6 – Adicionando componentes JButton ou botões a sua janela

A classe JButton herda de AbstractButton, uma classe que herda de JComponet e define o comportamento básico para os botões e itens de menu. Como ocorre com instâncias da classe JLabel, objetos da classe JButton podem conter texto, texto e imagem ou apenas imagens.Seguindo os mesmo métodos dos outros aplicativos de como adicionar componentes na janela esse também segue o mesmo modelo.

6.1 - Botão somente com texto: Veja o exemplo de botão sem imagem:

```
import javax.swing.*;
import java.awt.*;
public class ExemploBotao  extends JFrame{
  JButton botão;
  public ExemploBotao(){
    super("Exemplo com JButton");
    Container tela = getContentPane();
    setLayout(null);
    botao = new JButton ("Procurar");
    botao.setBounds(50,20,100,20);
    tela.add(botao);
    setSize(400, 250);
    setVisible(true);
  }
  public static void main(String  args[]){
        ExemploBotao  app = new ExemploBotao();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```

6.2 - Botão com texto e imagem: Exemplo de um botão com imagem

```
import java.awt.*;
import java.awt.event.*;
public class ExemploBotao  extends JFrame{
  JButton botao;
  ImageIcon  icone;
  public ExemploBotao(){
    super("Exemplo com JButton");
    Container tela = getContentPane();
    setLayout(null);
    icone = new ImageIcon("abrir.gif");
    botao = new JButton ("Abrir",icone);
    botao.setBounds(50,20,100,20);
    tela.add(botao);
    setSize(300, 150);
```

```
    setVisible(true);
  }
  public static void main(String args[]){
        ExemploBotao app = new ExemploBotao();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```

Analisando o programa, declaramos o ícone como Imagelcon, depois adicionamos a figura a ícone, icone = new Imagelcon("abrir.gif"); no final adicionamos o ícone junto ao texto do botão, botao = new JButton ("Abrir",icone); e com o objeto tela, tela.add(botao); adicionamos o botão a janela.

6.3 - Botão somente com imagem: Exemplo de botão somente com figura

```
import java.awt.event.*;
public class ExemploBotao extends JFrame{
  JButton botao;
  Imagelcon icone;
  public ExemploBotao(){
    super("Exemplo com JButton");
    Container tela = getContentPane();
    setLayout(null);
    icone = new Imagelcon("abrir.gif");
    botao = new JButton (icone);
    botao.setBounds(50,20,100,20);
    tela.add(botao);

    setSize(400, 250);
    setVisible(true);
    setLocationRelativeTo(null);
  }

  public static void main(String args[]){
        ExemploBotao app = new ExemploBotao();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```

Muito simples fazer esse tipo de botão com o aplicativo anterior de botão com imagem basta você retirar o texto de dentro da linha que somente a imagem será adicionada, veja a linha:

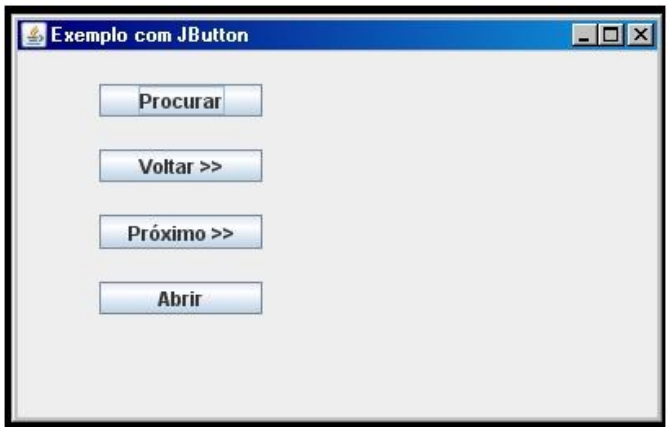
Linha do botão com texto e imagem: botão = new JButton (“Abrir”,icone);

Linha do botão somente com imagem: botão = new JButton (icone);

6.4 – Adicionando vários botões na janela

Veja o exemplo:

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

public class ExemploBotao extends JFrame{
  JButton botao1,botao2,botao3,botao4;
    public ExemploBotao(){
    super("Exemplo com JButton");
    Container tela = getContentPane();
    setLayout(null);
    botao1 = new JButton ("Procurar");
    botao2 = new JButton ("Voltar >>");
    botao3 = new JButton ("Próximo >>");
    botao4 = new JButton ("Abrir");

    botao1.setBounds(50,20,100,20);
    botao2.setBounds(50,60,100,20);
    botao3.setBounds(50,100,100,20);
    botao4.setBounds(50,140,100,20);

   tela.add(botao1);
   tela.add(botao2);
   tela.add(botao3);
   tela.add(botao4);
   setSize(400, 250);
   setVisible(true);
  }

  public static void main(String args[]){
        ExemploBotao app = new ExemploBotao();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```



6.5 – Definir a cor de fundo para um componente JButton

Para trocar a cor de fundo basta adicionar no aplicativo anterior as seguintes linhas:

