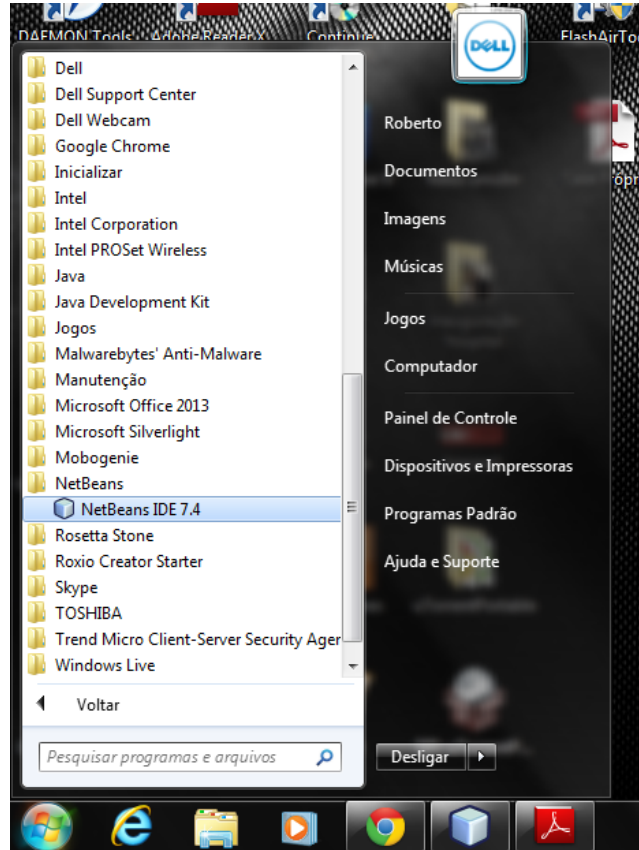


Programação Orientada a Objetos

Java - Aula 5

Criando um Form JFrame

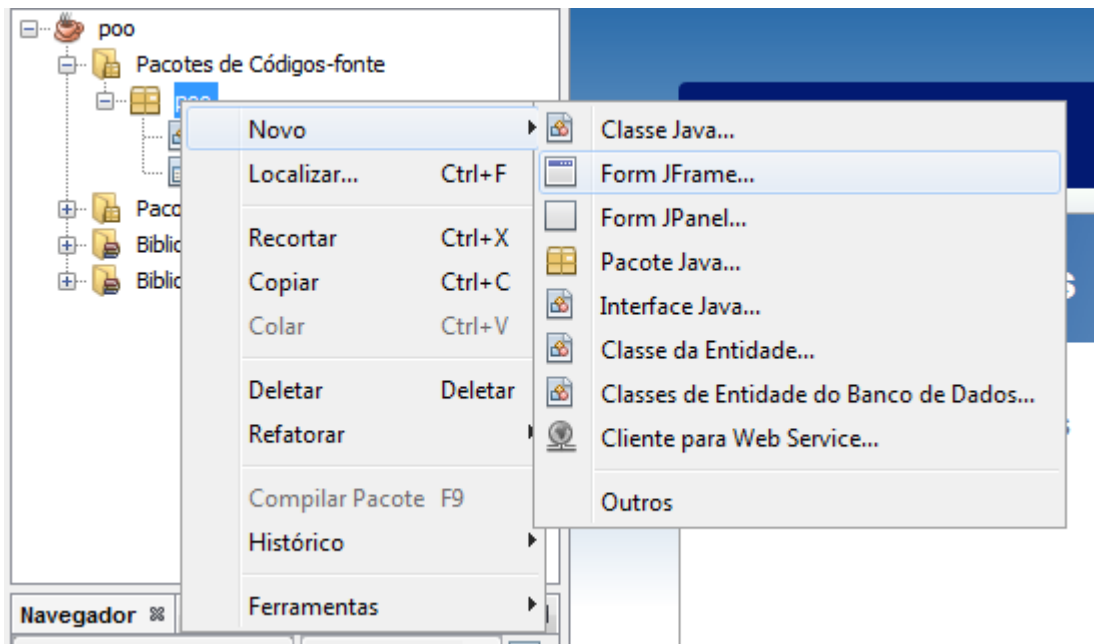
Abra o NetBeans:



Criando um Form JFrame

Se necessário crie um novo projeto em seguida dentro do pacote clique com o botão direito, selecione Novo -> Form JFrame.

