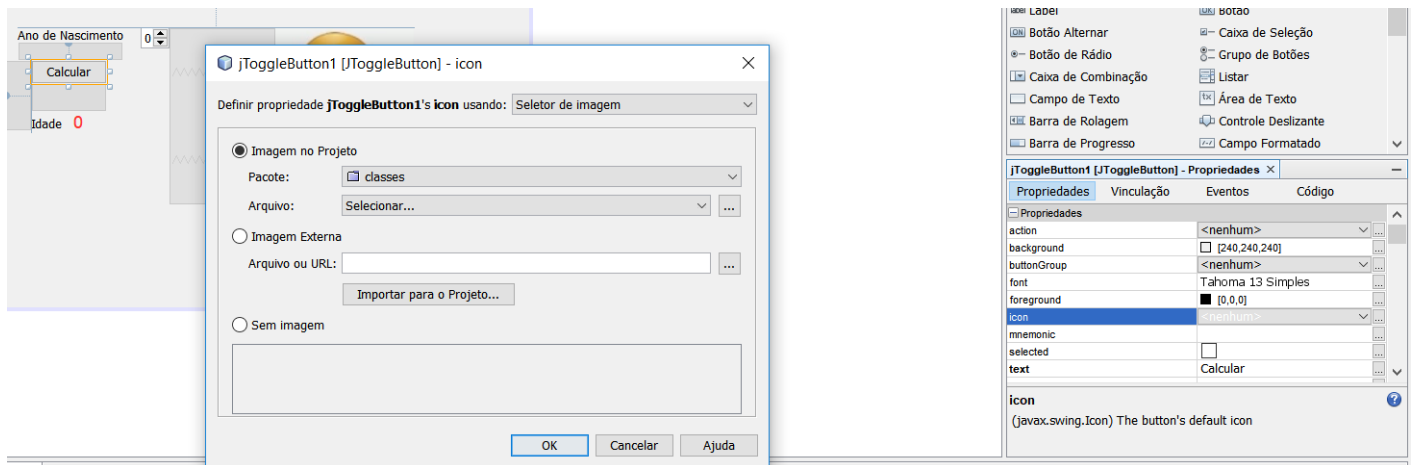


Selecione o ícone do usuário e clique em Ok.

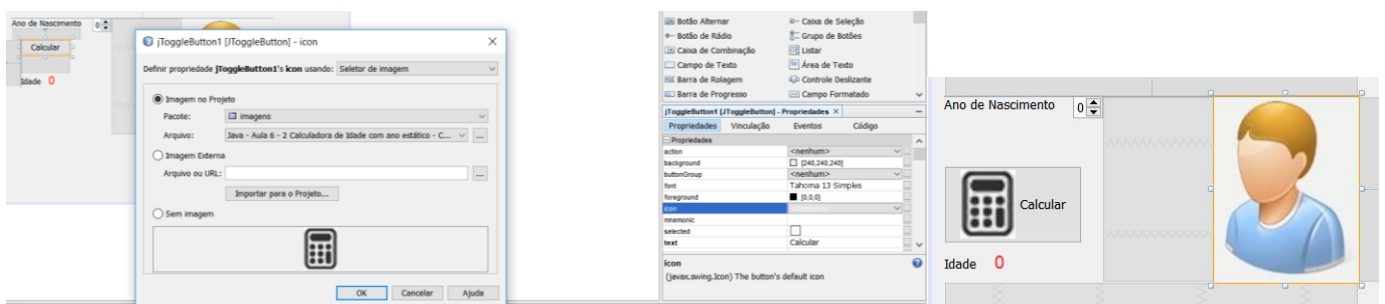


Adicione o desenho da calculadora dentro do botão de calcular, juntamente com o texto calcular.

Selecione o botão Calcular, clique na reticencias (...) de "icon", uma nova janela será exibida.

