

[< VOLTAR](#)

Projeto de um Applet Java

Applets em Java eram muito comuns no início da programação Java como um recurso poderoso de incluir pequenas aplicações Java dentro de sites. Muito cuidado, apenas, para não confundir programação de Applets com programação Java para Web, que possuem arquiteturas totalmente diferentes. Nesta aula você aprenderá a fazer deploy (criar executável) de sua aplicação e o implementar dentro de uma pequena página web.

NESTE TÓPICO



O que é um Applet Java?

Por definição, Applets programas desenvolvidos em alguma linguagem de alto nível (como Java, por exemplo), que rodam dentro do browser (navegador web).

Um Applet Java, então, é um programa feito em Java para rodar dentro de uma página Web. É importante não confundir Applet Java com desenvolvimento Web, que apresenta uma arquitetura cliente-servidor própria e possui suas próprias regras de desenvolvimento. Desenvolvimento Java para Web é uma disciplina específica que, certamente, você terá contato.

Atualmente Applets Java são bloqueados na maior parte dos browsers (inclusive *Internet Explorer!*), por motivos de segurança, pois, como um Applet é um programa executável que roda dentro de uma página Web, ele pode ter acesso ao seu computador inteiro. Infelizmente muitos “mal desenvolvedores” desenvolveram, ao longo da história, Applets ocultos que conseguiam captar informações secretas dos usuários, como senhas de bancos, números de cartões etc.

Isso quer dizer que, vamos ter de desbloquear nosso Applet para podermos executá-lo. Quer dizer, para executar qualquer Applet, você precisa dar autorização direta a ele. Este é um recurso relativamente novo do Java, implementado pela Oracle.

Desenvolvendo um Applet Java

Para desenvolvermos um pequeno Applet, vamos criar um novo projeto no NetBeans que é capaz de receber três valores (A, B e C) e calcular o Bhaskara. Isso mesmo, vamos desenvolver um pequeno Applet que dados os valores A, B e C de uma equação quadrática e calcular o valor de Delta e de suas raízes, se existirem.

O vídeo abaixo mostra o processo de implementação de nosso exemplo, desde a criação do projeto até a implementação da página web e desbloqueio para execução.

./videos/370303930.mp4



Para facilitar, abaixo você encontra a codificação das classes do vídeo:

Classe Bhaskara.Java:

```
1. package br.uninove.poo.matematica;
2.
3. public class Bhaskara {
4.
5.     private double a, b, c;
6.
7.     public double getA() {
8.         return a;
9.     }
10.
11.     public void setA(double a) {
12.         this.a = a;
13.     }
14.
15.     public double getB() {
16.         return b;
17.     }
18.
19.     public void setB(double b) {
20.         this.b = b;
21.     }
22.
23.     public double getC() {
24.         return c;
25.     }
26.
27.     public void setC(double c) {
28.         this.c = c;
29.     }
30.
31.     public double getDelta() {
32.         double delta = 0;
33.         delta = (b * b) - 4 * a * c;
34.         return delta;
35.     }
36.
37.     public double getRaizPositiva() {
38.         double raizPositiva = 0;
39.         raizPositiva = ((-1 * b) + Math.sqrt(getDelta())) / (2 * a);
40.         return raizPositiva;
41.     }
42.
43.     public double getRaizNegativa() {
44.         double raizPositiva = 0;
45.         raizPositiva = ((-1 * b) - Math.sqrt(getDelta())) / (2 * a);
46.         return raizPositiva;
47.     }
48. }
```



