
Introdução Aplicações Gráficas (SWING)

Werther

Classes do AWT

- Primeiro conjunto de classes gráficas do Java: AWT = **Abstract Window Toolkit**
 - Conjunto de Ferramentas para Abstração de Janelas
- Representam objetos e componentes de um programa visual, baseado em janelas
- Responsáveis pelo tratamento de diversos aspectos deste tipo de interface visual
- Herdeiras de **Component**, superclasse abstrata geradora de classes deste pacote
 - Todas as classes que utilizamos nas aplicações gráficas derivam destas classes (herdeiras)
 - Estão no **pacote java.awt**

Classes SWING

- Substituíram as classes **AWT**
 - Para corrigir problemas de aparência e comportamento de alguns objetos
 - Padrões de **Look and Feel**, independente de S.O.
- As classes *swing* (**pacote *javax.swing***) são derivadas de ***JComponent***
 - Que deriva também da classe ***Container*** do pacote *java.awt*
 - Exceto ***JFrame***, que é derivada de ***Frame*** (pacote *java.awt*)
- Algumas classes AWT, como ***Color***, ***Font***, ***Toolkit***, etc., ainda são utilizadas
 - ***Toolkit***: Classe que obtém informações sobre o sistema gráfico que está sendo utilizado a JVM
 - Ex.: Tamanho e resolução da tela, tamanho do cursor, área de transferência (clipboard), atalhos de menu, captura de imagens, eventos em fila, etc.

A Classe JFrame (SWING)

- Base sobre a qual serão desenvolvidas interfaces visuais em programas baseado em janelas
- Herdeira de *Frame* (AWT), porém mais complexa e com muitos outros recursos
- **RootPane** (painel raiz)
 - Permite mudanças na aparência e comportamento (*look and feel*) do JFrame
- **ContentPane** (painel de conteúdo)
 - Onde são instalados os componentes swing
 - As propriedades visuais são feitas neste painel

A Classe JFrame (SWING)

○ Exemplo (1):

```
import javax.swing.*; import java.awt.*;
public class Exemplo extends JFrame {
    public Exemplo() { // método construtor
        setTitle("Primeira Tela");
        setResizable(false);
        setSize(400,300);
        setLocation(100,100);
        getContentPane().setBackground(Color.blue);
    }
    public static void main (String[] args){
        Exemplo tela = new Exemplo(); // instancia o objeto tela
        tela.show(); // exibe a tela
    }
}
```

Introdução Aplicações Gráficas

Importando as classes *swing* e *awt*

Classe JFrame

Coloca o título na barra de título do JFrame

○ Exemplo (1):

```
import javax.swing.*; import java.awt.*;
```

```
public class Exemplo extends JFrame {
```

```
    public Exemplo() { // método construtor
```

```
        setTitle("Primeira Tela");
```

```
        setResizable(false);
```

```
        setSize(400,300);
```

```
        setLocation(100,100);
```

```
        getContentPane().setBackground(Color.blue);
```

```
    }
```

```
    public static void main (String[] args) {
```

```
        Exemplo tela = new Exemplo(); // instancia o objeto tela
```

```
        tela.show(); // exibe a tela
```

```
    }
```

```
}
```

Determina se a tela é redimensionável

Define tamanho da tela: largura x altura

Define a posição do frame na tela do computador

Define a cor de fundo da tela

A Classe JFrame

○ Exemplo (1):

```
import javax.swing.*; import java.awt.*;

public class Exemplo extends JFrame {

    public Exemplo() { // método construtor
        setTitle("Primeira Tela");
        setResizable(false);
        setSize(400,300);
        setLocation(100,100);
        getContentPane().setBackground(Color.blue);
    }

    public static void main (String[] args) {
        Exemplo tela = new Exemplo(); // instancia o objeto tela
        tela.show(); // exibe a tela
    }
}
```



A Classe JFrame (SWING)

○ Exemplo (2):

```
import javax.swing.*; import java.awt.*;
public class W_Frame extends JFrame {
    public W_Frame() {
        setTitle("Descobrindo JFrame");
        setBounds(200,150,400,300);
        getContentPane().setBackground(Color.cyan.darker());
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    public static void main(String args[]) {
        new W_Frame().show();
    }
}
```


Introdução Aplicações Gráficas

A Classe JFrame (SWING)

Classe W_Frame
derivada de **JFrame**

Exemplo (2):

Coloca a String na barra de
título do JFrame

```
import javax.swing.*; import java.awt.*;
public class W_Frame extends JFrame {
    public W_Frame() {
        setTitle("Descobrimos JFrame");
        setBounds(200,150,400,300);
        getContentPane().setBackground(Color.cyan.darker());
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    public static void main(String args[]) {
        new W_Frame().show();
    }
}
```

Define a posição e o tamanho
simultaneamente, ao invés de usar
também *setPosition* e *setSize*

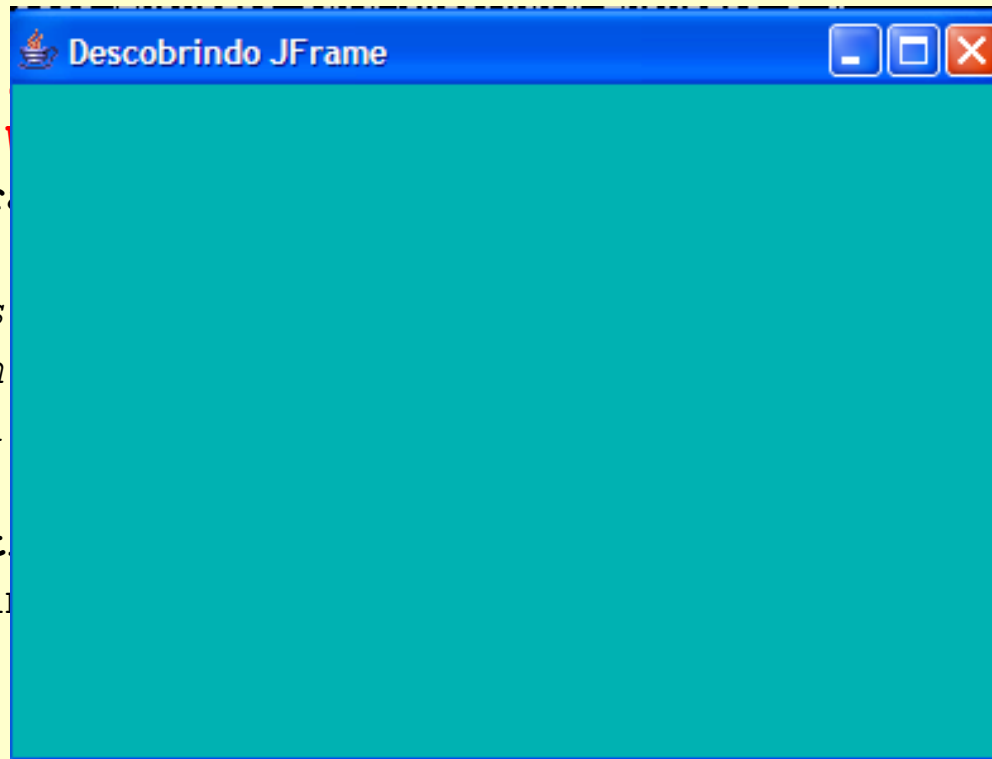
Define a operação padrão
quando é acionado o botão
de fechar da janela

HIDE_ON_CLOSE
DO_NOTHING_ON_CLOSE
DISPOSE_ON_CLOSE
EXIT_ON_CLOSE

A Classe JFrame (SWING)