Atribua Formulário para seu nome.



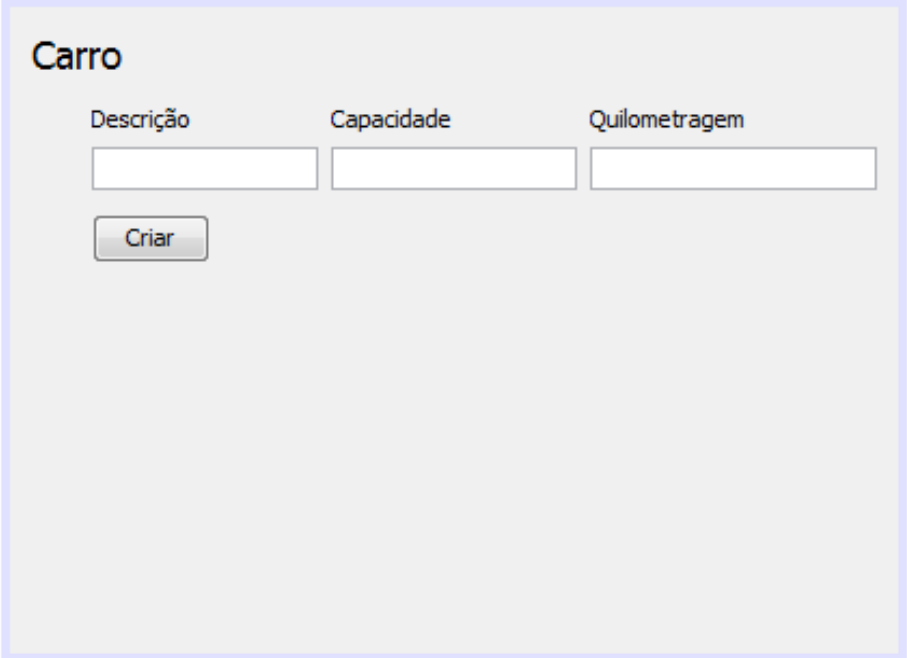
Criando um Form JFrame

Utilize os seguintes objetos para criar o formulário:

Label - Para adicionar um texto;

Campo de Texto - Para adicionar um texto de uma linha;

Botão - Para adicionar um botão.



The image shows a Java Swing window titled "Carro". Inside the window, there are three text input fields arranged horizontally. Above the first field is the label "Descrição", above the second is "Capacidade", and above the third is "Quilometragem". Below the first field, there is a button labeled "Criar".

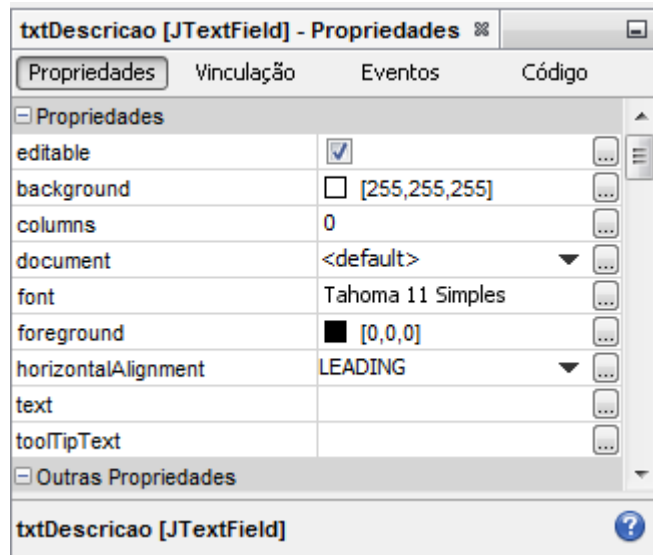
Criando um JFrame

Trabalhe com as seguintes propriedades para formatar o formulário:

Na aba Propriedades:

text - Apague o valor de text para limpar os campos de texto

font - para formatar o tamanho da font



Criando um JFrame

Trabalhe com as seguintes propriedades para formatar o formulário:

Na aba Código:

Nome da Variável - altera a instancia de cada objeto.

