



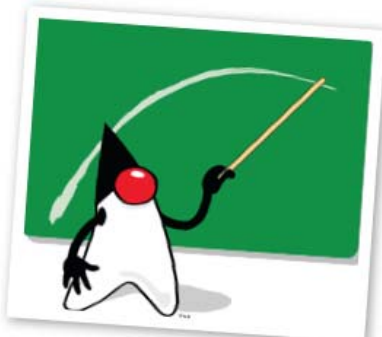


Componentes

Projeto *GUI04_Componentes*

Vamos exemplificar o uso dos componentes em diversas Classes neste projeto.

Crie o pacote controller e as referidas classes dentro dele.



A cartoon character with a red nose and black body is pointing a stick at a green board. The board has a white curved line on it. The character is standing on a white surface. In the bottom left corner of the slide, there is a small Java logo.

4

Classe *Botao*

Fazer a classe receber herança de JFrame

Dentro do método main

```
new Botao();
```

super("Botão");

Cria um objeto JButton

Msg ao passar o mouse sobre o botão

Listener vai ficar aguardando alguma ação no botão pelo usuário

Dentro deste método define-se o que será feito quando o botão for pressionado

```
JButton botao = new JButton("Botão 1");
botao.setToolTipText("Clica em mim !!!");
botao.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {

        JOptionPane.showMessageDialog(null, "Você clicou no botão 1.");

    }

});
```

Criando um painel e adicionando o botão ao painel

Adicionando o painel à janela e definindo propriedades da janela

```
JPanel painel = new JPanel();
painel.add(botao);

add(painel);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(300, 100);
setVisible(true);
```

Fazer todo este código dentro do método construtor

5

Classe *CaixaDeTexto*

Fazer a classe receber herança de JFrame

Dentro do método main

```
new CaixaDeTexto();
```

super("Caixa de Texto");

Adicionando um Listener à Caixa de Texto

Fazer todo este código dentro do método construtor

```
JTextField caixaDeTexto = new JTextField(20);
caixaDeTexto.setText("Digite algo aqui");
caixaDeTexto.setToolTipText("Digita logo !!!");

caixaDeTexto.addKeyListener(new KeyListener() {

    @Override
    public void keyTyped(KeyEvent e) {

    }

    @Override
    public void keyReleased(KeyEvent e) {
        caixaDeTexto.setText(caixaDeTexto.getText().toUpperCase());
    }

    @Override
    public void keyPressed(KeyEvent e) {



    }

});
```

Transformando tudo o que for digitado em maiúsculo automaticamente

Continua no próximo slide

6

INFORMA. FORMA. TRANSFORMA.

Classe *CaixaDeTexto*

Continuando dentro do método construtor

```

    JButton botao1 = new JButton("Botão 1");
    botao1.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null, caixaDeTexto.getText());
        }
    });


    JPanel painel = new JPanel();
    painel.add(caixaDeTexto);
    painel.add(botao1);

    add(painel);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(300, 100);
    setVisible(true);
    
```

Criando um botão e atribuindo suas funcionalidades

Criando um painel e adicionado a caixa de texto e o botão

Adicionando o painel à janela e definindo propriedades da janela


7




INFORMA. FORMA. TRANSFORMA.

Classe *CaixaDeSenha*

Fazer a classe receber herança de JFrame

```

    new CaixaDeSenha();

    super("Caixa de Senha");

    final JPasswordField caixaDeSenha = new JPasswordField(20);
    JLabel label = new JLabel("Digite sua senha aqui:");
    caixaDeSenha.setToolTipsText("Digita nimim");

    JButton botao1 = new JButton("Botão 1");
    botao1.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null,
                new String(caixaDeSenha.getPassword()));
        }
    });


    JPanel painel = new JPanel();
    painel.add(label);
    painel.add(caixaDeSenha);
    painel.add(botao1);

    add(painel);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(300, 200);
    setVisible(true);
    
```

