

Programação Orientada a Objetos

Interface gráfica em Java: Swing e SWT

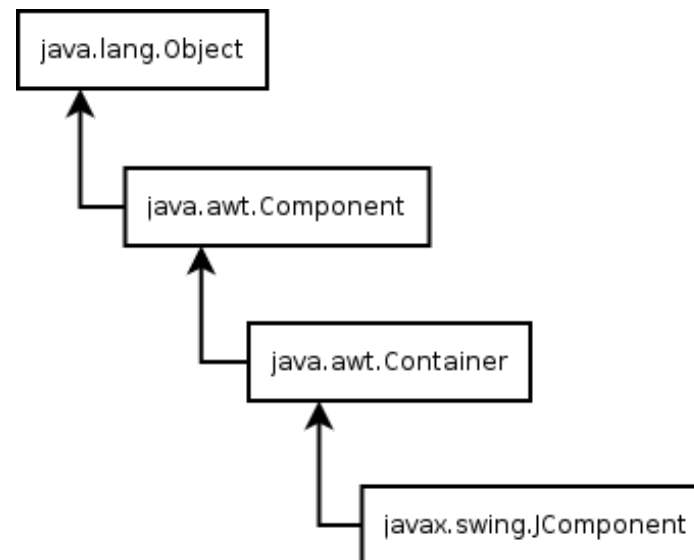
UNIFEI – 14/06/2012

Componentes gráficos em Java

- Os componentes gráficos em Java
 - Componentes de desenho – 2D e 3D
 - Componentes de interface – GUI (Graphical User Interface)
- AWT (Abstract Window Toolkit)
 - Biblioteca Java para componentes gráficos
 - Pacote **java.awt**
 - Disponível desde a primeira versão de Java
 - Diferentes especializações podem ser encontradas

Biblioteca Swing

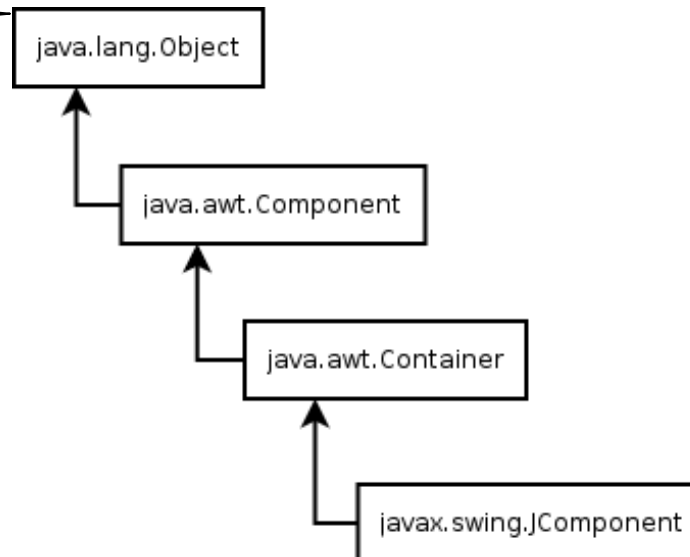
- Uma especialização da biblioteca AWT para interfaces GUI
 - Padroniza a exibição das interfaces em diferentes plataformas



