INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

AULA:

Interfaces Gráficas e Classes de Usuário

Programação Orientada a Objetos

Alba Lopes, Profa.

http://docentes.ifrn.edu.br/albalopes
alba.lopes@ifrn.edu.br

Criando a classe

- Crie uma classe Motocicleta, no pacote br.edu.ifrn.aula.dominio com as seguintes características:
  - Deve ter os atributos (estado):
    - Marca (String)
    - Modelo (String)
    - Velocidade (int)
  - Deve ter os métodos get e set para os 3 atributos acima
  - Deve ter os métodos (comportamento):
    - acelerar(int aceleracao): Incrementar o atributo velocidade
    - frear(int desaceleracao): Decrementar o atributo velocidade
    - parar(): Zerar o atributo velocidade
    - estaParada(): Verificar se a velocidade é zero

Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

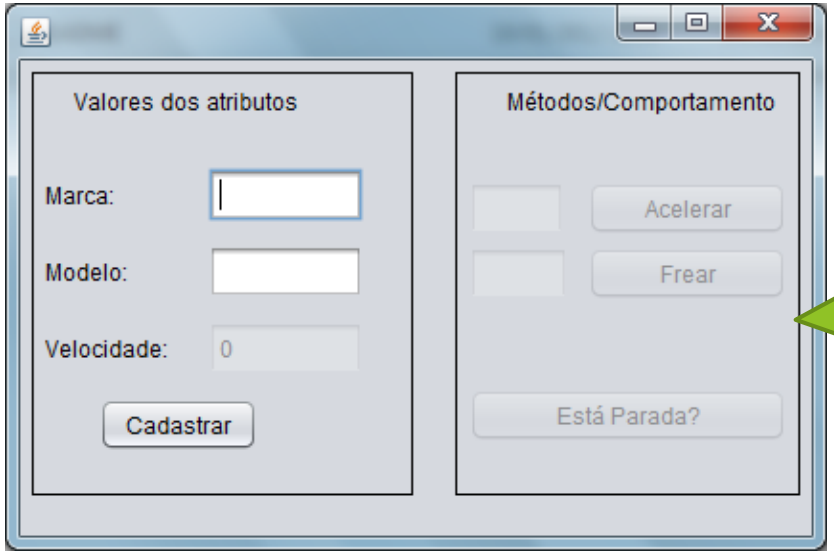
Criando a classe

```
public class Motocicleta {
    public String marca;
    public String modelo;
    public int velocidade;

    public void acelerar(int valor){
        velocidade+= valor;
    }
    public void frear(int valor){
        velocidade-= valor;
    }
    public void parar(){
        velocidade = 0;
    }

    public boolean estaParada(){
        return velocidade==0;
    }
    //gets e sets
```


Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

- Crie uma interface gráfica (InterfaceMotocicleta) no pacote br.edu.ifrn.aula.gui como mostrada a seguir:

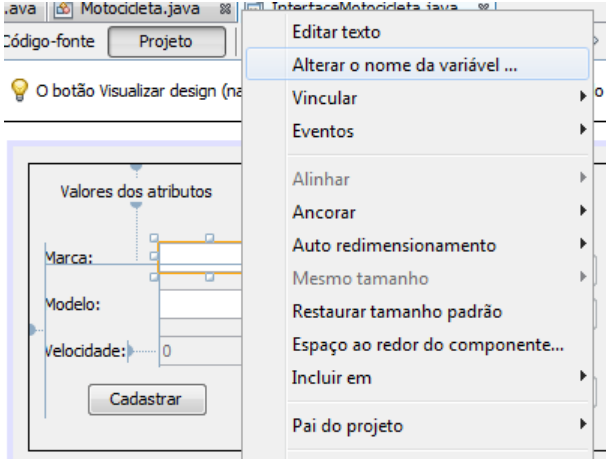


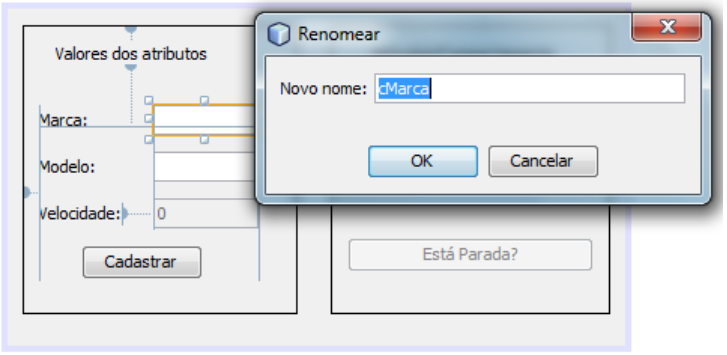
Note que os campos de texto e botões do lado esquerdo estão desabilitados. Desmarque a propriedade *enable* de cada um dos componentes na paleta de propriedades.

Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

- Para facilitar a manipulação dos campos da interface gráfica, altere os nomes dos campos (JTextField1 e demais campos) para nomes mais “amigáveis”:

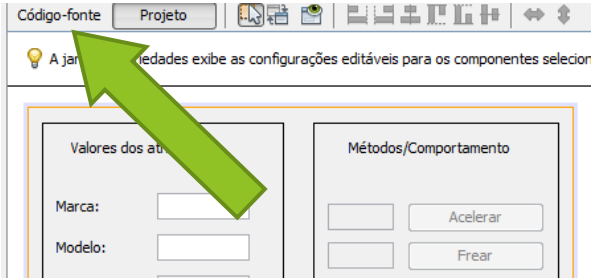




Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

- Antes de adicionar qualquer ação à nossa interface, precisamos criar um atributo do tipo Motocicleta no código fonte do arquivo InterfaceMotocicleta.java



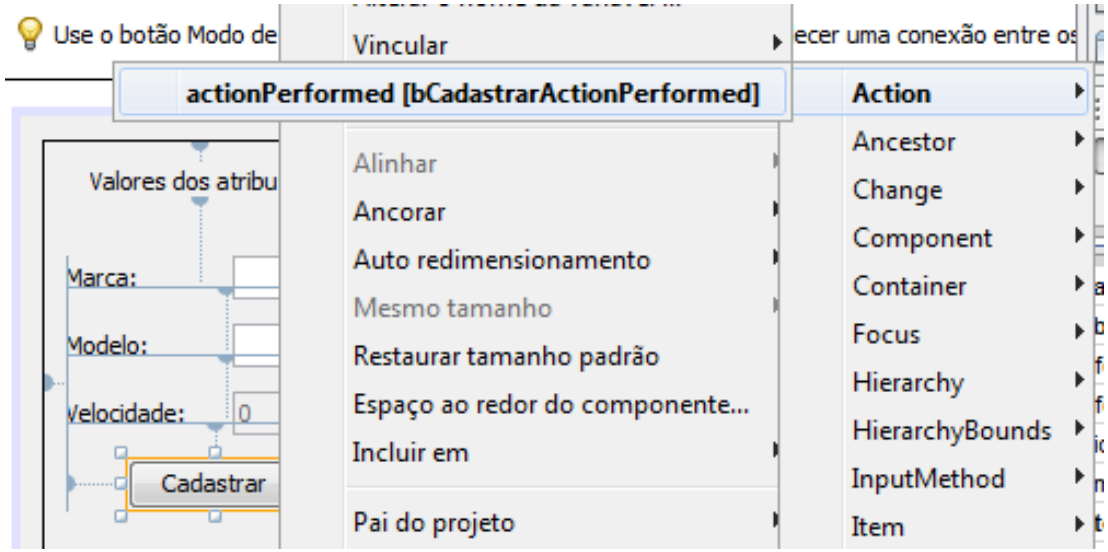

```
19  public class InterfaceMotocicleta extends javax.swing.JFrame {
20
21      private Motocicleta moto;
22
23      /** Creates new form InterfaceMoto
24      public InterfaceMotocicleta() {
25          initComponents();
26      }
```

Logo no início do código fonte, inclua a linha:

Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

- Vamos então, adicionar a primeira ação na nossa interface (no botão **Cadastrar**).



Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

- Vamos então, adicionar a primeira ação na nossa interface (no botão **Cadastrar**).

```
private void bCadastrarActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    moto = new Motocicleta();
    moto.setMarca(cMarca.getText());
    moto.setModelo(cModelo.getText());
    moto.setVelocidade(Integer.parseInt(cVelocidade.getText()));
}
```

Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

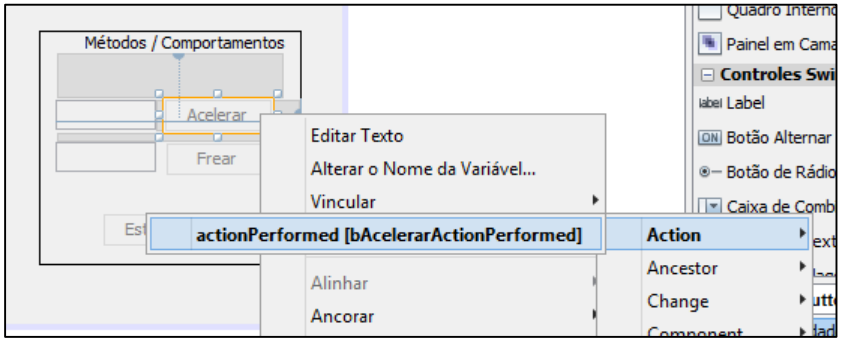
- Note que algumas outras ações acontecem quando o botão Cadastrar é pressionado. Ele desabilita alguns componentes, e habilita outros. Para isso, acrescente os seguintes comandos:

```
private void bCadastrarActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    moto = new Motocicleta();
    moto.setMarca(cMarca.getText());
    moto.setModelo(cModelo.getText());
    moto.setVelocidade(Integer.parseInt(cVelocidade.getText()));
    cMarca.setEnabled(false);
    cModelo.setEnabled(false);
    bCadastrar.setEnabled(false);
    bAcelerar.setEnabled(true);
    bFrear.setEnabled(true);
    bParada.setEnabled(true);
    aceleracao.setEnabled(true);
    desaceleracao.setEnabled(true);
}
```

Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

- Inclua a ação no botão acelerar:

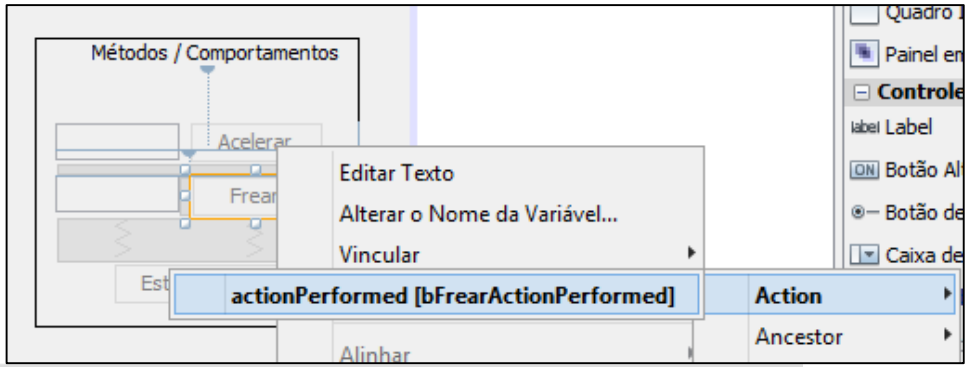



```
private void bAcelerarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    moto.acelerar(Integer.parseInt(aceleracao.getText()));
    cVelocidade.setText(String.valueOf(moto.getVelocidade()));
}
```


Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

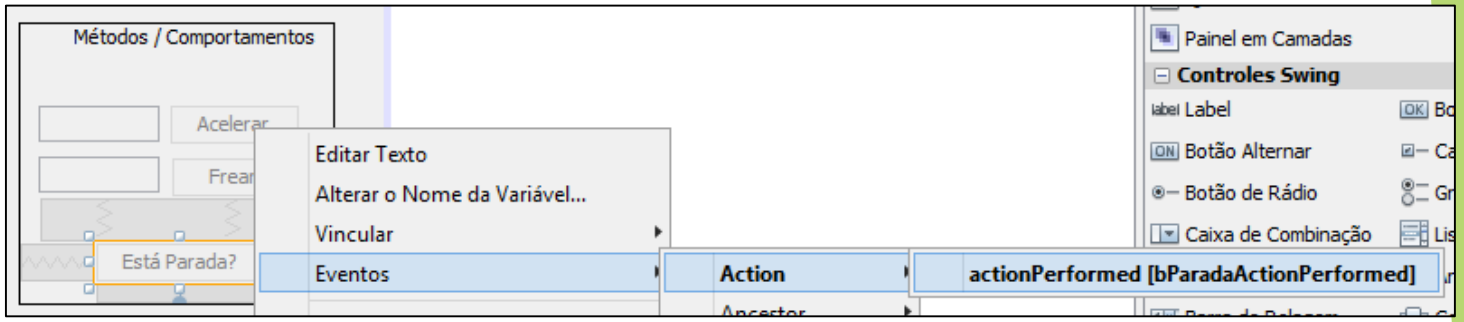
- Inclua a ação no botão frear




```
private void bFrearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    moto.frear(Integer.parseInt(desaceleracao.getText()));
    cVelocidade.setText(String.valueOf(moto.getVelocidade()));
}
```


Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Interface Gráfica

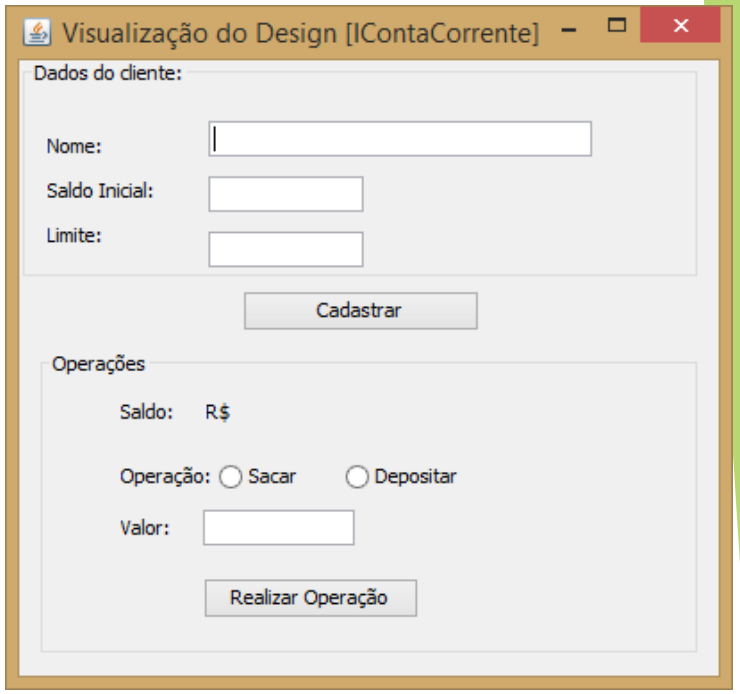



```
private void bParadaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (moto.estaParada()){
        JOptionPane.showMessageDialog(this, "A moto está parada!");
    }else{
        JOptionPane.showMessageDialog(this, "A moto não está parada!");
    }
}
```


Alba Lopes, Profa.
alba.lopes@ifrn.edu.br

Exercício

- Crie uma classe ContaCorrente que possua os atributos:
  - cliente – String
  - saldo – float
  - limite – float
- A classe deve possuir os métodos get e set dos atributos e os métodos abaixo:
  - sacar (float valor) – retorna um valor booleano para indicar se foi possível realizar ou não o saque
  - depositar(float valor) – retorna um valor booleano para indicar se foi possível ou não realizar o depósito (teste se o parâmetro valor é maior do que 0)
- Crie uma interface gráfica para realizar as operações da conta corrente:



Alba Lopes, Profa.
alba.lopes@ifrn.edu.br