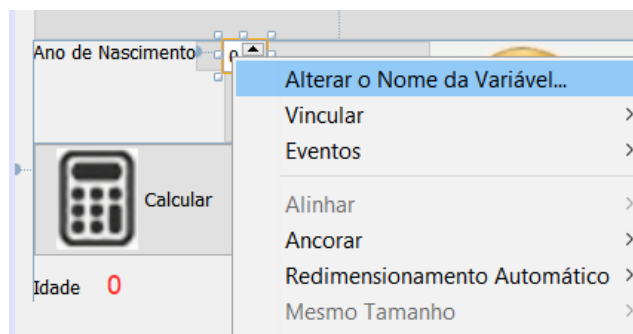


Em Pacotes, selecione "imagens" e em Arquivos, selecione a calculadora. Após, clique no botão Ok.

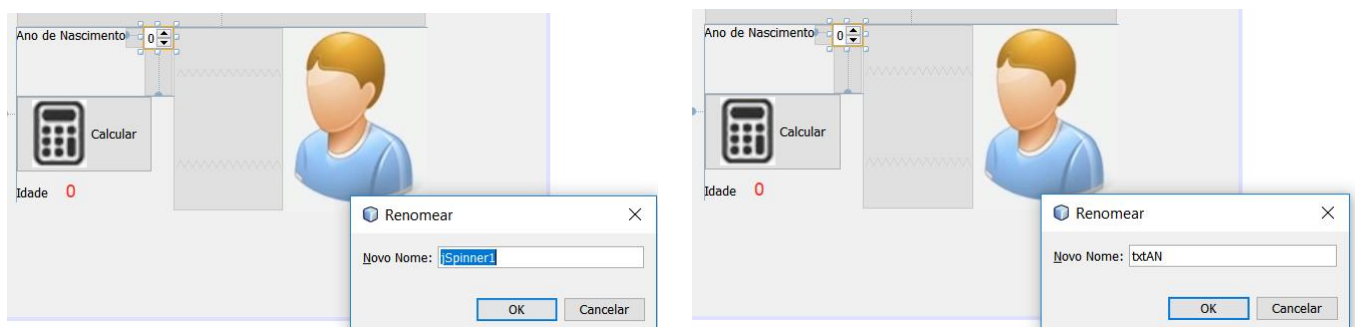


Nomeando as Variáveis

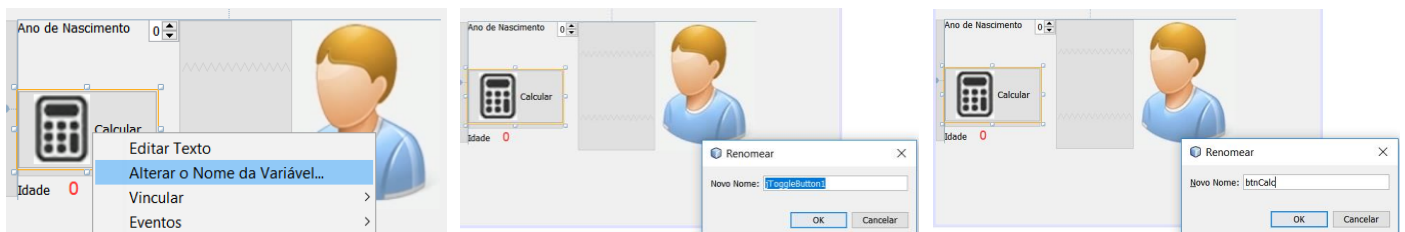
Clique no objeto Controle Giratório “0” com o botão direito do mouse => Clique em “**Alterar o Nome da Variável**”.