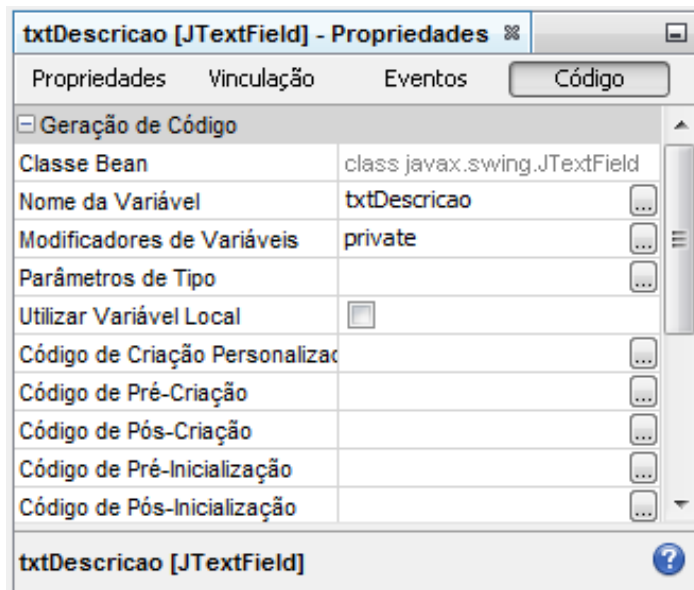
Para os campos de texto:

Descrição renomeie para txtDescCarro

Capacidade renomeie para txtCapacCarro

Quilometragem renomeie para txtKmCarro

Para o Botão Criar renomeie para btCriarCarro.



Criando um Form JFrame

Faça o mesmo para Avião, apenas altere o campo Quilometragem por horas:

Nome da Variável - altera a instancia de cada objeto.

Para os campos de texto de Avião::

Descrição renomeie para txtDescAviao

Capacidade renomeie para txtCapacAviao

Horas renomeie para txtHorasAviao

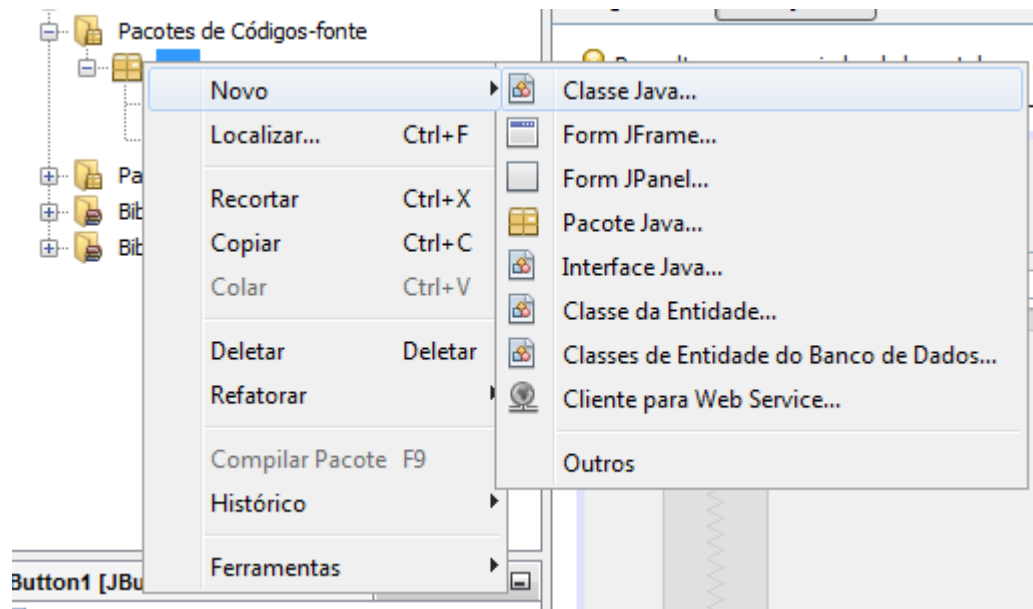
Para o Botão Criar renomeie para
btCriarCarro.

The image shows a Java Swing window with a light gray background and a blue border. It contains two sections: 'Carro' and 'Avião'. Each section has three text input fields and a 'Criar' button. The 'Carro' section has fields for 'Descrição', 'Capacidade', and 'Quilometragem'. The 'Avião' section has fields for 'Descrição', 'Capacidade', and 'Horas'. The 'Criar' button is located below the input fields in each section.