Dentro do método main

Fazer todo este código dentro do método construtor


8



Classe *AreaDeTexto*

`new AreaDeTexto();`

Dentro do método main

Fazer a classe receber herança de JFrame

```

super("Área de Texto");


JLabel label = new JLabel("Digite uma mensagem:");
JTextArea areaDeTexto = new JTextArea(10,30);

JPanel painel = new JPanel();
painel.add(label);
painel.add(areaDeTexto);


add(painel);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
pack();
setVisible(true);
    
```

Criando objeto de área de texto. Os números indicam altura e largura.

Fazer todo este código dentro do método construtor



9



Classe *AreaDeTextoComScroll*

`new AreaDeTextoComScroll();`

Dentro do método main

Fazer a classe receber herança de JFrame

```

super("Área de Texto com Scroll");

JLabel label = new JLabel("Digite uma mensagem:");
JTextArea areaDeTexto = new JTextArea(10, 30);


JScrollPane sp = new JScrollPane(areaDeTexto,
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
    JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);

JPanel painel = new JPanel();
painel.add(label);
painel.add(sp);



add(painel);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
pack();
setVisible(true);
    
```

Definindo barra de rolagem obrigatória na vertical e somente se for necessária na horizontal.

Fazer todo este código dentro do método construtor



10

INFORMA. FORMA. TRANSFORMA.

Classe *AreaDeTextoJanelaComScroll*

```

new AreaDeTextoJanelaComScroll();
super("Área de Texto Janela com Scroll");

JLabel label = new JLabel("Digite uma mensagem:");
JTextArea areaDeTexto = new JTextArea(10, 30);

JPanel painel = new JPanel();
painel.add(label);
painel.add(areaDeTexto);

JScrollPane sp = new JScrollPane(painel,
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);


add(sp);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(300, 200);
setVisible(true);
    
```

Dentro do método main



Fazer a classe receber herança de JFrame

Fazer todo este código dentro do método construtor

Definindo barra de rolagem desta vez ao painel, e não somente à área de texto.



11


INFORMA. FORMA. TRANSFORMA.


Classe *AreaDeTextoJanelaComScrollExercicio*

Exercício:

Copiar a classe *AreaDeTextoJanelaComScroll* e colar renomeando-a para *AreaDeTextoJanelaComScrollExercicio*

Adapte o código para que o texto digitado pelo usuário na área de texto seja exibido em uma caixa de mensagem quando o usuário clicar no botão.





12

Classe *ComboBox*

```
new ComboBox();
```

Dentro do método main

Fazer a classe receber herança de JFrame

```
super ("ComboBox");
```

```
final JLabel label = new JLabel();
```

Cria um vetor e o adiciona ao obj comboBox

```
String[] vetor = {"um", "dois", "três"};
```

```
JComboBox comboBox = new JComboBox(vetor);
```

Listener para comboBox

```
comboBox.addActionListener(new ActionListener() {
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    JComboBox cb = (JComboBox)e.getSource();
```

```
    String conteudo = (String)cb.getSelectedItem();
```

```
    label.setText(conteudo);
```

Variável **conteudo** recebe o valor selecionado pelo usuário no comboBox

```
}
```

```
});
```

conteudo é então exibido no label

```
JPanel painel = new JPanel();
```

```
painel.add(comboBox);
```

```
painel.add(label);
```

```
add(painel);
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
setSize(300, 200);
```

```
setVisible(true);
```



Fazer todo este código dentro do método construtor

13

Sistema
FIRJAN



INFORMA, FORMA, TRANSFORMA.

Parte
1 de 4

Classe *RadioButton*

Vamos utilizar as imagens homem.jpg e mulher.jpg, que devem estar na pasta img deste projeto.

Faça a classe ser herança de JFrame.