Biblioteca Swing

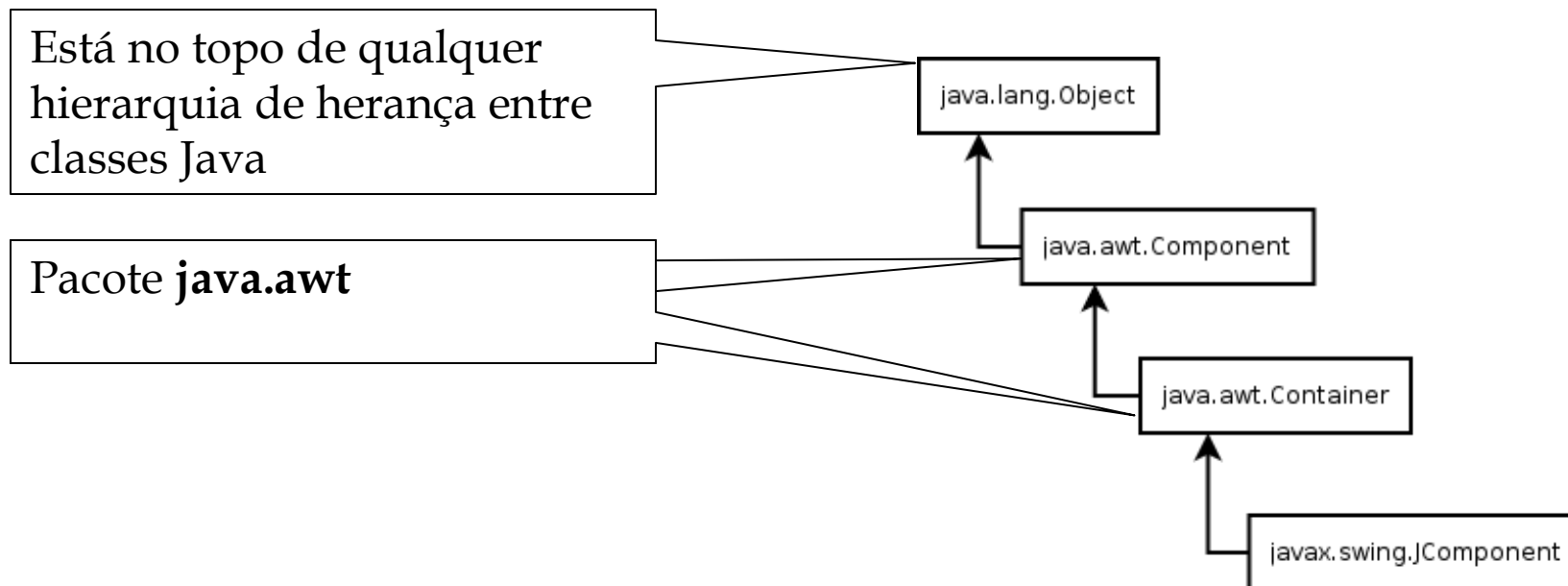
- Uma especialização da biblioteca AWT para interfaces GUI
 - Padroniza a exibição das interfaces em diferentes plataformas