Classe AppletJava.Java (criada pelo assistente de criação de layouts do NetBeans):

```

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.  package br.uninove.poo.applet;
7.
8.  import br.uninove.poo.matematica.Bhaskara;
9.
10. /**
11.  *
12.  * @author Thiago
13.  */
14. public class AppletBhaskara extends javax.swing.JApplet {
15.
16.     /**
17.      * Initializes the applet AppletBhaskara
18.      */
19.     @Override
20.     public void init() {
21.         /* Set the Nimbus look and feel */
22.         //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (o
23.         ptional) ">
24.         /* If Nimbus (introduced in Java SE 6) is not available, stay with the defau
25.         lt look and feel.
26.          * For details see http://download.oracle.com/javase/tutorial/uiswing/lookan
27.         dfeel/plaf.html
28.          */
29.         try {
30.             for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.
31.             getInstalledLookAndFeels()) {
32.                 if ("Nimbus".equals(info.getName())) {
33.                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
34.                     break;
35.                 }
36.             }
37.         } catch (ClassNotFoundException ex) {
38.             java.util.logging.Logger.getLogger(AppletBhaskara.class.getName()).log(j
39.             ava.util.logging.Level.SEVERE, null, ex);
40.         } catch (InstantiationException ex) {
41.             java.util.logging.Logger.getLogger(AppletBhaskara.class.getName()).log(j
42.             ava.util.logging.Level.SEVERE, null, ex);
43.         } catch (IllegalAccessException ex) {
44.             java.util.logging.Logger.getLogger(AppletBhaskara.class.getName()).log(j
45.             ava.util.logging.Level.SEVERE, null, ex);
46.         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
47.             java.util.logging.Logger.getLogger(AppletBhaskara.class.getName()).log(j
48.             ava.util.logging.Level.SEVERE, null, ex);
49.         }
50.         //</editor-fold>
51.
52.         /* Create and display the applet */
53.         try {
54.             java.awt.EventQueue.invokeLater(new Runnable() {
55.                 public void run() {
56.                     initComponents();
57.                 }
58.             });
59.         } catch (Exception ex) {
60.             ex.printStackTrace();
61.         }
62.
63.     /**
64.      * This method is called from within the init() method to initialize the
65.      * form. WARNING: Do NOT modify this code. The content of this method is
66.      * always regenerated by the Form Editor.
67.      */
68.     @SuppressWarnings("unchecked")
69.     // <editor-fold defaultstate="collapsed" desc="Generated Code">
70.     private void initComponents() {

```





```
126.                .addGroup(layout.createSequentialGroup())
127.                .addGap(52, 52, 52)
128.                .addComponent(jLbValorA)
129.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacem
ent.RELATED)
130.                .addComponent(jTxtValorA, javax.swing.GroupLayout.PREFER
RED_SIZE, 125, javax.swing.GroupLayout.PREFERRED_SIZE)))
131.                .addGap(49, 49, 49)
132.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
133.                .addGroup(layout.createSequentialGroup())
134.                .addComponent(jLabel3)
135.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
136.                .addComponent(jLbRaizneg))
137.                .addGroup(layout.createSequentialGroup())
138.                .addComponent(jLabel1)
139.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
140.                .addComponent(jLbResultadoDelta))
141.                .addGroup(layout.createSequentialGroup())
142.                .addComponent(jLabel2)
143.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
144.                .addComponent(jLbRaizPos)))
145.                .addContainerGap(63, Short.MAX_VALUE))
146.            );
147.            layout.setVerticalGroup(
148.                layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
149.                .addGroup(layout.createSequentialGroup())
150.                .addGap(27, 27, 27)
151.                .addComponent(jLbInformacao1)
152.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
153.                .addGroup(layout.createSequentialGroup())
154.                .addGap(18, 18, 18)
155.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.BASELINE)
156.                .addComponent(jLbValorA)
157.                .addComponent(jTxtValorA, javax.swing.GroupLayout.PREFER
RED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SI
ZE)))
158.                .addGroup(layout.createSequentialGroup())
159.                .addGap(13, 13, 13)
160.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.BASELINE)
161.                .addComponent(jLabel1)
162.                .addComponent(jLbResultadoDelta)))
163.                .addGap(18, 18, 18)
164.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.BASELINE)
165.                .addComponent(jLbValorA1)
166.                .addComponent(jTxtValorB, javax.swing.GroupLayout.PREFERRED_SIZ
E, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
167.                .addComponent(jLabel2)
168.                .addComponent(jLbRaizPos))
169.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
170.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.BASELINE)
171.                .addComponent(jLabel3)
172.                .addComponent(jLbRaizneg))
173.                .addGap(1, 1, 1)
174.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.BASELINE)
175.                .addComponent(jLbValorA2)
176.                .addComponent(jTxtValorC, javax.swing.GroupLayout.PREFERRED_SIZ
E, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
177.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
178.                .addComponent(jButton1)
179.                .addContainerGap(22, Short.MAX_VALUE))
180.            );
181.        } // </editor-fold>
```



```

182.
183.     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
184.         Bhaskara b = new Bhaskara();
185.         double aD, bD, cD, delta, rP ,rN;
186.         try {
187.             aD = Double.parseDouble(jTxtValorA.getText());
188.             bD = Double.parseDouble(jTxtValorB.getText());
189.             cD = Double.parseDouble(jTxtValorC.getText());
190.
191.             b.setA(aD);
192.             b.setB(bD);
193.             b.setC(cD);
194.
195.             delta = b.getDelta();
196.             rP = b.getRaizPositiva();
197.             rN = b.getRaizNegativa();
198.
199.             jLbResultadoDelta.setText(String.valueOf(delta));
200.             jLbRaizPos.setText(String.valueOf(rP));
201.             jLbRaizneg.setText(String.valueOf(rN));
202.
203.         } catch (Exception ex) {
204.
205.         }
206.     }
207.
208.     // Variables declaration - do not modify
209.     private javax.swing.JButton jButton1;
210.     private javax.swing.JLabel jLabel1;
211.     private javax.swing.JLabel jLabel2;
212.     private javax.swing.JLabel jLabel3;
213.     private javax.swing.JLabel jLbInformacao1;
214.     private javax.swing.JLabel jLbRaizPos;
215.     private javax.swing.JLabel jLbRaizneg;
216.     private javax.swing.JLabel jLbResultadoDelta;
217.     private javax.swing.JLabel jLbValorA;
218.     private javax.swing.JLabel jLbValorA1;
219.     private javax.swing.JLabel jLbValorA2;
220.     private javax.swing.JTextField jTxtValorA;
221.     private javax.swing.JTextField jTxtValorB;
222.     private javax.swing.JTextField jTxtValorC;
223.     // End of variables declaration
224. }