Declare os seguintes objetos globais:

```
ImageIcon iconHomem = new ImageIcon("img/homem.jpg");
```

```
ImageIcon iconMulher = new ImageIcon("img/mulher.jpg");
```

```
JLabel label = new JLabel(iconHomem);
```



14

Classe *RadioButton*

Dentro do método construtor:

Parte
2 de 4

```
super("Exemplo RadioButton");

label.setPreferredSize(new Dimension(150, 300));

JRadioButton radioHomem = new JRadioButton("Homem");
radioHomem.setMnemonic(KeyEvent.VK_H);
radioHomem.setSelected(true);

JRadioButton radioMulher = new JRadioButton("Mulher");
radioMulher.setMnemonic(KeyEvent.VK_M);

ButtonGroup group = new ButtonGroup();
group.add(radioHomem);
group.add(radioMulher);
```



15

Classe *RadioButton*

Ainda no método construtor, associe os Listeners aos componentes:

```
radioHomem.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        label.setIcon(iconHomem);
    }
});

radioMulher.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        label.setIcon(iconMulher);
    }
});
```

Parte
3 de 4



16

Classe *RadioButton*

Parte
4 de 4

Finalizando o conteúdo do método construtor:

```
JPanel radioPainel = new JPanel(new GridLayout(2, 1));
radioPainel.add(radioHomem);
radioPainel.add(radioMulher);
```

```
Container container = getContentPane();
```

```
container.add(radioPainel, BorderLayout.WEST);
container.add(label, BorderLayout.CENTER);
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(300, 400);
setVisible(true);
```



No método main, instancie o objeto utilizando o construtor da Classe *RadioButton*.



17

Sistema
FIRJAN



INFORMA, FORMA, TRANSFORMA.

Classe *RadioButton2*



Vamos exemplificar o uso de *RadioButton* associado a um botão, de forma que ao clicar no botão seja verificado qual item de *RadioButton* está selecionado.

Para isso copie a classe *RadioButton* e cole renomeando-a para *RadioButton2*.

Vamos realizar algumas adaptações conforme descrito nos próximos slides.



18

  INFORMA. FORMA. TRANSFORMA.

Parte 1 de 4

Classe *RadioButton2*


Será preciso colocar como global os objetos `radioHomem`, `radioMulher` e `group`.

Localize as seguintes linhas de código (**Não estão em sequência**):



```
JRadioButton radioHomem = new JRadioButton("Homem");
JRadioButton radioMulher = new JRadioButton("Mulher");
ButtonGroup group = new ButtonGroup();
```

Recorte-as e cole-as abaixo de

```
JLabel label = new JLabel(iconHomem);
```



19

  INFORMA. FORMA. TRANSFORMA.

Parte 2 de 4

Classe *RadioButton2*

Agora localize a seguinte linha de código


```
radioHomem.setMnemonic(KeyEvent.VK_H);
```

Digite abaixo dela:



```
radioHomem.setActionCommand("Homem");
```

Isto vai fazer com que ao clicar no botão, o *Listener* receba a palavra "Homem"

FAÇA O MESMO PARA O OBJETO `radioMulher`



20



INFORMA. FORMA. TRANSFORMA.

**Parte
3 de 4**

Classe *RadioButton2*

Vamos agora inserir os códigos referentes ao Botão.

Localize a linha

```
JPanel radioPainel = new JPanel(new GridLayout(2, 1));
```


Digite **ACIMA** dela:



```

JButton botao = new JButton("Ok");
botao.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null,
            group.getSelection().getActionCommand(),
            "Exemplo RadioButton2", JOptionPane.INFORMATION_MESSAGE);
    }
});

```


21



INFORMA. FORMA. TRANSFORMA.