Está no topo de qualquer hierarquia de herança entre classes Java



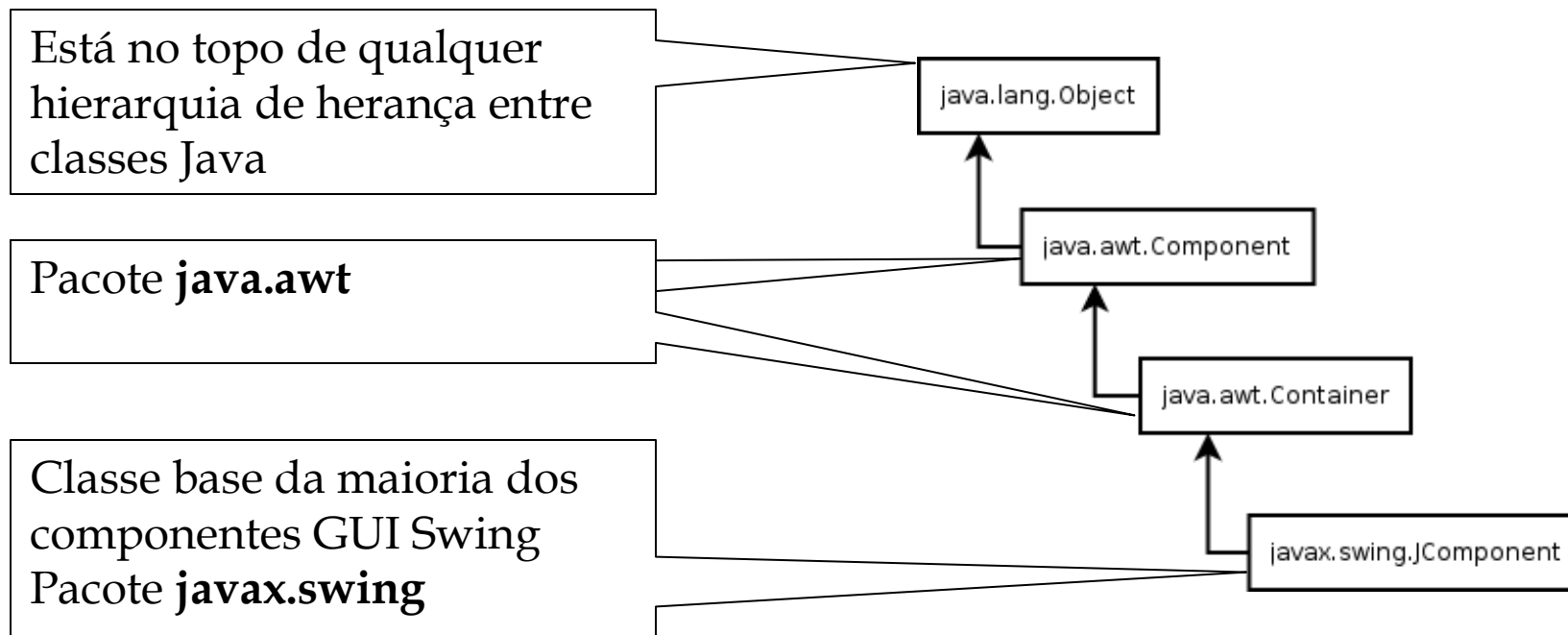
Biblioteca Swing

- Uma especialização da biblioteca AWT para interfaces GUI
 - Padroniza a exibição das interfaces em diferentes plataformas



Biblioteca Swing

- Uma especialização da biblioteca AWT para interfaces GUI
 - Padroniza a exibição das interfaces em diferentes plataformas



Biblioteca Swing

- O desenvolvimento de uma interface GUI em Swing se baseia no conceito de Janela
 - Uma janela é um contêiner de objetos gráficos
 - Os objetos devem ser anexados ao contêiner para que sejam exibidos
 - Existem diferentes classes que podem representar uma janela, no entanto, a classe **JFrame** fornece o padrão de janela comum da maioria dos aplicativos GUI

Um aplicativo GUI - Swing

- 1ª etapa:
 - Criação da janela que conterá os demais objetos gráficos da aplicação
- 2ª etapa:
 - Inserção dos componentes da interface
- 3ª etapa
 - Tratamento de eventos
- 4ª etapa
 - Lógica do programa

Um aplicativo GUI - Swing

- **1ª etapa:** Definição da janela que conterá os demais objetos gráficos da aplicação

```
public class ListaPresenca extends JFrame{  
    public ListaPresenca( ){  
        super("Lista de Presença");  
    }  
}
```

javax.swing.JFrame

Fornece uma janela com atributos padrão, como barra de títulos, botões de minimizar, maximizar e fechar.

Um aplicativo GUI - Swing

- **1ª etapa:** Definição da janela que conterá os demais objetos gráficos da aplicação

```
public class ListaPresenca extends JFrame{  
    public ListaPresenca( ){  
        super("Lista de Presença");  
    }  
}
```

Chamada ao construtor de JFrame,
para inicializar a janela com o título
'Lista de presença'

Um aplicativo GUI - Swing

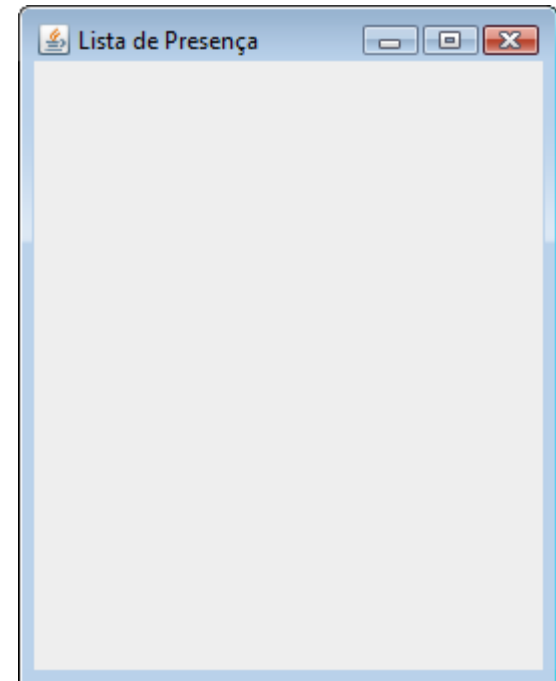
- **1ª etapa:** Definição da janela que conterá os demais objetos gráficos da aplicação

```
public static void main(String[] args){  
    ListaPresenca janela = new ListaPresenca();  
    janela.setSize(270,340);  
    janela.setVisible( true );  
}
```