○ Exemplo (2):

```
import javax.swing.*;
public class W_Frame {
    public W_Frame() {
        setTitle("Descobrindo JFrame");
        setBounds(100, 100, 400, 300);
        getContentPane().add(new JLabel("Hello World"));
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public static void main(String[] args) {
        new W_Frame().setVisible(true);
    }
}
```



```
marker());
```

A Classe JFrame (SWING)

○ Exemplo (3):

```
import javax.swing.*; import java.awt.*;
public class W_Frame extends JFrame {
    public W_Frame() {
        ...
        getRootPane().setBorder(
            BorderFactory.createLineBorder(Color.red, 3)
        );
        getRootPane().setWindowDecorationStyle(JRootPane.FRAME);
        setUndecorated(true);
    }
    public static void main(String args[]) {
        new W_Frame().setVisible(true);
    }
}
```

A Classe JFrame (SWING)

○ Exemplo (3):

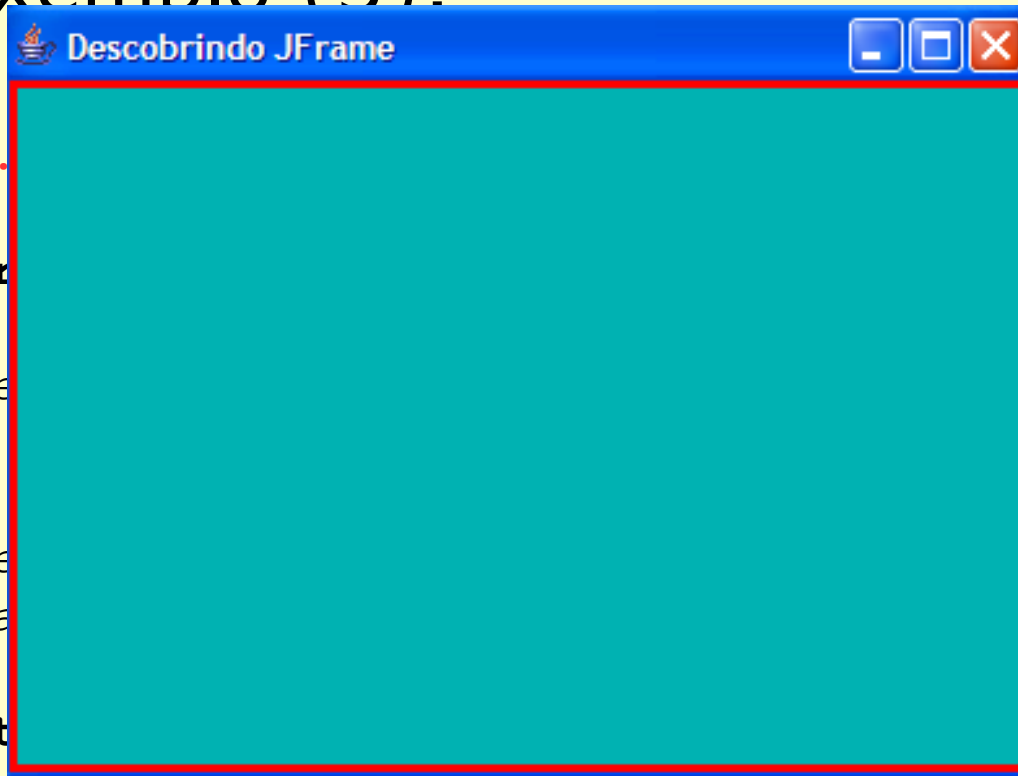
```
import javax.swing.*; import java.awt.*;
public class W_Frame extends JFrame {
    public W_Frame() {
        ...
        getRootPane().setBorder(
            BorderFactory.createLineBorder(Color.red, 3)
        );
        getRootPane().setWindowDecorationStyle(JRootPane.FRAME);
        setUndecorated(true);
    }
    public static void main(String args[]) {
        new W_Frame().setVisible(true);
    }
}
```

Definindo borda
através do painel raiz

A Classe JFrame (SWING)

○ Exemplo (3):

```
import javax.swing.*;
public class W_Frame
{
    ...
    getRootPane()
    ...
    getRootPane()
    setUndecorated(true);
}
public static void main(String[] args)
{
    new W_Frame().setVisible(true);
}
```



A Classe JFrame (SWING)

○ Exemplo (3):

```
import javax.swing.*; import java.awt.*;
public class W_Frame extends JFrame {
    public W_Frame() {
        ...
        getRootPane().setBorder(
            BorderFactory.createLineBorder(Color.red, 3)
        );
        getRootPane().setWindowDecorationStyle(JRootPane.FRAME);
        setUndecorated(true);
    }
    public static void main(String args[]) {
        new W_Frame().setVisible(true);
    }
}
```

Definindo estilo da janela,
através do painel raiz

NONE
FRAME
INFORMATION_DIALOG
ERROR_DIALOG
COLOR_CHOOSER_DIALOG
FILE_CHOOSER_DIALOG
QUESTION_DIALOG
WARNING_DIALOG

A Classe JFrame (SWING)

○ Exemplo (3):

```
import javax.  
public class
```

```
public W_Fr
```

```
...
```

```
getRootPane
```

```
getRootPane
```

```
setUndecora
```

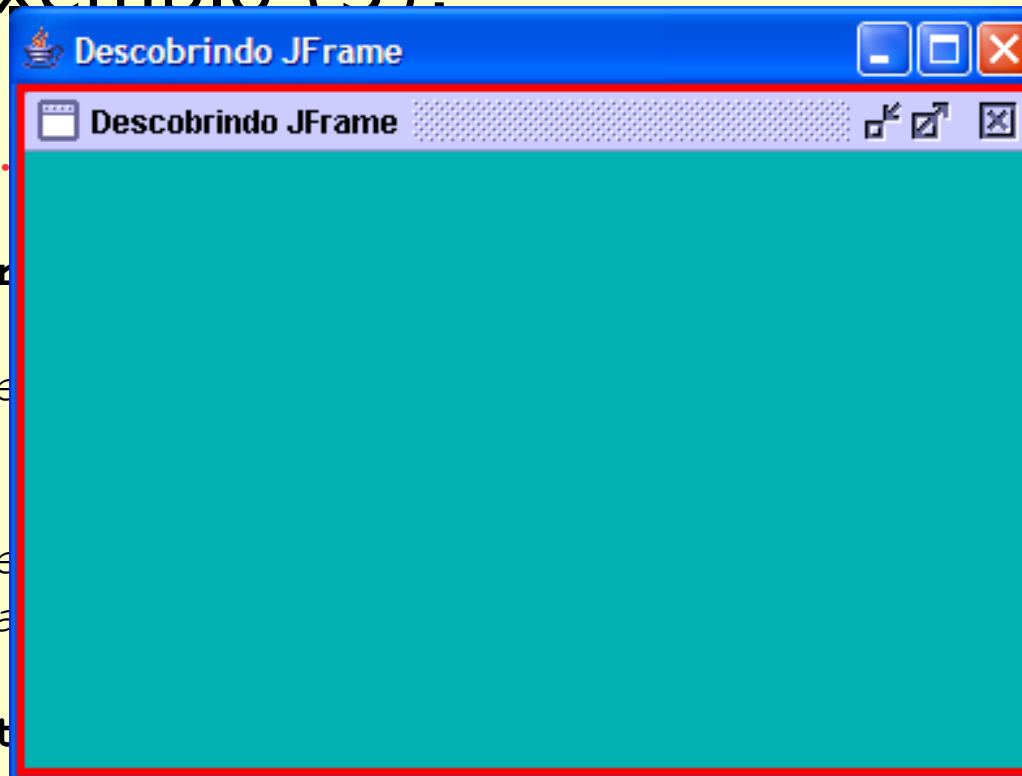
```
}
```

```
public stat
```

```
new W_Frame().setVisible(true);
```

```
}
```

```
}
```



```
color.red, 3)
```

```
ne.FRAME);
```