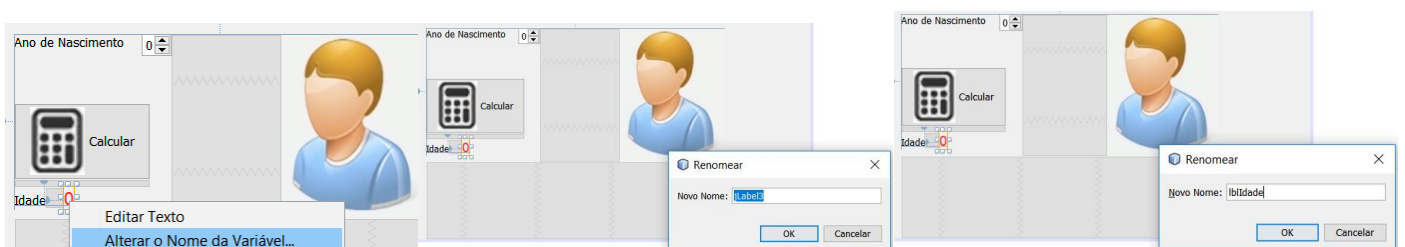
Altere o nome “jSpinner1” para “txtAN” e clique no botão Ok.



Altere o nome da variável do botão Calcular de “jToggleButton1” para “btnCalc” e clique no botão Ok.



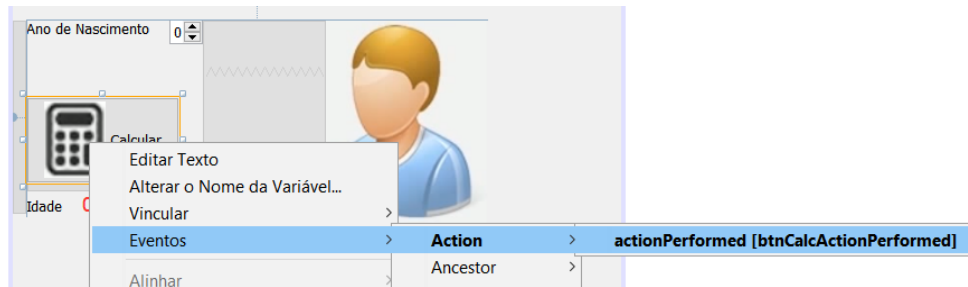
No valor da idade (0), clique com o botão direito do mouse => Clique em “Alterar o Nome da Variável...” e altere o nome de “jLabel3” para “lblIdade” e clique em Ok.



Programação em Java

Agora vamos iniciar a programação em Java.

Clique com o botão direito na calculadora => **Eventos => Action => actionPerformed [btnCalcActionPerformed]**.



Serão exibidos os códigos, conforme imagem.

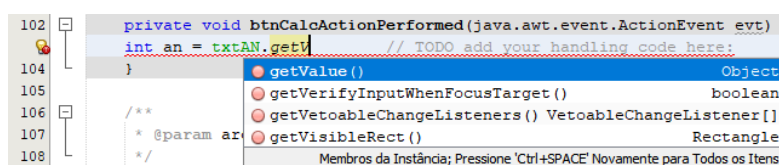
```
Código-Fonte  Projeto  Histórico
Generated Code
101
102 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {
103     // TODO add your handling code here:
104 }
105
106 /**
107  * @param args the command line arguments
108  */
109 public static void main(String args[]) {
110     /* Set the Nimbus look and feel */
111     Look and feel setting code (optional)
112
113     /* Create and display the form */
114     java.awt.EventQueue.invokeLater(new Runnable() {
115         public void run() {
116             new telaIdade().setVisible(true);
117         }
118     });
119 }
120
121 // Variables declaration - do not modify
122 private javax.swing.JToggleButton btnCalc;
123 private javax.swing.JLabel jLabel1;
124 private javax.swing.JLabel jLabel2;
125 private javax.swing.JLabel jLabel4;
126 private javax.swing.JLabel lblIdade;
127 private javax.swing.JSpinner txtAN;
128 // End of variables declaration
129 }
```

Crie uma variável inteira para receber o ano de nascimento (**int na =**).

```
102 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {
103     int an = // TODO add your handling code here:
104 }
```

Obs: Quanto utilizar os Campos de Texto é `getText`, mas **ao utilizarmos o Controle Giratório é `getValue`**.