Um aplicativo GUI - Swing

- **1ª etapa:** Definição da janela que conterá os demais objetos gráficos da aplicação

```
public static void main(String[] args){  
    ListaPresenca janela = new ListaPresenca();  
    janela.setSize(270,340);  
    janela.setVisible( true );  
}
```

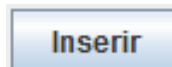


Um aplicativo GUI - Swing

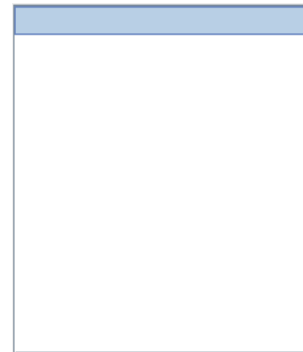
- **2ª etapa:** Inserção dos componentes da interface
- A biblioteca Swing fornece diversas classes que representam os elementos de interface padrão
 - Além disto, estas classes podem ser estendidas e novos componentes criados, personalizados.



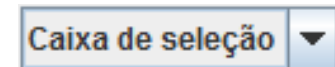
TextField



Button



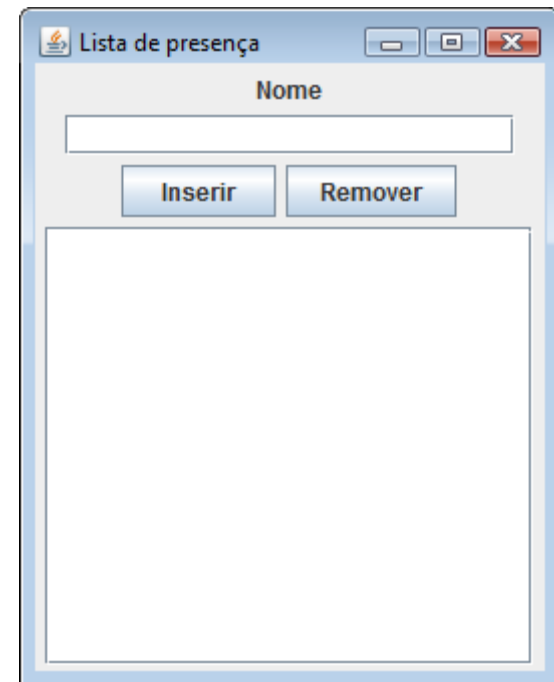
List



ComboBox

Um aplicativo GUI - Swing

- 2ª etapa: Inserção dos componentes da interface
- Inserção dos componentes na Janela
 - **LayoutManager**: Gerencia o posicionamento dos componentes na janela



Um aplicativo GUI - Swing

- 3ª etapa: Tratamento de eventos
- Os eventos permitem a interação entre usuários e interface, permitindo que o programa execute em função das ações do usuário
 - Movimentação de mouse
 - Pressionar teclas
 - Clicar em botões
 - Selecionar itens

Um aplicativo GUI - Swing

- 3^a etapa: Tratamento de eventos
 - Classes do pacote `java.awt.event`
 - Os componentes de interface disparam rotinas ao ‘ouvir’ um evento
 - Define-se uma classe que serve como ‘tratador de eventos’ de determinados objetos
 - Quando acionada, esta classe executa o método associado ao evento para tratá-lo
 - Diferentes objetos podem se vincular ao mesmo tratador de eventos