**Parte
4 de 4**


Classe *RadioButton2*

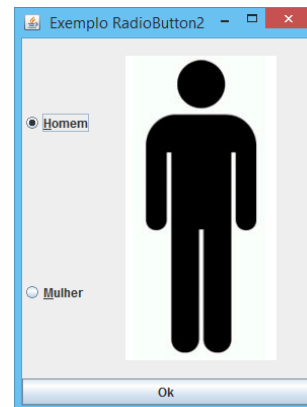
Localize a linha



```
container.add(label, BorderLayout.CENTER);
```

Digite abaixo dela:

```
container.add(botao, BorderLayout.SOUTH);
```


22





INFORMA. FORMA. TRANSFORMA.

**Parte
1 de 5**

Classe *CheckBox*

Vamos utilizar as imagens chocolate.jpg, coco.jpg e morango.jpg, que devem estar na pasta img deste projeto.


Faça a classe ser herança de JFrame.



Declare os seguintes objetos globais:

```

ImageIcon iconChocolate = new ImageIcon("img/chocolate.jpg");
ImageIcon iconCoco = new ImageIcon("img/coco.jpg");
ImageIcon iconMorango = new ImageIcon("img/morango.jpg");

JLabel labelChocolate = new JLabel(iconChocolate);
JLabel labelCoco = new JLabel(iconCoco);
JLabel labelMorango = new JLabel(iconMorango);
  
```


23



INFORMA. FORMA. TRANSFORMA.

**Parte
2 de 5**

Classe *CheckBox*

Dentro do método construtor:

```

super("Exemplo CheckBox");

labelChocolate.setPreferredSize(new Dimension(250, 250));
labelChocolate.setVisible(false);


labelCoco.setPreferredSize(new Dimension(250, 250));
labelCoco.setVisible(false);

labelMorango.setPreferredSize(new Dimension(250, 250));
labelMorango.setVisible(false);

JCheckBox checkChocolate = new JCheckBox("Chocolate");
checkChocolate.setMnemonic(KeyEvent.VK_H);

JCheckBox checkCoco = new JCheckBox("Coco");
checkCoco.setMnemonic(KeyEvent.VK_O);

JCheckBox checkMorango = new JCheckBox("Morango");
checkMorango.setMnemonic(KeyEvent.VK_M);
  
```


24

Classe *CheckBox*

Ainda no método construtor, associe os Listeners aos componentes:

```
checkChocolate.addItemListener(new ItemListener() {

    @Override
    public void itemStateChanged(ItemEvent e) {
        // TODO Auto-generated method stub
        if(e.getStateChange()==ItemEvent.DESELECTED){
            labelChocolate.setVisible(false);
        }else{
            labelChocolate.setVisible(true);
        }
    }
});
```

Parte
3 de 5

Este é o Listener para o Chocolate.
Faça abaixo deste os outros 2 Listeners
referentes ao Côco e ao Morango.



25

Classe *CheckBox*

Finalizando o conteúdo do método construtor:

```
JPanel checkPainel = new JPanel(new GridLayout(1, 3));
checkPainel.add(checkChocolate);
checkPainel.add(checkCoco);
checkPainel.add(checkMorango);

JPanel labelPainel = new JPanel(new GridLayout(1, 3));
labelPainel.add(labelChocolate);
labelPainel.add(labelCoco);
labelPainel.add(labelMorango);

Container container = getContentPane();

container.add(checkPainel, BorderLayout.NORTH);
container.add(labelPainel, BorderLayout.CENTER);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(800, 300);
setVisible(true);
```

Parte
4 de 5



26



INFORMA, FORMA, TRANSFORMA.



**Parte
5 de 5**

Classe *CheckBox*

Instancie o objeto utilizando o construtor da Classe *CheckBox*.




27




INFORMA, FORMA, TRANSFORMA.



Classe *CheckBox2*

Vamos exemplificar o uso de *CheckBox* associado a um botão, de forma que ao clicar no botão seja verificado quais itens de *CheckBox* estão selecionados.

Para isso copie a classe *CheckBox* e cole renomeando-a para *CheckBox2*.

Vamos realizar algumas adaptações conforme descrito nos próximos slides.