A Classe JFrame (SWING)

○ Exemplo (3):

```
import javax.swing.*; import java.awt.*;
public class W_Frame extends JFrame {
    public W_Frame() {
        ...
        getRootPane().setBorder(
            BorderFactory.createLineBorder(Color.red, 3)
        );
        getRootPane().setWindowDecorationStyle(JRootPane.FRAME);
        setUndecorated(true);
    }
    public static void main(String args[]) {
        new W_Frame().setVisible(true);
    }
}
```

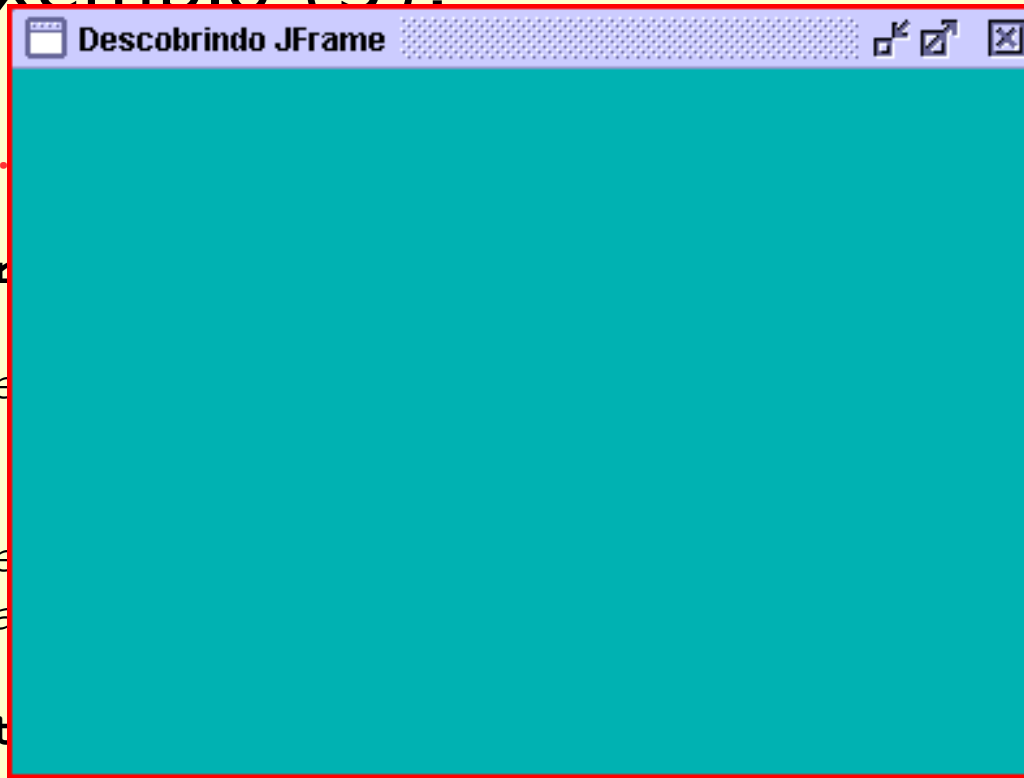
Desativa a barra de título e as bordas do **JFrame**, além do painel raiz

A Classe JFrame (SWING)

○ Exemplo (3):

```
import javax.swing.*;
public class W_Frame {
    public W_Frame() {
        ...
        getRootPane().setBackground(Color.red, 3);

        getRootPane().setUndecorated(true);
    }
    public static void main(String[] args) {
        new W_Frame().setVisible(true);
    }
}
```



A Classe JFrame (SWING)

○ Exemplo (3):

```
import javax.swing.*; import java.awt.*;
public class W_Frame extends JFrame {
    public W_Frame() {
        ...
        getRootPane().setBorder(
            BorderFactory.createLineBorder(Color.red, 3)
        );
        getRootPane().setWindowDecorationStyle(JRootPane.FRAME);
        setUndecorated(true);
    }
    public static void main(String args[]) {
        new W_Frame().setVisible(true);
    }
}
```

Exibindo a tela, tornando-a visível. Uso recomendado em relação ao *show()*

Classe Color (AWT)

- Gerencia o uso das cores em java
- Cores predefinidas:
 - *black* (preto); *blue* (azul); *cyan* (azul-esverdeado); *gray* (cinza); *darkGray* (cinza escuro); *lightGray* (cinza claro); *magenta* (roxo); *orange* (laranja); *pink* (rosa); *red* (vermelho); *white* (branco); *yellow* (amarelo)

`Color vermelho = Color.red;`

- Possui métodos:
 - *brighter()* – um tom mais claro da cor definida
`Color azulClaro = Color.blue.brighter();`
 - *darker()* – um tom mais escuro da cor definida
`Color vermelhoEscuro = Color.red.darker();`

Classe Color (AWT)

- É possível definir cores além dos padrões
 - Baseado no padrão RGB

`Color novaCor = new Color(r, g, b);`

- Onde **r**, **g** e **b** possuem valores entre 0 e 255
 - **r**: Padrão *red*
 - **g**: padrão *green*
 - **b**: padrão *blue*
 - Valores extremos:
 - 0, 0, 0 = preto (ausência de cor)
 - 255, 255, 255 = branco (todas as cores)

Classe Font (AWT)

- Administra as fontes
 - Tipo, tamanho, estilo
 - Exemplo básico de utilização:

...

```
JLabel L1 = new JLabel("Exemplo de Fonte");  
letra = new Font("Verdana", Font.PLAIN, 16);  
L1.setFont(letra);
```

...

Introdução Aplicações Gráficas

Classe Font (A

- Administra as
 - Tipo, tamanho
 - Exemplo básico

...

```
JLabel L1 = new JLabel("Exemplo de Fonte");  
letra = new Font("Verdana", Font.PLAIN, 16);  
L1.setFont(letra);
```

...

Para obter fonte:

```
System.out.println("From Toolkit:\n");  
Toolkit t = Toolkit.getDefaultToolkit();  
String s[] = t.getFontList();  
for (int i = 0; i < s.length; i++)  
    System.out.println(s[i]);  
  
System.out.println("\nFrom GraphicsEnvironment:\n");  
GraphicsEnvironment ge =  
    GraphicsEnvironment.getLocalGraphicsEnvironment();  
Font[] fs = ge.getAllFonts();  
for (int j = 0; j < fs.length; j++)  
    System.out.println(fs[j].getFontName() +  
        " (" + fs[j].getFamily() + ")");
```

Classe Font (AWT)

- Administra as fontes

- Tipo, tamanho, estilo
- Exemplo

...

```
JLabel L1 =
```

```
letra = new
```

```
L1.setFont(1
```

...