Um aplicativo GUI - Swing

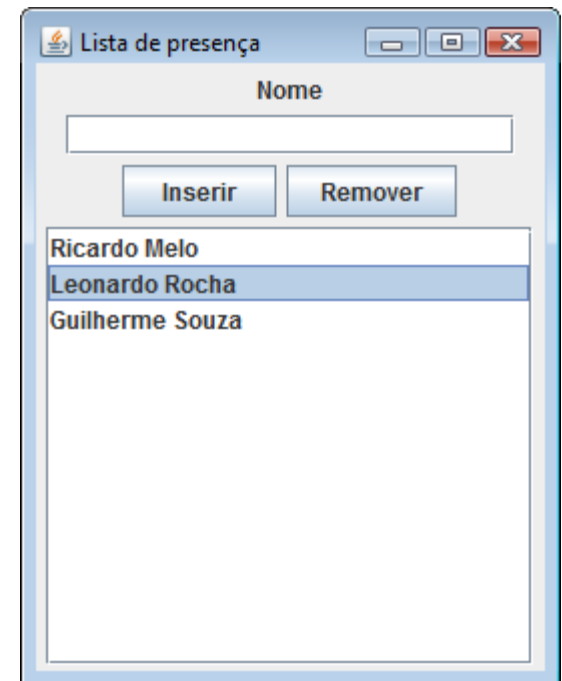
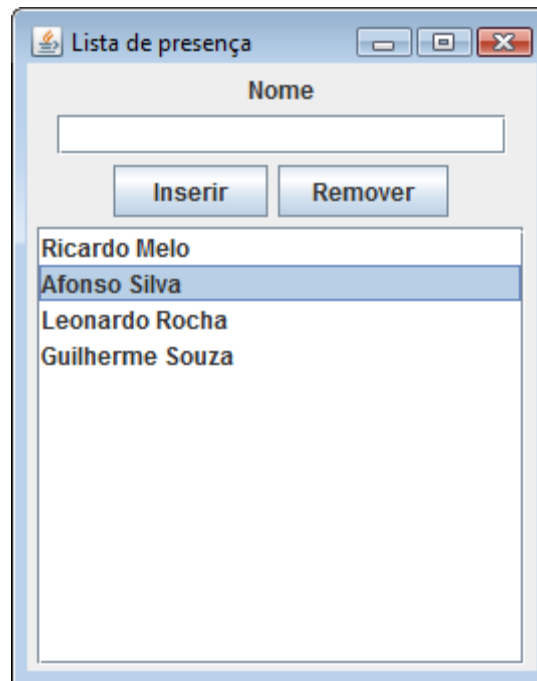
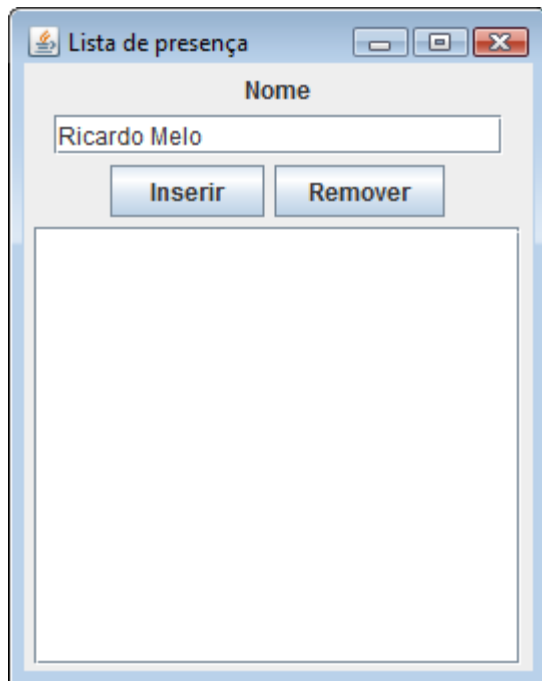
- 3ª etapa: Tratamento de eventos
- Interfaces comuns para o tratamento de eventos
 - **ActionListener** – eventos de ações do usuário como cliques em botões ou alteração de campos texto
 - **ItemListener** – eventos relacionados à manipulação de itens em lista de itens
 - **MouseListener** – eventos associados à ação do usuário no programa através do mouse

