



GUI

Graphical User Interface

Jose.wellington@uniceub.br

Calendário

◀ agosto de 2013 ▶

D	S	T	Q	Q	S	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
-	-	-	-	-	-	-

◀ setembro de 2013 ▶

D	S	T	Q	Q	S	S
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

◀ outubro de 2013 ▶

D	S	T	Q	Q	S	S
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

◀ novembro de 2013 ▶

D	S	T	Q	Q	S	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

◀ dezembro de 2013 ▶

D	S	T	Q	Q	S	S
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Agenda

- **GUI - Graphical User Interface**
- **Container**
- **JOptionPane**
 - Estrutura
 - showMessageDialog
 - showConfirmDialog
 - showInputDialog
- **JFrame**
- **Layout Frame**
 - BorderLayout
 - FlowLayout
 - GridLayout
- **Controles**
- **Exercício**

GUI

Graphical User Interface

GUI – Graphical User Interface

- ❑ Uma interface gráfica do usuário permite que os usuários interagem com seu computador através de ícones e um ponteiro em vez de digitando no texto na linha de comando.
- ❑ Possuem muitas funcionalidades para prover a comunicação **homem-máquina gráfica**



The screenshot shows a graphical user interface window titled "Candidato a emprego". The window has a menu bar with "Arquivo", "Formatar", and "Sobre". The main content area is titled "Ficha de Avaliação" and contains several form fields:

- Identificação:** A text field for "Nome" containing "nome do candidato" and radio buttons for "Sexo" with options "Feminino" and "Masculino".
- Curriculum Vitae:** A large text area with a placeholder text "Digite aqui os dados do Curriculum Vitae resumido".
- Áreas:** Two dropdown menus labeled "Interesse" (with "Professor" selected) and "Atuação" (with "Informática" selected).

At the bottom of the window, there is a green bar containing three buttons: "Documentos Entregues", "Salva", and "Cancela".

GUI – Graphical User Interface

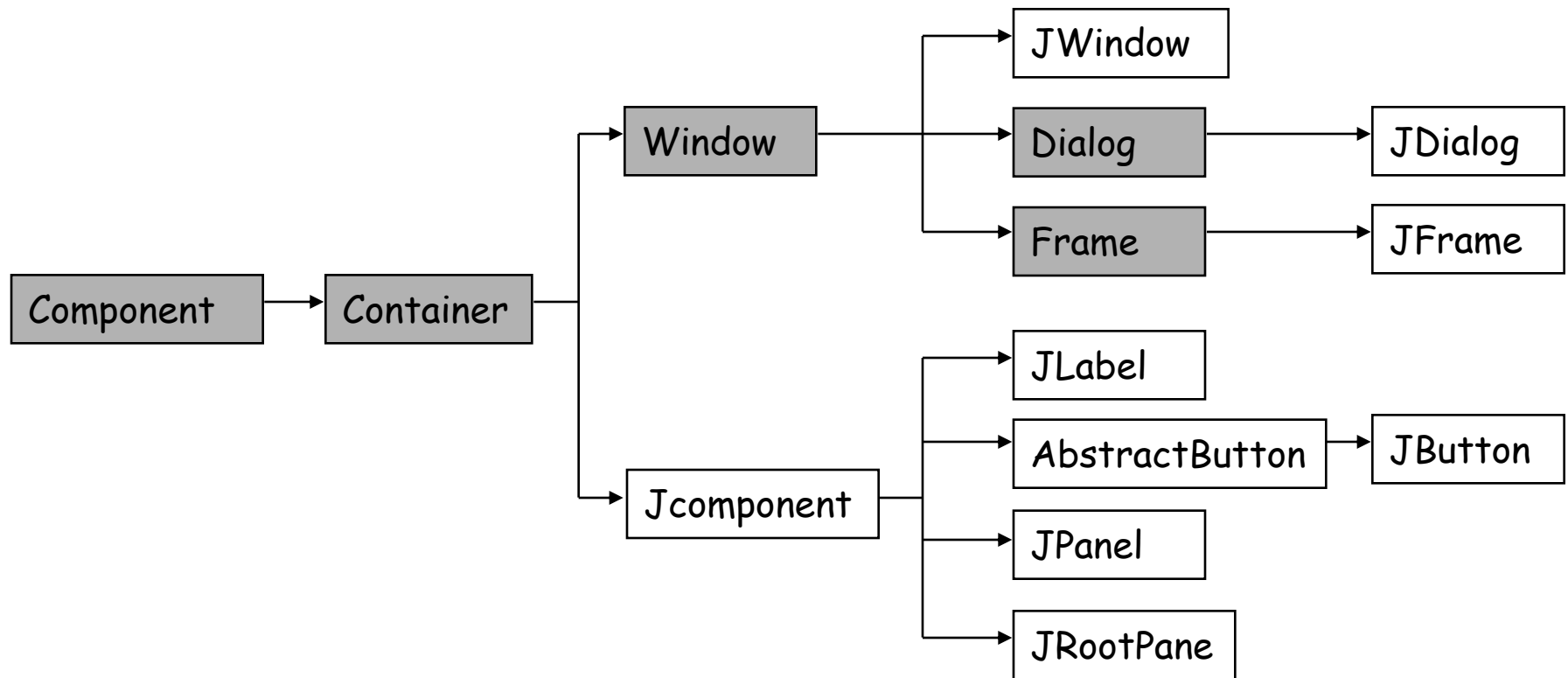
Os elementos básicos necessários para criar uma interface gráfica, ou GUI, residem em dois pacotes:

- **java.awt**: Abstract Windowing Toolkit (classes básicas)
- **javax.swing**: Swing Components (criado em 1997)

GUI – Graphical User Interface

- A **AWT (Abstract Windowing Toolkit)** sempre esteve presente na linguagem Java. Os seus defeitos a muito tempo são bem conhecidos e os desenvolvedores tem gasto muito tempo e esforço na criação de componentes.
- A meta do projeto **Swing (criado em 1997)** foi acrescentar novas funcionalidades, através de uma biblioteca de classes, a fim de suprir as demandas dos usuários.

GUI – Graphical User Interface



- *Os retângulos sombreados representam os controles da AWT.*

Container

GUI – Graphical User Interface

Uma interface gráfica em Java é baseada em dois elementos:

■ Containers

Contém um grupo de componentes (controles)

Servem para agrupar e exibir outros componentes

Exemplo: Panel, Dialog, controles e etc.

■ Controles

É um controle visual independente, como um botão de ação ou campo texto.

São os botões, text box, labels, scrollbars, etc.

GUI – Graphical User Interface

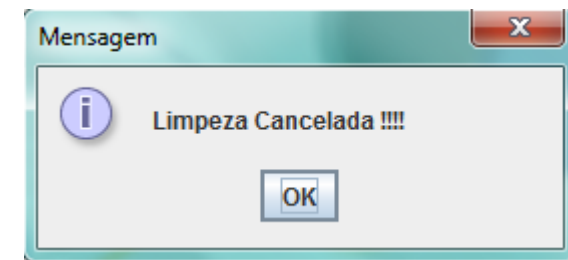
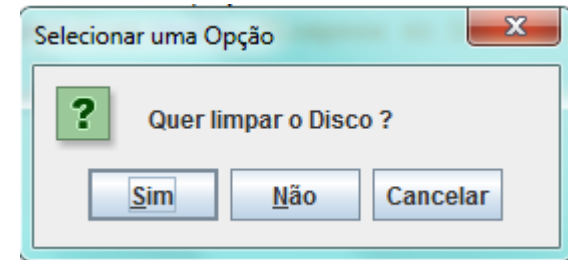
```
java.lang.Object
+--java.awt.Component
    +--java.awt.Container
        |
        +--javax.swing.JComponent
            |
            +--javax.swing.JButton
            +--javax.swing.JLabel
            +--javax.swing.JMenuBar
            +--javax.swing.JOptionPane
            +--javax.swing.JPanel
            +--javax.swing.JTextArea
            +--javax.swing.JTextField
        +--java.awt.Window
            +--java.awt.Frame
                +--javax.swing.JFrame
```