Introdução Aplicações Gráficas

A Classe JLabel

- Utilizado para acrescentar títulos, rótulos e mensagens fixas em uma janela.

```
import javax.swing.*; import java.awt.*;
public class ExLabel extends W_Frame {
    public ExLabel() {
        super();
        setTitle("Adicionando Rotulos");
        setBounds(200,150,400,300);
        Font f = new Font("SansSerif",
                          Font.BOLD +
                          Font.ITALIC,18);

        JLabel lb1 = new JLabel();
        lb1.setText("Este é um \"JLabel\"");
        lb1.setSize(250,15);
        lb1.setFont(f);
        lb1.setLocation(60,50);
        lb1.setForeground(Color.blue.darker());
```

```
        JLabel lb2 = new JLabel();
        lb2.setText("Este é outro \"JLabel\"");
        lb2.setBounds(80,150,300,15);
        lb2.setFont(new Font("Monospaced",
                             Font.PLAIN,16));
        lb2.setForeground(Color.red.darker());

        getContentPane().setLayout(null);
        getContentPane().add(lb1);
        getContentPane().add(lb2);
    }
    public static void main(String args[]) {
        new ExLabel().show();
    }
}
```


Introdução Aplicações Gráficas

A Classe JLabel

- Utilizado para mensagens fixas, rótulos e

Executando o construtor de *W_Frame*

```
import javax.swing.*; import java.awt.*;
public class ExLabel extends W_Frame {
    public ExLabel() {
        super();
        setTitle("Adicionando Rotulos");
        setBounds(200,150,400,300);
        Font f = new Font("SansSerif",
                           Font.BOLD +
                           Font.ITALIC,18);

        JLabel lb1 = new JLabel();
        lb1.setText("Este é um \"JLabel\"");
        lb1.setSize(250,15);
        lb1.setFont(f);
        lb1.setLocation(60,50);
        lb1.setForeground(Color.blue.darker());
    }
}
```

Classe exemplo derivada de *W_Frame*, classe que definimos no exemplo anterior

A definição de Layout *null* é para o painel de conteúdo

```
JLabel lb2 = new JLabel();
lb2.setText("Este é outro \"JLabel\"");
lb2.setBounds(80,150,300,15);
lb2.setFont(new Font("Monospaced",
                      Font.PLAIN,16));
lb2.setForeground(Color.red.darker());

getContentPane().setLayout(null);
getContentPane().add(lb1);
getContentPane().add(lb2);
}

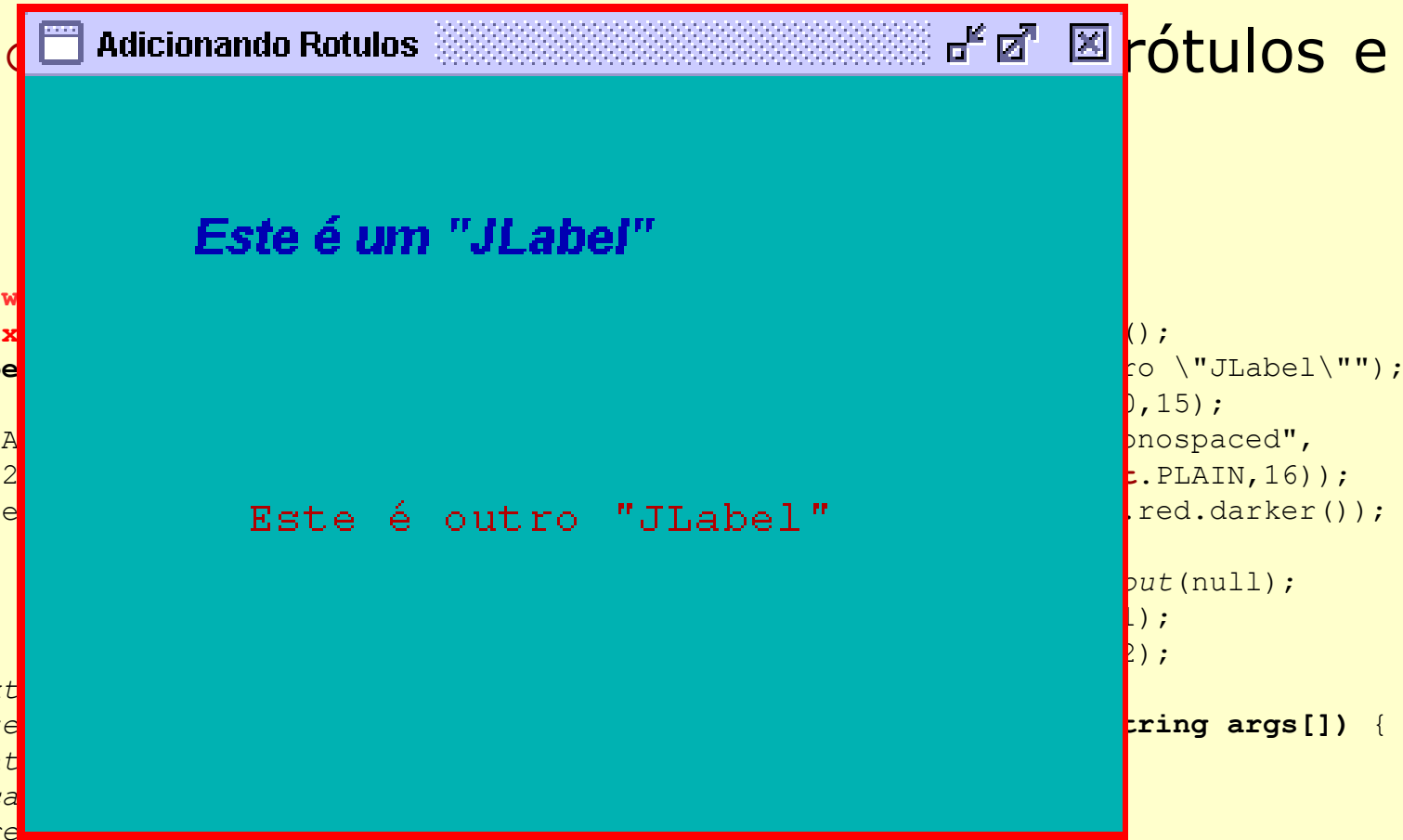
public static void main(String args[]) {
    new ExLabel().show();
}
```

Instanciando um objeto de *JLabel* e definindo suas propriedades

Adicionando objetos visuais ao painel de conteúdo

Introdução Aplicações Gráficas

A Classe JLabel



A Classe JTextField

- Gera objetos que são caixas para digitação
 - Texto em **uma única linha**
- Métodos principais:
 - `setSize(int largura, int altura)` ou `setSize(Dimension dim)`
 - `setLocation(int x, int y)`
 - `setBounds(int x, int y, int largura, int altura)`
 - `setText(String texto)`
 - `setFont(Font fonte)`
 - `setForeground(Color corEmPrimeiroPlano)`
 - `setBackground(Color corDeFundoPlano)`
 - `getText()`

Introdução Aplicações Gráficas

A Classe JTextField

```
import javax.swing.*; import java.awt.*;
public class ExJTextField extends JFrame {
    public JTextField tx1, tx2;
    public JLabel l1, l2, l3;

    public ExJTextField() {
        setTitle("Adicionando Campos de Texto");
        setBounds(200,150,400,300);
        Font f1 = new Font("SansSerif",Font.ITALIC,14);
        Font f2 = new Font("SansSerif",Font.ITALIC,10);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        l1 = new JLabel();
        l1.setText("Ficha de Cadastro");
        l1.setBounds(105,25,200,15);

        l2 = new JLabel("Digite seu Nome :");
        l2.setFont(f1);
        l2.setForeground(Color.black);
        l2.setBounds(10,50,120,15);

        tx1 = new JTextField();
        tx1.setFont(f2);
        tx1.setText("Seu Nome :");
        tx1.setLocation(140,50);
        tx1.setSize(200,20);

        l3 = new JLabel("Digite seu e-mail :");
        l3.setFont(f1);
        l3.setForeground(Color.black);
        l3.setBounds(10,90,120,15);

        tx2 = new JTextField("Seu e-mail :");
        tx2.setFont(f2);
        tx2.setBackground(Color.yellow);
        tx2.setForeground(Color.red);
        tx2.setBounds(140,90,200,20);
        tx2.setToolTipText
            ("Use apenas letras minúsculas");

        getContentPane().setLayout(null);
        getContentPane().add(l1);
        getContentPane().add(l2);
        getContentPane().add(l3);
        getContentPane().add(tx1);
        getContentPane().add(tx2);
    }

    public static void main(String args[]) {
        new ExJTextField().show();
    }
}
```

