

## Programação Java – II

### Jbuilder

URI – Campus de Erechim  
Curso de Ciência da Computação  
Prof. Neilor

### O que é o JBuilder

- O Jbuilder é uma IDE gráfica, desenvolvida na linguagem Java, que ajuda o desenvolvedor a criar aplicativos com a linguagem Java.
- Trabalhar com o Jbuilder é relativamente simples quando se trata de um desenvolvedor que conhece outra ferramenta de desenvolvimento como por exemplo o Borland Delphi.
- O Jbuilder é um software proprietário, mas é muito utilizado por possibilitar o desenvolvimento rápido de um sistema portátil, em Java, sem complicações.
- O que mais pesa na hora de desenvolver em Java é o tempo de desenvolvimento, que sem uma ferramenta visual, fica pouco provável que se consiga um sistema pronto em um curto espaço de tempo.

### Iniciando no JBuilder

- Para criar um projeto chama do de MinhaPrimeiraTela, você iniciará um assistente que o ajudará a configurar seu projeto.
  - Vá ao menu **File** e clique em **New Project**
  - Digite o nome de seu projeto "MinhaPrimeiraTela"
  - Na segunda etapa serão especificados os caminhos do projeto. Note que existem várias subpastas, que guardarão cada uma um tipo de arquivo (Output path (saída dos arquivos compilados), backup path (backup dos arquivos), working directory (diretório dos arquivos)).
  - Clicando em **Next**, você terá a tela para configurar as informações adicionais para arquivos de documentação do Java.
  - Quando terminado o assistente clique no botão **Finish**.

### Iniciando no JBuilder

- Para gerar os arquivos de código .java, clique no menu **File, New** ou clique no botão **New** na barra de ferramentas principal, onde você terá acesso ao diálogo **Object Gallery**, no qual você pode escolher que tipo de projeto poderá ser iniciado.
- Clique em **Application** e depois em **OK**
- Novamente será aberto um assistente, no qual o auxiliará na criação de seu novo aplicativo. Na primeira etapa digite em **Class name**, PrimeiroAplicativo. O Jbuilder usa esse nome para criar o arquivo chamado de PrimeiroAplicativo.java, nesse arquivo se encontra a classe Main do seu aplicativo.
- Na segunda etapa do assistente, clique em **Next** depois da primeira etapa, você irá digitar em **Class** o texto MeuPrimeiroAplicativo, no qual gerará um arquivo chamado MeuPrimeiroAplicativo.java. A classe gerada nesse passo será chamada pela classe PrimeiroAplicativo, a qual contém o método main para rodar aplicativo.
- O título dado em **Title** é o texto que aparecerá na barra de título do seu aplicativo.

### Iniciando no JBuilder

- Para rodar os arquivos gerados até aqui pelo Jbuilder, basta clicar no menu **Run**, opção **Run Project**, ou na barra de ferramentas em **Run Project**.
- Quando rodar seu aplicativo, aparecerá uma janela com o título MeuPrimeiroAplicativo, com os botões de maximizar, minimizar e fechar.
- Para quem conhece Java, sabe que alguns comportamentos são padrões na estrutura do Jframe. O evento para fechar é criado automaticamente pelo Jbuilder, assim como a centralização na tela e a aparência do SO no desenvolvimento.

### Iniciando no JBuilder

- Elementos do Visualizador de Arquivos:
  - Barra de ferramentas do projeto
  - Painel de projetos
  - Barra de ferramentas do painel estrutura
  - Painel de estrutura
  - Guia de arquivo
  - Painel de conteúdo
  - Guias de visão de arquivos
  - Painel de mensagens

## Iniciando no JBuilder

- Clicando na guia **Design**, você visualizará seu aplicativo, no qual poderá editar suas propriedades e inserir objetos como botões, caixas de texto entre outras coisas.
- Experimente alterar a cor de fundo da janela e o título do aplicativo, usando o painel **Properties**.
- Após as alterações, clique em **Run Project**.
- Depois de ter conhecido um pouco do Design do Jbuilder, clique em Source para conhecer o código gerado pelo programa.
- Volte para a guia **Design**. Na parte superior do aplicativo existe uma caixa de ferramentas, conhecida como Paleta de Componentes. Na caixa de ferramentas são encontradas várias guias que contém diversos objetos.