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

JOptionPane

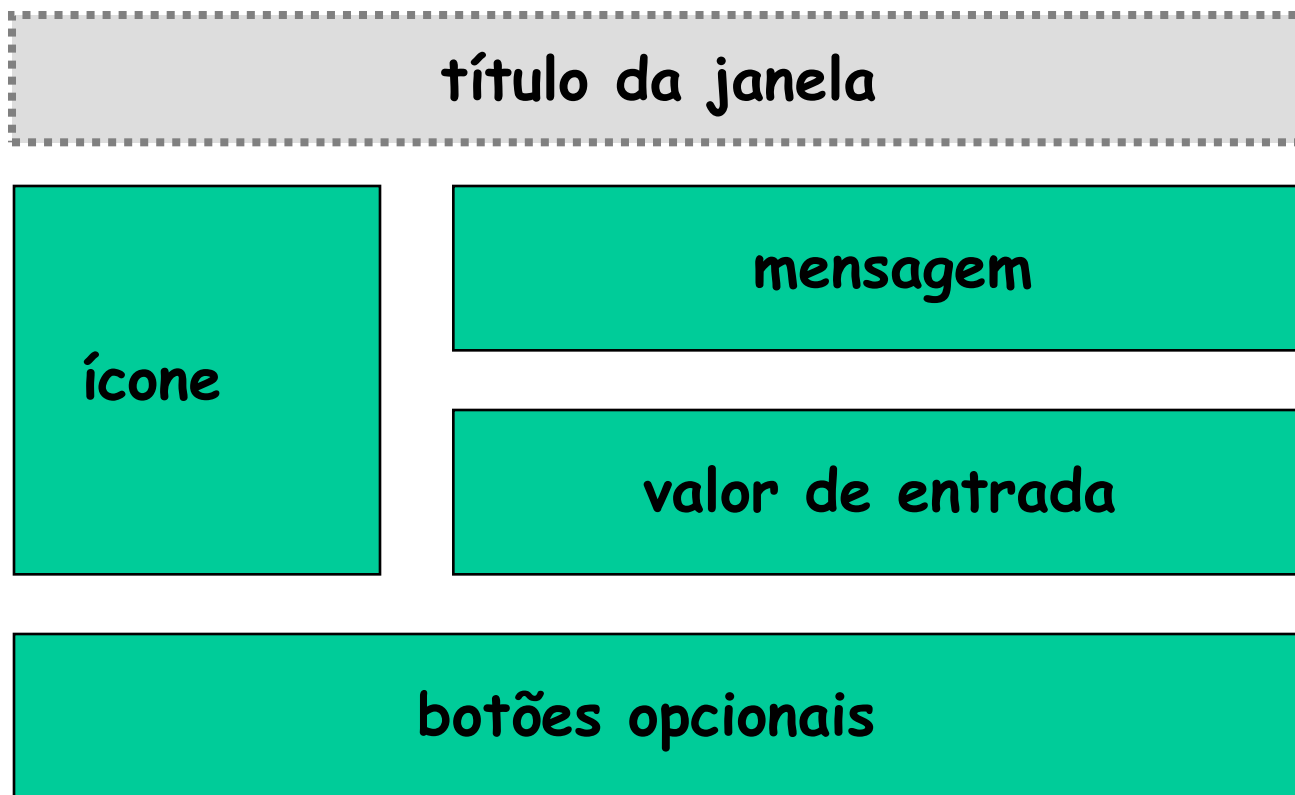
Classe JOptionPane

- Através da classe **JOptionPane** é fácil exibir uma caixa de diálogo padrão que induza o usuário a um valor ou o informa de alguma coisa.
- Na sua forma mais básica, ela exibe uma linha de mensagem e um botão rotulado com “OK”.
- Uma vez que o diálogo tenha sido exibido, ele fica na tela até o usuário pressionar o botão “OK”. Enquanto o diálogo continuar na tela, o usuário não poderá interagir com mais nada na aplicação.



Classe JOptionPane - Estrutura

- A aparência de uma das caixas de diálogo é similar à figura abaixo:



Classe JOptionPane

Constantes para tipos de mensagens:

ERROR_MESSAGE
INFORMATION_MESSAGE
WARNING_MESSAGE
QUESTION_MESSAGE
PLAIN_MESSAGE

Constantes para opções de botões:

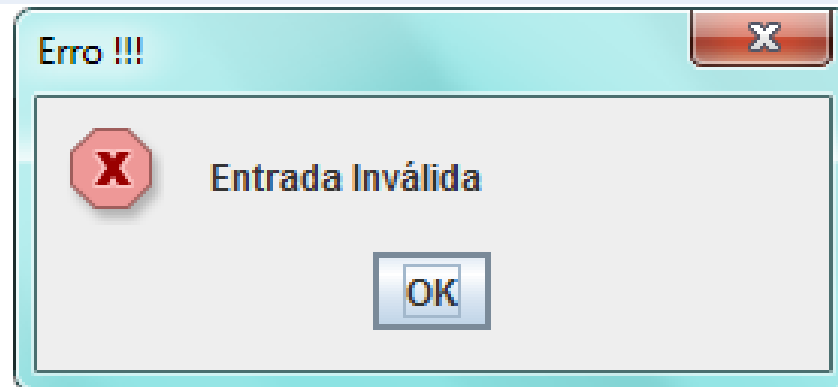
YES_NO_OPTION
YES_NO_CANCEL_OPTION
OK_CANCEL_OPTION

Constantes inteiras de retorno:

JOptionPane.YES_OPTION
JOptionPane.NO_OPTION
JOptionPane.CANCEL_OPTION
JOptionPane.OK_OPTION
JOptionPane.CLOSED_OPTION