Um aplicativo GUI - Swing

- 4^a etapa: Lógica do programa
- A aplicação é concluída após inserir a lógica do programa nos os eventos de clique nos botões

Um aplicativo GUI - Swing

- 4ª etapa: Lógica do programa
- A aplicação é concluída após inserir a lógica do programa nos os eventos de clique nos botões



A biblioteca SWT

- O SWT (*Standard Widget Toolkit*) é uma alternativa proposta ao desenvolvimento GUI com Swing
- Diferenças entre o desenvolvimento de aplicativos GUI com Swing e SWT
 - Aparência
 - Eficiência
 - Estrutura de desenvolvimento
 - **NÃO HÁ SUPORTE NATIVO JAVA PARA O SWT**
 - Necessidade de baixar e instalar a biblioteca swt

Um aplicativo GUI - SWT

- Será desenvolvido o mesmo aplicativo – Lista de Presença
- As mesmas 4 etapas serão consideradas
- As principais diferenças se concentram na elaboração da interface
 - O tratamento de erros e a inserção da lógica do programa têm grande semelhança com Swing

Um aplicativo GUI - SWT

- **1ª etapa:** Definição da janela que conterá os demais objetos gráficos da aplicação

```
public class ListaPresencaSWT {  
    public static void main(String[] args){  
        Display display = new Display();  
        Shell shell = new Shell(display);  
        shell.setText("Lista de Presença");  
        shell.open();  
    }  
}
```

Um aplicativo GUI - SWT

- **1ª etapa:** Definição da janela que conterá os demais objetos gráficos da aplicação

```
public class ListaPresencaSWT {  
    public static void main(String[] args){  
        Display display = new Display();  
        Shell shell = new Shell(display);  
        shell.setText("Lista de Presença");  
        shell.open();  
    }  
}
```

Um objeto Display é um contêiner de Janelas

Um objeto Shell representa uma janela de interface com o usuário

Um aplicativo GUI - SWT

- **2ª etapa:** Inserção dos componentes da interface
- Como na biblioteca Swing, diversas classes podem ser instanciadas para gerar a interface

```
public class ListaPresencaSWT {  
    public static void main(String[] args){  
        Display display = new Display();  
        Shell shell = new Shell(display);  
        shell.setText("Lista de Presença");  
  
        Button btnIns = new Button(shell, SWT.PUSH);  
        btnIns.setBounds(70, 70, 60, 20);  
        btnIns.setText("Inserir");  
        ...  
  
        shell.open();  
    }  
}
```


Um aplicativo GUI - SWT

- **2ª etapa:** Inserção dos componentes da interface
- Como na biblioteca Swing, diversas classes podem ser instanciadas para gerar a interface

```
public class ListaPresencaSWT {  
    public static void main(String[] args){  
        Display display = new Display();  
        Shell shell = new Shell(display);  
        shell.setText("Lista de Presença");  
  
        Button btnIns = new Button(shell, SWT.PUSH);  
        btnIns.setBounds(70, 70, 60, 20);  
        btnIns.setText("Inserir");  
        ...  
        shell.open();  
    }  
}
```

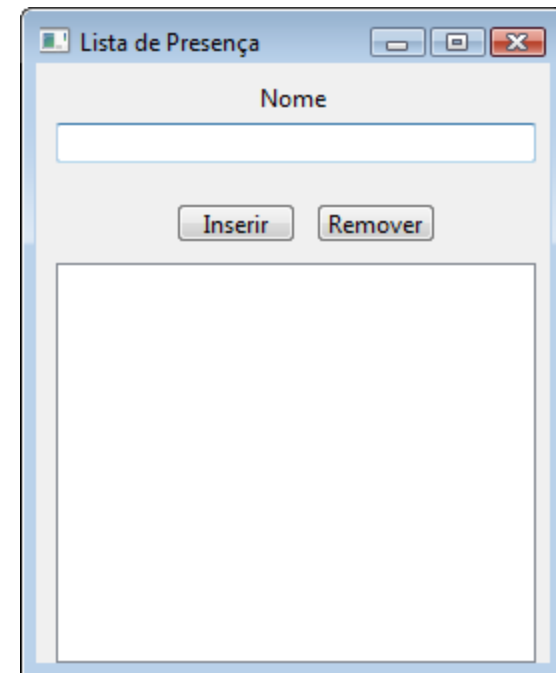
Passa-se como parâmetro o objeto de janela onde o componente será anexado