## Iniciando no JBuilder

- Na guia **Swing**, clique no primeiro objeto, chamado de **Jbutton**, depois clique bem ao centro de seu design.
- No inspetor **Properties**, vá até a propriedade **text** e altere para Fechar Janela.
- Em **Properties** altere a propriedade **layout**, selecionando Flow Layout. (Content Pane)
- Para finalizar, selecionando JButton1, clique em **Events**, guia que está ao lado de Properties.
- Em **actionPerformed** dê um duplo clique e adicione o seguinte código `System.exit(0);`
- Rode o programa e clique no botão para fechar o aplicativo.
- Feche o projeto: **File... Close Projects**

## Trabalhando com Layouts

- Um gerenciador de layout utiliza um editor de diálogos para arrastar e colocar os componentes na superfície do diálogo e depois usam-se ferramentas para alinhar, espaçar, etc.
- O problema é que o layout resultante precisa ser manualmente refeito caso o tamanho dos componentes mude. Por exemplo, existem programas onde se o usuário mudar o tamanho de uma fonte no seu sistema operacional, um simples botão mal planejado pode não exibir seu texto, porque a fonte aumenta, mas o botão não.

## Trabalhando com Layouts

- Um gerenciador de layout determina como os componentes serão organizados ao serem inseridos em um contêiner.
- O gerenciador de layout padrão para painéis é a classe `FlowLayout`. Essa classe permite que os componentes fluam da esquerda para direita na ordem em que são inseridos em um contêiner. Quando não houver mais espaço, uma nova linha de componentes será iniciada imediatamente abaixo da primeira e a ordem da esquerda para a direita continuará.

## Trabalhando com Layouts

- A linguagem Java inclui outros gerenciadores de layout, onde os principais são:
  - `GridLayout`
  - `BorderLayout`
  - `CardLayout`
  - `GridBagLayout`

## Trabalhando com Layouts

### Layout de Fluxo:

- Inicie um novo projeto. Neste novo projeto crie uma aplicação. Para os nomes das classes, utilize **TrabComLayouts** e **Layouts** respectivamente, colocando como título "Trabalhando com Layouts"
- Na guia design, mude esse layout para `FlowLayout` no painel Properties.
- Para demonstrar esse layout, altere o tamanho de sua janela para 350 de largura por 100 de altura (deve-se estar posicionado no item `this` na árvore de componentes(`Structure`)).
- Adicione 7 JButtons denominados: Botão1, Botão2, e assim por diante até o Botão7.
- Rode o programa e veja que a ordem colocada de cada botão é a ordem mostrada neste tipo de layout. Arrastando para aumentar ou diminuir a dimensão da janela, você notará que os botões mudam de linha automaticamente de acordo com a largura e altura disponíveis.

## Trabalhando com Layouts

- No painel Structure, altere o alinhamento dos componentes clicando no item **FlowLayout** e usando o painel Properties/alignment.
- Você pode também alterar a distância horizontal entre os componentes e a vertical em hgap e vgap respectivamente, em pixels.
- **Layout de Borda:**
  - O layout padrão para o Jbuilder é o BorderLayout, que são criados usando a classe de mesmo nome, e que se divide em um contêiner de cinco seções: norte, sul, leste, oeste e centro.
  - No layout BorderLayout os componentes dos quatro pontos cardeais ocuparão o espaço máximo necessário, o centro ocupará o espaço restante.
  - Apague os botões 6 e 7 e troque o layout para **BorderLayout**

## Trabalhando com Layouts

- **Layout de Grade:**
  - O gerenciador GridLayout organiza os componentes em uma grade de linhas e colunas. Os componentes são incluídos na primeira linha superior da grade, começando pela célula mais à esquerda e continuando para a direita na grade. Quando todas as células da linha superior estiverem preenchidas, o próximo componente será inserido na linha seguinte e assim por diante.
  - Inicie um novo projeto. Neste novo projeto, crie uma aplicação.
  - Selecione contentPane em Structure e altere a propriedade para GridLayout. Selecione depois GridLayout1 no painel Structure e altere as propriedades como mostrada na tabela a seguir:

## Trabalhando com Layouts

Columns	3
Hgap	5
Rows	3
Vgap	5

- Adicione 6 Jbuttons.