Adicionando uma “dica”
ao campo Tx2

Introdução Aplicações Gráficas

A classe JTextField

Definindo atributos
(objeto) da classe

Instanciando outro
objeto de
JTextField com
outro método
construtor

```
import javax.swing.*; import java.awt.*;
public class ExJTextField extends JFrame {
    public JTextField tx1, tx2;
    public JLabel l1, l2, l3;

    public ExJTextField() {
        setTitle("Adicionando Campos de Texto");
        setBounds(200,150,400,300);
        Font f1 = new Font("SansSerif",Font.ITALIC,14);
        Font f2 = new Font("SansSerif",Font.ITALIC,10);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        l1 = new JLabel();
        l1.setText("Ficha de Cadastro");
        l1.setBounds(105,25,200,15);

        l2 = new JLabel("Digite seu Nome :");
        l2.setFont(f1);
        l2.setForeground(Color.black);
        l2.setBounds(10,50,120,15);

        tx1 = new JTextField();
        tx1.setFont(f2);
        tx1.setText("Seu Nome :");
        tx1.setLocation(140,50);
        tx1.setSize(200,20);
```

Instanciando um
objeto de
JTextField e
definindo suas
propriedades

```
l3 = new JLabel("Digite seu e-mail :");
l3.setFont(f1);
l3.setForeground(Color.black);
l3.setBounds(10,90,120,15);
```

```
tx2 = new JTextField("Seu e-mail :");
tx2.setFont(f2);
tx2.setBackground(Color.yellow);
tx2.setForeground(Color.red);
tx2.setBounds(140,90,200,20);
tx2.setToolTipText
    ("Use apenas letras minúsculas");
```

```
getContentPane().setLayout(null);
getContentPane().add(l1);
getContentPane().add(l2);
getContentPane().add(l3);
getContentPane().add(tx1);
getContentPane().add(tx2);
```

```
}
public static void main(String args[]) {
    new ExJTextField().show();
}
```

Adicionando objetos visuais ao
painel de conteúdo

Introdução Aplicações Gráficas

A Classe JTextField

```
import javax.swing.*; import java.awt.*;
```

```
public class ExJText
```

```
    public JTextField  
    public JLabel l1,
```

```
public ExJTextFiel
```

```
    setTitle("Adicio
```

```
    setBounds(200,15
```

```
    Font f1 = new Fo
```

```
    Font f2 = new Fo
```

```
    setDefaultCloseOperation
```

```
    l1 = new JLabel (
```

```
    l1.setText("Fich
```

```
    l1.setBounds(105
```

```
    l2 = new JLabel (
```

```
    l2.setFont(f1);
```

```
    l2.setForeground
```

```
    l2.setBounds(10,
```

```
    tx1 = new JTextE
```

```
    tx1.setFont(f2);
```

```
    tx1.setText("Seu
```

```
    tx1.setLocation(140,50);
```

```
    tx1.setSize(200,20);
```

Adicionando Campos de Texto

Ficha de Cadastro

Digite seu Nome :

Seu Nome :

Digite seu e-mail :

Seu e-mail :

Use apenas letras minúsculas

Seu e-mail :");

ack);

);

e-mail : ");

ellow);

ed);

20);

inúsculas");

t(null);

ing args[]) {

;

Introdução Aplicações Gráficas

A Classe JPasswordField

- Derivada de **JTextField**
 - Específico para introdução de campos de senha
- Máscara default → ***** (*asterisco*)
 - Método *setEchoChar*(**char** caractereQueApareceNoCampo)

```
import javax.swing.*; import java.awt.*;
public class ExPass extends W_Frame {
    public JPasswordField sx ;
    public JLabel l1;

    public ExPass() {
        super();
        setTitle("Adicionando Campos de Senha");

        l1 = new JLabel();
        l1.setText("Digite sua senha:");
        l1.setBounds(20,120,200,15);
```

```
        sx = new JPasswordField();
        sx.setBounds(140,120,200,20);
        sx.setToolTipText("Escreva aqui sua senha");
        sx.setEchoChar('X');

        getContentPane().setLayout(null);
        getContentPane().add(l1);
        getContentPane().add(sx);
    }
    public static void main(String args[]) {
        new ExPass().show();
    }
}
```

Introdução Aplicações Gráficas

A Classe JPasswordField

- Derivada de **JTextField**
 - Específico para introdução de campos de senha
- Máscara default → ***** (*asterisco*)
 - Método `setEchoChar(char caractereQueAparece)`

Atribuindo caractere **X** como máscara de senha deste campo

Executando construtor de **W_Frame**

Instanciando objeto de **JPasswordField**

```
import javax.swing.*; import java.awt.*;
public class ExPass extends W_Frame {
    public JPasswordField sx ;
    public JLabel l1;

    public ExPass() {
        super();
        setTitle("Adicionando Campos de Senha");

        l1 = new JLabel();
        l1.setText("Digite sua senha:");
        l1.setBounds(20,120,200,15);
    }
}
```

```
sx = new JPasswordField();
sx.setBounds(140,120,200,20);
sx.setToolTipText("Escreva aqui sua senha");
sx.setEchoChar('X');
```

```
getContentPane().setLayout(null);
getContentPane().add(l1);
getContentPane().add(sx);
}
```

```
public static void main(String args[]) {
    new ExPass().show();
}
```