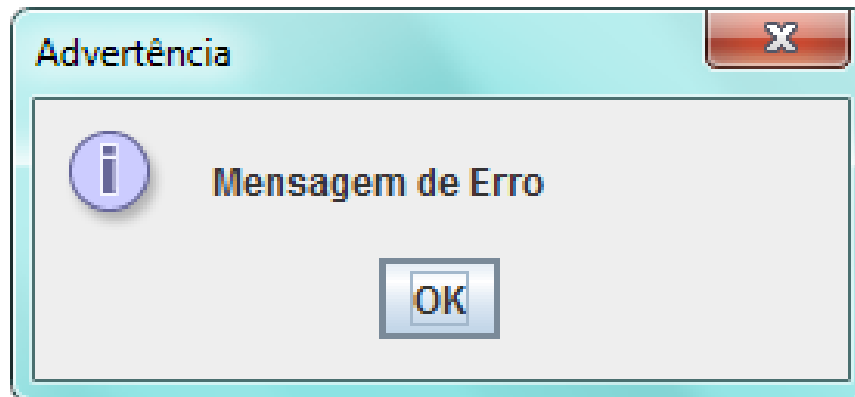
JOptionPane – showMessageDialog - exemplo 1

```
import javax.swing.*;  
public class Msg01 {  
  
    public static void main(String[] args) {  
  
        JOptionPane.showMessageDialog(null, "Entrada Inválida",  
            "Erro !!!", JOptionPane.ERROR_MESSAGE);  
    }  
}
```



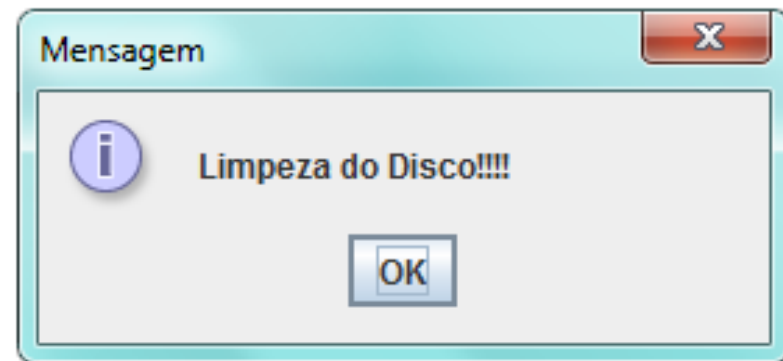
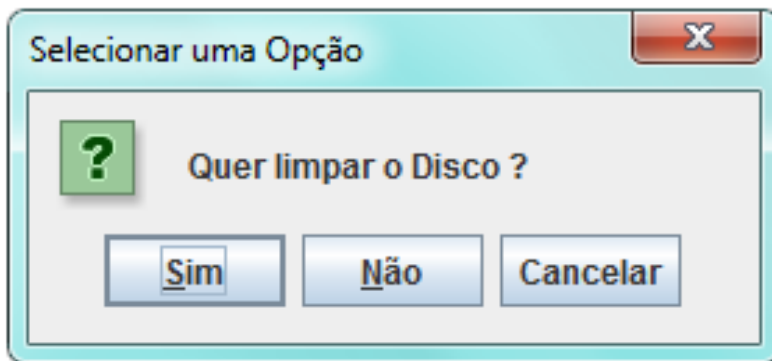
JOptionPane – showMessageDialog - exemplo 2

```
import javax.swing.*;  
public class Msg01 {  
  
    public static void main(String[] args) {  
  
        JOptionPane.showMessageDialog(null, "Mensagem de Erro",  
            "Advertência", JOptionPane.INFORMATION_MESSAGE);  
    }  
}
```



JOptionPane – showConfirmDialog - exemplo 3

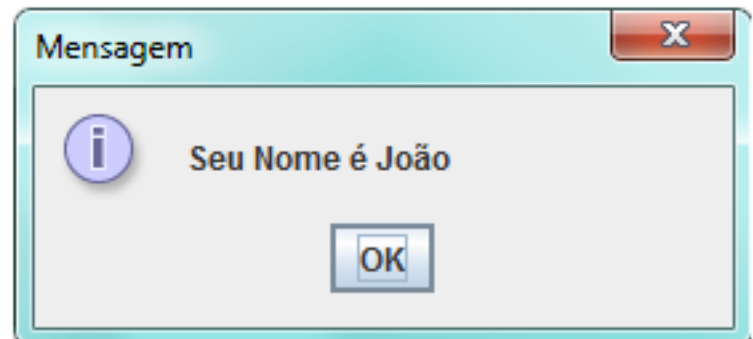
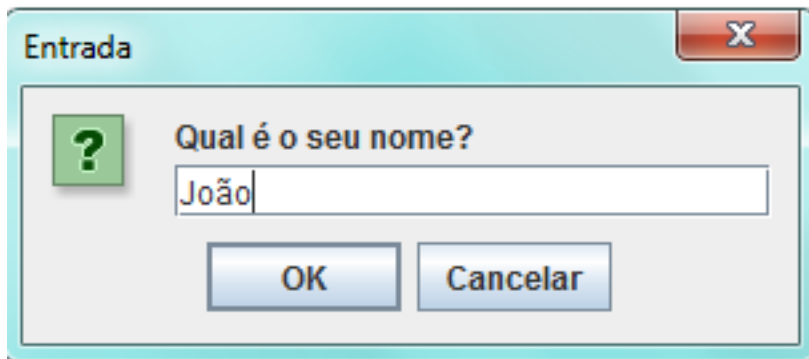
```
import javax.swing.*;
public class Msg {
    public static void main(String[] args) {
        int opcao = JOptionPane.showConfirmDialog (null,
            "Quer limpar o Disco ?");
        if (opcao == JOptionPane.YES_OPTION) {
            JOptionPane.showMessageDialog (null, "Limpeza do Disco!!!!");
        }
        else {
            JOptionPane.showMessageDialog (null, "Limpeza Cancelada !!!!!");
        }
    }
}
```