## Misturando Gerenciadores de Layouts

- Até agora vimos gerenciadores de layouts separados. Para se ter o layout perfeito, você geralmente terá que combinar layouts.
- Isso é feito inserindo contêineres em um contêiner principal, como um quadro ou janela, e fornecendo a cada um desses contêineres menores seus próprios gerenciadores de layout.
- O contêiner que deve ser usado nesses contêineres menores é o painel, que é criado a partir da classe JPanel..
- O JPanel, assim como outros painéis podem ser encontrados na guia Swing Containers, na paleta de componentes.

## Misturando Gerenciadores de Layouts – Layout de Carta

- Inicie um novo projeto. Neste novo projeto, crie uma aplicação (File... New... Application...)
  - Na árvore de componentes, em Structure selecione o componente contentPane e modifique seu layout para GridLayout.
  - Selecione o componente GridLayout1 e altere sua propriedade para 2 colunas e 1 linha.
  - Na guia de componentes **Swing Containers**, adicione 2 JPanel's no seu projeto através do uso da árvore de componentes, clicando sobre o componente contentPane em Structure (clique em contentPane após selecionar o JPanel em Swing Containers).
  - Nos dois painéis adicionados, um terá o layout XYLayout e o outro CardLayout.
  - No primeiro painel, chamado de JPanel1, adicione 2 Jbuttons, clicando sobre o painel em Structure

## Misturando Gerenciadores de Layouts – Layout de Carta

- No JButton1 mude a propriedade **text** para Vermelho e no JButton2 mude a propriedade **text** para Laranja.
- No painel JPanel2 adicione mais 2 Jpanels, onde você terá JPanel3 e JPanel4.
- Em JPanel3 altere a cor de fundo para Orange e em JPanel4 altere a cor de fundo para Red.
- Para colocar ações nos botões para que sejam mudados os painéis ao clicá-los, dê um duplo clique no botão JButton1 no design ou selecione JButton1 e na guia Events dê um duplo clique em actionPerformed.
- Em Source, adicione a seguinte linha:  
cardLayout1.show(JPanel2,"JPanel4");
- Faça o mesmo com o botão JButton2 e adicione a seguinte linha:  
cardLayout1.show(JPanel2,"JPanel3");
- Rode o programa e clique no botão escrito Vermelho e veja a mudança no layout lateral direito. O mesmo ocorre com o botão escrito Laranja.

## Misturando Gerenciadores de Layouts – Layout de Grade

- O gerenciador GridBagLayout é uma extensão do gerenciador GridLayout.
- Vamos fazer uma tela com campos de preenchimento para Nome e Senha e um botão Entrar.
- Crie uma nova aplicação (BagLayout–AplicBagLayout)
  - Para o nome do Frame (Frame Class) use FrameBagLayout.
  - Selecione na árvore de componentes contentPane e altere a propriedade layout para null. Altere também o tamanho da janela para 350, 180 (this).
  - Adicione os seguintes componentes: 2 JLabel, 1 JTextField, 1 JPasswordField e 1 JButton. Altere as propriedades de cada componente como mostrado na tabela a seguir:

## Misturando Gerenciadores de Layouts – Layout de Grade

Componente	Propriedade	Valor
jLabel1	Text	Nome:
jLabel2	Text	Senha:
jTextField1	Text	Deixe vazio aqui
jPasswordField1	Text	Deixe vazio aqui
jButton1	Text	Entrar

- Segurando o **shift** pressionado, clique nos Labels Nome e Senha e clique com o botão direito. Escolha a opção "Align Right" no menu. Selecione os campos do usuário e senha e repita o processo, alinhando à esquerda. Alinhe o botão entrar no Centro.
- Depois de ajustados os componentes, selecione o componente contentPane e altera a propriedade layout para GridBagLayout.
- Para finalizar, você vai fazer pequenos ajustes no layout usando as restrições do GridBagLayout. Selecione um componente, por exemplo JLabel1 e selecione **Constraints**. Pode-se fazer vários ajustes através da caixa de diálogo apresentada.