

Componentes da Interface Gráfica com o Usuário (GUI)

Parte I

Objetivos

- Entender princípios de projeto.
- Capacitação para construir GUIs.
- Entender os pacotes que contém componentes de GUI, e as classes e interfaces de tratamentos de eventos.
- Criar e manipular botões, rótulos, listas, campos de texto e painéis.
- Entender eventos de mouse e eventos de teclado.
- Entender e utilizar os gerenciadores de layout.

Introdução

- Uma GUI apresenta uma forma com que os usuários de programas podem executar componentes ou programas, entrar com dados para programas e receber dados de programas,

Introdução

- dando ao usuário um nível básico de familiaridade, reduzindo o tempo para aprender a usar um programa e aumentando a habilidade de usar um programa de maneira produtiva.

Introdução

- Proporciona a um programa uma “aparência” e um “comportamento”: o aspecto e a maneira com que o usuário interage com um programa.
- Fornecendo a diferentes aplicativos um conjunto de componentes intuitivos, pré-construídos, segundo um Modelo de Componentes definido para a linguagem Java.

Componentes GUI

- Uma janela com componentes rotulados da GUI.
- Barra de menus (File, Edit, View, Insert, Format, Tools, Presentations, Windows, Help, other applications).
- Conjunto de botões e campo de texto.
- Os menus, botões, campos de texto e rótulos fazem parte da GUI.
- Permitem que o usuário interaja com o aplicativo.

Componentes GUI

- Também chamados de *controles* ou “*widgets – windows gadgets*”.
- O componente GUI é um objeto com o qual o usuário interage através do mouse, do teclado, ou outra forma de entrada, como o reconhecimento de voz.

Pacotes Java para GUIs

- A linguagem Java apresenta dois pacotes (de classes e interfaces) apropriados para a construção de GUIs.
- Pacote original: *AWT – Abstract Windowing Toolkit*
java.awt
- Pacote conhecido como *Swing*: **javax.swing** contendo a classe **JComponent** e os componentes JFC (Java Foundation Classes) do Swing.

AWT e Swing

- Ressaltar as classes importantes e a hierarquia dessas classes.
- Ver figura Classes AWT e Swing
(Livro do Orfali)

Componentes AWT

- Os componentes AWT estão diretamente associados com os recursos para interface gráfica da plataforma local.

Componentes AWT

- Quando um programa Java com uma GUI AWT é executado em diferentes plataformas de sistemas operacionais, os componentes GUI são exibidos com uma aparência diferente em cada plataforma.

Componentes AWT

- Juntos, o **aspecto e a maneira** como o usuário interage com o programa, são conhecidos como a *aparência e comportamento* desse programa.

Componentes AWT

- Considere um programa que exibe um componente **Button** do pacote `java.awt`.
- Em uma máquina com o sistema operacional Windows, o **Button** terá a mesma *aparência e comportamento* de outras aplicações **Windows**.

Componentes AWT

- O mesmo para o sistema operacional **Apple Macintosh**.
- O mesmo para os sistemas operacionais oriundos do **UNIX**.
- Além das **diferenças na aparência** e a maneira com que o usuário interage com os componentes GUI em uma plataforma particular também difere de uma plataforma para outra.

Componentes AWT

- As subclasses de `java.awt.Window` (como a `JFrame`) que exibem janelas na tela, e as subclasses de `java.applet.Applet` (`JApplet`) ainda exigem interação direta com o sistema de janelas local (da plataforma de SO sendo usada).

Componentes Swing

- Os componentes Swing permitem que o programador especifique uma **aparência e um comportamento uniformes** em todas as plataformas.

Aparência e Comportamento

- Como os **componentes Swing** são escritos em Java, eles oferecem um maior nível de portabilidade e flexibilidade que os componentes GUI Java originais do pacote `java.awt`.

Superclasses dos Componentes Swing

- `java.lang.Object`
- `java.awt.Component`
- `java.awt.Container`
- `javax.swing.JComponent`

Componentes Swing

- Hierarquia de classes que definem os atributos e os comportamentos que são comuns para a maioria dos componentes Swing.

Componentes Swing

- A classe que herda da classe `Component` é um componente.
- A classe `Container` herda da classe `Component`.
- O `Container` é um `Component`.

Componentes Swing

- JComponent é um Container.
- A classe JComponent é a superclasse para a maioria dos componentes Swing.
- Define os atributos e os comportamentos comuns a todas as subclasses de JComponent.

A classe JOptionPane

- Classe do Swing `JOptionPane` que permite um programa ler dados do teclado e gerar como saída os dados para a tela utilizando o diálogo de entrada e o diálogo de mensagem.

A classe JOptionPane

- `import javax.swing.JOptionPane`
- Métodos: `showInputDialog(...);`
`showMessageDialog(...);`

A classe JOptionPane

- Pacote: `javax.swing,`
- Classe: `JOptionPane,`
- Método: `static`
`ShowMessageDialog(arg1, arg2)`
- `arg1` pode ser `null` e a caixa de diálogo aparecerá no centro da tela.
- `arg2` é a string a exibir na caixa de diálogo.

A Classe JFrame

- JFrame

Classe do pacote `javax.swing` que fornece os atributos e componentes básicos de uma janela – uma barra de título e botões para minimizar, maximizar e fechar a janela.

O método Super

- `super()`
- `super(arg1, arg2, ..., argN)`
- `// chamada explícita ao construtor default da superclasse.`

O Container

- É uma coleção de componentes relacionados que definem uma interface.
- Em aplicativos com `JFrame` e em *applets*, anexamos os componentes ao **painel de conteúdo (content pane)**, que é um objeto da classe `Container`.

Alguns métodos de Container

- Métodos: `setLayout() ;`
- `add() ;`
- `getContentPane() ;`

Gerenciadores de Layout

- São classes fornecidas para organizar (**posicionar e dimensionar**) os componentes GUI em um container para fins de apresentação.
- Fornecem recursos básicos de layout, e são mais fáceis de usar do que determinar a posição e o tamanho exatos de cada componente GUI.

Gerenciadores de Layout

- O programador se concentra na aparência e no comportamento.
- E os gerenciadores processam a maioria dos detalhes do layout.

Gerenciadores de Layout

- `FlowLayout`
- Gerenciador de layout mais básico.
- Os componentes são colocados em um container da esquerda para à direita, na ordem em que são gerados pelo programa.

Gerenciadores de Layout

- Quando se alcança a borda do container, os componentes continuam na próxima linha.
- A classe `FlowLayout` herda da classe `Object` e implementa a interface *LayoutManager*.