Desta forma, o ano de nascimento virá diretamente de **`txtAN.getValue()`**; ficando da seguinte forma: **Int na = `txtAN.getValue()`**.



```
102 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {
103     int an = txtAN.getValue() // TODO add your handling code here:
104 }
```

É exibida uma incompatibilidade no código pois não é um valor inteiro e sim um tipo, por isso a incompatibilidade de tipo, já que ele é do tipo objeto. Para início a solução, adicione “**toString()**” ao final.

```
102 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
104     int an = txtAN.getValue().to  
    }  
    toString() String
```

```
102 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
104     int an = txtAN.getValue().toString() // TODO add your handling  
    }
```

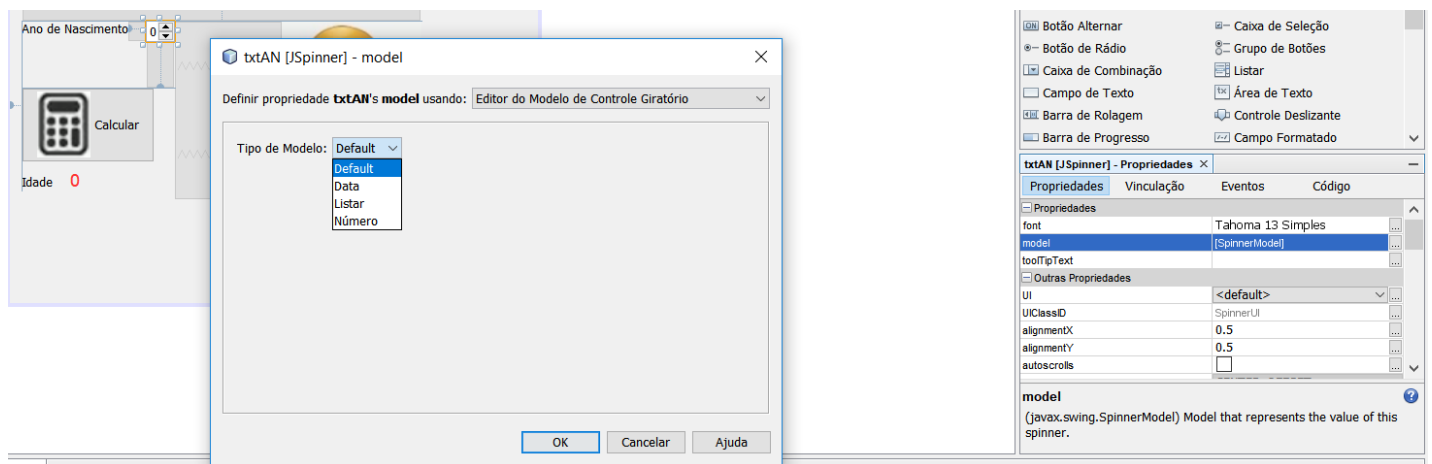
Ao colocar “**toString()**” ele é uma string. É necessário converter essa string para número, adicionando “**Integer.parseInt()**”.

```
102 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
104     int an = Integer.parseInt(txtAN.getValue().toString()) // TODO add you  
    }  
    /**  
    * @param args  
    */  
    parseInt(String string) int  
    parseInt(String string, int i) int  
    parseInt(String string) int  
    parseInt(String string, int i) int
```

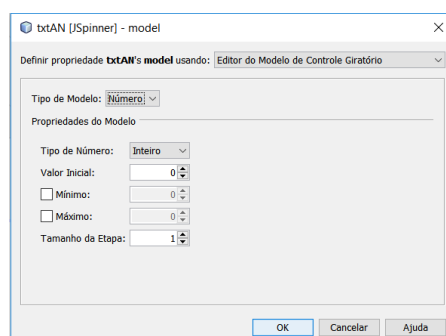
```
102 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
103     int an = Integer.parseInt(txtAN.getValue().toString());  
104 }
```

Com isso, já temos o ano de nascimento.