Instancia-se todos os objetos que compõe a interface e os anexa à janela

Um aplicativo GUI - SWT

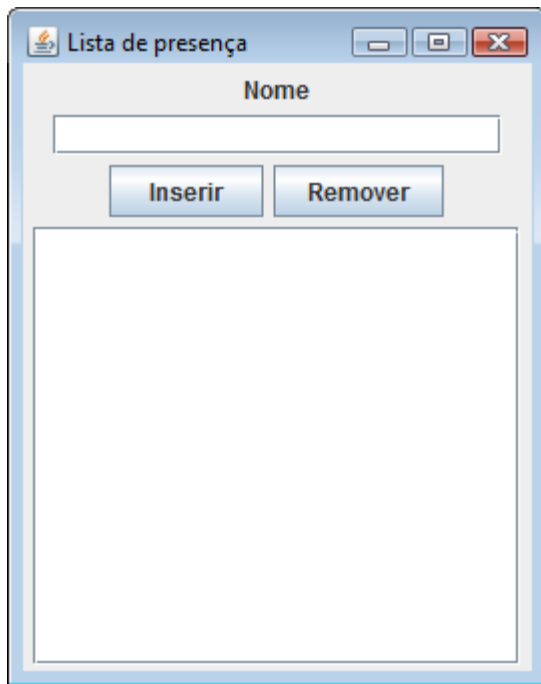
- **2ª etapa:** Inserção dos componentes da interface
- Como na biblioteca Swing, diversas classes podem ser instanciadas para gerar a interface

```
public class ListaPresencaSWT {  
    public static void main(String[] args){  
        Display display = new Display();  
        Shell shell = new Shell(display);  
        shell.setText("Lista de Presença");  
  
        Button btnIns = new Button(shell, SWT.PUSH);  
        btnIns.setBounds(70, 70, 60, 20);  
        btnIns.setText("Inserir");  
        ...  
  
        shell.open();  
    }  
}
```

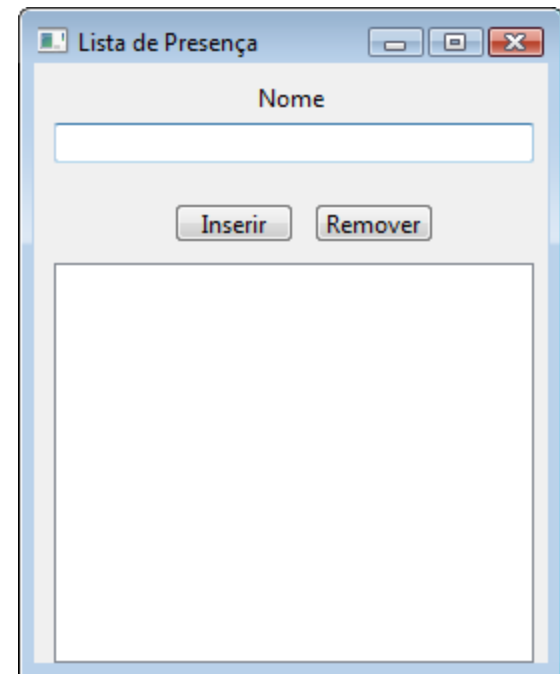


Um aplicativo GUI - SWT

- **2ª etapa:** Inserção dos componentes da interface
- É possível notar a diferença entre as interfaces geradas utilizando o Swing e SWT



biblioteca Swing



biblioteca SWT

Um aplicativo GUI - SWT

- 3^a e 4^a etapas: Eventos e Lógica do programa
- O mecanismo para o tratamento de eventos é semelhante
 - Todos aqueles componentes que desejarem tratar um dado evento, definem uma classe para ‘ouvi-los’

Um aplicativo GUI - SWT

- 3ª e 4ª etapas: Eventos e Lógica do programa