Adicionando objetos visuais ao painel de conteúdo

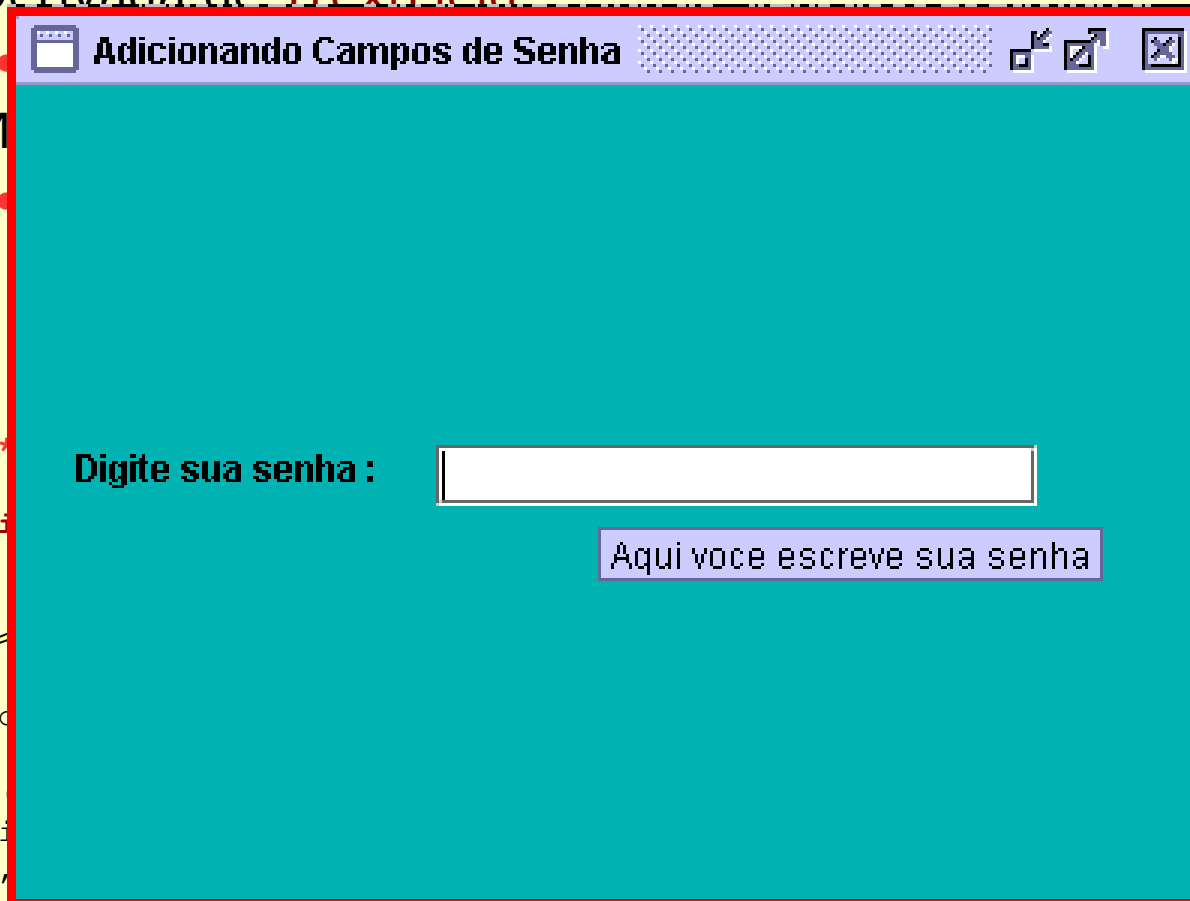
Introdução Aplicações Gráficas

A Classe JPasswordField

- Derivada de `JTextField`

- M

```
import javax.swing.*;  
public class ExPass  
    public JPasswordField  
    public JLabel l1;  
  
    public ExPass() {  
        super();  
        setTitle("Adicionando Campos de Senha")  
  
        l1 = new JLabel("Digite sua senha :");  
        l1.setBounds(20, 580, 200, 620);  
    }  
}
```



ampo)

sua senha");

);

gs[])

A Classe JTextArea

- Derivada também de **JTextField**
 - Permite digitação em toda a sua área e não somente em uma linha
- Alguns métodos:
 - *setWrapLine*(**boolean** wrap)
 - Se verdadeiro, a digitação, quando chega ao final da linha, continua na próxima linha
 - Default: falso
 - *setWrapStyleWord*(**boolean** wordWrap)
 - Se verdadeiro, durante a digitação, a palavra que não couber no final da linha é transportada para a próxima linha
 - Default: falso
 - *setEditable*(**boolean** visível)
 - Se falso, não permite edição no seu conteúdo
 - Default: verdadeiro

Introdução Aplicações Gráficas

A Classe JTextArea

```
import javax.swing.*; import java.awt.*;
public class WExTextArea extends JFrame {
    public JTextArea ta;

    public WExTextArea() {
        setTitle("Usando a classe JTextArea");
        setBounds(100,100,500,400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        ta = new JTextArea();
        ta.setBounds(10,40,450,300);
        ta.setBorder(BorderFactory.createLoweredBevelBorder());
        ta.setBackground(Color.blue);
        ta.setForeground(Color.white);
        ta.setLineWrap(true);

        getContentPane().setLayout(null);
        getContentPane().add(ta);
    }
    public static void main(String[] args) {
        new WExTextArea().setVisible(true);
    }
}
```

Instanciando um objeto de *JTextArea*

Definindo borda do objeto

Permitindo *wrap* de linhas

Equivalente ao método *show()*, que já está depreciado

Introdução Aplicações Gráficas

A Classe JTextArea

Instanciando um objeto de *JTextArea*

```
import java.awt.*;  
public class JAreaExemplo {  
    public J
```

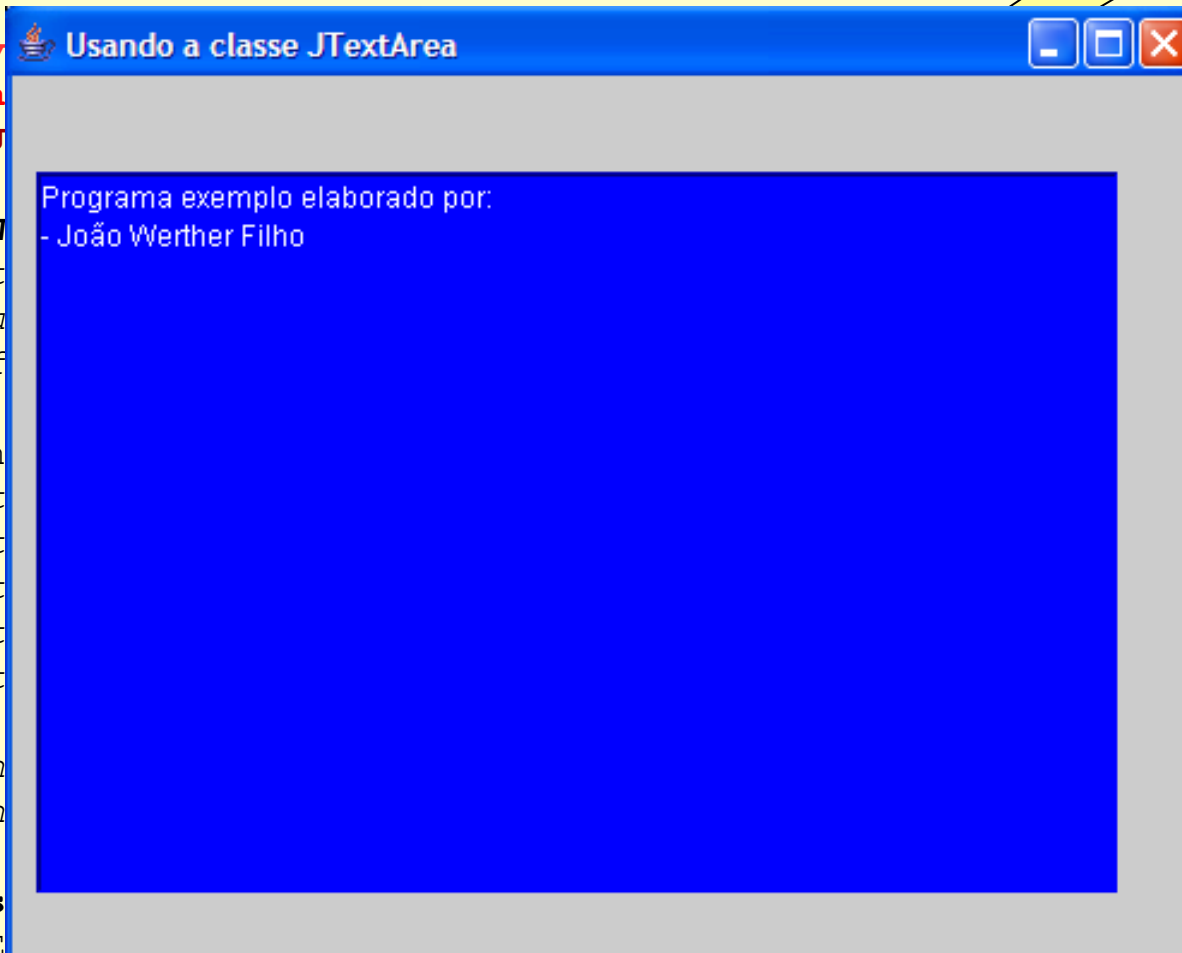
```
    public WindowExemplo() {  
        setTitle("Exemplo JTextArea");  
        setBounds(100, 100, 400, 300);  
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
        setDefaultLookAndFeelDefaults();  
        JTextArea ta = new JTextArea(10, 20);  
        ta.setLineWrap(true);  
        ta.setWrapStyleWord(true);  
        ta.setCaretColor(Color.red);  
        ta.setFont(new Font("Serif", Font.PLAIN, 14));  
        getContentPane().add(ta, BorderLayout.CENTER);  
        getCon
```

```
        ta = new JTextArea(10, 20);  
        ta.setLineWrap(true);  
        ta.setWrapStyleWord(true);  
        ta.setCaretColor(Color.red);  
        ta.setFont(new Font("Serif", Font.PLAIN, 14));  
        getContentPane().add(ta, BorderLayout.CENTER);  
        getCon
```

```
        getContentPane().add(ta, BorderLayout.CENTER);  
        getCon
```

```
    }
```

```
    public static void main(String[] args) {  
        new WindowExemplo().setVisible(true);  
    }
```