```
botao1.setBackground(Color.yellow);
botao2.setBackground(Color.red);
botao3.setBackground(Color.blue);
botao4.setBackground(Color.white);
```

6.6 – Definir teclas de atalho para objetos JButton

Para definir as teclas de atalhos temos que importar o pacote:
import java.awt.event.*; no inicio do programa

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

E adicione as seguintes linhas dentro do aplicativo anterior:

```
botao1.setMnemonic(KeyEvent.VK_P);
botao2.setMnemonic(KeyEvent.VK_V);
botao3.setMnemonic(KeyEvent.VK_X);
botao4.setMnemonic(KeyEvent.VK_S);
```

6.7 – Qual botão reagirá ao pressionamento da tecla enter

Você pode passar o foco para um determinado botão assim que pressionar a tecla enter, Com o comando getRootPane().setDefaultButton(). Basta digitar o comando getRootPane().setDefaultButton(botao);

botão é o nome do botão que você definiu no programa, nesse exemplo definir o último botão.

Quando você pressionar a tecla Enter o botão que receberá o foco piscará.

Veja o resultado:

6.8 - Alterando a cor do texto de um componente JButton

Basta adicionar as seguintes linhas dentro do aplicativo:

```
botao1.setForeground(Color.blue);
botao2.setForeground(Color.green);
botao3.setForeground(Color.red);
botao4.setForeground(Color.pink);
```

Resultado do programa:

7 - Adicionando componentes JPasswordField

A classe JPasswordField possibilita a criação de caixas de texto para a digitação de senhas. Esta classe herda diretamente de JTextField e, portanto, permite a aplicação de todas as técnicas já vista na classe JTextField.

No aplicativo abaixo vamos ver como funciona esse componente:

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

public class ExemploJPasswordField extends JFrame{
  JPasswordField caixa;
  JLabel rotulo;
    public ExemploJPasswordField(){
    super("Exemplo com JPasswordField");
    Container tela = getContentPane();
    setLayout(null);
    rotulo = new JLabel("Senha: ");
    caixa = new JPasswordField(10);
    rotulo.setBounds(50,20,100,20);
    caixa.setBounds(50,60,100,20);
    tela.add(rotulo);
    tela.add(caixa);
    setSize(400, 250);
    setVisible(true);
  }

  public static void main(String args[]){
        ExemploJPasswordField app = new ExemploJPasswordField();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

  }
}
```

7.1 – Alterando a cor de fonte de um JPasswordField

Você poderá trocar a cor da letra da caixa de texto. No caso essa letra vai ficar azul.

```
caixa.setForeground(Color.blue);
```

7.2 – Definir uma cor personalizada para o componente JPasswordField

Você também pode definir uma cor para letra conforme seu gosto usando essa linha `caixa.setForeground(new Color(115,99,128));`

7.3 – Alterando a cor de fundo de um componente JPasswordField

Você pode digitar essa linha no aplicativo anterior:

```
caixa.setBackground(Color.yellow);
```

A caixa de texto dentro ficará com a cor amarela por dentro

7.4 – Cor de fundo personalizada para o componente JPasswordField

Ao digitar essa linha no aplicativo dentro da caixa de texto ficará com a cor rosa. Onde você mesmo pode definir sua cor.

```
caixa.setBackground(new Color(255,128,128));
```

7.5 – Alterar tipo de fonte, estilo e tamanho da letra do componente

Com essa linha você pode alterar o estilo da fonte o tipo e o tamanho da fonte.

```
caixa.setFont(new Font("Ariel",Font.BOLD,20));
```

Exemplos para essa fonte você pode aproveitar o exemplo do JTextField

7.6 – Alterando o caracter que aparece no componente JPasswordField

Você pode personalizar qual caracter vai aparecer na sua caixa de texto basta você digitar essa linha `caixa.setEchoChar('*');`

Dentro dos parênteses você pode adicionar qualquer caracter.

