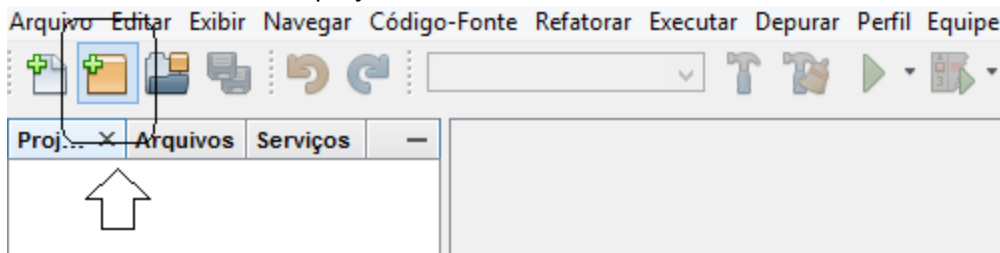
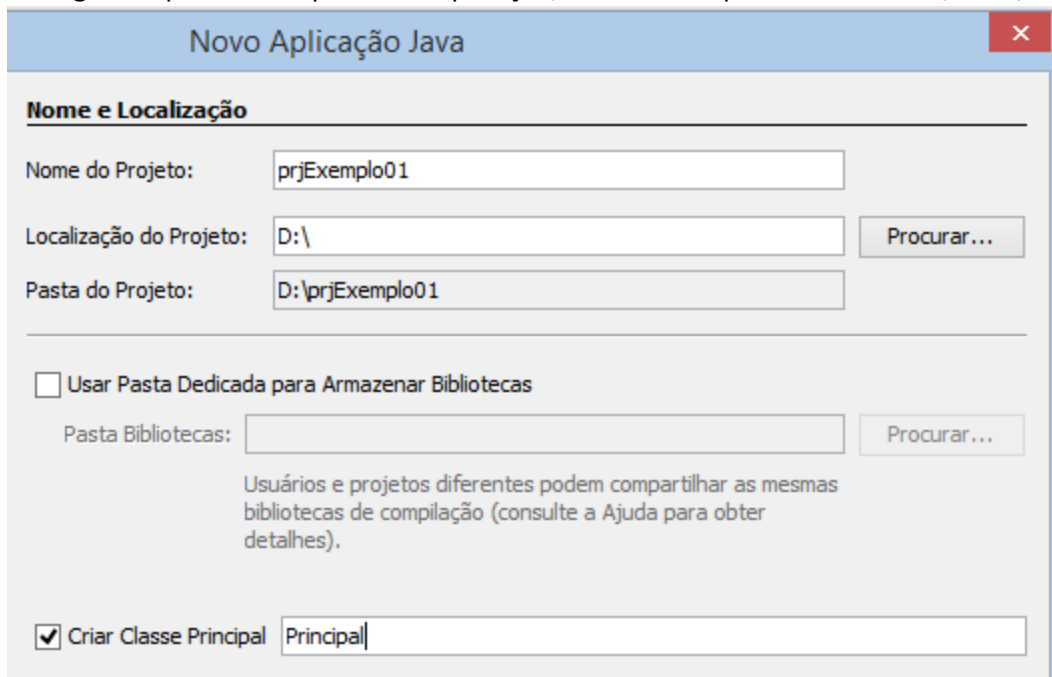


# Linguagem de Programação Orientada a Objetos

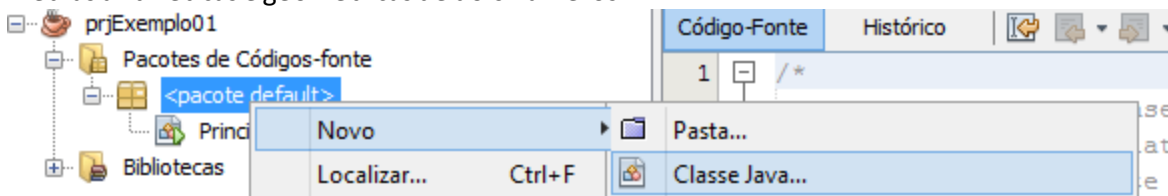
1. Vamos iniciar criando um projeto em netBeans:



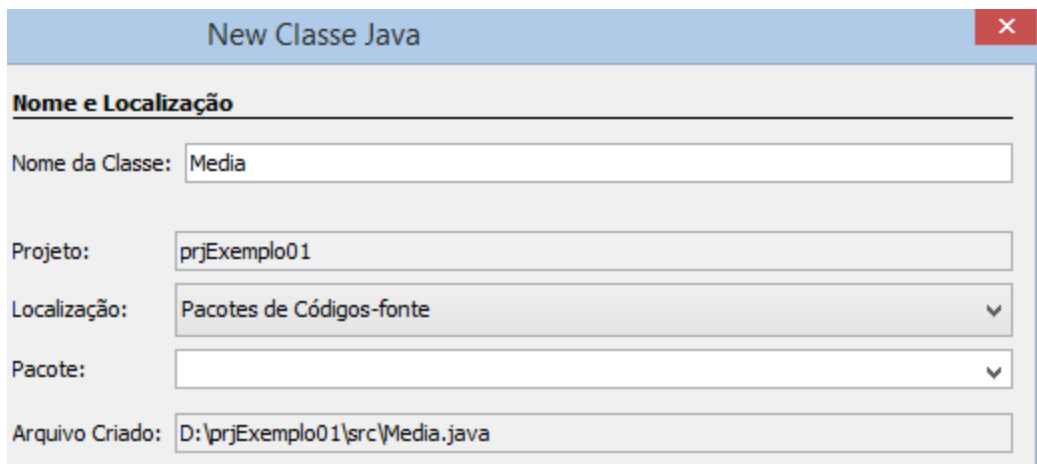
Configure os parâmetros para a sua aplicação, modificando para a unidade D:\ ou E:\ do seu computador:



Adicione a seu pacote default onde está a classe principal a classe Média, que será usada para calcular médias aritméticas e geométricas de dois números:



Nomeie a classe conforme abaixo:



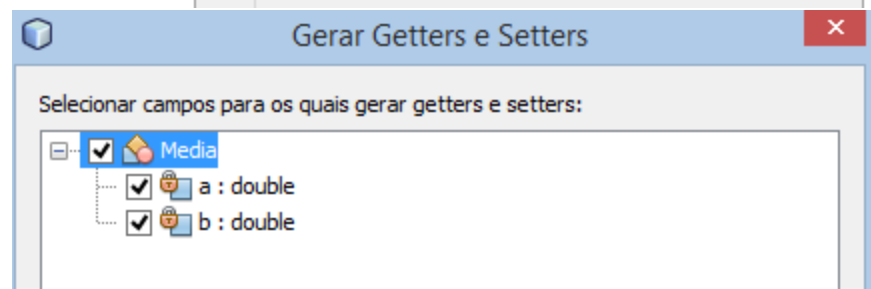
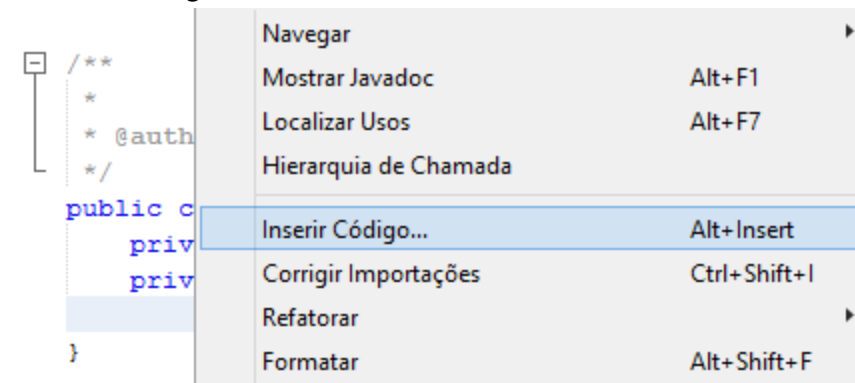
Codifique as propriedades da classe:

```

public class Media {
    private double a;
    private double b;
}