Adicionando borda ao objeto

Permitindo *wrap* de linhas

Equivalente ao *show()*, que depreciado

A Classe JButton

- Responsável pela criação de botões de ação na interface de janelas
- Alguns métodos importantes
 - *setEnabled*(**boolean** disponivel);
 - *setMnemonic*(**int** atalho)
 - Proveniente da classe **KeyEvent**
 - *addActionListener*(**ActionListener** ouvinte)
 - Para ouvir eventos de ação

Introdução Aplicações Gráficas

A Classe JButton

○ Exemplo (1):

```
import java.awt.*; import javax.swing.*;
import javax.swing.border.*;
public class WExJButton extends W_Frame {
    public JButton b1, b2;

    public ExJTextField() {
        setTitle("Acrescentando botões");

        b1 = new JButton();
        b1.setText("Sair");
        b1.setBounds(50,200,100,25);
        b1.setBackground(Color.orange);
        b1.setForeground(Color.blue);
        b1.setBorder(
            BorderFactory.createBevelBorder(
                BevelBorder.LOWERED,
                Color.white,Color.black)
            );
        b1.setEnabled(true);
        b1.setToolTipText("saindo do sistema");
```

```
        b2 = new JButton("Cadastrar");
        b2.setBounds(200,200,100,25);
        b2.setBorder(
            BorderFactory.createLineBorder(
                Color.black,2)
            );
        b2.setEnabled(true);
        b2.setToolTipText("indo ao cadastro");

        getContentPane().setLayout(null);
        getContentPane().add(b1);
        getContentPane().add(b2);
    }

    public static void main(String args[]) {
        new WExJButton().setVisible(true);
    }
}
```

Introdução Aplicações Gráficas

A Classe JButton

Exemplo (1):

Instanciando objeto
b1 de **JButton**

Derivando de
W_Frame

Definindo atributos
(objetos) da classe

Instanciando objeto
b2 de **JButton**

Definindo
borda de **b2**

Definindo
borda de **b1**

Deixando o
botão **b1**
disponível

```
import java.awt.*; import javax.swing.*;
import javax.swing.border.*;
public class WExJButton extends W_Frame {
    public JButton b1, b2;

    public ExJTextField() {
        setTitle("Acrescentando botões");

        b1 = new JButton();
        b1.setText("Sair");
        b1.setBounds(50,200,100,25);
        b1.setBackground(Color.orange);
        b1.setForeground(Color.blue);
        b1.setBorder(
            BorderFactory.createBevelBorder(
                BevelBorder.LOWERED,
                Color.white,Color.black
            );
        b1.setEnabled(true);
        b1.setToolTipText("saindo do sistema");
    }
}
```

```
b2 = new JButton("Cadastrar");
b2.setBounds(200,200,100,25);
b2.setBorder(
    BorderFactory.createLineBorder(
        Color.black,2
    );
b2.setEnabled(true);
b2.setToolTipText("indo ao cadastro");

getContentPane().setLayout(null);
getContentPane().add(b1);
getContentPane().add(b2);

public static void main(String args[]) {
    new WExJButton().setVisible(true);
}
```

Introdução Aplicações Gráficas

A Classe JButton

Instanciando objeto

b1 de *JButton*

Acrescentando botões

Instanciando objeto

b2 de *JButton*

```
import java.awt.*;
import javax.swing.*;

public class WExJButton {
    public JButton b1;

    public ExJTextField
        setTitle("Acrescentando botões");

    b1 = new JButton("Sair");
    b1.setText("Sair");
    b1.setBounds(50, 750, 150, 100);
    b1.setBackground(Color.white);
    b1.setForeground(Color.black);
    b1.setBorder(
        new BorderFactory.
            BevelBorder(
                BorderFactory.
                    Color.white
            );
    b1.setEnabled(true);
    b1.setToolTipText("Saindo do sistema");
```



Deixando o
botão **b1**
disponível

A Classe JOptionPane

- Contém diversos painéis de aviso (caixas de mensagem) para o usuário
 - Favorecendo uma interação mais rápida e imediata
- Existem 4 tipos possíveis de painéis, chamados por seus métodos:
 - **JOptionPane.showMessageDialog** → abre um painel exibindo alguma informação útil ao usuário
 - **JOptionPane.showConfirmDialog** → abre painel para confirmação de uma operação
 - **JOptionPane.showInputDialog** → abre painel com uma caixa de texto para entrada de dados
 - **JOptionPane.showOptionDialog** → abre um painel personalizado, fazendo a junção das características dos painéis anteriores

A Classe JOptionPane

- Todos os métodos descritos trabalham com os seguintes parâmetros configuráveis:
 - Mensagem: texto que vai aparecer no painel
 - Título: texto que aparece no título do painel
 - Botões: os botões que aparecerão no painel
 - **JOptionPane.YES_NO_OPTION** ou 0
 - **JOptionPane.YES_NO_CANCEL_OPTION** ou 1
 - **JOptionPane.OK_CANCEL_OPTION** ou 2
 - **JOptionPane.DEFAULT_OPTION** ou -1

A Classe JOptionPane

- Ícone: ícone representativo do painel



- **JOptionPane.ERROR_MESSAGE** ou 0;

- Usado para mensagens de erro



- **JOptionPane.INFORMATION_MESSAGE** ou 1;

- Usado para mensagens de caráter informativo



- **JOptionPane.WARNING_MESSAGE** ou 2;

- Usado para mostrar avisos



- **JOptionPane.QUESTION_MESSAGE** ou 3;

- Usado para perguntas e questões

- **JOptionPane.PLAIN_MESSAGE** ou -1;