Carro	Avião
Descrição	Descrição
Capacidade	Capacidade
Quilometragem	Horas
Criar	Criar

Criando a classe Carro

Clique com o botão direito,
Novo -> Classe Java



Criando a classe Carro

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package poo;
8
9  /**
10   *
11   * @author Roberto
12   */
13  public class Carro {
14      public String Descricao;
15      public int Capacidade;
16      public int Quilomentragem;
17  }
```

Criando um objeto dentro do formulário

```
- public formulario() {  
    initComponents();  
}  
  
- /**  
    * This method is called from within the constructor to initialize the form.  
    * WARNING: Do NOT modify this code. The content of this method is always  
    * regenerated by the Form Editor.  
    */  
@SuppressWarnings("unchecked")  
+ Generated Code  
private Carro CarroObj;  
  
- private void btCriarCarroActionPerformed(java.awt.event.ActionEvent evt) {
```

Dê um duplo clique no botão criar, perceba que foi alterado para a aba Código-fonte. Adicione a seguinte linha em uma linha acima a classe criada para o botão:

```
private Carro CarroObj;
```

Criando um objeto dentro do formulário

Anteriormente criamos o objeto que iremos trabalhar, então agora é hora de instanciar o objeto e inserir os valores do formulário dentro do objeto.

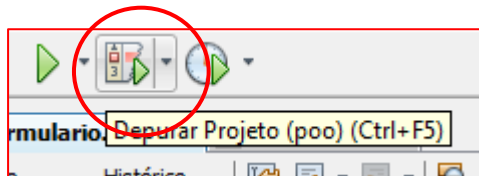
```
177 private void btCriarCarroActionPerformed(java.awt.event.ActionEvent evt) {  
178     CarroObj = new Carro();  
179     CarroObj.Descricao = txtDescCarro.getText();  
180     CarroObj.Capacidade = Integer.parseInt(txtCapacCarro.getText());  
181     CarroObj.Quilomentragem = Integer.parseInt(txtKmCarro.getText());  
182  
183     // TODO add your handling code here:  
184 }
```

Debugando

Dê um duplo clique no botão criar do Carro. Marque um break point na linha que instancia o objeto CarroObj, Para marcar o break point é necessário clicar no número da linha.