JOptionPane - `showInputDialog` exemplo 4

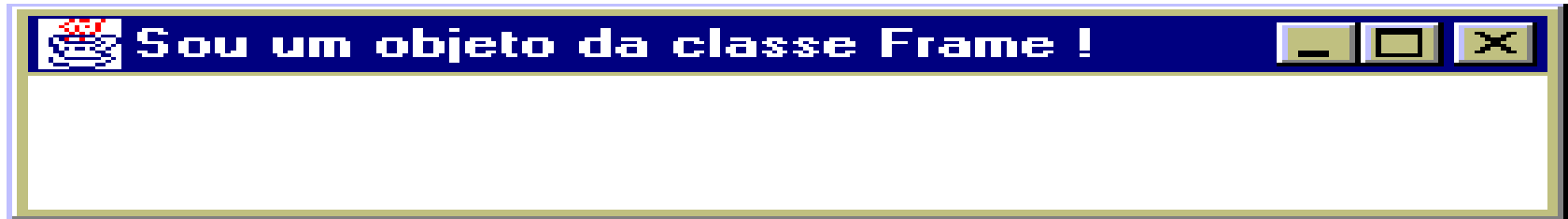
```
import javax.swing.*;
public class Msg01 {

    public static void main(String[] args) {
        String nome = JOptionPane.showInputDialog(null,
            "Qual é o seu nome?");
        JOptionPane.showMessageDialog(null, "Seu Nome é " + nome);
    }
}
```



JFrame

Frame



- ❑ Uma *Frame* é uma janela com uma *barra de título* e uma *borda*.
- ❑ A classe *Frame* especializa a classe *Window*, que por sua vez, especializa a classe *container*.
- ❑ Se uma aplicação tem uma janela dependente de outra (que desaparece quando a outra é iconificada, por exemplo), então deve-se utilizar *Dialog* ou *Window* para esta janela.

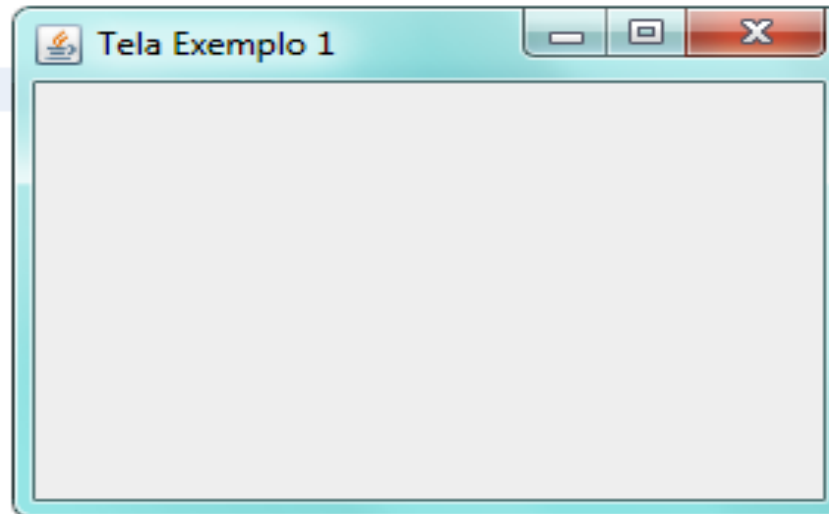
JFrame - exemplo 1

```
import javax.swing.*;
public class Frame {

    public static void main(String[] args) {

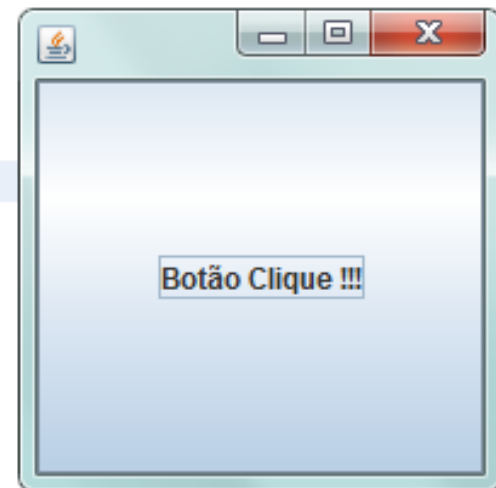
        JFrame tela = new JFrame("Tela Exemplo 1");

        tela.setSize(200,200);
        tela.setVisible(true);
    }
}
```