- Indica que a mensagem não possui ícone associado

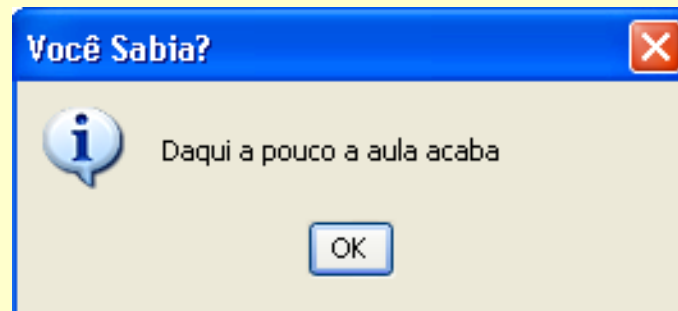
A Classe JOptionPane

- Método ***showMessageDialog***
 - Para mensagens ao usuário. Exemplos:

```
JOptionPane.showMessageDialog(null, "Daqui a pouco a aula acaba", "Você Sabia?",  
                               JOptionPane.INFORMATION_MESSAGE);
```

Ou

```
JOptionPane.showMessageDialog(null, "Daqui a pouco a aula acaba", "Você Sabia?", 1);
```



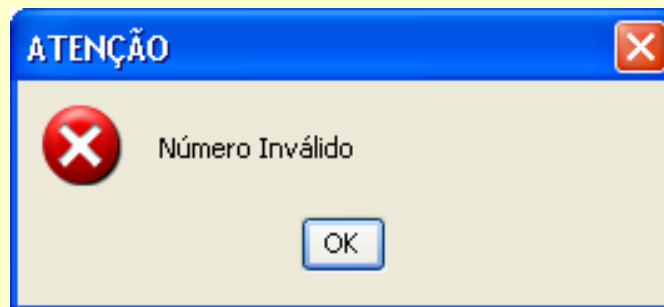
A Classe JOptionPane

- Método ***showMessageDialog***
 - Para mensagens ao usuário. Exemplos:

```
JOptionPane.showMessageDialog(null, "Número Inválido", "ATENÇÃO",  
                                JOptionPane.ERROR_MESSAGE);
```

Ou

```
JOptionPane.showMessageDialog(null, "Número Inválido", "ATENÇÃO", 0);
```



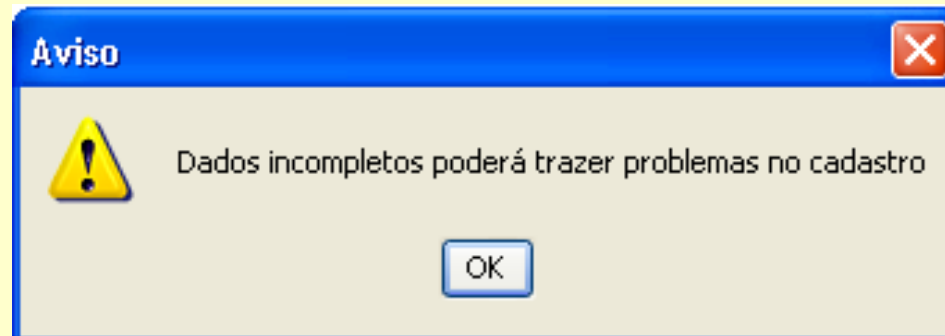
A Classe JOptionPane

- Método ***showMessageDialog***
 - Para mensagens ao usuário. Exemplos:

```
JOptionPane.showMessageDialog(null,  
                                "Dados incompletos poderá trazer problemas no cadastro",  
                                "Aviso",JOptionPane.WARNING_MESSAGE);
```

ou

```
JOptionPane.showMessageDialog(null,  
                                "Dados incompletos poderá trazer problemas no cadastro",  
                                "Aviso",2);
```



Introdução Aplicações Gráficas

A Classe JOptionPane

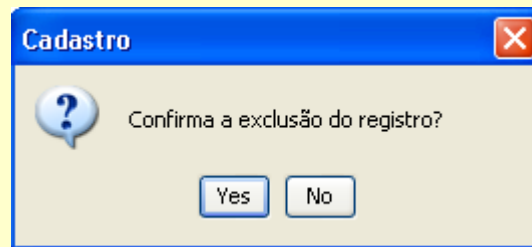
resp é uma variável do tipo *int*, que vai ser utilizado para armazenar a resposta do usuário

- Método ***showConfirmDialog***
 - Para diálogos de confirmação. Exemplos:
 - Botões utilizados:
 - **JOptionPane.YES_NO_OPTION** ou 0
 - **JOptionPane.YES_NO_CANCEL_OPTION** ou 1
 - Exemplos:

```
int resp = JOptionPane.showConfirmDialog(null, "Confirma a exclusão do registro?",  
                                         "Cadastro", JOptionPane.YES_NO_OPTION);
```

ou

```
int resp = JOptionPane.showConfirmDialog(null, "Confirma a exclusão do registro?", "Cadastro", 0);
```



ATENÇÃO:

Se o usuário clicar em “**YES**”, o valor de *resp* é **0** → **JOptionPane.YES_OPTION**

Se o usuário clicar em “**NO**”, o valor de *resp* é **1** → **JOptionPane.NO_OPTION**

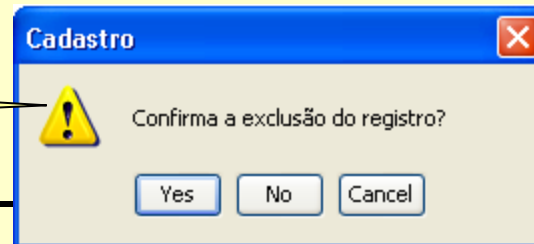
Introdução Aplicações Gráficas

A Classe JOptionPane

- Método ***showConfirmDialog***
 - Para diálogos de confirmação. Exemplos:
 - Botões utilizados:
 - **JOptionPane.YES_NO_OPTION** ou 0
 - **JOptionPane.YES_NO_CANCEL_OPTION** ou 1
 - Exemplos:

```
int resp = JOptionPane.showConfirmDialog(null, "Confirma a exclusão do registro?", "Cadastro",  
                                         JOptionPane.YES_NO_CANCEL_OPTION, JOptionPane.WARNING_MESSAGE);  
ou  
int resp = JOptionPane.showConfirmDialog(null, "Confirma a exclusão do registro?", "Cadastro", 1, 2);
```

Note a mudança do ícone



ATENÇÃO:

Se o usuário clicar em “**YES**”, o valor de *resp* é **0** → **JOptionPane.YES_OPTION**

Se o usuário clicar em “**NO**”, o valor de *resp* é **1** → **JOptionPane.NO_OPTION**

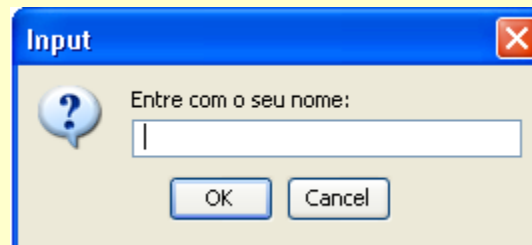
Se o usuário clicar em “**CANCEL**”, o valor de *resp* é **2** → **JOptionPane.CANCEL_OPTION**

A Classe JOptionPane

- Método ***showInputDialog***
 - Para diálogos de entrada de dados
 - Exemplos:

```
String nome = JOptionPane.showInputDialog(null, "Entre com o seu nome:");
```

nome é um objeto da classe ***String*** que vai ser utilizado para armazenar a entrada de dados do usuário



A Classe JOptionPane

- Método ***showInputDialog***
 - Para diálogos de entrada de dados
 - Exemplos:

```
String nome = JOptionPane.showInputDialog(null, "Entre com o seu nome:",  
                                           "Cadastro", JOptionPane.INFORMATION_MESSAGE);  
ou  
String nome = JOptionPane.showInputDialog(null, "Entre com o seu nome:", "Cadastro", 1);
```

Note a mudança do
ícone e do título