```



E, finalmente, nosso arquivo HTML, externo ao projeto:

```

1.  <html>
2.      <head>
3.          <title>Applet Java</title>
4.      </head>
5.      <body>
6.          <center>
7.              <h1>Testando nosso Applet Java de Bhaskara</h1>
8.              <br/><hr><br/>
9.              <applet code="br.uninove.poo.applet.AppletBhaskara" archive="AppletBhask
ara.jar" width="100%" height="50%"/>
10.          </center>
11.      </body>
12. </html>

```

Compile o código e execute seu Applet. Não esqueça de desbloquear o uso de Applets no browser o qual você está rodando a aplicação.

Resumo da Aula

Chegamos ao final de mais uma aula de programação orientada a objetos. Nesta aula você aprendeu a desenvolver Applets Java, desbloqueando o recurso nas configurações do Java.

Você aprendeu, também, que o processo de deploy (construção) da aplicação gera um arquivo “jar” que fica armazenado na pasta “dist” de seu projeto. Este processo de deploy pode ser aplicado a qualquer projeto que você já tenha desenvolvido.

Pratique bastante a programação de Applets e todos os conceitos de Java aprendidos até aqui. Boa programação e bons estudos.

Quiz

Exercício Final

Projeto de um Applet Java



INICIAR ➤

Referências

Teruel, E. C., 2015, Programação Orientada a Objetos com Java - sem mistérios - 1ª Ed., Editora Uninove

Deitei P. e Deitel H., 2010, Java : Como programar, 8ª Edição, Pearson Pretice Hall

Schildt, H., 2015, Schildt, Java para iniciantes : crie, compile e execute programas Java rapidamente, Bookman



Avalie este tópico



ANTERIOR

Projeto de GUI conectada a Bases de Dados

Biblioteca



Índice

(<https://www.uninove.br/conhec-a->

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

© Todos os direitos reservados

Ajuda?
(<https://ava.uninove.br/ava/curso.php?idCurso=>)