```

Acrescente os getters e setters a sua classe:



Clique em gerar:

```

public double getA() {
    return a;
}

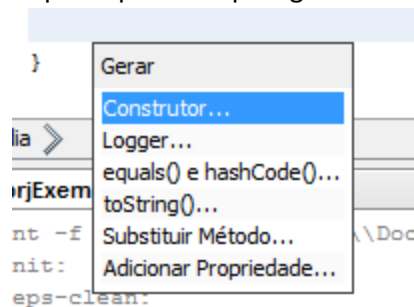
public void setA(double a) {
    this.a = a;
}

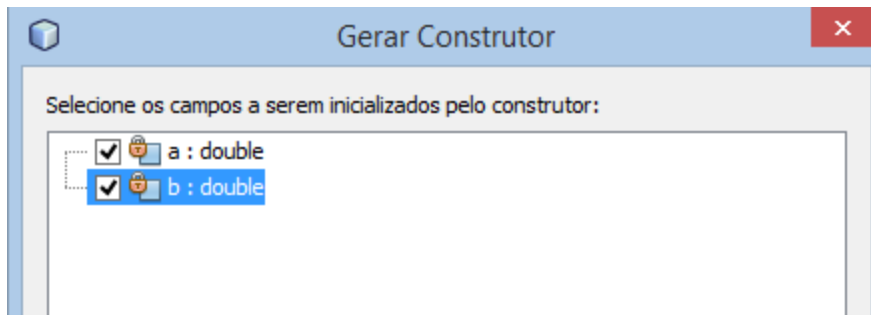
public double getB() {
    return b;
}

public void setB(double b) {
    this.b = b;
}

```

Repita o processo para gerar o construtor:





Abaixo o construtor gerado:

```
public Media(double a, double b) {
    this.a = a;
    this.b = b;
}
```

Abaixo do construtor criar os seus métodos para o cálculo das médias:

```
public String calcular() {

    double r = (this.a+this.b)/2;
    return String.valueOf(r);
}
```

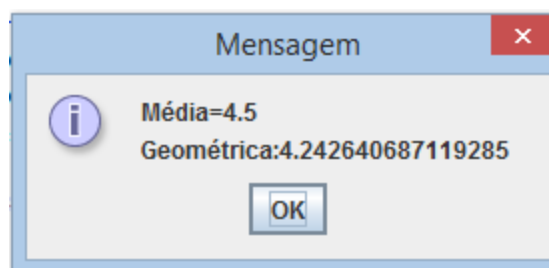
O método para a média geométrica:

```
public String geometrica() {
    double r = Math.sqrt(this.a*this.b);
    return String.valueOf(r);
}
```

Na classe Principal testaremos a nossa classe Média:

```
public static void main(String[] args) {
    double a = Double.parseDouble(JOptionPane.showInputDialog("Digite A:"));
    double b = Double.parseDouble(JOptionPane.showInputDialog("Digite b:"));
    Media m = new Media(a,b);
    JOptionPane.showMessageDialog(null, "Média="+ m.calcular()+
        "\nGeométrica:" + m.geometrica());
}
```

O resultado será:

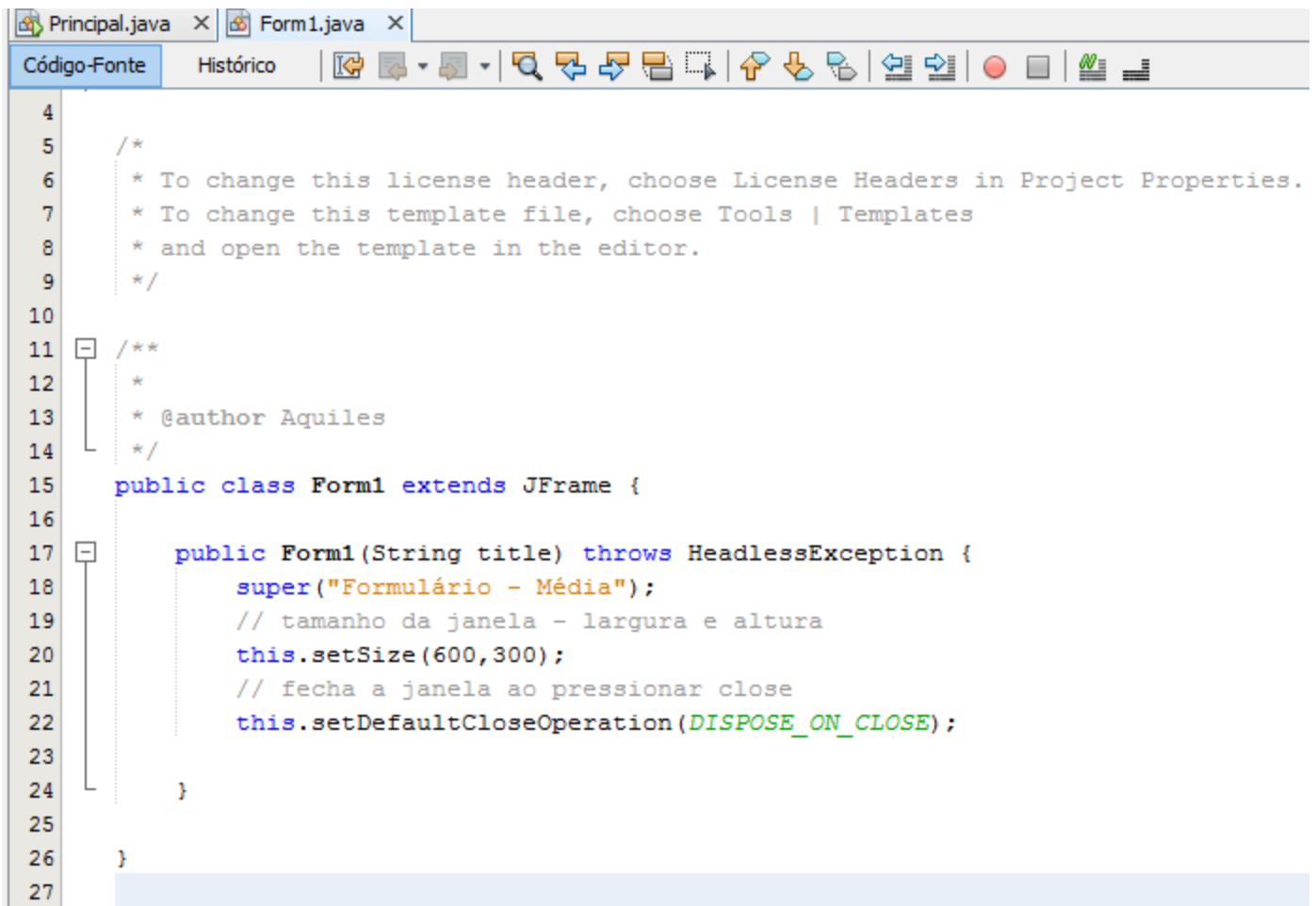


Criando uma interface para a execução a classe, para isso iremos adicionar uma classe chamada Form1:

```
public class Form1 {

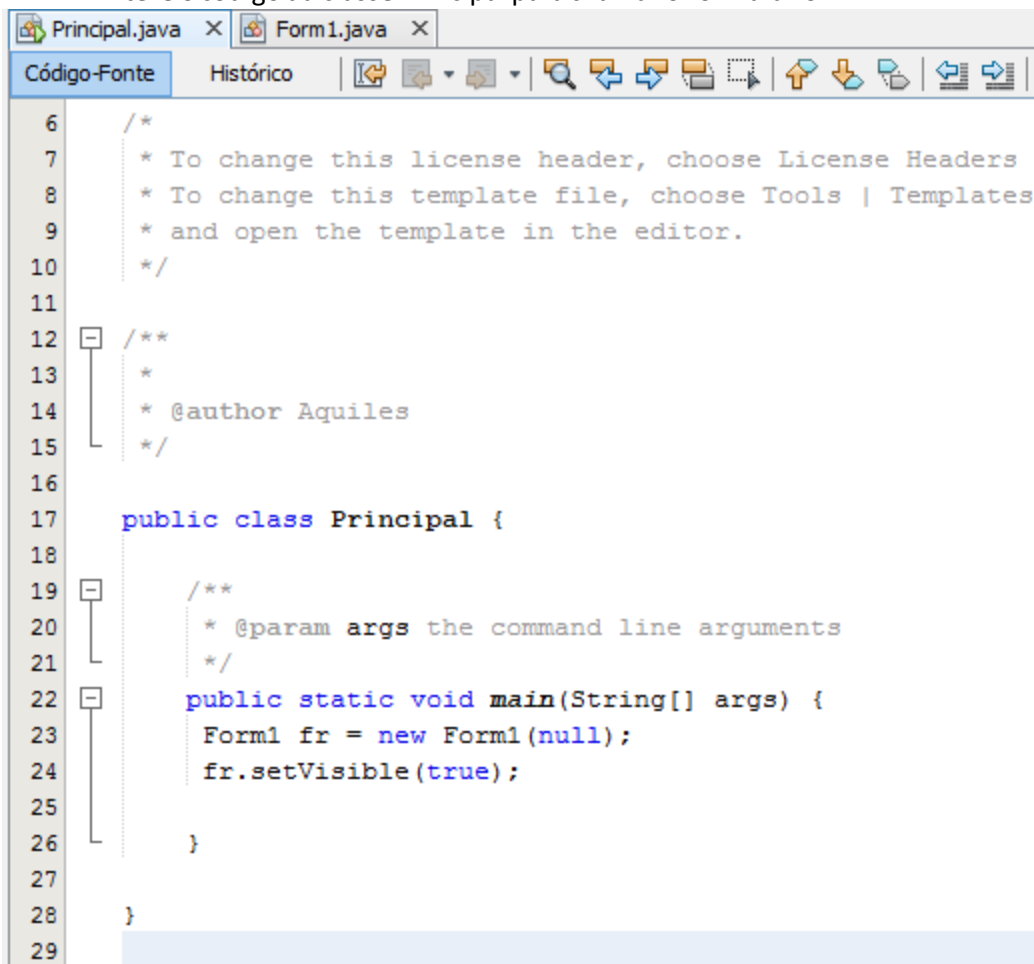
}
```

Modifique a classe para importar a estrutura de um formulário:



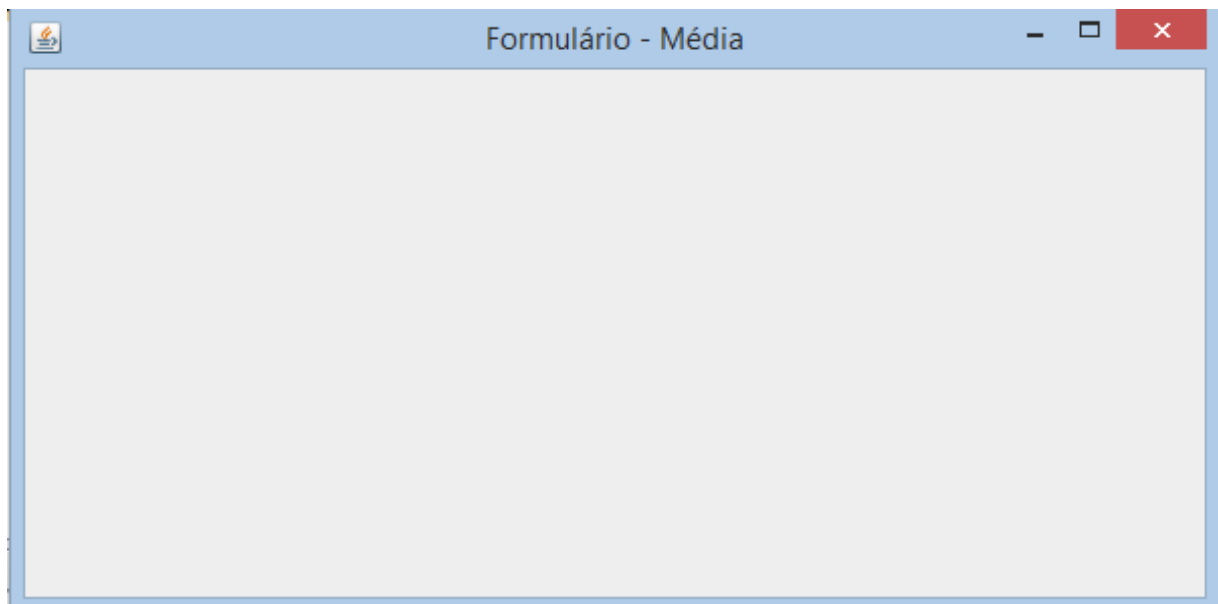
```
4
5  /*
6   * To change this license header, choose License Headers in Project Properties.
7   * To change this template file, choose Tools | Templates
8   * and open the template in the editor.
9   */
10
11  /**
12   *
13   * @author Aquiles
14   */
15  public class Form1 extends JFrame {
16
17      public Form1(String title) throws HeadlessException {
18          super("Formulário - Média");
19          // tamanho da janela - largura e altura
20          this.setSize(600,300);
21          // fecha a janela ao pressionar close
22          this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
23
24      }
25
26  }
27
```

Altere o código da classe Principal para chamar o formulário:



```
6
7  /*
8   * To change this license header, choose License Headers :
9   * To change this template file, choose Tools | Templates
10   * and open the template in the editor.
11   */
12
13  /**
14   *
15   * @author Aquiles
16   */
17  public class Principal {
18
19      /**
20       * @param args the command line arguments
21       */
22      public static void main(String[] args) {
23          Form1 fr = new Form1(null);
24          fr.setVisible(true);
25
26      }
27
28  }
29
```

A visualização será a seguinte:

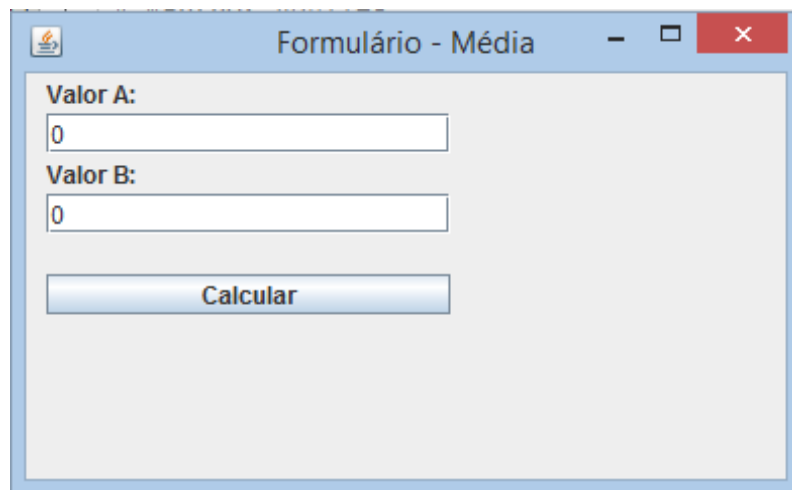


Criando a estrutura de botão e caixas de texto com rótulos no formulário. O `setBounds` é usado para desenhar os componentes no formulário, os dois primeiros valores são as coordenadas horizontal e vertical, os parâmetros seguintes se referem ao comprimento e altura.

```
public class Form1 extends JFrame {  
  
    public Form1(String title) throws HeadlessException {  
  
        super("Formulário - Média");  
  
        JButton btCalcular = new JButton("Calcular");  
        JTextField txtA = new JTextField("0");  
        JTextField txtB = new JTextField("0");  
        JLabel lbA = new JLabel("Valor A:");  
        JLabel lbB = new JLabel("Valor B:");  
        JPanel p = new JPanel();  
        p.setLayout(null);  
        this.add(p);  
        lbA.setBounds(10,0,100,20); txtA.setBounds(10,20,200,20);  
        lbB.setBounds(10,40,100,20); txtB.setBounds(10,60,200,20);  
        btCalcular.setBounds(10,100,200,20);  
        p.add(lbA);  
        p.add(lbB);  
        p.add(txtA);  
        p.add(txtB);  
  
        p.add(btCalcular);  
  
        // tamanho da janela - largura e altura  
        this.setSize(600,300);  
    }  
}
```



O resultado visual será:



Agora criaremos o evento para controlar o pressionamento do botão e nele calcular a média dos valores digitados na caixa de texto:

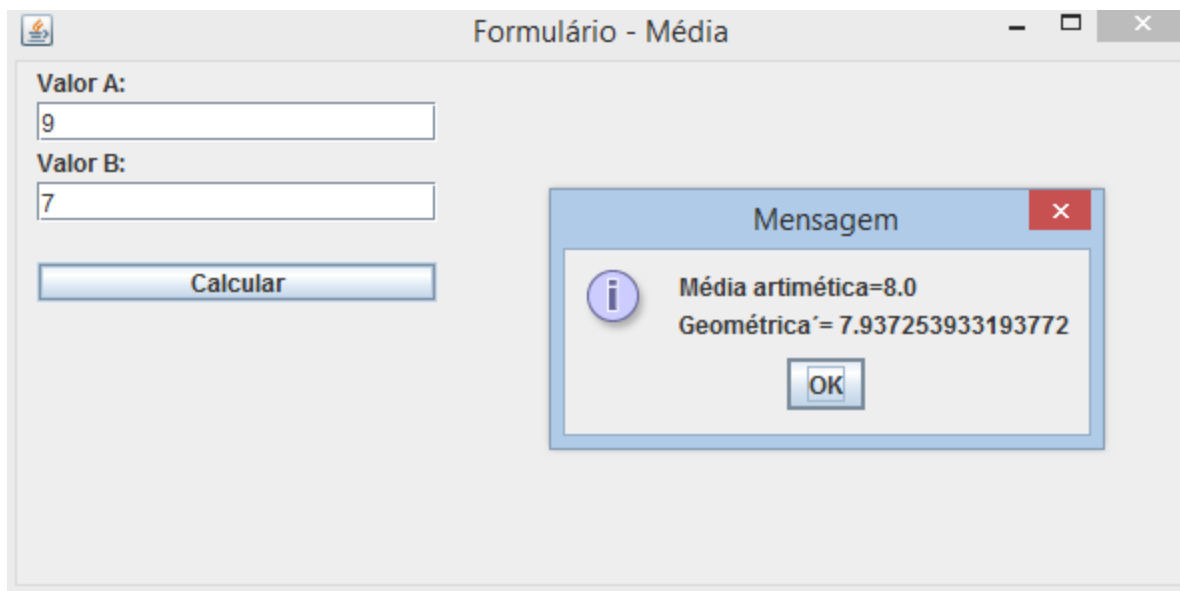
```
p.add(btCalcular);
```

```
btCalcular.addActionListener((ActionEvent e) -> {
    double a = Double.parseDouble(txtA.getText());
    double b = Double.parseDouble(txtB.getText());
    Media m = new Media(a,b);
    JOptionPane.showMessageDialog(null,"Média aritmética=" + m.calcular()
    + "\nGeométrica'=" + m.geometrica());
});
```

```
// tamanho da janela - largura e altura
```

```
this.setSize(600,300);
```

O resultado será:



#### Exercício

1. Codificar um método para calcular a média harmônica de dois números e exibir a resposta na Caixa de Mensagem.