

Janelas

- Originada da Classe *JFrame*.
- Nível hierárquico mais alto.
- Os demais componentes serão colocados dentro dela.
- Tudo que se deseja exibir deve ser colocado em uma janela. Sem janela nada pode ser exibido.

Painel



Painel na Janela



Resultado de exibição



Projeto *GUI03_JanelasEPaineis*

Crie a Classe **Janela** dentro do pacote controller. Digite isto dentro do método main e interprete o resultado ao executar:

```
JFrame janela = new JFrame();

janela.setTitle("Título da Janela");
janela.setSize(500, 300);
janela.setResizable(false);
janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

ImageIcon icone = new ImageIcon("img/ic_curtir.gif");
janela.setIconImage(icone.getImage());

janela.setLocationRelativeTo(null);
janela.setVisible(true);
```

Criando objeto da Classe JFrame

500 px horizontal x 300 px vertical

Copiar a pasta **img** do projeto anterior e colar neste projeto

False bloqueia o redimensionamento da janela por mouse

Encerrar o programa ao fechar a janela

Centraliza a janela na tela

Exibe a janela na tela

Substitua a linha `janela.setLocationRelativeTo(null);` pela linha `janela.setLocation(100, 200);` e interprete o resultado



5

Projeto *GUI03_JanelasEPaineis*


- Copie a Classe *Janela* e cole renomeando-a para **JanelaHeranca**.
- Faça a Classe receber herança de JFrame.
- Crie um método construtor público onde o seu conteúdo seja tudo que estava originalmente no método main.
- Insira como 1ª linha do método construtor a chamada ao método construtor da Super Classe:
`super("Título: Janela com Herança");`
- Apague o termo `janeLa`. que inicia diversas linhas contidas neste método construtor.
- Dentro do método main() deverá existir apenas a chamada ao método construtor:



```
new JanelaHeranca();
```

6

Sistema
FIRJAN




INFORMA. FORMA. TRANSFORMA.

Sobre o uso de JFrame

Vimos que é possível fazer uso da Classe JFrame através de import (exemplo da classe Janela) e também através de herança (exemplo da classe JanelaHeranca) .


É interessante ressaltar que o uso através de herança ativa as funcionalidades práticas da orientação a objetos e facilita a padronização MVC.

Por estes motivos adotaremos como padrão a utilização de JFrame através de herança.



7

Sistema
FIRJAN





INFORMA. FORMA. TRANSFORMA.

Painéis

1. Os itens que vão aparecer na tela para o usuário serão inseridos em painéis;
2. Estes painéis, por sua vez, serão inseridos na Janela;
3. E esta janela se tornará visível para o usuário;

Tudo funciona numa hierarquia de camadas.





8

Classe *Painel*

Digitar o código abaixo dentro do método construtor

```

super("Título");

ImageIcon icone1 = new ImageIcon("img/ic_youtube.png");
ImageIcon icone2 = new ImageIcon("img/ic_onoff.jpg");
ImageIcon icone3 = new ImageIcon("img/ic_download.gif");

JLabel labelTexto = new JLabel("Texto e imagem no painel");
JLabel label1 = new JLabel(icone1);
JLabel label2 = new JLabel(icone2);
JLabel label3 = new JLabel(icone3);


JPanel painel = new JPanel();
painel.add(labelTexto);
painel.add(label1);
painel.add(label2);
painel.add(label3);

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

Annotations:

- Criando objeto de ImageIcon
- Etiqueta que recebe texto
- Etiqueta que recebe imagem
- Criando um painel
- Adicionando as etiquetas ao painel
- Adicionando o painel à janela
- A janela terá o menor tamanho possível
- Encerrar o programa ao fechar a janela
- Torna a janela visível

Digitar dentro do método main():
 new Painel();



9

FIRJAN | **INFORMA, FORMA, TRANSFORMA.**

Classe *PainelGridLayout1*

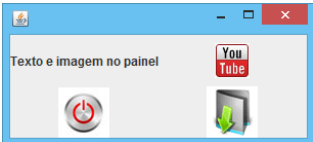
Copie a Classe Painel e cole renomeando-a para PainelGridLayout1.

Localize a seguinte linha de código:
 JPanel painel = new JPanel();

Abaixo desta linha, insira esta:
 painel.setLayout(new GridLayout(2,2));

Fazer com que o método main acione o construtor desta classe.

Annotation:
 O painel foi dividido em 2 linhas e 2 colunas. Cada item que for adicionado ao painel será posicionado em uma célula desta tabela.



10

Classe *PainelGridLayout2*

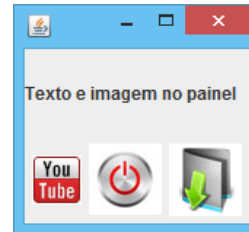
Copie a Classe PainelGridLayout1 e cole renomeando-a para PainelGridLayout2.

Localize e apague as linhas referentes ao JPanel (total de 6 linhas)

Digite estas no lugar das que foram apagadas:

```
JPanel painelIcones = new JPanel();
painelIcones.add(label1);
painelIcones.add(label2);
painelIcones.add(label3);

JPanel painelPrincipal = new JPanel();
painelPrincipal.setLayout(new GridLayout(2,1));
painelPrincipal.add(labelTexto);
painelPrincipal.add(painelIcones);
```



Na linha add, mude o conteúdo entre parênteses para `painelPrincipal`

Fazer com que o método main acione o construtor desta classe.

Execute e interprete como obtivemos este resultado.



11

Classe *PainelBorderLayout*

Copie a Classe PainelGridLayout2 e cole renomeando-a para PainelBorderLayout.

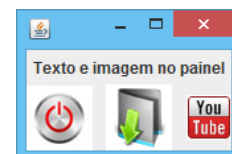
Localize e apague as linhas referentes ao JPanel (total de 8 linhas)

Digite estas no lugar das que foram apagadas:

```
JPanel painelTexto = new JPanel();
painelTexto.add(labelTexto);

JPanel painelPrincipal = new JPanel();
painelPrincipal.setLayout(new BorderLayout());

painelPrincipal.add(painelTexto, BorderLayout.NORTH);
painelPrincipal.add(label1, BorderLayout.EAST);
painelPrincipal.add(label2, BorderLayout.WEST);
painelPrincipal.add(label3, BorderLayout.CENTER);
```



Apague a linha `add(painelPrincipal);`



Digite no lugar desta que foi apagada: `setContentPane(painelPrincipal);`

Fazer com que o método main acione o construtor desta classe.

Execute e interprete como obtivemos este resultado.






12

  INFORMA. FORMA. TRANSFORMA.

Layouts para painéis:

- **Border Layout** – Define a organização dos itens em 5 regiões (norte, sul, leste, oeste e centro).
- **Flow Layout** – Define a organização dos itens da esquerda para a direita. Não cabendo mais à direita então passa para a próxima linha e continua novamente da esquerda para a direita. (Na ordem de inserção dos componentes) – Default do JPanel.
- **Grid Layout** – Define a organização por linhas e colunas. É indicado a quantidade de linhas e colunas, e os itens são inseridos nesta estrutura de grade.

 13


  INFORMA. FORMA. TRANSFORMA.

Abas

Para trabalhar com abas entenda que utilizaremos vários painéis, e cada painel será visualizado através de uma aba.

Crie a classe Abas e coloque-a como herança de JFrame.

Digite o código presente nos próximos slides.

 14

Classe *Abas*

Instancie os seguintes objetos globais:

```
JTabbedPane abas = new JTabbedPane();

JLabel label1 = new JLabel("Painel 1");
JPanel painel1 = new JPanel();

JLabel label2 = new JLabel("Painel 2");
JPanel painel2 = new JPanel();

JLabel label3 = new JLabel("Painel 3");
JPanel painel3 = new JPanel();

ImageIcon iconAjuda = new ImageIcon("img/ajuda.png");
ImageIcon iconImprimir = new ImageIcon("img/imprimir.png");
ImageIcon iconSair = new ImageIcon("img/sair.png");
```



15



Classe *Abas*

Agora digite isso dentro do método construtor:

```
super("Exemplo Abas");

painel1.add(label1);
painel2.add(label2);
painel3.add(label3);

abas.addTab("Ajuda", iconAjuda, painel1, "Primeiro painel");
abas.addTab("Imprimir", iconImprimir, painel2, "Segundo painel");
abas.addTab("Sair", iconSair, painel3, "Terceiro painel");



Container container = getContentPane();
container.add(abas);


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



16


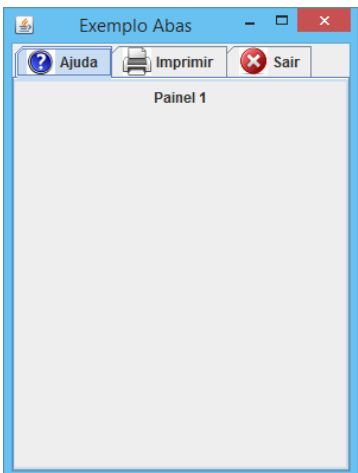




  **FIRJAN** | **INFORMA, FORMA, TRANSFORMA.**




Classe *Abas*



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

17

  **FIRJAN** | **INFORMA, FORMA, TRANSFORMA.**


Dúvidas?

18




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

19