```
176  
177 private void btCriarCarroActionPerformed(java.awt.event.ActionEvent evt) {  
178     CarroObj = new Carro();  
179     CarroObj.Descricao = txtDescCarro.getText();
```

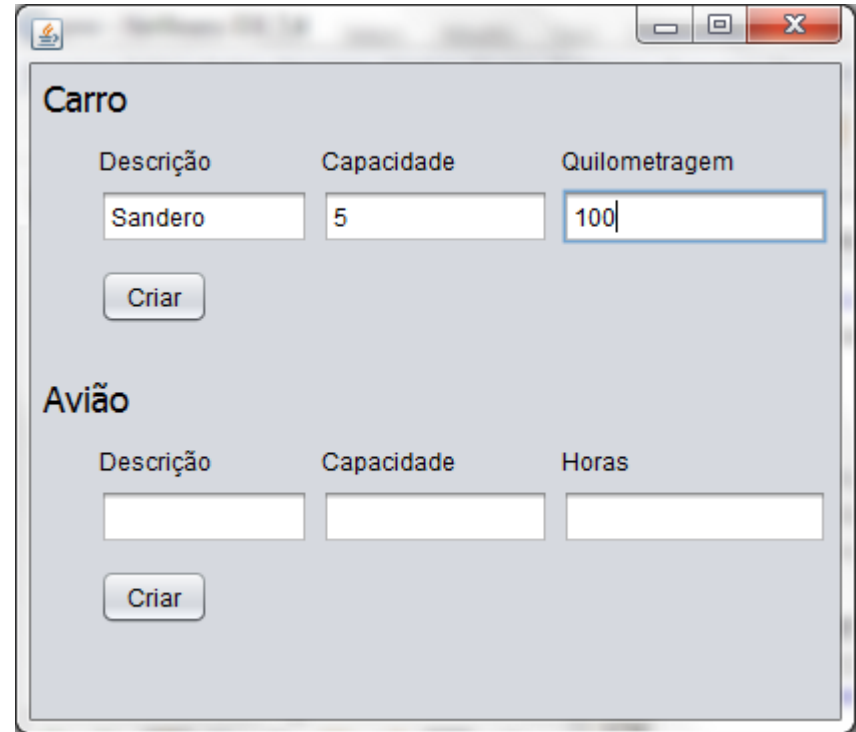
Clique no botão depurar projeto:



Debugando

Digite a Descrição, Capacidade e Quilometragem.

Clique no botão Criar



The image shows a Java Swing window with a light gray background. It contains two sections: 'Carro' and 'Avião'. The 'Carro' section has three input fields labeled 'Descrição', 'Capacidade', and 'Quilometragem'. The 'Descrição' field contains the text 'Sander', the 'Capacidade' field contains '5', and the 'Quilometragem' field contains '100'. Below these fields is a button labeled 'Criar'. The 'Avião' section has three empty input fields labeled 'Descrição', 'Capacidade', and 'Horas'. Below these fields is a button labeled 'Criar'. The window has a standard title bar with minimize, maximize, and close buttons.

Descrição	Capacidade	Quilometragem
Sander	5	100

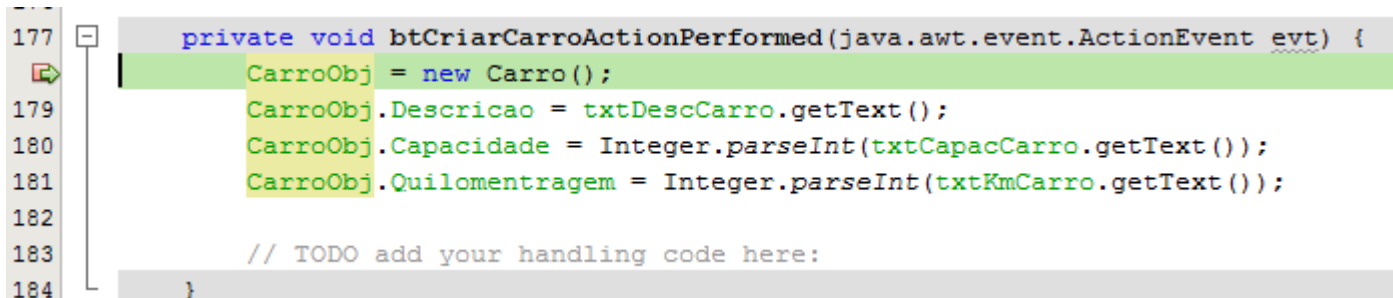
Criar

Descrição	Capacidade	Horas

Criar

Debugando

Ao clicar no botão Criar, o java já indica que chegou na linha demarcada, perceba que existe uma seta verde sobre o quadrado vermelho.

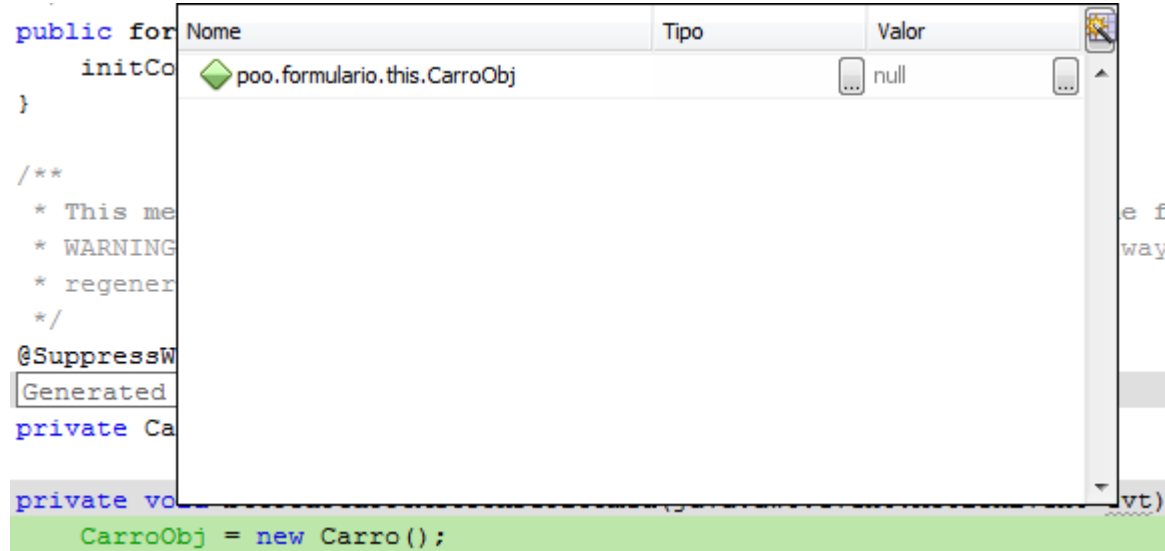


