JAVA SWING & AWT

FORMULÁRIOS

UM BREVE TUTORIAL

(Versão 1.0)

ALCIDES CALSAVARA

PUCPR

2018

# Criação de diálogos de entrada e de mensagem

Crie um projeto Java.

Crie a classe **Login**:

import javax.swing.JOptionPane;  
  
public class Login {

public static void main(String[] args) {  
 String usuario = JOptionPane.*showInputDialog*(null, "Usuário");  
 String senha = JOptionPane.*showInputDialog*(null, "Senha");  
  
 System.*out*.println(usuario);  
 System.*out*.println(senha);  
  
 if (usuario != null && senha != null && usuario.equals("pucpr") && senha.equals("poo"))  
 {  
 JOptionPane.*showMessageDialog*(null, "login autorizado");  
 }  
 else  
 {  
 JOptionPane.*showMessageDialog*(null, "login recusado");  
 }  
 }  
}

Compile e execute o projeto a partir da classe Login.

Forneça os dados solicitados nas janelas com título "Entrada".

Observe a janela exibida no final com título "Mensagem".

A classe **JOptionPane** permite:

* criar diálogo para entrada de uma string: *showInputDialog*
* criar diálogo para exibir uma string: *showMessageDialog*

# Entrada de valor numérico

Crie a classe **SomarNumerosInteiros**:

import javax.swing.JOptionPane;  
  
public class SomarNumerosInteiros {

public static void main(String[] args) {  
 String s1 = JOptionPane.*showInputDialog*(null, "Primeiro número inteiro");  
 while (!*NumeroInteiroValido*(s1)) {  
 s1 = JOptionPane.*showInputDialog*(null,  
 "Entrada inválida! Digite um número inteiro.");  
 }  
  
 String s2 = JOptionPane.*showInputDialog*(null, "Segundo número inteiro");  
 while (!*NumeroInteiroValido*(s2)) {  
 s2 = JOptionPane.*showInputDialog*(null,  
 "Entrada inválida! Digite um número inteiro.");  
  
 }  
  
 int soma = new Integer(s1) + new Integer(s2);  
 JOptionPane.*showMessageDialog*(null, "A soma de "  
 + s1 + " e " + s2 + " é " + soma);  
  
 }  
  
 private static boolean NumeroInteiroValido(String s) {  
 boolean resultado;  
 try {  
 Integer.*parseInt*(s);  
 resultado = true;  
 } catch (NumberFormatException e) {  
 resultado = false;  
 }  
 return resultado;  
 }  
}

Compile e execute o projeto a partir da classe SomarNumerosInteiros.

Observe que o programa exige que as entradas sejam números inteiros.

O método NumeroInteiroValido verifica se a string fornecida como parâmetro representa um número inteiro.

* A classe Integer, que faz parte da biblioteca de Java, armazena um valor inteiro.
* O método parseInt converte a string fornecida como parâmetro no respectivo valor inteiro. Caso a conversão não seja possível, é gerada uma exceção do tipo NumberFormatException.

As repetições com uso do comando while garantem que as entradas conterão strings que representam valores inteiros. Isso é feito antes do uso dessas strings, no cálculo da soma.

# Criação de formulários com rótulos, campos e botões

Crie as classes **Formulario** e **LeitorBotoes**:

import javax.swing.\*;  
  
import java.awt.event.ActionListener;  
  
public class Formulario {

public static void main(String[] args) {  
 JFrame frame = new JFrame("Login");  
 frame.setSize(300, 150);  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel painel = new JPanel();  
 frame.add(painel);  
 *inserir\_componentes*(painel);  
  
 frame.setVisible(true);  
 }  
  
 private static void inserir\_componentes(JPanel painel) {  
  
 painel.setLayout(null);  
  
 // rótulos e campos  
  
 JLabel rotulo\_usuario = new JLabel("Usuário");  
 rotulo\_usuario.setBounds(10, 10, 80, 25);  
 painel.add(rotulo\_usuario);  
  
 JTextField campo\_usuario = new JTextField(20);  
 campo\_usuario.setBounds(100, 10, 160, 25);  
 painel.add(campo\_usuario);  
  
 JLabel rotulo\_senha = new JLabel("Senha");  
 rotulo\_senha.setBounds(10, 40, 80, 25);  
 painel.add(rotulo\_senha);  
  
 JPasswordField campo\_senha = new JPasswordField(20);  
 campo\_senha.setBounds(100, 40, 160, 25);  
 painel.add(campo\_senha);  
  
 // botões:  
  
 JButton botao\_login = new JButton("login");  
 botao\_login.setBounds(10, 80, 80, 25);  
 painel.add(botao\_login);  
  
 JButton botao\_cadastro = new JButton("cadastro");  
 botao\_cadastro.setBounds(180, 80, 80, 25);  
 painel.add(botao\_cadastro);  
  
 ActionListener leitor\_botoes = new LeitorBotoes();  
 botao\_login.addActionListener(leitor\_botoes);  
 botao\_cadastro.addActionListener(leitor\_botoes);  
 }  
}

import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
import javax.swing.JButton;  
import javax.swing.JOptionPane;  
  
public class LeitorBotoes implements ActionListener {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JButton origem = (JButton) e.getSource();  
 JOptionPane.*showMessageDialog*(origem, origem.getText() + " foi apertado");  
 }  
}

Compile e execute o projeto a partir da classe Formulario.

Forneça os dados solicitados e acione algum dos botões.

Feche a janela com título "Login".

Um formulário é construído por uma instância de JFrame e uma instância de JPanel, devidamente vinculadas.

O método inserir\_componentes criar instâncias de JLabel, JTextField, JPasswordField e JButton para construir o formulário. Observe os valores das coordenadas e das dimensões de cada componente que permitiram organizá-los dentro do frame.

A classe JTextField possui os métodos getText e setText para ler e escrever, respectivamente, uma string. (Não utilizado no código.)

A classe LeitorBotoes é responsável por detectar quando algum botão é apertado. Essa classe implementa a interface ActionListener que, por sua vez, especifica o método actionPerformed. A implementação desse método na classe LeitorBotoes define a ação correspondente ao evento de acionamento de algum botão. Esse evento é fornecido como parâmetro do método e permite identificar o botão que o originou (chamada do método getSource).

Todo botão de um formulário tem associada uma instância de LeitorBotoes.

**REFERÊNCIAS**:

http://www.edu4java.com/en/swing/swing0.html