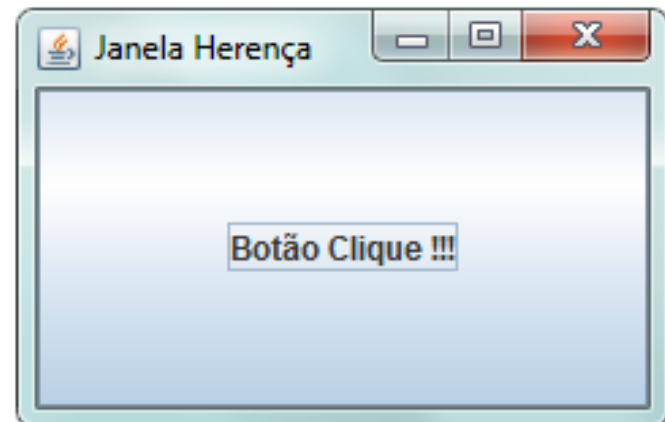
JFrame - exemplo 2

```
package janelas;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
  
public class Janela {  
  
    public Janela() {  
  
        JFrame frame = new JFrame();  
        JButton botao = new JButton("Botão Clique !!!");  
        frame.getContentPane().add(botao);  
        frame.setSize(200,200);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setVisible(true); }  
  
    public static void main(String[] args) {  
        new Janela(); }  
}
```



JFrame - exemplo 3

```
import javax.swing.JButton;  
import javax.swing.JFrame;  
public class Janela01 extends JFrame {  
  
    public Janela01() {  
        //JFrame frame = new JFrame();  
        super("Janela Herança");  
        JButton botao = new JButton("Botão Clique !!!");  
        getContentPane().add(botao);  
        setSize(200,200);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setVisible(true); }  
    public static void main(String[] args) {  
        new Janela01(); }  
}
```