```
177 private void btCriarCarroActionPerformed(java.awt.event.ActionEvent evt) {  
178     CarroObj = new Carro();  
179     CarroObj.Descricao = txtDescCarro.getText();  
180     CarroObj.Capacidade = Integer.parseInt(txtCapacCarro.getText());  
181     CarroObj.Quilomentragem = Integer.parseInt(txtKmCarro.getText());  
182  
183     // TODO add your handling code here:  
184 }
```

The image shows a code editor window with a Java method. A red square is located on line 178, and a green arrow points to it from the left margin. The code is as follows:

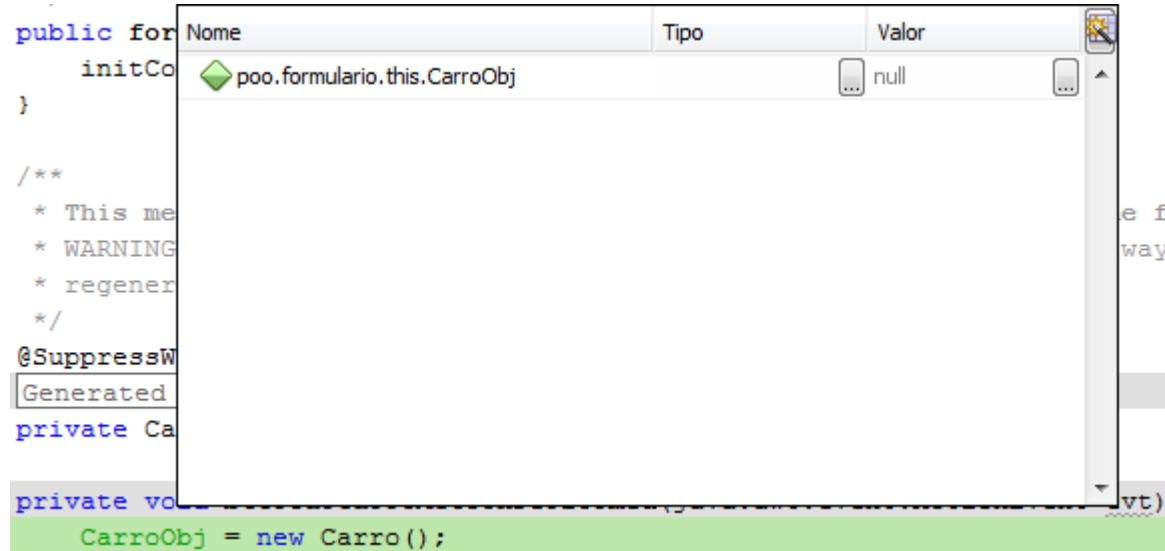
Debugando

Pare o mouse sobre o nome CarroObj, clique no + que aparecerá na caixa de diálogo flutuante, perceba que o objeto foi criado, mas ainda está vazio.



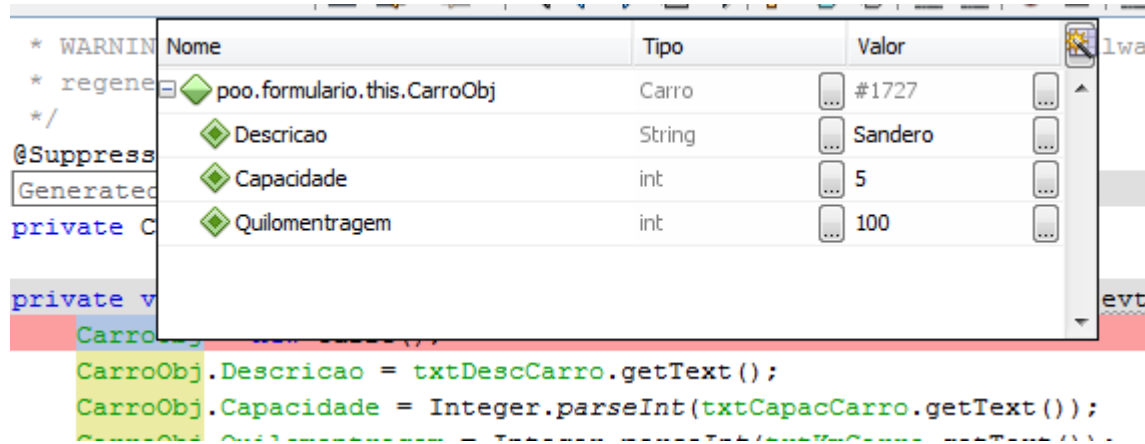
Debugando

Para continuar navegando clique em  , ou Shift + F8.



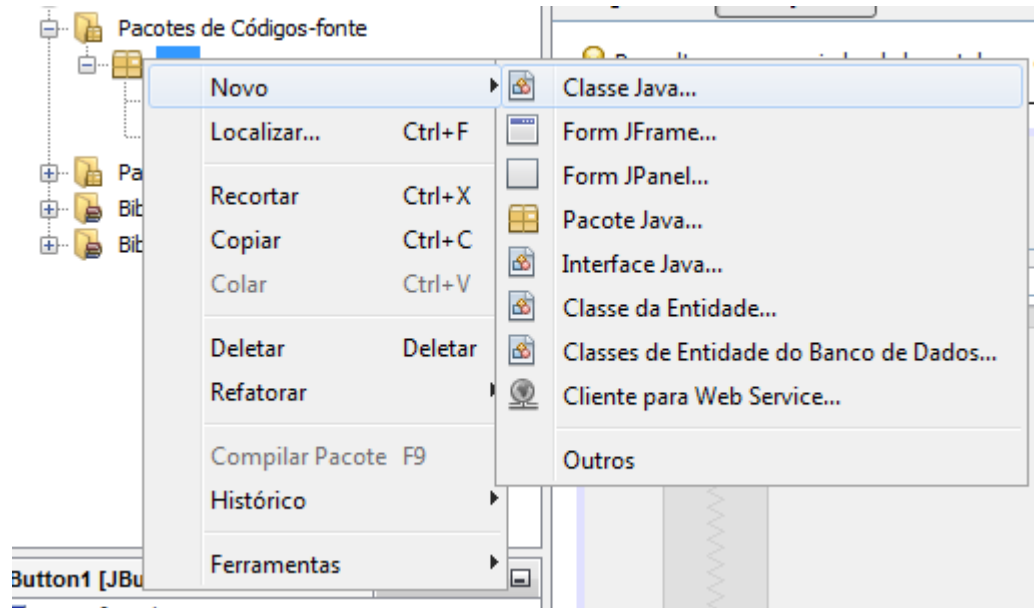
Debugando

Ao final da execução acesse novamente o objeto e verifique ele todo preenchido.





Criando a classe Aviao

Clique com o botão direito,
Novo -> Classe Java



Criando a classe Aviao