28




INFORMA. FORMA. TRANSFORMA.

Classe *CheckBox2*

1 – Localize estas 3 linhas abaixo (elas não estão em sequência):

```
JCheckBox checkChocolate = new JCheckBox("Chocolate");
JCheckBox checkCoco = new JCheckBox("Coco");
JCheckBox checkMorango = new JCheckBox("Morango");
```

Transforme estes 3 objetos em objetos globais, retirando-os de dentro do método construtor e colocando-os no início da Classe abaixo dos outros objetos globais que já existiam.


29

2 – Localize esta linha: `checkMorango.setMnemonic(KeyEvent.VK_M);`
 Insira abaixo dela o seguinte código:

```
JButton botao = new JButton("Ok");
botao.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        StringBuffer stringBuffer = new StringBuffer();

        if(checkChocolate.isSelected()){
            stringBuffer.append("-Chocolate\n");
        }
        if(checkCoco.isSelected()){
            stringBuffer.append("-Coco\n");
        }
        if(checkMorango.isSelected()){
            stringBuffer.append("-Morango\n");
        }
        String retorno = stringBuffer.toString();
        if(retorno.equals("")){
            retorno = "Nenhuma opção selecionada";
        }
        JOptionPane.showMessageDialog(null, retorno,
            "Exemplo CheckBox2", JOptionPane.INFORMATION_MESSAGE);
    }
});
```


30

INFORMA. FORMA. TRANSFORMA.

Classe *CheckBox2*

3 – Localize esta linha:

```
container.add(labelPainel, BorderLayout.CENTER);
```

Insira abaixo dela:

```
container.add(botao, BorderLayout.SOUTH);
```

31

INFORMA. FORMA. TRANSFORMA.

Classe *Menu*

Faça a classe ser herança de JFrame.

Declare o seguinte objeto global:

```
JLabel output = new JLabel();
```

Dentro do método construtor:

```
super("Exemplo Menu");
```

Atenção: Os próximos slides ainda são códigos que devem ficar dentro do construtor.

32

Classe *Menu*

1 - Criando a barra de menu e inserindo um menu:

```
JMenuBar menuBar = new JMenuBar();

JMenu menu = new JMenu("Um Menu");
menu.setMnemonic(KeyEvent.VK_A);
menuBar.add(menu);

class Acao implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        output.setText(e.getActionCommand());
    }
}

Acao acao = new Acao();
```

Fazer todo este
código dentro
do método
construtor



33

Classe *Menu*



2 - Inserindo um item de Menu somente com texto

```
JMenuItem menuItem1 = new JMenuItem("Item de menu somente com  
texto");
menuItem1.setMnemonic(KeyEvent.VK_T);
menuItem1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_1,  
ActionEvent.ALT_MASK));
menuItem1.setActionCommand  
("Clicou em item de menu somente com texto");
menuItem1.addActionListener(acao);
menu.add(menuItem1);
```

Fazer todo este
código dentro
do método
construtor



34



INFORMA. FORMA. TRANSFORMA.


Classe *Menu*

3 - Inserindo um item de Menu com texto e ícone

```

ImageIcon icon = new ImageIcon("img/abrir.png");
JMenuItem menuItem2 = new JMenuItem("Texto e ícone juntos",
    icon);
menuItem2.setMnemonic(KeyEvent.VK_J);
menuItem2.setActionCommand
    ("Clicou em item de menu com texto e ícone");
menuItem2.addActionListener(acao);
menu.add(menuItem2);
    
```

Fazer todo este código dentro do método construtor


35



INFORMA. FORMA. TRANSFORMA.