Layout Frame

Layout


Layouts

 Absolute layout


 FlowLayout

 BorderLayout

 GridLayout


 GridBagLayout


 CardLayout

 BoxLayout

 SpringLayout

 FormLayout

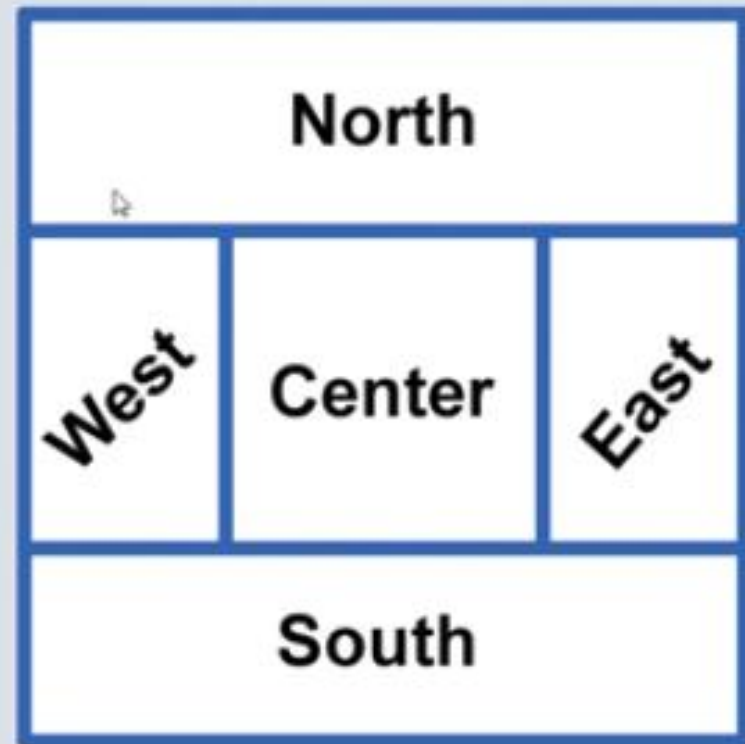
 MigLayout

 GroupLayout



Layout

BorderLayout



Layout BorderLayout

```
package layout;
import java.awt.BorderLayout;
import java.awt.Container;
import javax.swing.JButton;
import javax.swing.JFrame;
public class Layout extends JFrame{

    public Layout() {
        super("Layout");
        //getContentPane().add(new JButton("1"));

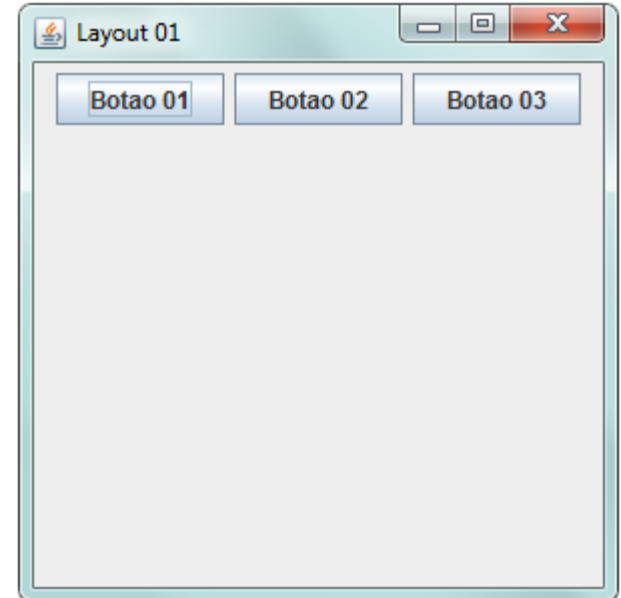
        Container c = getContentPane();
        c.add(BorderLayout.NORTH, new JButton(" 1 Norte"));
        c.add(BorderLayout.SOUTH, new JButton(" 2 Sul"));
        c.add(BorderLayout.CENTER, new JButton(" 3 Centro"));
        c.add(BorderLayout.EAST, new JButton(" 4 Leste"));
        c.add(BorderLayout.WEST, new JButton(" 5 Oeste"));
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300,300);
        setVisible(true); }

    public static void main(String[] args) {
        new Layout(); }
}
```



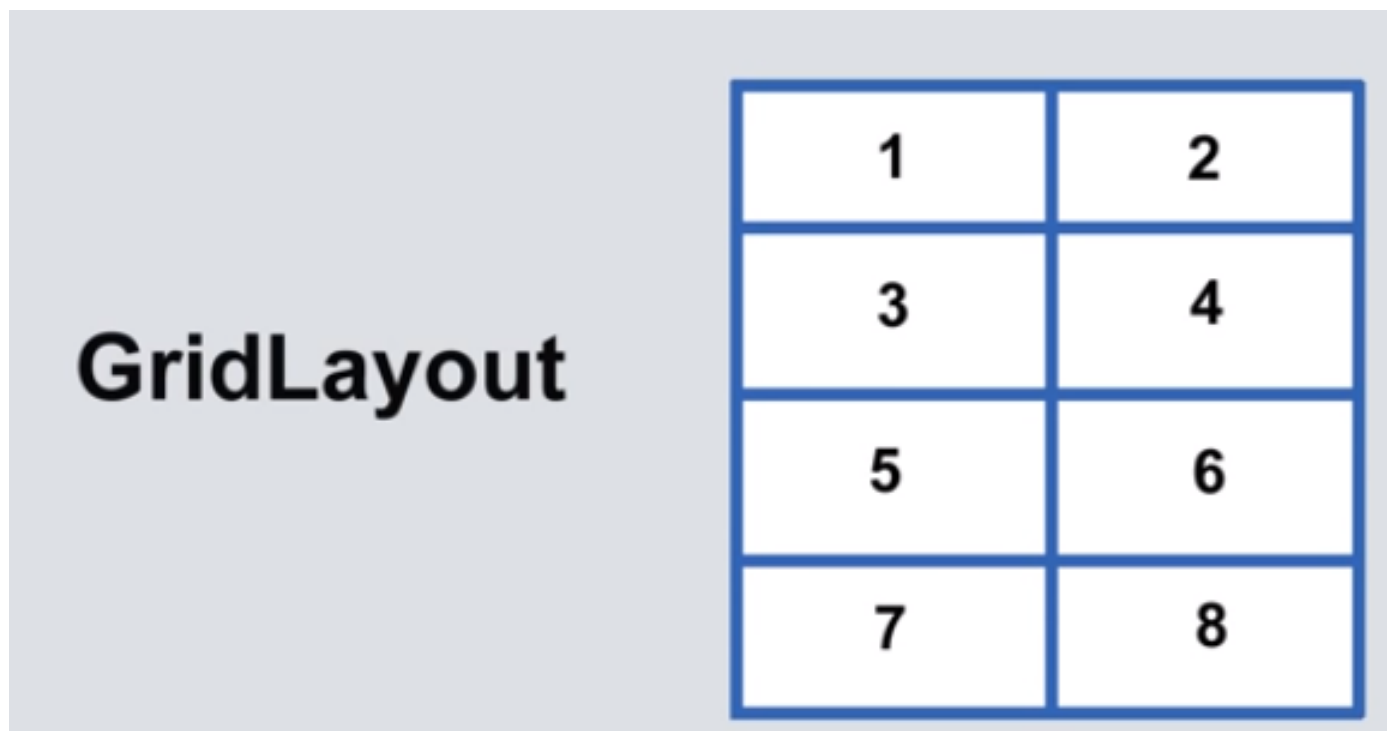
Layout FlowLayout

```
import java.awt.BorderLayout;  
import java.awt.Container;  
import java.awt.FlowLayout;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
  
public class Layout01 extends JFrame{  
  
    public Layout01() {  
        super("Layout 01");  
        Container c = getContentPane();  
        c.setLayout(new FlowLayout());  
  
        c.add(new JButton("Botao 01"));  
        c.add(new JButton("Botao 02"));  
        c.add(new JButton("Botao 03"));  
  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setSize(300,300);  
        setVisible(true); }  
    public static void main(String[] args) {  
        new Layout01(); }  
}
```



