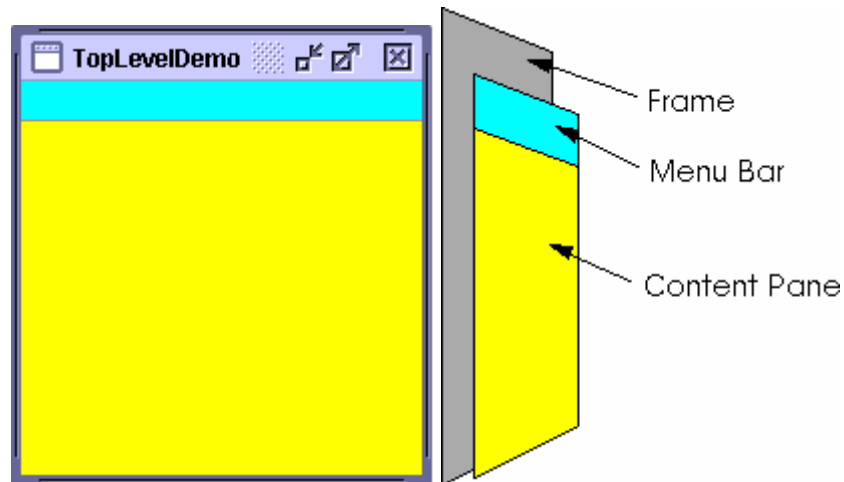


Montagem do conteúdo de uma janela

A classe JFrame discutida na última aula é um Top-Level container. Todo Top-Level container possui um painel de conteúdo (Content Pane) que armazena os componentes visíveis da janela. A figura abaixo (obtida a partir do tutorial da SUN) mostra como é a estrutura básica de um JFrame:



É importante notar que "Content Pane" também é um container, e sendo assim é capaz de armazenar referências a outros objetos. Utilizamos esse container para armazenar os componentes que desejamos apresentar dentro da janela. Existem vários componentes de IHM prontos em Java. Entre eles iremos inicialmente abordar os descritos abaixo:

Componente	Apresenta dentro da janela
JLabel	Rótulo de texto
TextField	Campo para digitação de texto
Button	Botão

Para apresentar tais componentes em uma janela, basta criar objetos do tipo desejado, e inseri-lo dentro do "Content Pane". Para obtermos acesso ao "Content Pane" de um JFrame, basta utilizar o método "getContentPane()". Desse modo, a apresentação de um texto em uma janela poderia ser feito da seguinte maneira:

```

public class FormPrincipal extends JFrame{
    public FormPrincipal() {
        setTitle("Janela Principal");
        JLabel lblNome = new JLabel("Nome");
        getContentPane().add(lblNome);
        pack();
    }
}

```

A exibição deste JFrame resultaria em:



Repare na utilização do método "pack()" no programa acima. Este método ajusta automaticamente o tamanho da janela de acordo com a disposição dos componentes inseridos.

Acontece porém que o "Content Pane" de um JFrame utiliza um gerenciador de layout chamado de "Border Layout", que nem sempre é o mais adequado para organizarmos os componentes dentro de uma janela. Se novos componentes fossem adicionados da mesma forma ao "Content Pane" acima, somente o último inserido seria apresentado na janela. Isso acontece porque o "Border Layout" irá sobrepor os componentes na parte central de sua estrutura. Para entender melhor esse comportamento é necessário analisar mais detalhadamente o funcionamento deste tipo de gerenciador de layout.

Existem vários modos de organizar os componentes em uma janela. A utilização de painéis (JPane) costuma ser caracterizado como um modo bastante elegante, além de gerar ótimos resultados de modo facilitado.

Um "Jpane" também é um container, e utiliza um gerenciador de layout chamado de "Flow Layout". Esse gerenciador alinha os componentes horizontalmente, e quando uma linha torna-se totalmente preenchida, ele dispõe os componentes seguintes na próxima linha.

Exemplo:

```

public class FormPrincipal extends JFrame{

    public FormPrincipal() {
        setTitle("Janela Principal");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel lblNome = new JLabel("Nome");
        JTextField txtNome = new JTextField("",20);
        JLabel lblSenha = new JLabel("Senha");
        JTextField txtSenha = new JTextField("",20);
        JButton btnConfirma = new JButton("Confirma");
        JButton btnCancela = new JButton("Cancela");

        JPanel painel = new JPanel();
    }
}

```

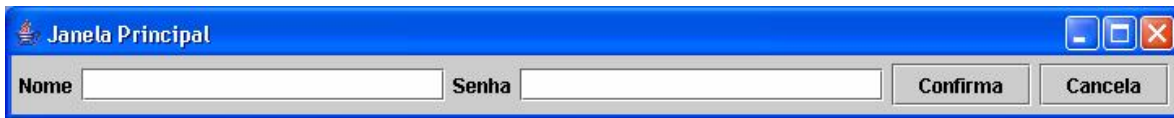
```

        painel.add(lblNome);
        painel.add(txtNome);
        painel.add(lblSenha);
        painel.add(txtSenha);
        painel.add(btnConfirma);
        painel.add(btnCancela);

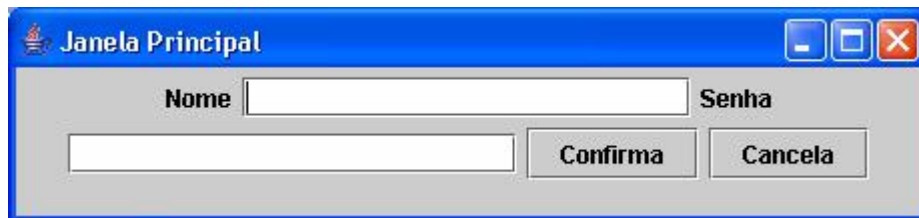
        getContentPane().add(painel);
        pack();
    }
}

```

A exibição deste JFrame resultaria em:



Ao redimensionarmos manualmente a janela podemos perceber o trabalho do gerenciador de layout. No exemplo abaixo, a largura foi diminuída através do mouse:



Uma forma de resolver esse inconveniente é utilizar outros gerenciadores de layout disponíveis na biblioteca do Java.