Clique na reticencias (...) da Propriedade **model**.



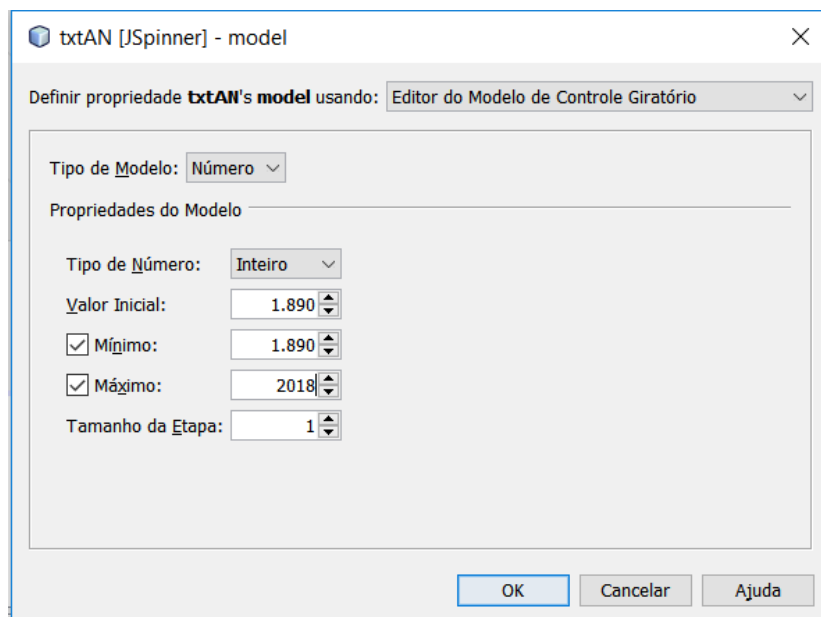
Neste exemplo, selecione Número.



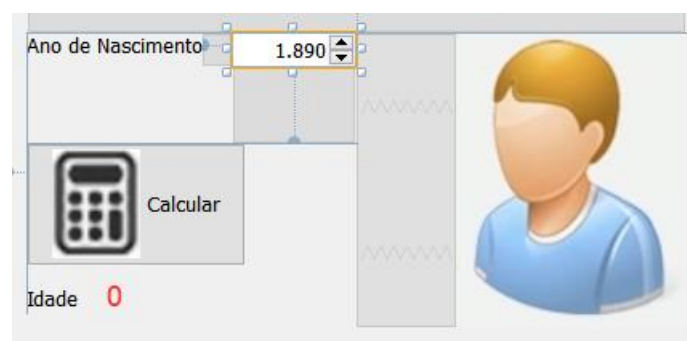
É possível definir um Valor Inicial, por exemplo, 1890.

Também é possível informar o valor Mínimo e o valor Máximo permitido no campo. Por exemplo, valor Mínimo 1890, pois dificilmente alguém nasceu antes, e valor máximo 2018, pois é do ano atual.

Em Tamanho da Etapa é informado quanto será acrescentado cada vez que a seta para aumentar ou diminuir for clicada.



Após selecionar as opções desejadas, clique no botão Ok.



Para calcular a idade, crie uma variável (id).

```
104 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
105     int an = Integer.parseInt(txtAN.getValue().toString());  
106     int id =  
107     // TODO add your handling code here:  
108 }
```

Esta variável idade (id) receberá o ano atual subtraído do ano de nascimento (an).

```
104 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
105     int an = Integer.parseInt(txtAN.getValue().toString());  
106     int id = 2018 - an;  
107     // TODO add your handling code here:  
108 }
```