```
1   /*
2      * To change this license header, choose License Headers in Project Properties.
3      * To change this template file, choose Tools | Templates
4      * and open the template in the editor.
5  */
6
7  package poo;
8
9   /**
10     *
11     * @author Roberto
12  */
13  public class Aviao {
14      public String Descricao;
15      public int Capacidade;
16      public int Horas;
17  }
18
19  }
```

Criando o objeto AviaObj no formulário

```
28  +  Generated Code  
175  private Carro CarroObj;  
176  
177  private Aviao AviaoObj;
```

Criando o objeto AviaObj no formulário

Dê um duplo Clique no botão Criar em Avião, e entre com o seguinte código:

```
188 private void btCriarAviaoActionPerformed(java.awt.event.ActionEvent evt) {  
189     AviaoObj = new Aviao();  
190     AviaoObj.Descricao = txtDescAviao.getText();  
191     AviaoObj.Capacidade = Integer.parseInt(txtCapacCarro.getText());  
192     AviaoObj.Horas = Integer.parseInt(txtHorasAviao.getText());  
193 }
```

Atividade

Atividade 1: Aproveite todo o conhecimento estudado e crie um novo projeto com duas classes: Funcionário e Aluno. Denomine o projeto como Academico, analise pelo menos 5 atributos paracada classe. Construa um front-end em Windows form.

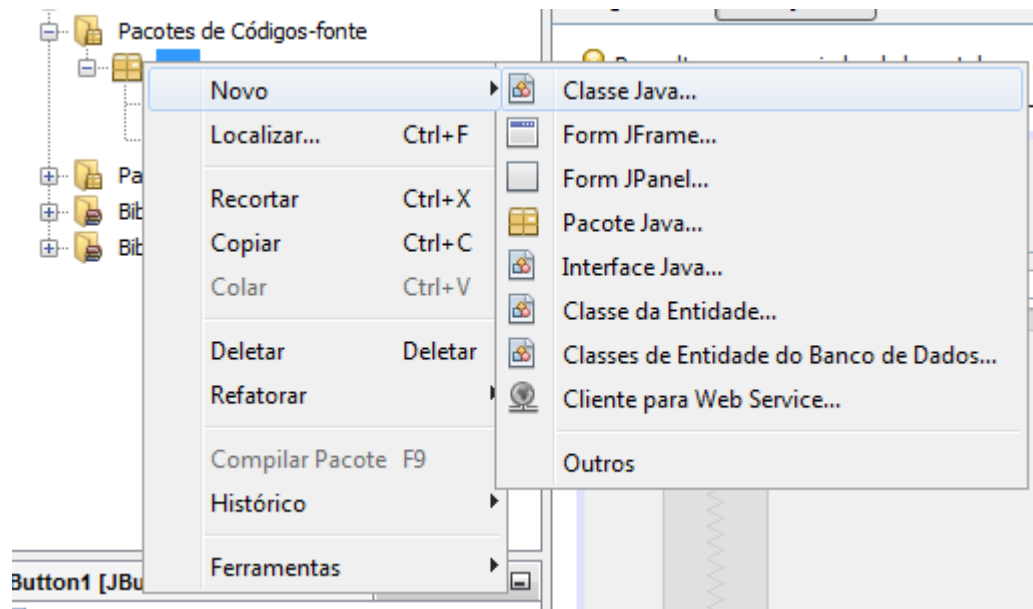
Herança

Seu principal objetivo é permitir a reutilização de código. A Herança tem a finalidade de herdar funcionalidades mais básicas. Agora de forma prática iremos pegar nossas classes: carro e avião e criar herança entre elas.




Como você pode ver um carro tem uma Descrição, uma Capacidade e uma Quilometragem e o avião tem uma Descrição, uma Capacidade e Horas, ambas tem dois atributos em comum. Para aperfeiçoar o processo precisamos generalizar, que significa criar uma classe básica para atender os dois atributos em comum, então, vamos criar a classe meioTransporte.

Criando a classe MeioTransporte

Clique com o botão direito,
Novo -> Classe Java



Criando a classe MeioTransporte

```
1   /*  
2      * To change this license header, choose License Headers in Project Properties.  
3      * To change this template file, choose Tools | Templates  
4      * and open the template in the editor.  
5  */  
6  
7  package poo;  
8  
9   /**  
10     *  
11     * @author Roberto  
12 */  
13  public class MeioTransporte {  
14     public String Descricao;  
15     public int Capacidade;  
16 }
```

Aplicando Herança

Assim acrescentamos os dois atributos em comum entre as duas classes em MeioTransporte.

É preciso que a classe MeioTransporte seja publica para que a classe Carro e Avião acesse a ela, então adicione public.

É preciso ajustar as classes Carro e Avião: apague os dois atributos que tem em comum entre as duas classes e adicione herança dentro de cada classe. No java é preciso adicionar a palavra extends e o nome da classe que está sendo herdada.

Aplicando Herança na Classe Carro

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7   package poo;
8
9   /**
10    *
11    * @author Roberto
12    */
13   public class Carro extends MeioTransporte {
14       public int Quilomentragem;
15   }
16
```

Aplicando Herança na Classe Aviao

```
1  /*
2  |  * To change this license header, choose License Headers in Project Properties.
3  |  * To change this template file, choose Tools | Templates
4  |  * and open the template in the editor.
5  |  */
6
7  package poo;
8
9  /**
10 |  *
11 |  * @author Roberto
12 |  */
13 public class Aviao extends MeioTransporte{
14     public int Horas;
15 }
```

Atividade

Atividade 2: Construa a classe Pessoa e trabalhe a herança entre as classes: Funcionário e Aluno.