8 – Fazendo um JButton executar uma ação

8.1 – Clicando no botão para fechar uma janela

Exemplo:


```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class ExemploBotaoSair extends JFrame{
 JButton botaosair;
   public ExemploBotaoSair(){
   super("Exemplo com JButton");
   Container tela = getContentPane();
   setLayout(null);
   botaosair = new JButton ("Sair");
   botaosair.setBounds(100,50,100,20);
  botaosair.addActionListener(
        new ActionListener(){
              public void actionPerformed(ActionEvent e){
                    System.exit(0);
              }

        }
    );
     tela.add(botaosair);
     setSize(300, 150);
     setVisible(true);
   }
   public static void main(String args[]){
         ExemploBotaoSair app = new ExemploBotaoSair();
         app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
  }
```

Como funciona o aplicativo sempre lembrando que temos que importar os pacotes:

import javax.swing.*; - pacote que exibe conteúdos na janela
import java.awt.*; - pacote que exibe conteúdos na janela
import java.awt.event.*; - pacote que trabalha com os eventos

Como já visto em outros aplicativos:

Declarar o componente como JButton: JButton botaosair;

Atribuir o texto ao botão: botaosair = new JButton ("Sair");

Posicionar o botão na janela: botaosair.setBounds(100,50,100,20);

E chama o objeto tela para exibir o componente: tela.add(botaosair);

Explicando o funcionamento do botão:

```
botaosair.addActionListener
(
      new ActionListener(){
            public void actionPerformed(ActionEvent e){
                  System.exit(0);
            }
      }
);
```

addActionListener – Adicionar ação a lista que estará dentro do botão.

new ActionListener – Chamando nova ação a ser listada.

public void actionPerformed(ActionEvent e) –
performance da ação que o evento público chamara que será o exit.

System.exit(0); - Finaliza o aplicativo

Use sempre essa lista para fazer qualquer botão funcionar ou chama algum evento:

```
nomebotao.addActionListener(
      new ActionListener(){
            public void actionPerformed(ActionEvent e){
          }
      }
);
```