Iremos utilizar um gerenciador chamado de Box-Layout. Esse tipo de gerenciador é bastante simples de utilizar, e permite a criação rápida de ótimas interfaces.

O Box-Layout permite a criação de caixas de organização vertical e horizontal. Uma técnica bastante usada é criar uma caixa horizontal para os componentes de cada linha da janela, e no final agrupar todas as linhas em uma caixa vertical. Exemplo:

```

public class FormPrincipal extends JFrame{

    public FormPrincipal() {
        setTitle("Janela Principal");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel lblNome = new JLabel("Nome");
        JTextField txtNome = new JTextField("",20);
        JLabel lblSenha = new JLabel("Senha");
        JTextField txtSenha = new JTextField("",20);
        JButton btnConfirma = new JButton("Confirma");
        JButton btnCancela = new JButton("Cancela");
    }
}

```

```

Box primeiraLinha = Box.createHorizontalBox();
primeiraLinha.add(lblNome);
primeiraLinha.add(txtNome);

Box segundaLinha = Box.createHorizontalBox();
segundaLinha.add(lblSenha);
segundaLinha.add(txtSenha);

Box terceiraLinha = Box.createHorizontalBox();
terceiraLinha.add(btnConfirma);
terceiraLinha.add(btnCancela);

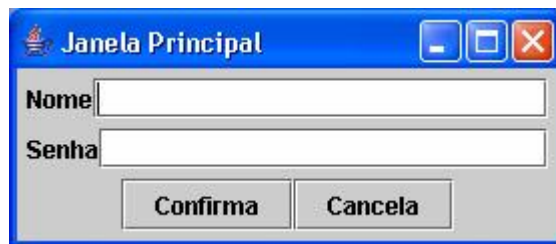
Box linhas = Box.createVerticalBox();
linhas.add(primeiraLinha);
linhas.add(Box.createVerticalStrut(5));
linhas.add(segundaLinha);
linhas.add(Box.createVerticalStrut(5));
linhas.add(terceiraLinha);

JPanel painel = new JPanel();
painel.add(linhas);

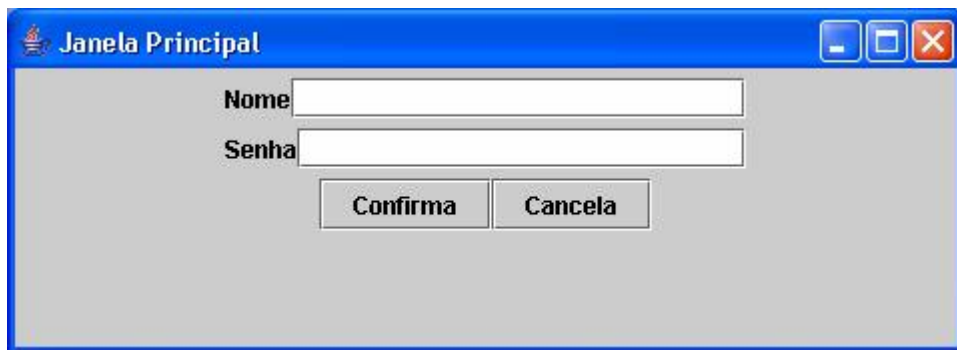
getContentPane().add(painel);
pack();
}
}

```

A exibição deste JFrame resultará em:

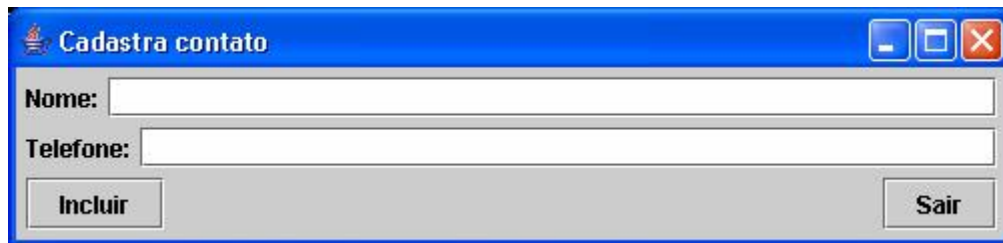


Agora, mesmo redimensionando a janela manualmente, o lay-out desejado é mantido. Exemplo:



Exercícios

- 1) Criar um projeto chamado "Java2_0004_JPaneJLabelJTextJButtonBoxLayout" que apresente a janela de login (nome, senha, botão confirma e cancela) utilizada como exemplo anteriormente.
- 2) Criar um novo projeto chamado "Java2_0004_FormCadastraContato" que apresente a seguinte tela para cadastro de contatos (Perceba que os botões estão posicionados nas laterais da linha).



- 3) Centralizar automaticamente a janela criada acima. Utilize o método **size()** de JFrame para saber o tamanho da janela.

Links interessantes

"Using Top-Level Containers" em:

<http://java.sun.com/docs/books/tutorial/uiswing/components/toplevel.html>

Final do documento

Prof.Fernando J.C.Branquinho

<http://br.geocities.com/brangs>