Classe *Menu*

4 - Inserindo um item de Menu somente com ícone

```

icon = new ImageIcon("img/salvar.png");
JMenuItem menuItem3 = new JMenuItem(icon); //Somente com ícone
menuItem3.setMnemonic(KeyEvent.VK_I);
menuItem3.setActionCommand
    ("Clicou em item de menu somente com ícone");
menuItem3.addActionListener(acao);
menu.add(menuItem3);

menu.addSeparator();
    
```

Fazer todo este código dentro do método construtor


36

5 - Inserindo RadioButton no menu

```

JRadioButtonMenuItem rbMenuItem1 = new JRadioButtonMenuItem(
    "Um item de menu radio button");
rbMenuItem1.setSelected(true);
rbMenuItem1.setMnemonic(KeyEvent.VK_R);
rbMenuItem1.setActionCommand(
    "Clicou em um item de menu radio button");
rbMenuItem1.addActionListener(acao);
menu.add(rbMenuItem1);

JRadioButtonMenuItem rbMenuItem2 = new JRadioButtonMenuItem(
    "Outro item de menu radio button");
rbMenuItem2.setMnemonic(KeyEvent.VK_O);
rbMenuItem2.setActionCommand(
    "Clicou em outro item de menu radio button");
rbMenuItem2.addActionListener(acao);
menu.add(rbMenuItem2);

ButtonGroup group = new ButtonGroup();
group.add(rbMenuItem1);
group.add(rbMenuItem2);

menu.addSeparator();

```



37

Classe *Menu*

6 - Inserindo um Sub-menu

```

JMenu submenu = new JMenu("Submenu");
submenu.setMnemonic(KeyEvent.VK_S);



JMenuItem subMenuItem1 = new JMenuItem("Um item no submenu");
subMenuItem1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_2,
    ActionEvent.ALT_MASK));
subMenuItem1.setActionCommand("Clicou em um item no submenu");
subMenuItem1.addActionListener(acao);
submenu.add(subMenuItem1);

JMenuItem subMenuItem2 = new JMenuItem("Outro item no submenu");
subMenuItem2.setActionCommand("Clicou em outro item no submenu");
subMenuItem2.addActionListener(acao);
submenu.add(subMenuItem2);
menu.add(submenu);

```



38



INFORMA. FORMA. TRANSFORMA.

Classe *Menu*

7 - Inserindo um Menu sem itens de menu

```


JMenu menuVazio = new JMenu("Outro Menu");
menuVazio.setMnemonic(KeyEvent.VK_N);
menuBar.add(menuVazio);



setJMenuBar(menuBar);

Container container = getContentPane();
container.add(output, BorderLayout.CENTER);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(300, 200);
setVisible(true);

```

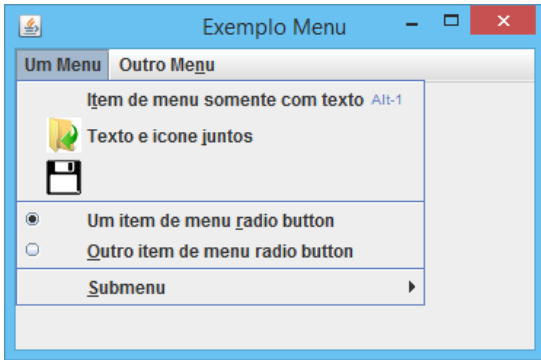
Fazer todo este código dentro do método construtor



39





INFORMA. FORMA. TRANSFORMA.

Classe *Menu*

Agora finalizando a Classe, no método main, instancie o objeto utilizando o construtor da Classe Menu.





40





INFORMA. FORMA. TRANSFORMA.

Dúvidas?




41




INFORMA. FORMA. TRANSFORMA.


Bibliografia



Java Como Programar 8ª Edição
Paul Deitel e Harvey Deitel
Ed. Pearson



Java 8 Programação de Computadores
José Augusto N. G. Manzano & Roberto Affonso da Costa Junior
Ed. Érica | Saraiva



Fundamentos de Computação e Orientação a Objetos Usando Java
Francisco A. C. Pinheiro
Ed. LTC

42