8.2 – Clicando no mostrar para descobrir a senha digitada

Um aplicativo simples mais muito interessante de ser observado, o usuário digita a senha e clicando em mostrar será habilitado um JLabel mostrando qual a senha que foi digitada. Veja o exemplo:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class SenhaDigitada extends JFrame{
      JPasswordField caixa;
      JLabel rotulo, exibir;
     JButton mostrar;
   public SenhaDigitada(){
   super("Exemplo com JPasswordField");
   Container tela = getContentPane();
   setLayout(null);
   rotulo = new JLabel("Senha: ");
   caixa = new JPasswordField(10);
   exibir = new JLabel("A senha digitada é: ");
   mostrar = new JButton("Mostrar");
   exibir.setVisible(false);
   rotulo.setBounds(50,20,100,20);
   caixa.setBounds(50,60,200,20);
   exibir.setBounds(50,120,200,20);
   mostrar.setBounds(150,100,80,20);
   mostrar.addActionListener(
        new ActionListener(){
              public void actionPerformed(ActionEvent e){
               String senha = new String(caixa.getPassword());
                    exibir.setVisible(true);
                    exibir.setText(senha);
              }
        }
 );
  tela.add(rotulo);
  tela.add(caixa);
  tela.add(exibir);
  tela.add(mostrar);
  setSize(400, 250);
  setVisible(true);
 }
 public static void main(String args[]){
      SenhaDigitada app = new SenhaDigitada();
      app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }
}
```

Um aplicativo muito fácil de ser compreendido, de diferente ao anterior que já

foi visto na parte JPasswordField só acrescentamos um novo JLabel que quando iniciado vem oculto, e dentro do evento do botão criamos uma variável chamada senha do tipo String, onde ela recebe uma String que vem da caixa da senha e então é convertida em caracteres normal. E depois será habilitado novamente o JLabel.

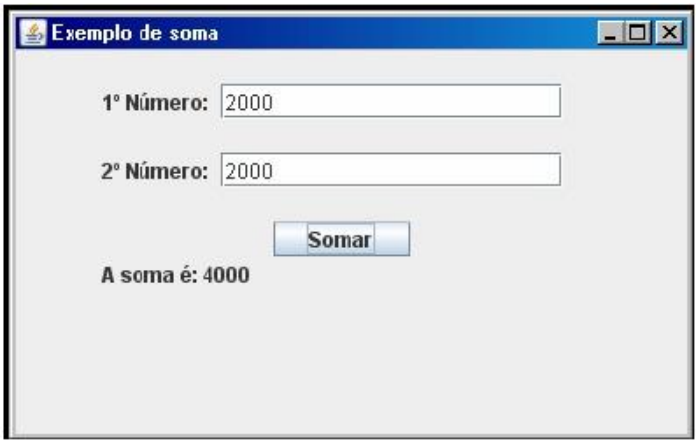
8.3 – Clicando no botão somar e será mostrada a soma dos números

Já nesse aplicativo alguns métodos novos foram aplicados, esse pede um pouquinho de nossa atenção, alguns componentes foram colocados ao lado para não ficar muito extenso e ser mais fácil de compreender.

Veja o exemplo:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Soma extends JFrame{
   JLabel rotulo1, rotulo2,exibir;
   JTextField texto1,texto2;
   JButton somar;
   public Soma(){
   super("Exemplo de soma");
   Container tela = getContentPane();
   setLayout(null);
   rotulo1 = new JLabel("1º Número: ");
   rotulo2 = new JLabel("2º Número: ");
   texto1 = new JTextField(5);
   texto2 = new JTextField(5);
   exibir = new JLabel("");
   somar = new JButton("Somar");

   rotulo1.setBounds(50,20,100,20); rotulo2.setBounds(50,60,100,20);
   texto1.setBounds(120,20,200,20); texto2.setBounds(120,60,200,20);
   exibir.setBounds(50,120,200,20); somar.setBounds(150,100,80,20);

 somar.addActionListener(
             new ActionListener(){
               public void actionPerformed(ActionEvent e){
               int numero1,numero2,soma;
               soma=0;
               numero1 = Integer.parseInt(texto1.getText());
               numero2 = Integer.parseInt(texto2.getText());
               soma = numero1 + numero2;
               exibir.setVisible(true);
               exibir.setText("A soma é: "+soma);
                                }
       }
     );
   exibir.setVisible(false);

   tela.add(rotulo1); tela.add(rotulo2);
   tela.add(texto1);  tela.add(texto2);
   tela.add(exibir);  tela.add(somar);
```

```
        setSize(400, 250);
        setVisible(true);
    }

    public static void main(String args[])
    {
        Soma app = new Soma();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



A habilitação do botão já foi explicado no aplicativo anterior, o que foi acrescentado nesse aplicativo, foram os comandos que trabalham com conversão de números.

Foram declaradas três variáveis do tipo inteira: int numero1, numero2, soma;
Foi atribuído zero a variável soma, para a variável ser iniciada vazia: soma=0;
Como texto1, texto2 são String, melhor dizer texto para armazená-las nas variáveis numero1 e numero2, será necessário converte a String em números.

Nesse caso estamos convertendo o texto1 em inteiro para isso usamos o Integer.parseInt(), e armazenamos na variável numero1 que é inteira.
Isso ocorrerá para a texto2 e a variável numero2.

```
numero1 = Integer.parseInt(texto1.getText());
numero2 = Integer.parseInt(texto2.getText());
```

Depois de convertido os valores agora é só fazer a soma e armazenar na variável correspondente a soma, veja: soma = numero1 + numero2;

Chamamos o comando para habilitar o JLabel exibir e jogamos o valor de soma dentro do texto do Jlabel exibir para se apresentado no aplicativo.

```
exibir.setVisible(true);
exibir.setText("A soma é: "+soma);
```

Para variáveis do tipo Float, usaremos: Float.parseFloat();

Para variáveis do Double, usaremos Doublé.parseDouble();

8.4 – Clicando no botão limpar e as caixas de texto serão limpas

Depois de usadas nada como limpar suas caixas de texto, esta ai o exemplo que irá ajudá-lo(a).

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

public class Limpar extends JFrame{
   JLabel rotulo1, rotulo2;
   JTextField texto1,texto2;
   JButton limpar;
   public Limpar(){
   super("Exemplo Limpar");
   Container tela = getContentPane();
   setLayout(null);
   rotulo1 = new JLabel("1º Número: ");
   rotulo2 = new JLabel("2º Número: ");
   texto1 = new JTextField(5);
   texto2 = new JTextField(5);
   limpar = new JButton("Limpar");
   rotulo1.setBounds(50,20,100,20);
   rotulo2.setBounds(50,60,100,20);
   texto1.setBounds(120,20,200,20);
   texto2.setBounds(120,60,200,20);
   limpar.setBounds(150,100,80,20);
   limpar.addActionListener(
              new ActionListener(){
                 public void actionPerformed(ActionEvent e){
              texto1.setText(null);
              texto2.setText(null);
              texto1.requestFocus();
            }
         }
      );
   tela.add(rotulo1);
   tela.add(rotulo2);
   tela.add(texto1);
   tela.add(texto2);
   tela.add(limpar);
   setSize(400, 250);
   setVisible(true);
 }

 public static void main(String args[]){
         Limpar app = new Limpar();
         app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }
}
```