Para mostrarmos a idade na tela, adicione o código `lblIdade.setText`, conforme imagem.

```
104 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
105     int an = Integer.parseInt(txtAN.getValue().toString());  
106     int id = 2018 - an;  
107     lblIdade.setT  
108     // TODO ad  
109 }  
110
```

setText(String string)	void
setToolTipText(String string)	void
setTransferHandler(TransferHandler th)	void

Mas para mostrar na tela, será ainda necessário converter a idade para string da seguinte forma:

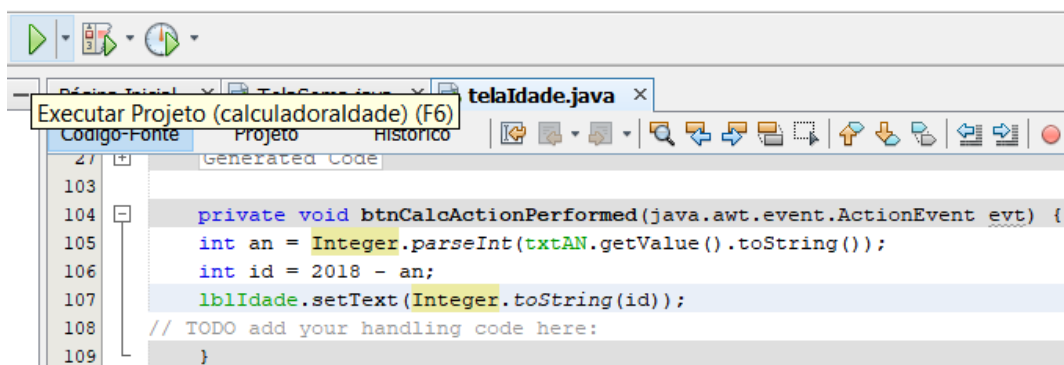
Sobrescreva digitando “id”, de idade.

```
104 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
105     int an = Integer.parseInt(txtAN.getValue().toString());  
106     int id = 2018 - an;  
107     lblIdade.setText(id);  
108     // TODO add your handling code here:  
109 }
```

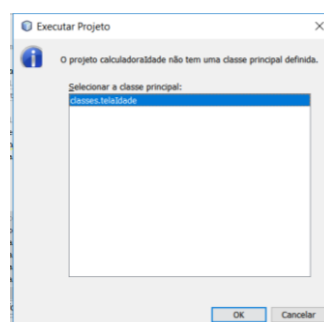
Converta a idade para string utilizando: **Integer.toString**, conforme imagem.

```
104 private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
105     int an = Integer.parseInt(txtAN.getValue().toString());  
106     int id = 2018 - an;  
107     lblIdade.setText(Integer.toString(id));  
108     // TODO add your handling code here:  
109 }
```

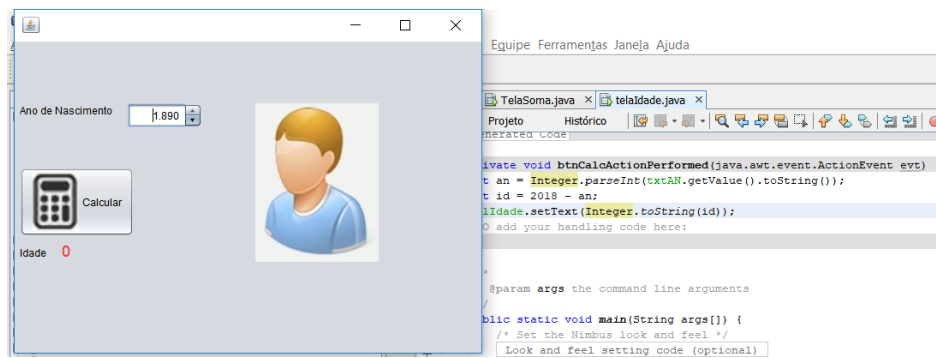
Agora o programa já está pronto, para finalizar aperte o Play.



O NetBeans questiona qual é a classe principal. Aperte Ok.



A janela será exibida.



Realize os testes.