Layout **GridLayout**

Defini a quantidade de Linhas e Colunas



Layout GridLayout

Defini a quantidade de Linhas e Colunas

```
package layout02;
import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import javax.swing.*.*;

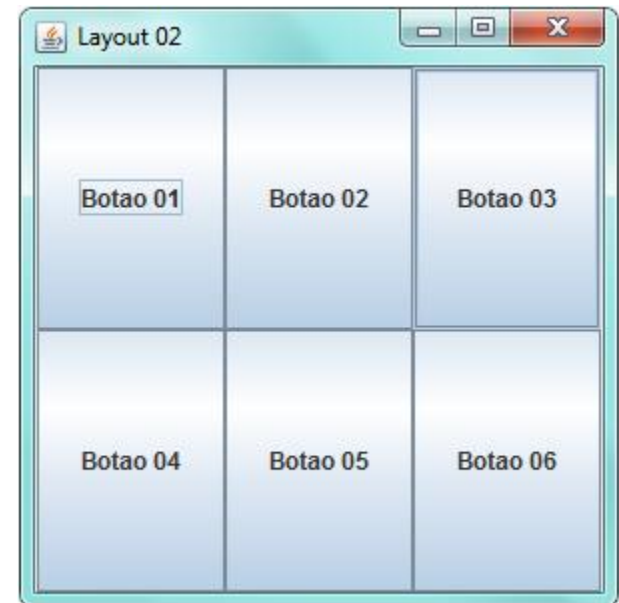
public class Layout02 extends JFrame {

    public Layout02() {
        super("Layout 02");

        Container c = getContentPane();
        c.setLayout(new GridLayout(2,3));
        c.add(new JButton("Botao 01"));
        c.add(new JButton("Botao 02"));
        c.add(new JButton("Botao 03"));
        c.add(new JButton("Botao 04"));
        c.add(new JButton("Botao 05"));
        c.add(new JButton("Botao 06"));

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300,300);
        setVisible(true); }





















    public static void main(String[] args) {
        new Layout02(); }
}
```



Controles

Controles

Controles javax.swing

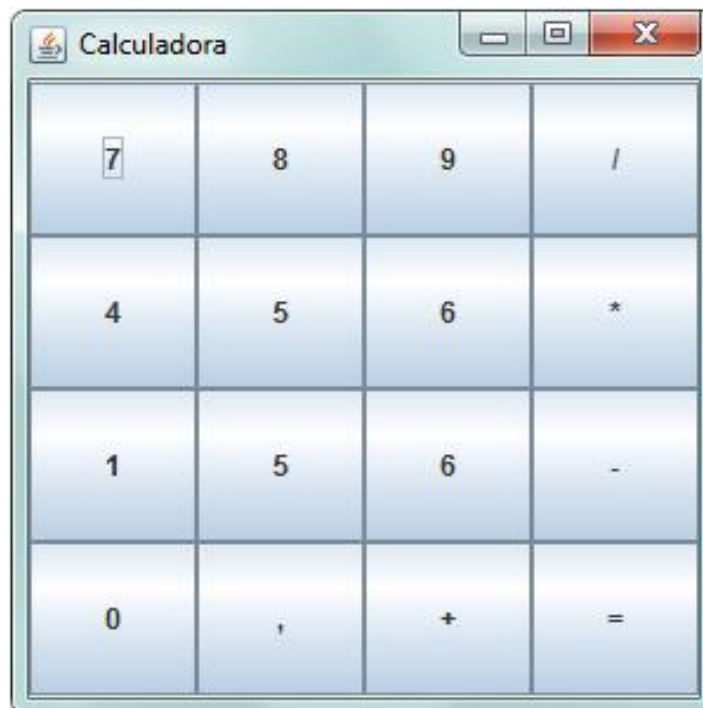
 JLabel	 JTextPane	 JTextArea
 JTextField	 JEditorPane	 JFormattedTextField
 JComboBox	 JSpinner	 JPasswordField
 JButton	 JList	 JScrollBar
 JCheckBox	 JTable	 JSeparator
 JRadioButton	 JTree	 JSlider
 JToggleButton	 JProgressBar	

Exercício

Exercício 22

Fazer um teclado de uma Calculadora

- 1) Usando GridLayout 4 x 4
- 2) Tamanho 200 x 200)



Agenda

- **GUI - Graphical User Interface**
- **Container**
- **JOptionPane**
 - Estrutura
 - showMessageDialog
 - showConfirmDialog
 - showInputDialog
- **JFrame**
- **Layout Frame**
 - BorderLayout
 - FlowLayout
 - GridLayout
- **Controles**
- **Exercício**