Nada de diferente no aplicativo somente as linhas que vão limpar as caixas de texto e passar o foco para a primeira caixa vejam:

```
texto1.setText(null);  ou texto1.setText("");
texto2.setText(null);  ou texto2.setText("");
texto1.requestFocus();
```

É atribuído null dentro das caixas, pois nulo significa vazio e requestFocus() passa o foco para a caixa selecionada pelo usuário.

8.5 – Ocultar e Exibir componentes

Esse aplicativo é só para teste não tem muito que ser explicado, pois a única coisa diferente é que quando você clica em ocultar os JLabels some e você clica em exibir os JLabels aparecem novamente na janela.
Isso ocorre por que o comando setVisible(false), oculta e o comando setVisible(true), exibe mostra na janela. Veja o exemplo:

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

public class OcultarExibir extends JFrame{
   JLabel rotulo1, rotulo2;
   JButton ocultar,exbir;
   public OcultarExibir(){
   super("Exemplo de ocultar e exibir componente");
   Container tela = getContentPane();
   setLayout(null);
   rotulo1 = new JLabel("Rótulo 1");  rotulo2 = new JLabel("Rótulo 2");
   ocultar = new JButton("Ocultar ");  exbir = new JButton("Exibir ");
   rotulo1.setBounds(50,20,100,20);  rotulo2.setBounds(50,60,100,20);
   ocultar.setBounds(100,100,80,20);  exbir.setBounds(250,100,80,20);
   ocultar.addActionListener(
            new ActionListener(){
               public void actionPerformed(ActionEvent e){
                        rotulo1.setVisible(false);
                        rotulo2.setVisible(false);
      }
      }
     );

 exbir.addActionListener(
            new ActionListener(){
               public void actionPerformed(ActionEvent e){
                        rotulo1.setVisible(true);
                        rotulo2.setVisible(true);
      }
      }
     );

   tela.add(rotulo1);  tela.add(rotulo2);
   tela.add(ocultar);   tela.add(exbir);
```

```
    setSize(400, 250);
    setVisible(true);
  }

  public static void main(String args[]){
        OcultarExibir app = new OcultarExibir();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```

8.6 – Desabilitar e Habilitar Exibir componentes

Nada de diferente também somente o comando setEnabled(true) que habilita e o setEnabled(false) eu desabilita. Veja o exemplo:

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

public class DesabilitarHabilitar extends JFrame{
    JLabel rotulo1, rotulo2;
    JButton desabilitar,habilitar;
    public DesabilitarHabilitar(){
    super("Exemplo de Desabilitar e Habilitar componentes");
    Container tela = getContentPane();
    setLayout(null);
    rotulo1 = new JLabel("Rótulo 1"); rotulo2 = new JLabel("Rótulo 2");
    desabilitar = new JButton("Desabilitar"); habilitar = new JButton("Habilitar");
    rotulo1.setBounds(50,20,100,20); rotulo2.setBounds(50,60,100,20);
    desabilitar.setBounds(80,100,100,20); habilitar.setBounds(250,100,100,20);

   desabilitar.addActionListener(
              new ActionListener(){
                 public void actionPerformed(ActionEvent e){
                                rotulo1.setEnabled(false);
                                rotulo2.setEnabled(false);
          }
         }
       );

    habilitar.addActionListener(
              new ActionListener(){
                 public void actionPerformed(ActionEvent e){
                        rotulo1.setEnabled(true);
                        rotulo2.setEnabled(true);
          }
         }
       );
    tela.add(rotulo1); tela.add(rotulo2);
    tela.add(desabilitar); tela.add(habilitar);
```

```
    setSize(400, 250);
    setVisible(true);
  }

  public static void main(String args[]){
        DesabilitarHabilitar app = new DesabilitarHabilitar();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```